WenQuanYi Micro Hei [Scale=0.9]WenQuanYi Micro Hei Mono song-WenQuanYi Micro Hei sfWenQuanYi Micro Hei "zh" = 0pt plus 1pt

ãĂŁPython CookbookãĂŃçňňäÿL'çL'Ĺ

Release 3.0.0

çĘŁèČ¡

Dec 20, 2018

Contents

Contents:

1 Copyright

æŻťåďŽåŘŚåÿČä£ąæĄŕèŕůåŘĆèĂČ
http://oreilly.com/catalog/errata.csp?isbn=9781449340377

2 åĽ■èÍĂ

2.1 éąźçŻőäÿżéąţ

https://github.com/yidao620c/python3-cookbook

2.2 èŕŚèĂĚçŽĎèŕİ

äžžçŤ§èŃęç§■ïijŇæĹŚçŤÍ PythonïijĄ

èŕŚèĂĚäÿĂçŻťåİŽæŇĄä¡£çŤÍ Python 3ïijŇåŻääÿžåőČäżčèąĺäžĘ Python
çŽĎæIJłæİěãĂĆèŹ¡çĎűåŘŚåŘŐåĚijåőźæŸŕåőČçŽĎçąňäijďïijŇä¡ĘæŸŕè£ŹäÿłåśĂéİćè£§æŮľäijŽæŤźåŔŸçŽ
èĂŇäÿŤ Python 3 çŽĎæIJłæİěéIJĂèęĄæŕŔäÿłäžžçŽĎåÿőåŁľåŠŇæŤŕæŇĄãĂĆ
çŻőåĽ■åÿĆéİćäÿŁçŽĎæŢŹçÍŃäźęçś■ïijŇç¡ŚäÿŁçŽĎæĽŃåĘŇåďğéČÍåĹĘå§žæIJňéČ¡æŸŕ
2.x çşżåĹŮçŽĎïijŇäÿŞéŮÍå§žäžŐ 3.x çşżåĹŮçŽĎäźęçś■åřŚçŽĎåŕŕæĂIJãĂĆ

æIJĂè£ŚçIJŃåĹřäÿĂæIJňãĂŁPython CookbookãĂŃ3rd Edi-
tionïijŇåőŇåĚÍå§žäžŐ Python 3ïijŇåĘźçŽĎäź§å¿Ĺäÿ■éŤŹãĂĆ äÿžäžĘ Python 3
çŽĎæŹőåŔŁïijŇæĹŚäź§äÿ■èĞłéĞŔåŁŻïijŇæČşåĄŽçĆźäżĂäźĹäžŃæČĚãĂĆäžŐæŸŕäźŐïijŇåřśæIJĽäžĘç£
è£Źäÿ■æŸŕäÿĂéąźè¡żæİ¿çŽĎåůěä¡IJïijŇå■ťæŸŕäÿĂäżűåĂijå¿ŮåĄŽçŽĎåůěä¡IJïijŽäÿ■äżĚæŰźä¿£äžĘåĹńä

èŕŚèĂĚäijŽåİŽæŇĄåŕźèĞłåůśæŕŔäÿĂåŔěçŽĎç£żèŕŚèť§èťčïijŇåŁŻæśĆéńŸèťÍéĞŔãĂĆä¡ĘåŔŮèČ¡åŁŽ
åęĆæđIJèŕŚæŰĞäÿ■æIJĽäżĂäźĹéŤŹæijŘçŽĎåIJřæŰźèŕůåďğåőűèğĄèřĚïijŇäź§æňćè£ŐåďğåőűéŽŔæŮűæŇ
yidao620@gmail.com

2.3 ä¡IJèĂĚçŽĎèŕİ

èĞłäžŐ 2008 åźťäžěæİěïijŇPython 3 æÍłçľžåĞžäÿŰåźűæĚćæĚćè£ŻåŇŰãĂĆPython
3 çŽĎæţĄèąŇäÿĂçŻťèćńèőďäÿžéIJĂèęĄå¿ĹéŢ£äÿĂæőţæŮűéŮťãĂĆ
äžŃåőđäÿŁïijŇåĹřæĹŚåĘŹè£ŹæIJňäźęçŽĎ 2013 åźťïijŇçżİåďğéČÍåĹĘçŽĎ Python
çÍŃåžŘåŚŸäż■çĎűåIJÍçŤ§äžğçŐŕåćČäÿ■ä¡£çŤÍçŽĎæŸŕçĽĹæIJň 2 çşżåĹŮïijŇ
æIJĂäÿżèęĄæŸŕåŻääÿž Python 3 äÿ■åŘŚåŘŐåĚijåőźãĂĆæŕňæŮăçŰŚéŮőïijŇåŕźäžŐåůěä¡IJåIJÍéĄŮçŢŹäżčç
ä¡ĘæŸŕæŤ¿çIJijæIJłæİěïijŇä¡ååřśäijŽåŔŚçŐř Python 3 çżŹä¡åäÿęæİěäÿ■äÿĂæäůçŽĎæČŁåŰIJãĂĆ

æ■čåęĆ Python 3 äżčèąĺæIJłæİěäÿĂæäůïijŇæŰřçŽĎãĂŁPython Cook-
bookãĂŃçĽĹæIJňçŻÿæŕŤè¿ČäźŃåĽ■çŽĎçĽĹæIJňæIJĽäżĘäÿĂäÿłåĚÍæŰřçŽĎæŤźåŔŸãĂĆ
éęŰåĚĹïijŇäź§æŸŕæIJĂéĞ■èęĄçŽĎïijŇè£ŹæĎŔåŚşçÌĂæIJňäźęæŸŕäÿĂæIJňéİđåÿÿåĽ■æš£çŽĎåŔĆèĂČäź
Python 3.3 çĽĹæIJňäÿŃéİćçijŰåĘźåŠŇæţŃèŕŢçŽĎïijŇ åźűæšąæIJĽèĂČèŹŚäźŃåĽ■èĂĄçĽĹæIJňçŽĎåĚijå
ä¡ĘæŸŕæĹŚäżňæIJĂçżĹçŽĎçŻőçŽĎæŸŕåĘŹäÿĂæIJňáőŇåĚÍå§žäžŐçŐřäżčáůěåĚůåŠŇèŕ■èÍĂçŽĎäźęçś■ãĂ
æĹŚäżňåÿŇæIJŻæIJňäźęèČ¡åď§æŇĞåŕijäžžäżňä¡£çŤÍ Python 3
çijŰåĘźæŰřçŽĎäżčçăĄæĹŰèĂĚå■Ğçžğäź ŃåĽ■çŽĎéĄŮçŢŹäżčçăĄãĂĆ

æŕňæŮăçŰŚéŮőïijŇçijŰåĘźäÿĂæIJňè£ŹæăůçŽĎäźęçżŹçijŰè¿Śåůěä¡IJåÿęæİěäÿĂáőŽçŽĎæŇŚæĹŸãĂ
Python çğŸçś■çŽĎèŕİïijŇäijŽåIJÍèŕÿåęĆ ActiveStateâĂŹs Python recipes æĹŰèĂĚ Stack
Overflow çŽĎç¡ŚçńŹäÿŁæŘIJåĹřæŢřäżěå■Čèőąç ŽĎæIJĽçŤÍçŽĎçğŸçś■ïijŇä¡ĘæŸŕåĚűäÿ■çżIJåďğéČÍåĹĘé
è£ŹäżŽçğŸçś■éŹďäżĘæŸŕå§žäžŐ Python 2 çijŰåĘźäźŃåďŰïijŇåŕŕèČ¡è£ŸæIJĽå¿Ĺåď Žèğčå ĘşæŰźæą ĹåI
ïijĹæŕŤåęĆ 2.3 åŠŇ 2.4 çĽĹæIJňïijĽãĂĆ åŔęåďŰïijŇåőČäżňè£ŸäijŽçżŔåÿÿä¡£çŤÍäÿĂäżŽè£ĞæŮűçŽĎæŁ

Python 3.3 éĞŇéİćåŐżäžĘãĂĆåŕżæĽ¿åőŇåĚĺå§žäžŐ Python 3 çŽĎçğŸçś■çIJ§çŽĎéŽ¿äÿŁåŁäéŽ¿åŢŁãĂĆ

è£ŹæIJňäźęçŽĎæĽĂæIJĽäÿżéćŸéČ¡æŸŕå§žäžŐåůšçżŔå■ŸåIJÍçŽĎäżčçăĄåŠŇæŁĂæIJŕïijŇèĂŇäÿ■æ
Python 3 çĽźæIJĽçŽĎçğŸçś■ãĂĆ åIJÍåŐ§æIJĽäžčçăĄå§žçąĂäÿŁïijŇæĹŚäžňåőŇåĚĺä¡£çŤÍæIJĂæŰřçŽĎ
Python æŁĂæIJŕåŐżæŤźéĂäăĂĆ æĽĂäżěïijŇäżżä¡ŢæČşä¡£çŤÍæIJĂæŰřæŁĂæIJŕçijŰåĘŹäżčçăĄçŽĎçÍŇåž

åIJÍéĂĽæŃľèęĄåŇĚåŘńåŞłäžŻçğŸçś■æŰźéİćïijŇå¿ĹæŸŐæŸ¿äÿ■åŘŕèČ¡çijŰåĘŹäÿĂæIJňäźęåŻŁæŃ
Python écĘå§§æĽĂæIJĽçŽĎäÿIJèě£ãĂĆ åŻäæ■d'ïijŇæĹŚäžňäijŸåĚĹéĂĽæŃľäžĘ Python
èŕ■èÍĂæäÿå£ČéČÍåĹĘïijŇäżěåŘŁéĆčäžŻæIJĽçİĂåź£æşŻåžŤçŤÍéćĘå§§çŽĎéŰőécŸãĂĆ
åŘęåď'ŰïijŇåĚűäÿ■æIJĽå¿Ĺåď'ŽçğŸçś■çŤÍæİěåśŢçď'ž Python 3 çŽĎæŰřçĽźæĂğïijŇ
è£ŹåŕźäžŐå¿Ĺåď'Žäžżæİěèŕť'æŸŕæŕŤè¿ČéŹŇçŤ§çŽĎïijŇåŞłæĂŢæŸŕä¡£çŤÍ
Python èĂĄçĽĹæIJňçŽĎçżŘéłŇäÿřåŕŇçŽĎçÍŇåžŘåŚŸãĂĆ
è£ŹäžŻçď'žä¿ŃçÍŇåžŘäź§äijŽåĄŘåŘŚäžŐåśŢçď'žäÿĂäżŻæIJĽçİĂåź£æşŻåžŤçŤÍçŽĎçijŰçÍŇæŁĂæIJŕ
ïijĹå■şçijŰçÍŇæĺąåijŘïijĽïijŇ èĂŇäÿ■æŸŕäżĚäżĚåőŽä¡■åIJÍäÿĂäżŻåĚůä¡ŞçŽĎéŰőécŸäÿŁãĂĆåŕ¡çőąäź§æ
Python èŕ■èÍĂæäÿå£ČåŠŇæăĞåĞĘåžŞãĂĆ

2.4 è£ŹæIJňäźęéĂĆåŘĹèřĄ

è£ŹæIJňäźęçŽĎçŻőæăĞèŕżèĂĚæŸŕéĆčäžŻæČşæůśåĚěçŘĘèğč Python
èŕ■èÍĂæIJżåĹűåŠŇçŐřäżčçijŰçÍŇéčŐæäijçŽĎæIJĽçżŘéłŇçŽĎ Python çÍŇåžŘåŚŸãĂĆ
æIJňäźęåď'ğéČÍåĹĘåĘĚåőźéŻĘäÿ■äžŐåIJÍæăĞåĞĘåžŞïijŇæąĘæđűåŠŇåžŤçŤÍçÍŇåžŘäÿ■åź£æşŻä¡£çŤÍçŽĎ
æIJňäźęæĽĂæIJĽçď'žä¿ŇåİĞåĄĞěő¿èŕżèĂĚåĚůæIJĽäÿĂäőŽçŽĎçijŰçÍŇèČŇæŹŕåźűäÿŤåŘŕäżěèŕżæĞĆçŽ
ïijĹæŕŤåęĆå§žæIJňçŽĎèőąçőŮæIJżçğŚå■ęç§ěèŕĘïijŇæŢŕæ■őçżŞæđĎç§ěèŕĘïijŇçőŮæşŢåď'■æİĆåžęïijŇçşżç
èŕ■èÍĂçijŰçÍŇç■Ľïij'ãĂĆ åŘęåď'ŰïijŇæŕŘäÿłçď'žä¿ŇéČ¡åŘłæŸŕäÿĂäÿłåĚěéŮÍæŇĞåŕijïijŇåęĆæđIJèŕżèĂĚ
æĹŚäžňåĄĞåőŽèŕżèĂĚåŘŕäżěå¿ĹçĘ§çżČçŽĎä¡£çŤÍæŘIJçť'ćåijŢæŞŐäżěåŘŁç§ěéĄŞæĂŐæăůæ§ěèŕćåIJÍçž£
Python æŰĞæąčãĂĆ

æIJĽäÿĂäžŻæŻť'åŁäéńŸçžğçŽĎçğŸçś■ïijŇåęĆæđIJèĂŘå£ČéŸĚèŕżïijŇåřĘæIJĽåŁľäžŐçŘĘèğč
Python åžŢåśĆçŽĎåůěä¡IJåŐ§çŘĘãĂĆ äżŐäÿ■ä¡äåřĘå■ęåĹŕäÿĂäžŻæŰřçŽĎæŁĂåůğåŠŇæŁĂæIJŕïijŇåźűåž

2.5 è£ŹæIJňäźęäÿ■éĂĆåŘĹèřĄ

è£ŹæIJňäźęäÿ■éĂĆåŘĹ Python çŽĎåĹİå■ęèĂĚãĂĆäžŃåđäÿŁïijŇæIJňäźęåĄĞåŐŽèŕżèĂĚåĚůæIJĽ
Python æŢźçÍŇæĹŮåĚěéŮÍäźęçś■äÿ■æĽĂæŢźæŐĹçŽĎå§žçąĂç§ěèŕĘãĂĆ
æIJňäźęäź§äÿ■æŸŕéĆčçğ■å£ńéĂ§åŘĆèĂČæĽŃåĘŃ ïijĹä¿ŃåęĆå£ńéĂ§æ§ěèŕćæ§ŘäÿłæĺąåİŮäÿŃçŽĎæ§Řä
æIJňäźęæŮĺåIJÍèĄŽçĎęåĞääÿłæIJĂéĞ■èęĄçŽĎäÿżéćŸïijŇæijŤçď'žåĞäçğ■åŘŕèČ¡çŽĎèğčåĘşæŰźæąĹïijŇ
æŘŘä¿ŻäÿĂäÿłèůşæİ£åijŢåŕijèŕżèĂĚè£ŻåĚěäÿĂäžŻæŻť'éńŸçžğçŽĎåĘĚåőźïijĹè£ŹäžŻåŘŕäżěåIJÍç¡ŚäÿŁæĹ

2.6 åIJÍçž£çď'žä¿ŃäżčçăĄ

æIJňäźęåĞääźŐæĽĂæIJĽæžŘäżčçăĄåİĞåŘŕäżěåIJÍ http://github.com/dabeaz/python-cookbook äÿŁéİćæĽ'¿åĹřãĂĆ ä¡IJèĂĚæňćè£ŐåŘĎä¡■èŕżèĂĚä£őæ■č bugïijŇæŤźè£ŻäżčçăĄåŠŇèŕĎèőžãĂĆ

2.7 ä¡£çŤÍçď žä¿Ńäžčçă Ą

æIJňäźęářśæŸŕåÿőåŁľä¡ăáőŇæĹŘä¡ăçŽĎáůěä¡IJçŽĎãĂĆ
äÿĂèĹňæĺěèőšïijŇåŘłèęĄæŸŕæIJňäźęäÿŁéĺćçŽĎçď žä¿Ńäžčçă Ąïij Ňä¡ăéČ¡åŘŕäžěéŽŘæŮűæŃ£è£ĞåŐžåIJÍä
éŹď éĺđä¡ăä¡£çŤÍäžĘáď ğéĞŘçŽĎäžčçă Ąïij ŇåŘęåĹŹäÿ■éIJĂèęĄåŘŚæĹŚäžňçŤşèŕůèőÿåŘŕãĂĆ
ä¿ŃäęĆïijŇä¡£çŤÍåĞääÿłäžčçă AçĽĞæőţåŐžáőŇæĹŘäÿĂäÿłçÍŃäžŘäÿ■éIJĂèęĄèőÿåŘŕïijŇèť ľ å■ŰæĹŰèĂĚ
åijŢçŤÍæIJňäźęäŠŇçď žä¿Ńäžčçă ĄáŐžç¡ŚäÿŁäŽđç■ŤäÿĂäÿłéŮőéćŸäÿ■éIJĂèęĄèőÿåŘŕïijŇä¡ĘæŸŕåŘĹäźűä

æĹŚäžňäÿ■äijŽèęĄæśĆä¡ăæůžäŁääžčçă AçŽĎäĞžäď ĎïijŇä¡ĘæŸŕäęĆæđIJä¡ăè£ŹäźĹäĄŽäžĘïijŇæĹŚäź
åijŢçŤÍéĂŽäÿÿäŇĚäŘńæăĞéćŸïijŇä¡IJèĂĚïijŇäĞžçĽĹçď ¿ïijŇISBNãĂĆ ä¿ŃäęĆïijŽ*Python Cookbook*, 3rd edition, by David Beazley and Brian K. Jones (OâĂŹReilly). Copyright 2013 David Beazley and Brian Jones, 978-1-449-34037-7.

äęĆæđIJä¡ăèğĽä¿Ůä¡äářźçď žä¿Ńäžčçă AçŽĎä¡£çŤÍèűĚäĞžäžĘäŘĹçŘĘä¡£çŤÍæĹŰèĂĚäÿŁè£řäĹŮäĞžç
èŕůéŽŘæŮűèĄŤçşżæĹŚäžňïijŇæĹŚäžňçŽĎéĆőçőśæŸŕ permissions@oreilly.comãĂĆ

2.8 èĄŤçşżæĹŚäžň

èŕůářĘäĚşäžŐæIJňäźęçŽĎèŕĎèőžäŠŇéŮőéćŸäŘŚéĂĄçżŹäĞžçĽĹçď ¿ïijŽ

OâĂŹReilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

æĹŚäžňäÿžæIJňäźęäžžçńŃäžĘäÿĂäÿłç¡Śéąţïij Ň åĚűäÿ■äŇĚäŘńäŃŸèŕŕèąĺïijŇçď žä¿ŃäŠŇäÿĂäžŻäĚűä
äŘŕäžěéĂŽè£ĞéŞ¿æŐě http://oreil.ly/python_cookbook_3e èő£éŮőãĂĆ

äĚşäžŐæIJňäźęçŽĎäžžèőőäŠŇæŁĂæIJŕæĂğéŮőéćŸïijŇèŕůäŘŚéĂĄéĆőäžűèĞşïijŽ bookquestions@oreilly.com

äĚşäžŐæĹŚäžňçŽĎäźęçś■ïijŇèőĺèőžäijŽïijŇæŰŕéŮżçŽĎæŻť äď Žä£ąæĄŕïijŇ
èŕůèő£éŮőæĹŚäžňçŽĎç¡ŚçńŹïijŽ http://www.oreilly.com

äIJÍ Facebook äÿŁæĽ¿äĹŕæĹŚäžňïijŽhttp://facebook.com/oreilly

äIJÍ Twitter äÿŁäĚşæşĺæĹŚäžňïijŽhttp://twitter.com/oreillymedia

äIJÍ YouTube äÿŁèğĆçIJŃæĹŚäžňïijŽhttp://www.youtube.com/oreillymedia

2.9 èĞťèřć

æĹŚäžňèąůå£ČæĎ§èřćæIJňäźęçŽĎæŁĂæIJŕæăąåőąäžžåŚŸ Jake Vanderplasïij Ňrobert Kern åŠŇ Andrea Crotti éİđåÿÿæIJL'çŤĺçŽĎèřĎèőžåŠŇåžžèőőïijŇ è£ŸæIJL' Python çď¿åŇžçŽĎåÿőåŁľåŠŇéijŞåŁśãĂĆæĹŚäžňåŘŇæăůæĎ§èřćäÿŁäÿĂäÿłçL'ĹæIJňçŽĎçijŰè¿Ś Alex MartelliïijŇAnna Ravenscroft åŠŇ David AscherãĂĆ
åř¡çőąè£ŹäÿłçL'ĹæIJňæŸŕæŰřåĹŻä¡IJçŽĎïijŇä¡ĘæŸŕåL'■äÿĂäÿłçL'ĹæIJňäÿžæIJňäźęæŘŘä¿ŻäžĘäÿĂäÿłæŇ
æIJĂåŘŐäź§æŸŕæIJĂéĞ■èęĄçŽĎïijŇæĹŚäžňèęĄæĎ§èřćæL'ĂæIJL'æŮľæIJ§écĎèğĹçL'ĹæIJňçŽĎèřżèĂĚïij

3 çňňäÿĂçńăïijŽæŢřæ■őçżŞæđĎåŠŇçőŮæşŢ

Python æŘŘä¿ŻäžĘåď'ğéĞŘçŽĎåĘĚç¡őæŢřæ■őçżŞæđĎïijŇåŇĚæŃňåĹŮèąĺïijŇéŻĘåŘĹäžěåŘŁå■ŮåĚ
ä¡ĘæŸŕïijŇæĹŚäžňäź§äijŽçżŘåÿÿçćřåĹřåĹřèřÿåęĆæ§ěèřćïijŇæŐŠåžŘåŠŇè£ĞæżďçL'ç■L'è£ŹäžŻæŹőéĄ■
åŻăæ■ď'ïijŇè£ŹäÿĂçńăçŽĎçŻőçŽĎářśæŸŕèőĺèőžè£ŹäžŻæŕŤè¿Čåÿÿèğ ĄçŽĎéŮőécŸåŠŇçőŮæşŢãĂĆ
åŘęåď'ŰïijŇæĹŚäžňäź§äijŽçżŹåĞžåIJĺéŻĘåŘĹæĺąåİŮ `collections`
å¡Şäÿ■æŞ■ä¡IJè£ŹäžŻæŢřæ■őçżŞæđĎçŽĎæŰźæşŢãĂĆ

3.1 1.1 èğčåŐŃåžŘåĹŮèţŃåĂijçżŹåď'ŽäÿłåŘŸéĞŘ

éŮőécŸ

çŐřåIJĺæIJL'äÿĂäÿłåŇĚåŘń N äÿłåĚČçť'äçŽĎåĚČçżĎæĹŰèĂĚæŸŕåžŘåĹŮïijŇæĂŐæăůåřĘåőČéĞŇéİ
N äÿłåŘŸéĞŘïij§

èğčåĘşæŰźæąĹ

äżżä¡ŢçŽĎåžŘåĹŮïijĹæĹŰèĂĚæŸŕåŘŕè£■äżčåŕźèśąïijL'åŘŕäżěéĂŽè£ĞäÿĂäÿłçőĂå■ŢçŽĎèţŃåĂijèŕ■
åŤŕäÿĂçŽĎåL'■æŘŘåřśæŸŕåŘŸéĞŘçŽĎæŢřéĞŘå£Ĕéążèů§åžŘåĹŮåĚČçť'äçŽĎæŢřéĞŘæŸŕäÿĂæăůçŽĎã

äžčçăĄçď'žä¿ŃïijŽ

```
>>> p = (4, 5)
>>> x, y = p
>>> x
4
>>> y
5
>>>
>>> data = [ 'ACME', 50, 91.1, (2012, 12, 21) ]
>>> name, shares, price, date = data
>>> name
'ACME'
>>> date
(2012, 12, 21)
```

(continues on next page)

(continued from previous page)

```
>>> name, shares, price, (year, mon, day) = data
>>> name
'ACME'
>>> year
2012
>>> mon
12
>>> day
21
>>>
```

åęĆæđIJåŘŸéĞŘäÿłæŢřåŠŇåžŘåĹŮåĚČçť ăçŽĎäÿłæŢřäÿ■åŇźéĚ■ïijŇäijŽäžğçŤ§äÿĂäÿłåijĆåÿÿãĂĆ

äżčçăĄçď žä¿ŃïijŽ

```
>>> p = (4, 5)
>>> x, y, z = p
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
ValueError: need more than 2 values to unpack
>>>
```

èőÍèőž

åőđéŹĚäÿŁïijŇè£Źçğ■èğčåŐŃèţŃåĂijåŘŕäżěçŤÍåIJÍäżżä¡ŢåŘŕè£■äżčåŕźèśąäÿŁéİćïijŇèĂŇäÿ■äżĚäżĚæ
åŇĚæŃňå■ŮçňęäÿšïijŇæŰĞäżűåŕźèśąïijŇè£■äżčåŹÍåŠŇçŤ§æĹŘåŹÍãĂĆ

äżčçăĄçď žä¿ŃïijŽ

```
>>> s = 'Hello'
>>> a, b, c, d, e = s
>>> a
'H'
>>> b
'e'
>>> e
'o'
>>>
```

æIJL'æŮűåĂŹïijŇä¡ăåŘŕèČ¡åŘłæČşèğčåŐŃäÿĂéČÍåĹĘïijŇäÿćåijČåĚűäżŰçŽĎåĂijãĂĆåŕźäžŐè£Źçğ■a
Python åźűæšąæIJL'æŘŘä¿ŻçL'źæőŁçŽĎèŕ■æşŢãĂĆ ä¡ĘæŸŕä¡ăåŘŕäżěä¡£çŤÍäżżæĎŘåŘŸéĞŘåŘ■åŐżå■ăä¡

äżčçăĄçď žä¿ŃïijŽ

```
>>> data = [ 'ACME', 50, 91.1, (2012, 12, 21) ]
>>> _, shares, price, _ = data
>>> shares
50
>>> price
```

(continues on next page)

(continued from previous page)

```
91.1
>>>
```

ä¡ăå£Ěéążä£ÌèŕĄä¡ăéĂL'çŤĺçŽĎéĆčäžŻå■ăä¡■åŔŸéĞŔåŘ■åIJĺåĚűäżŰåIJřæŰźæšąèćńä¡£çŤĺåĹřãĂĆ

3.2 1.2 èğčåŐŃåŔŕè£■äžčåŕźèśąèţŃåĂijçżŹåďŽäÿłåŔŸéĞŔ

éŮőéćŸ

åęĆæđIJäÿĂäÿłåŔŕè£■äžčåŕźèśąçŽĎåĚČçť ääÿłæŢřèűĚè£ĞåŔŸéĞŔäÿłæŢřæŮűïijŃäijŽæŁŻåĞžäÿĂäÿł
`ValueError` ãĂĆ éĆčäźĹæĂŐæăůæL'■èČ¡äžŐè£ŹäÿłåŔŕè£■äžčåŕźèśąäÿ■èğčåŐŃåĞž N
äÿłåĚČçť äåĞžæÌěïij§

èğčåĘşæŰźæąĹ

Python çŽĎæŸ§åŔůèąĺè¿¿åijŔåŔŕäžěçŤĺæÌěèğčåĘşè£ŹäÿłéŮőéćŸãĂĆæŕŤåęĆïijŃä¡äåIJĺå■ęäźääÿĂéŮĺ
ä¡äæČşçż§èőąäÿŃåőűåž■ä¡IJäÿŽçŽĎåźşåÌĞæĹŘçżl'ïijŃä¡ĘæŸŕæŐŠéŹď æŐL'çňňäÿĂäÿłåŠŃæIJĂåŘŐäÿĂä
ä¡ĘåęĆæđIJæIJL' 24 äÿłåŚćïij§è£ŹæŮűåĂŹæŸ§åŔůèąĺè¿¿åijŔåřśæť¿äÿŁçŤĺåIJžäžĘïijŽ

```
def drop_first_last(grades):
    first, *middle, last = grades
    return avg(middle)
```

åŔęåď ŰäÿĂçğ■æČĚåĘţïijŃåĄĞèő¿ä¡äçŐřåIJĺæIJL'äÿĂäžŻçŤĺæĹůçŽĎèőřå¡ŢåĹŮèąĺïijŃæŕŘæÌąèőřå¡Ţ
ä¡äåŔŕäžěåČŔäÿŃéÌćè£ŹæăůåĹĘèğčè£ŹäžŻèőřå¡ŢïijŽ

```
>>> record = ('Dave', 'dave@example.com', '773-555-1212', '847-555-
↪1212')
>>> name, email, *phone_numbers = record
>>> name
'Dave'
>>> email
'dave@example.com'
>>> phone_numbers
['773-555-1212', '847-555-1212']
>>>
```

åĂijå¿ŮæşĺæĎŔçŽĎæŸŕäÿŁéÌćèğčåŐŃåĞžçŽĎ `phone_numbers`
åŔŸéĞŔæřÿè£IJéČ¡æŸŕåĹŮèąĺçśżåđŃïijŃäÿ■çőąèğčåŐŃçŽĎçŤţèŕÌåŘůçăĄæŢřéĞŔæŸŕåď ŽåřŚïijĹåŃĚæŃ
0 äÿłïijL'ãĂĆ æL'Ääžěïij Ńäżżä¡Ţä¡£çŤĺåĹř `phone_numbers`
åŔŸéĞŔçŽĎäžčçăĄåřśäÿ■éIJĂèęĄåĄŽåď Žä¡ŹçŽĎçśżåđŃæčĂæ§ěåŐżçąőèőď åőČæŸŕåŘęæŸŕåĹŮèąĺçśżåd

æŸ§åŔůèąĺè¿¿åijŔäź§èČ¡çŤĺåIJĺåĹŮèąĺçŽĎåijĂåğŃéČĺåĹĘãĂĆæŕŤåęĆïijŃä¡äæIJL'äÿĂäÿłåĚňåŔŸåL'■
8 äÿłæIJĺéŤĂåŤőæŢřæ■őçŽĎåžŘåĹŮïijŃ ä¡ĘæŸŕä¡äæČşçIJŃäÿŃæIJĂè£ŚäÿĂäÿłæIJĹæŢřæ■őåŠŃåL'■éÌć
7 äÿłæIJĹçŽĎåźşåÌĞåĂijçŽĎåŕźæŕŤãĂĆä¡äåŔŕäžěè£ŹæăůåĄŽïijŽ

```
*trailing_qtrs, current_qtr = sales_record
trailing_avg = sum(trailing_qtrs) / len(trailing_qtrs)
return avg_comparison(trailing_avg, current_qtr)
```

äÿŃéİćæŸŕåIJÍ Python èğčéĞŁåŹÍäÿ■æĽğèąŃçŽĎçżŞæđIJïijŽ

```
>>> *trailing, current = [10, 8, 7, 1, 9, 5, 10, 3]
>>> trailing
[10, 8, 7, 1, 9, 5, 10]
>>> current
3
```

èőÍèőž

æĽľåśŢçŽĎè£■äžčèğčåŐŃèŕ■æşŢæŸŕäÿŞéŮÍäÿžèğčåŐŃäÿ■çąőåőŽäÿłæŢřæĹŰäżżæĎŔäÿłæŢřåĚČçť
éĂŽåÿÿïijŃè£ŹäžŽåŘŕè£■äžčåŕźèśąçŽĎåĚČçť ăçżŞæđĎæIJĽçąőåőŽçŽĎèğĎåĹŹïijĹæŕŤåęĆçňň
1 äÿłåĚČçť ăåŘŐéİćéČ¡æŸŕçŤţèŕİåŘůçăĄïijĽïijŃ æŸ§åŘůèąĺè¿¿åijŘèőľ åijĂåŘŚäžžåŚŸåŘŕäżěå¿ĹåőźæŸŞç
èĂŃäÿ■æŸŕéĂŽè£ĞäÿĂäžŽæŕŤè¿Čåď■æİĆçŽĎæĽŃæőţåŐżèŐůåŘŰè£ŹäžŽåĚşèĄŤçŽĎåĚČçť ăåĂijãĂĆ

åĂijå¿ŮæşĺæĎŘçŽĎæŸŕïijŃæŸ§åŘůèąĺè¿¿åijŘåIJÍè£■äžčåĚČçť ääÿžåŘŕåŘŸéŢ£åĚČçżĎçŽĎåžŘåĹŮæ
æŕŤåęĆïijŃäÿŃéİćæŸŕäÿĂäÿłåÿęæIJĽæăĞç■¿çŽĎåĚČçżĎåžŘåĹŮïijŽ

```
records = [
    ('foo', 1, 2),
    ('bar', 'hello'),
    ('foo', 3, 4),
]

def do_foo(x, y):
    print('foo', x, y)

def do_bar(s):
    print('bar', s)

for tag, *args in records:
    if tag == 'foo':
        do_foo(*args)
    elif tag == 'bar':
        do_bar(*args)
```

æŸ§åŘůèğčåŐŃèŕ■æşŢåIJÍlå■ŮçňęäÿšæŞ■ä¡IJçŽĎæŮűåĂŹäź§äijŽå¿ĹæIJĽçŤÍïijŃæŕŤåęĆå■Ůçňęäÿšç

äžčçăĄçď žä¿ŃïijŽ

```
>>> line = 'nobody:*:-2:-2:Unprivileged User:/var/empty:/usr/bin/
↪false'
>>> uname, *fields, homedir, sh = line.split(':')
>>> uname
'nobody'
```

(continues on next page)

(continued from previous page)

```
>>> homedir
'/var/empty'
>>> sh
'/usr/bin/false'
>>>
```

æIJL'æŮűåĂŹïijŇä¡ăæČşèğčåŐŃäÿĂäžŻåĚČçť'ăåŘŐäÿćåijČåőČäžňïijŇä¡ăäÿ■èČ¡çőĂå■Ţåřśä¡£çŤÍ
* ïijŇ ä¡ĘæŸŕä¡ăåŘŕäżěä¡£çŤÍäÿĂäÿłæŹőéĂŽçŽĎåž§åijČåŘ■çğřïijŇæŕŤåęĆ _ æĹŰèĂĚ
ign ïijĹignoreïijL'ãĂĆ

äžčçăĄçď'žä¿ŃïijŽ

```
>>> record = ('ACME', 50, 123.45, (12, 18, 2012))
>>> name, *_, (*_, year) = record
>>> name
'ACME'
>>> year
2012
>>>
```

åIJÍå¿Ĺåď'ŽåĞ¡æŢřåijŘèŕ■èÍĂäÿ■ïijŇæŸ§åŘůèğčåŐŃèŕ■æşŢèů§åĹŮèąĺåď'ĎçŘĘæIJL'èőÿåď'ŽçŻÿäijijä
ä¡ăåŘŕäżěå¿Ĺåőźæ'ŸŞçŽĎåřĘåőČåĹĘåL'šæĹŘåL'■åŘŐäÿď'éČÍåĹĘïijŽ

```
>>> items = [1, 10, 7, 4, 5, 9]
>>> head, *tail = items
>>> head
1
>>> tail
[10, 7, 4, 5, 9]
>>>
```

åęĆæđIJä¡ăåď'§èĄłæŸŐçŽĎèŕÍïijŇè£ŸèČ¡çŤÍè£Źçğ■åĹĘåL'šèŕ■æşŢåŐżåůğåęŹçŽĎåőđçŐřéĂŠå¡ŠçőŮ

```
>>> def sum(items):
...     head, *tail = items
...     return head + sum(tail) if tail else head
...
>>> sum(items)
36
>>>
```

çĎűåŘŐïijŇçŤśäžŐèŕ■èÍĂåśĆéÍćçŽĎéŹŘåĹűïijŇéĂŠå¡Šåźűäÿ■æŸŕ Python
æŞĚéŢ£çŽĎãĂĆ åŻäæ■ď'ïijŇæIJĂåŘŐéĆčäÿłéĂŠå¡ŠæijŤçď'žäžĚäżĚæŸŕäÿłåě¡åěĞçŽĎæŐćçť'ćç¡ćäžĘïijŇå

3.3 1.3 ä£ÍçŢŹæIJĂåŘŐ N äÿłåĚČçť'ä

éŮőéćŸ

åIJÍè£■äžčæŞ■ä¡IJæĹŰèĂĚåĚűäżŰæŞ■ä¡IJçŽĎæŮűåĂŹïijŇæĂŐæăůåŘłä£ÍçŢŹæIJĂåŘŐæIJL'éŹŘåĞă

èğčåĘşæŰźæąĹ

ä£ÍçŢŹæIJĽéŹŘåŐĘåŘšèőřå¡Ţæ■čæŸŕ `collections.deque` åď ğæŸ¿èžńæĽŃçŽĎæŮűåĂŹãĂĆæŕŤåęĆïijŇäÿŃéÍćçŽĎäžčçăĄåIJÍåď ŽèąŇäÿŁéÍćåĄŽçőĂå■ŢçŽĎæŰĞæ åźűè£ŤåŻđåŇźéĚ■æĽĂåIJÍèąŇçŽĎæIJĂåŘŐNèąŇïijŽ

```
from collections import deque


def search(lines, pattern, history=5):
    previous_lines = deque(maxlen=history)
    for line in lines:
        if pattern in line:
            yield line, previous_lines
        previous_lines.append(line)

# Example use on a file
if __name__ == '__main__':
    with open(r'../../cookbook/somefile.txt') as f:
        for line, prevlines in search(f, 'python', 5):
            for pline in prevlines:
                print(pline, end='')
            print(line, end='')
            print('-' * 20)
```

èőÍèőž

æĹŚäžňåIJÍåĘŹæ§ěèŕćåĚČçť ăçŽĎäžčçăĄæŮűïijŇéĂŽåÿÿäijŽä¡£çŤÍåŇĚåŘń `yield` èąÍè¿¿åijŘçŽĎçŤ§æĹŘåŹÍåĞ¡æŢřïijŇäź§åřśæŸŕæĹŚäžňäÿŁéÍćçď žä¿ŃäžčçăĄäÿ■çŽĎéĆčæăůãĂĆ è£ŹæăůåŘŕäžěåřĘæŘIJçť ćè£ĞçÍŃäžčçăĄåŠŇä¡£çŤÍæŘIJçť ćçżŞæđIJäžčçăĄèğčèĂęãĂĆåęĆæđIJä¡ăè£Ÿäÿ■a 4.3 èŁĆãĂĆ

ä¡£çŤÍ `deque(maxlen=N)` æđĎéĂåĞ¡æŢřäijŽæŰřåžžäÿĂäÿłåŽžåőŽåď ğåřŘçŽĎéŸ§åĹŮãĂĆå¡ŞæŮ æIJĂèĂĄçŽĎåĚČçť ăäijŽèĞłåŁÍèćńçğžéŹď æŐĽãĂĆ

äžčçăĄçď žä¿ŃïijŽ

```
>>> q = deque(maxlen=3)
>>> q.append(1)
>>> q.append(2)
>>> q.append(3)
>>> q
deque([1, 2, 3], maxlen=3)
>>> q.append(4)
>>> q
deque([2, 3, 4], maxlen=3)
>>> q.append(5)
>>> q
deque([3, 4, 5], maxlen=3)
```

åř¡çőąä¡ääź§åŘŕäżěæĽŃåŁĺåIJĺäÿĂäÿłåĹŮèąĺäÿŁåőđçŐřè£ŹäÿĂçŽĎæŞ■ä¡IJïijĹæŕŤåęĆåćđåŁäãĂąåĹ

æŻť äÿĂèĹňçŽĎïijŇ `deque` çśżåŘŕäżěèćńçŤĺåIJĺäżżä¡Ţä¡äåŘłéIJĂèęĄäÿĂäÿłçőĂå■ŢéŸ§åĹŮæŢřæ■ő
åęĆæđIJä¡ääÿ■èő¿ç¡őæIJĂåď ğéŸ§åĹŮåď ğåřŘïijŇéĆčäźĹåřśäijŽå¿ŮåĹřäÿĂäÿłæŮäéŹŘåď ğåřŘéŸ§åĹŮïijŇ

äžčçăĄçď žä¿ŃïijŽ

```
>>> q = deque()
>>> q.append(1)
>>> q.append(2)
>>> q.append(3)
>>> q
deque([1, 2, 3])
>>> q.appendleft(4)
>>> q
deque([4, 1, 2, 3])
>>> q.pop()
3
>>> q
deque([4, 1, 2])
>>> q.popleft()
4
```

åIJĺéŸ§åĹŮäÿď çńŕæŘŠåĚěæĹŰåĹäéŹď åĚČçť äæŮűéŮť åď ■æİĆåžęéČ¡æŸŕ `O(1)` ïijŇåŇžåĹńäžŐåĹŮèąĺïijŇåIJĺåĹŮèąĺçŽĎåijĂåď ť æŘŠåĚěæĹŰåĹäéŹď åĚČçť äçŽĎæŮűéŮť åď ■æİĆåžęäÿž `O(N)` ãĂĆ

3.4 1.4 æ§ěæĽ¿æIJĂåď ğæĹŰæIJĂåřŘçŽĎ N äÿłåĚČçť ä

éŮőécŸ

æĂŐæäůäżŐäÿĂäÿłéŻĘåŘĹäÿ■èŐůå¿ŮæIJĂåď ğæĹŰèĂĚæIJĂåřŘçŽĎ N äÿłåĚČçť äåĹŮèąĺïij§

èğčåĘşæŰźæąĹ

heapq æĺąåİŮæIJĽäÿď äÿłåĞ¡æŢřïijŽ`nlargest()` åŠŇ `nsmallest()` åŘŕäżěåőŇç¿Őèğčå Ęşè£ŹäÿłéŮőécŸãĂĆ

```
import heapq
nums = [1, 8, 2, 23, 7, -4, 18, 23, 42, 37, 2]
print(heapq.nlargest(3, nums)) # Prints [42, 37, 23]
print(heapq.nsmallest(3, nums)) # Prints [-4, 1, 2]
```

äÿď äÿłåĞ¡æŢřéČ¡èČ¡æŐěåŘŮäÿĂäÿłåĚşéŤőå■ŮåŘĆæŢřïijŇçŤĺäžŐæŻť åď ■æİĆçŽĎæŢřæ■őçżŞæđĎä

```
portfolio = [
    {'name': 'IBM', 'shares': 100, 'price': 91.1},
```

(continues on next page)

(continued from previous page)

```
    {'name': 'AAPL', 'shares': 50, 'price': 543.22},
    {'name': 'FB', 'shares': 200, 'price': 21.09},
    {'name': 'HPQ', 'shares': 35, 'price': 31.75},
    {'name': 'YHOO', 'shares': 45, 'price': 16.35},
    {'name': 'ACME', 'shares': 75, 'price': 115.65}
]
cheap = heapq.nsmallest(3, portfolio, key=lambda s: s['price'])
expensive = heapq.nlargest(3, portfolio, key=lambda s: s['price'])
```

èŕŚèĂĚæşĺïijŽäÿŁéİćäžčçăĄåIJĺåŕźæŕŘäÿłåĚČçť ăè£ŻèąŇåŕźæŕŤçŽĎæŮűåĂŹïijŇäijŽäžě `price` çŽĎåĂijè£ŻèąŇæŕŤè¿ČãĂĆ

èőĺèőž

åęĆæđIJä¡ăæČşåIJĺäÿĂäÿłéŻĘåŘĹäÿ■æ§ěæĽ¿æIJĂåŕŘæĹŰæIJĂåď ğçŽĎ N äÿłåĚČçť ăïijŇåźűäÿŤ N åŕŘäžŐéŻĘåŘĹåĚČçť ăæŢŕéĞŘïijŇéĆčäźĹè£ŹäžŽåĞ¡æŢŕæŘŘä¿ŻäžĘå¿Ĺåě¡çŽĎæ åŽääÿžåIJĺåžŢåśĆåőđçŐŕéĞŇéİćïijŇéęŰåĚĹäijŽåĚĹåŕĘéŻĘåŘĹæŢŕæ■őè£ŻèąŇåăĘæŐŠåžŘåŘŐæŤ¿åĚěäÿ

```
>>> nums = [1, 8, 2, 23, 7, -4, 18, 23, 42, 37, 2]
>>> import heapq
>>> heap = list(nums)
>>> heapq.heapify(heap)
>>> heap
[-4, 2, 1, 23, 7, 2, 18, 23, 42, 37, 8]
>>>
```

åăĘæŢŕæ■őçżŞæđĎæIJĂéĞ■èęĄçŽĎçĽźå¿ĄæŸŕ `heap[0]` æřÿè£IJæŸŕæIJĂåŕŘçŽĎåĚČçť ăăĂĆåźűäÿŤåĽľä¡ŹçŽĎåĚČçť ăåŘŕäžěå¿ĹåőźæŸ ŞçŽĎéĂŽè£ĞèřČçŤĺ `heapq.heappop()` æŰźæşŢå¿ŮåĹřïijŇ èŕěæŰźæşŢäijŽåĚĹåŕĘçňňäÿĂäÿłåĚČçť ăåijźåĞžæİěïijŇçĎűåŘŐ O(log N)ïijŇN æŸŕåăĘåď ğåřŘïijĽăĂĆ æŕŤåęĆïijŇåęĆæđIJæČşèęĄæ§ěæĽ¿æIJĂåŕŘçŽĎ 3 äÿłåĚČçť ăïijŇä¡ăåŘŕäžěè£ŹæăůåĄŽïijŽ

```
>>> heapq.heappop(heap)
-4
>>> heapq.heappop(heap)
1
>>> heapq.heappop(heap)
2
```

å¡ŞèęĄæ§ěæĽ¿çŽĎåĚČçť ääÿłæŢŕçŻÿåŕźæŕŤè¿ČåŕŘçŽĎæŮűåĂŹïijŇåĞ¡æŢŕ `nlargest()` åŠŇ `nsmallest()` æŸŕå¿ĹåŘĹéĂĆçŽĎãĂĆ åęĆæđIJä¡äažĚäžĚæČşæ§ěæĽ¿åŤŕäÿĂçŽĎæIJĂåŕŘæĹŰæIJĂåď ğïijĹN=1ïijĽçŽĎåĚČçť ăçŽĎèŕİïijŇéĆčäźĹ `min()` åŠŇ `max()` åĞ¡æŢŕäijŽæŻť å£ńäžŽãĂĆ çśżäijijçŽĎïijŇåęĆæđIJ N çŽĎåď ğåřŘåŠŇéŻĘåŘĹåď ğåřŘæŐěè£ŚçŽĎæŮűåĂŹïijŇéĂŽåÿÿåĚĹæŐŠåžŘè£ŹäÿłéŻĘåŘĹçĎűåŘŐåĘ■ä¡ ïijĹ `sorted(items)[:N]` æĹŰèĂĚæŸŕ `sorted(items)[-N:]` ïijĽăĂĆ éIJĂèęĄåIJĺæ■čçąőåIJžåŘĹä¡£çŤĺåĞ¡æŢŕ `nlargest()` åŠŇ `nsmallest()` æĽ■èČ¡åŘŚæŇěåőČäžňçŽĎäijŸåŁ£ ïijĹåęĆæđIJ N å£ńæŐěè£ŚéŻĘåŘĹåď ğåřŘäžĘïijŇéĆčäźĹä¡£çŤĺæŐŠåžŘæŞ■ä¡IJäijŽæŻť åě¡äžŻïijĽăĂĆ

ář¡çőąä¡ăæšąæIJL’å£ĚèęĄäÿĂåőŽä¡£çŤÍè£ŹéĞŇçŽĎæŰźæşŢïijŇä¡ĘæŸŕåăĘæŢŕæ■őçżŞæđĎçŽĎåőđçŰ
å§žæIJňäÿŁåŘłèęĄæŸŕæŢŕæ■őçżŞæđĎåŠŇçőŮæşŢäźęçś■éĞŇéĺćéČ¡äijŽæIJL’æŘŘåŘŁåĹřãĂĆ
heapq æĺąåİŮçŽĎåőŸæŰźæŰĞæąčéĞŇéĺćäź§èŕęçżĘçŽĎäżŃçż■äžĘåăĘæŢŕæ■őçżŞæđĎåžŢåśĆçŽĎåőđçŰ

3.5 1.5 åőđçŐŕäÿĂäÿłäijŸåĚĹçžğéŸ§åĹŮ

éŮőéćŸ

æĂŐæăůåőđçŐŕäÿĂäÿłæŇL’äijŸåĚĹçžğæŐŠåžŘçŽĎéŸ§åĹŮïij§
åźűäÿŤåIJĺè£ŹäÿłéŸ§åĹŮäÿŁéĺćæŕŘæňą pop æŞ■ä¡IJæĂżæŸŕè£ŤåŽđäijŸåĚĹçžğæIJĂéńŸçŽĎéĆčäÿłåĚČçt

èğčåĘşæŰźæąĹ

äÿŃéĺćçŽĎçśżåĹl’çŤÍ heapq æĺąåİŮåőđçŐŕäžĘäÿĂäÿłçőĂå■ŢçŽĎäijŸåĚĹçžğéŸ§åĹŮïijŽ

```
import heapq

class PriorityQueue:
    def __init__(self):
        self._queue = []
        self._index = 0

    def push(self, item, priority):
        heapq.heappush(self._queue, (-priority, self._index, item))
        self._index += 1

    def pop(self):
        return heapq.heappop(self._queue)[-1]
```

äÿŃéĺćæŸŕåőČçŽĎä¡£çŤÍæŰźåijŘïijŽ

```
>>> class Item:
...     def __init__(self, name):
...         self.name = name
...     def __repr__(self):
...         return 'Item({!r})'.format(self.name)
...
>>> q = PriorityQueue()
>>> q.push(Item('foo'), 1)
>>> q.push(Item('bar'), 5)
>>> q.push(Item('spam'), 4)
>>> q.push(Item('grok'), 1)
>>> q.pop()
Item('bar')
>>> q.pop()
Item('spam')
>>> q.pop()
```

(continues on next page)

(continued from previous page)

```
Item('foo')
>>> q.pop()
Item('grok')
>>>
```

äżŤçżĘèğĆåŕ§åŘŕäżěåŘŚçŐřïijŇçňňäÿĂäÿł pop() æŞ■ä¡IJè£ŤåŻđäijŸåĚĹçžğæIJĂéńŸçŽĎåĚČçť'äăĂ
åŘęåď'ŰæşĺæĎŘåĹřåęĆæđIJäÿď'äÿłæIJL'çĬĂçŻÿåŘŇäijŸåĚĹçžğçŽĎåĚČçť'äïijĹ foo åŠŇ
grok ïijL'ïijŇpop æŞ■ä¡IJæŇL'çĚğåőČäżňèćńæŘŠåĚěåĹřéŸ§åĹŮçŽĎéąžåžŘè£ŤåŻđçŽĎãĂĆ

èőĺèőž

è£ŹäÿĂåřŘèŁĆæĹŚäżňäÿżèęĄåĚşæşĺ heapq æĺąåİŮçŽĎä¡£çŤĺãĂĆ
åĞ¡æŢř heapq.heappush() åŠŇ heapq.heappop() åĹĘåĹńåIJĺéŸ§åĹŮ
_queue äÿŁæŘŠåĚěåŠŇåĹäéŹď'çňňäÿĂäÿłåĚČçť'äïijŇ åźűäÿŤéŸ§åĹŮ
_queue ä£ÌèŕĄçňňäÿĂäÿłåĚČçť'äæŃěæIJL'æIJĂéńŸäijŸåĚĹçžğïijĹ
1.4 èŁĆåůšçżŘèőĺèőžè£ĞĚ£ŹäÿłéŮőéćŸïijL'ãĂĆ heappop()
åĞ¡æŢřæĂżæŸŕè£ŤåŻđâĂİæIJĂåřŘçŽĎâĂİçŽĎåĚČçť'äïijŇè£ŹåŕśæŸŕä£ÌèŕĄéŸ§åĹŮpopæŞ■ä¡IJè£ŤåŻđæ
åŘęåď'ŰïijŇçŤśäžŐ push åŠŇ pop æŞ■ä¡IJæŮűéŮť'åď'■æİĆåžęäÿž
O(log N)ïijŇåĚűäÿ■ N æŸŕåăĘçŽĎåď'ğåŕŘïijŇåŽäæ■ď'åŕśçőŮæŸŕ N
å¿Ĺåď'ğçŽĎæŮűåĂŹåőČäżňè£ŘèąŇéĂ§åžęäź§ä¿ÌæŮğå¿Ĺå£ńãĂĆ

åIJĺäÿŁéİćäżčçăĄäÿ■ïijŇéŸ§åĹŮåŇĚåŘńäžĘäÿĂäÿł (-priority, index,
item) çŽĎåĚČçżĎãĂĆ äijŸåĚĹçžğäÿžèť'§æŢřçŽĎçŻőçŽĎæŸŕä¡£å¿ŮåĚČçť'äæŇL'çĚğäijŸåĚĹçžğäżŐéńŸ
è£Źäÿłèů§æŹőéĂŽçŽĎæŇL'äijŸåĚĹçžğäżŐä¡ŐåĹřéńŸæŐŠåžŘçŽĎåăĘæŐŠåžŘæĄřåůğçŻÿåŘ■ãĂĆ

index åŘŸéĞŘçŽĎä¡IJçŤĺæŸŕä£ÌèŕĄåŘŇç■L'äijŸåĚĹçžğåĚČçť'äçŽĎæ■čçąőæŐŠåžŘãĂĆ
éĂŽè£ĞĚ£Ìå■ŸäÿĂäÿłäÿ■æŰ■åćđåŁäçŽĎ index äÿŃæăĞåŘŸéĞŘïijŇåŕäżěçąőä£ÌåĚČçť'äæŇL'çĚğäőČäż
èĂŇäÿŤïijŇ index åŘŸéĞŘäź§åIJĺçŻÿåŘŇäijŸåĚĹçžğåĚČçť'äæŕŤè¿ČçŽĎæŮűåĂŹèţůåĹřéĞ■èęĄä¡IJçŤĺã

äÿžäžĘéŸŘæŸŐè£ŹäžŻïijŇåĚĹåĄĞäőŽ Item äőđä¿ŃæŸŕäÿ■æŤŕæŇĄæŐŠåžŘçŽĎïijŽ

```
>>> a = Item('foo')
>>> b = Item('bar')
>>> a < b
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
TypeError: unorderable types: Item() < Item()
>>>
```

åęĆæđIJä¡ää¡£çŤĺåĚČçżĎ (priority, item) ïijŇåŘłèęĄäÿď'äÿłåĚČçť'äçŽĎäijŸåĚĹçžğäÿ■åŘŇåŕ
ä¡ĘæŸŕåęĆæđIJäÿď'äÿłåĚČçť'ääijŸåĚĹçžğäÿĂæäůçŽĎèŕÌïijŇéĆčäźĹæŕŤè¿ČæŞ■ä¡IJåŕśäijŽèů§äźŇåĽ■äÿĂ

```
>>> a = (1, Item('foo'))
>>> b = (5, Item('bar'))
>>> a < b
True
>>> c = (1, Item('grok'))
>>> a < c
```

(continues on next page)

(continued from previous page)

```
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
TypeError: unorderable types: Item() < Item()
>>>
```

éĂŽè£ĞåijŢåĚěåŘęåď'ŰçŽĎ index åŘŸéĞŘçżĎæĹŘäÿL'åĚČçżĎ (priority, index, item) ïijŇåŕśèČ¡å¿Ĺåě¡çŽĎéĄ£åĚ■äÿŁéİćçŽĎéŤŹèŕŕïijŇ åŻääÿžäÿ■åŘŕèČ¡æIJL'äÿď'äÿłåĚČçť'äæIJL'çŻÿåŘŇçŽĎ index åĂijãĂĆPython åIJĺåĄŽåĚČçżĎæŕŤè¿ČæŮűåĂŹïijŇåęĆæđIJåL'■éİćçŽĎæŕŤè¿ČåůšçżŘåŘŕäžěçąőåőŽçżŞæđIJäžĘïijŇ åŘŐéİćçŽĎæŕŤè¿ČæŞ■ä¡IJåŕśäÿ■äijŽåŘŚçŤ§äžĘïijŽ

```
>>> a = (1, 0, Item('foo'))
>>> b = (5, 1, Item('bar'))
>>> c = (1, 2, Item('grok'))
>>> a < b
True
>>> a < c
True
>>>
```

åęĆæđIJä¡äæČşåIJĺåď'Žäÿłçž£çĺŃäÿ■ä¡£çŤĺåŘŇäÿĂäÿłéŸ§åĹŮïijŇéĆčäźĹä¡äéIJĂèęĄåćđåŁäéĂĆå¡ŞçŽ
åŘŕäžěæ§ěçIJŃ 12.3 åřŘèŁĆçŽĎä¿Ňå■ŘæijŤçď'žæŸŕæĂŐæäůåĄŽçŽĎãĂĆ

heapq æĺąåİŮçŽĎåőŸæŰźæŰĞæąčæIJL'æŻť'èŕęçżEçŽĎä¿Ňå■ŘçĺŃåžŘäžěåŘŁåŕźäžŐåäEçŘĘèőžåŘŁ

3.6 1.6 å■ŮåĚÿäÿ■çŽĎéŤőæŸäåřĎåď'Žäÿłå Ăij

éŮőéćŸ

æĂŐæäůåőđçŐřäÿĂäÿłéŤőåŕźåžŤåď'Žäÿłå Ăijç ŽĎå■ŮåĚÿïijĹäź§åŘń multidictïijL'ïij§

èğčåĘşæŰźæąĹ

äÿĂäÿłå■ŮåĚÿåŕśæŸŕäÿĂäÿłéŤőåŕźåžŤäÿĂäÿłå■ŢåĂijçŽĎæŸäåřĎãĂĆåęĆæđIJä¡äæČşèęĄäÿĂäÿłéŤőá
æŕŤåęĆåĹŮèąĺæĹŰèĂĚéŻĘåŘĹéĞŃéİćãĂĆæŕŤåęĆïijŇä¡äåŘŕäžěåČŘäÿŃéİćè£ŹæäůæđĎéĂäè£ŹæäůçŽĎå■

```
d = {
    'a' : [1, 2, 3],
    'b' : [4, 5]
}
e = {
    'a' : {1, 2, 3},
    'b' : {4, 5}
}
```

éĂĽæŃľä¡£çŤĺåĹŮèąĺè£ŸæŸŕéŻĘåŘĹåŔŰåĘşäžŐä¡äçŽĎåőđéŹĚéIJĂæśĆãĂĆåęĆæđIJä¡äæČşä£ÍæŇ
åęĆæđIJæČşåŐżæŐĽľéĞ■åď■åĚČçť äářšä¡£çŤĺéŻĘåŘĹïijĹåźűäÿŤäÿ■åĚşå£ČåĚČçť äçŽĎéąžåžŘéŮőéćŸïijĽ

ä¡ääŘŕäżěå¿ĹæŰźä¿£çŽĎä¡£çŤĺ `collections` æĺąåİŮäÿ■çŽĎ `defaultdict` æİěæđĎéĂäè£ŹæäůçŽĎå■ŮåĚÿãĂĆ `defaultdict` çŽĎäÿĂäÿłçĽźå¿ĄæŸŕäőČäijŽèĞłåŁĺåĹİåğŃåŇŰæŕŘäÿł `key` åĹŽåijĂäğŃåŕźåžŤçŽĎåĂijïijŇæĽĂäžěä¡ääŘłéIJĂèęĄåĚşæşĺæůżåŁäåĚČçť äæŞ■ä¡IJäžĘãĂĆæŕŤåęĆïijŽ

```
from collections import defaultdict

d = defaultdict(list)
d['a'].append(1)
d['a'].append(2)
d['b'].append(4)

d = defaultdict(set)
d['a'].add(1)
d['a'].add(2)
d['b'].add(4)
```

éIJĂèęĄæşĺæĎŘçŽĎæŸŕïijŇ `defaultdict` äijŽèĞłåŁĺäÿžåřĘèęĄèő£éŮőçŽĎéŤőïijĹåřśçőŮçŻőåĽ■ä
åęĆæđIJä¡ääźűäÿ■éIJĂèęĄè£ŹæäůçŽĎçĽźæĂğïijŇä¡ääŘŕäżěåIJĺäÿĂäÿłæŹőéĂŽçŽĎå■ŮåĚÿäÿŁä¡£çŤĺ `setdefault()` æŰźæşŢæİěäžčæŻ£ãĂĆæŕŤåęĆïijŽ

```
d = {} # äÿĂäÿłæŹőéĂŽçŽĎå■ŮåĚÿ
d.setdefault('a', []).append(1)
d.setdefault('a', []).append(2)
d.setdefault('b', []).append(4)
```

ä¡ĘæŸŕå¿Ĺåď ŽçÍŃåžŘåŚŸèğĽå¿Ů `setdefault()` çŤĺèţůæİěæIJĽçĆźåĹńæĽ■ãĂĆåŻääÿžæŕŘæňąèř `[]` ïijĽãĂĆ

èőĺèőž

äÿĂèĹňæİěèőšïijŇåĹŻåžžäÿĂäÿłåď ŽåĂijæŸäářĎå■ŮåĚÿæŸŕå¿ĹçőĂå■ŢçŽĎãĂĆä¡ĘæŸŕïijŇąęĆæđIJä
ä¡ääŘŕèČ¡äijŽåČŘäÿŃéĺćè£ŹæäůæİěäőđçŐŕïijŽ

```
d = {}
for key, value in pairs:
    if key not in d:
        d[key] = []
    d[key].append(value)
```

åęĆæđIJä¡£çŤĺ `defaultdict` çŽĎèŕİäžčçăĄářśæŻť åŁäçőĂæť ĄäžĘïijŽ

```
d = defaultdict(list)
for key, value in pairs:
    d[key].append(value)
```

è£ŹäÿĂåřŔèŁĆæĽĂèőĺèőžçŽĎéŮőéćŸèů§æŢřæ■őåď ĎçŘĘäÿ■çŽĎèőřå¡Ţå¡ŠçśżéŮőéćŸæIJĽåď ğçŽĎ
1.15 åřŔèŁĆçŽĎä¿Ńå■ŘãĂĆ

3.7 1.7 å■ŮåĚÿæŐŠåžŔ

éŮőéćŸ

ä¡ăæČşåĹŻåżžäÿĂäÿłå■ŮåĚÿïijŇåźűäÿŤåIJÍè£■äżčæĹŰåžŔåĹŮåŇŰè£Źäÿłå■ŮåĚÿçŽĎæŮűåĂŹèČ¡åd

èğčåĘşæŰźæąĹ

äÿžäžĘèČ¡æŐğåĹűäÿĂäÿłå■ŮåĚÿäÿ■åĚČçť ăçŽĎéąžåžŔïijŇä¡ăåŔŕäżěä¡£çŤÍ `collections` æĺąåİŮäÿ■çŽĎ `OrderedDict` çśżãĂĆ åIJÍè£■äżčæŞ■ä¡IJçŽĎæŮűåĂŹåőČäijŽä£İæŇĄåĚČçť ăèćńæŔŠåĚěæŮűçŽĎéąžåžŔïijŇçď žä¿ŃåęĆäÿŃïijŽ

```
from collections import OrderedDict

d = OrderedDict()
d['foo'] = 1
d['bar'] = 2
d['spam'] = 3
d['grok'] = 4
# Outputs "foo 1", "bar 2", "spam 3", "grok 4"
for key in d:
    print(key, d[key])
```

å¡Şä¡ăæČşèęĄæđĎåżžäÿĂäÿłåřĘæİěéIJĂèęĄåžŔåĹŮåŇŰæĹŰçijŰçăĄæĹŘåĚűäżŰæăijåijŔçŽĎæŸăåřŔ `OrderedDict` æŸŕéİđåÿÿæIJĽçŤÍçŽĎãĂĆ æŕŤåęĆïijŇä¡ăæČşçš¿çąőæŐğåĹűäżě JSON çijŰçăĄåŘŐå■ŮæőţçŽĎéąžåžŔïijŇä¡ăåŔŕäżěåĚĹä¡£çŤÍ `OrderedDict` æİěæđĎåżžè£ŹæăůçŽĎæŢřæ■őïijŽ

```
>>> import json
>>> json.dumps(d)
'{"foo": 1, "bar": 2, "spam": 3, "grok": 4}'
>>>
```

èőÍèőž

`OrderedDict` åĘĚéČÍçżť æŁď çİĂäÿĂäÿłæăźæ■őéŤőæŔŠåĚěéąžåžŔæŐŠåžŔçŽĎåŔŇåŘŠéŞ¿èąÍãĂĆ åőČäijŽèćńæŤ¿åĹřéŞ¿èąÍçŽĎåř¿éČÍãĂĆåŕźäžŐäÿĂäÿłåůšçżŔå■ŸåIJÍçŽĎéŤőçŽĎéĞ■åď ■èţŃåĂijäÿ■äijŽæŤ

éIJĂèęĄæşÍæĎŔçŽĎæŸŕïijŇäÿĂäÿł `OrderedDict` çŽĎåď ğåřŔæŸŕäÿĂäÿłæŹőéĂŽå■ŮåĚÿçŽĎäÿď å æĽĂäżěåęĆæđIJä¡äèęĄæđĎåżžäÿĂäÿłéIJĂèęĄåď ğéĞŔ `OrderedDict` åőđä¿ŃçŽĎæŢřæ■őçżŞæđĎçŽĎæŮűåĂŹïijĹæŕŤåęĆèŕżåŔŰ 100,000 èąŇ CSV æŢřæ■őåĹřäÿĂäÿł `OrderedDict` åĹŮèąÍäÿ■åŐżïijĽïijŇ éĆčäźĹä¡ăåřšå¿ŮäżŤçżĘæĬČèąąäÿĂäÿŃæŸŕåŘęä¡£çŤÍ `OrderedDict` åÿęæİěçŽĎåě¡åď ĎèęĄåď ğè£Ğéćİåď ŰåĘĚå■ŸæűĹèĂŮçŽĎå¡śåŞ■ãĂĆ

3.8 1.8 å■ŮåĚÿçŽĎè£ŘçőŮ

éŮőéćŸ

æĂŐæăůåIJÍæŢřæ■őå■ŮåĚÿäÿ■æĽğèąŇäÿĂäžŻèőąçőŮæŞ■ä¡IJïijĹæŕŤåęĆæśĆæIJĂåřŘåĂijãĂĄæIJĂå

èğčåĘşæŰźæąĹ

èĂČèŹŚäÿŇéİćçŽĎèĆąçĕĺåŘ■åŠŇäżůæăijæŸăåřĎå■ŮåĚÿïijŽ

```
prices = {
    'ACME': 45.23,
    'AAPL': 612.78,
    'IBM': 205.55,
    'HPQ': 37.20,
    'FB': 10.75
}
```

äÿžäžĘåŕźå■ŮåĚÿåĂijæĽğèąŇèőąçőŮæŞ■ä¡IJïijŇéĂŽåÿÿéIJĂèęĄä¡£çŤÍ `zip()` åĞ¡æŢřåĚĹåřĘéŤőåŠŇåĂijåŘ■è¡ňè£ĞæİĕãĂĆ æŕŤåęĆïijŇäÿŇéİćæŸŕæ§ĕæĽ¿æIJĂåřŘåŠŇæIJĂåď ğèĆąçĕĺ

```
min_price = min(zip(prices.values(), prices.keys()))
# min_price is (10.75, 'FB')
max_price = max(zip(prices.values(), prices.keys()))
# max_price is (612.78, 'AAPL')
```

çśżäijijçŽĎïijŇåŘŕäżĕä¡£çŤÍ `zip()` åŠŇ `sorted()` åĞ¡æŢřæİĕæŐŠåĹŮå■ŮåĚÿæŢřæ■őïijŽ

```
prices_sorted = sorted(zip(prices.values(), prices.keys()))
# prices_sorted is [(10.75, 'FB'), (37.2, 'HPQ'),
#                   (45.23, 'ACME'), (205.55, 'IBM'),
#                   (612.78, 'AAPL')]
```

æĽğèąŇè£ŹäžŻèőąçőŮçŽĎæŮűåĂŹïijŇéIJĂèęĄæşĺæĎŘçŽĎæŸŕ `zip()` åĞ¡æŢřåĹŻåżžçŽĎæŸŕäÿĂäÿłåŘłèČ¡èő£éŮőäÿĂæňąçŽĎè£■äżčåŹĺãĂĆ æŕŤåęĆïijŇäÿŇéİćçŽĎäžčçăĄåřśäijŽäžğçŤ§éŤŹèŕŕïijŽ

```
prices_and_names = zip(prices.values(), prices.keys())
print(min(prices_and_names)) # OK
print(max(prices_and_names)) # ValueError: max() arg is an empty␣
↪sequence
```

èőĺèőž

åęĆæđIJä¡ăåIJĺäÿĂäÿłå■ŮåĚÿäÿŁæĽğèąŇæŹőéĂŽçŽĎæŢřå■ęè£ŘçőŮïijŇä¡ăäijŽåŘŚçŐřåőČäżňäżĚäż

```
min(prices) # Returns 'AAPL'
max(prices) # Returns 'IBM'
```

è£ŹäÿłçżŞæđIJåźűäÿ■æŸŕä¡ăæČşèęĄçŽĎïijŇåŽääÿžä¡ăæČşèęĄåIJÍå■ŮåĚÿçŽĎåĂijéŻĘåŘĹäÿŁæĽ'ğèą
æĹŰèőÿä¡ăäijŽåřÌèŕŢçÌĂä¡£çŤÍå■ŮåĚÿçŽĎ values() æŰźæşŢæÌěèğčåĘşè£ŹäÿłéŮőéćŸïijŽ

```
min(prices.values()) # Returns 10.75
max(prices.values()) # Returns 612.78
```

äÿ■åźÿçŽĎæŸŕïijŇéĂŽåÿÿè£ŹäÿłçżŞæđIJåŘŇæăůäź§äÿ■æŸŕä¡ăæČşèęĄçŽĎãĂĆ
ä¡ăåŘŕèČ¡è£ŸæČşèęĄç§ěéĄŞåŕźåžŤçŽĎéŤőçŽĎä£ąæĄŕïijĹæŕŤåęĆéĆčçğ■èĆąçěÍäżůæăijæŸŕæIJĂä¡ŐçŽĎï

ä¡ăåŘŕäżěåIJÍ min() åŠŇ max() åĞ¡æŢřäÿ■æŘŘä¿Ż key
åĞ¡æŢřåŘĆæŢřæÌěèŐůåŘŰæIJĂåřŘåĂijæĹŰæIJĂåď'ğåĂijåŕźåžŤçŽĎéŤőçŽĎä£ąæĄŕãĂĆæŕŤåęĆïijŽ

```
min(prices, key=lambda k: prices[k]) # Returns 'FB'
max(prices, key=lambda k: prices[k]) # Returns 'AAPL'
```

ä¡ĘæŸŕïijŇåęĆæđIJè£ŸæČşèęĄå¿ŮåĹřæIJĂåřŘåĂijïijŇä¡ăåŘĹå¿ŮæĽ'ğèąŇäÿĂæňąæ§ěæĽ'¿æŞ■ä¡IJãĂ

```
min_value = prices[min(prices, key=lambda k: prices[k])]
```

åĽ■éÌćçŽĎ zip() åĞ¡æŢřæŰźæąĹéĂŽè£ĞåřĘå■ŮåĚÿâĂÌåŘ■è¡ňâĂÌäÿž
(åĂijïijŇéŤő) åĚČçżĎåžŘåĹŮæÌěèğčåĘşäžĘäÿŁè£řéŮőéćŸãĂĆ
å¡ŞæŕŤè¿Čäÿď'äÿłåĚČçżĎçŽĎæŮűåĂŹïijŇåĂijäijŽåĚĹè£ŻèąŇæŕŤè¿ČïijŇçĎűåŘŐæĽ'■æŸŕéŤőãĂĆ
è£ŹæăůçŽĎèŕÌä¡ăåřśèČ¡éĂŽè£ĞäÿĂæÌąçőĂå■ŢçŽĎèŕ■åŘěåřśèČ¡å¿Ĺè¡żæÌ¿çŽĎåőđçŐřåIJÍå■ŮåĚÿäÿŁçŽĎ

éIJĂèęĄæşĺæĎŘçŽĎæŸŕåIJÍèőąçőŮæŞ■ä¡IJäÿ■ä¡£çŤÍåĹřäžĘ (åĂijïijŇéŤő)
åŕźãĂĆå¡Şåď'Žäÿłåőđä¡ŞæŇěæIJĽçŻÿåŘŇçŽĎåĂijçŽĎæŮűåĂŹïijŇéŤőäijŽåĘşåőŽè£ŤåŻđçżŞæđIJãĂĆ
æŕŤåęĆïijŇåIJÍæĽ'ğèąŇ min() åŠŇ max() æŞ■ä¡IJçŽĎæŮűåĂŹïijŇåęĆæđIJæĄŕåůğæIJĂåřŘæĹŰæIJĂåď'

```
>>> prices = { 'AAA' : 45.23, 'ZZZ': 45.23 }
>>> min(zip(prices.values(), prices.keys()))
(45.23, 'AAA')
>>> max(zip(prices.values(), prices.keys()))
(45.23, 'ZZZ')
>>>
```

3.9 1.9 æ§ěæĽ'¿äÿď'å■ŮåĚÿçŽĎçŻÿåŘŇçĆź

éŮőéćŸ

æĂŐæăůåIJÍäÿď'äÿłå■ŮåĚÿäÿ■åŕżåŕżæĽ'¿çŻÿåŘŇçĆźïijĹæŕŤåęĆçŻÿåŘŇçŽĎéŤőãĂĄçŻÿåŘŇçŽĎåĂij

èğčåĘşæŰźæąĹ

èĂČèŹŚäÿŇéÌćäÿď'äÿłå■ŮåĚÿïijŽ

```
a = {
    'x' : 1,
    'y' : 2,
    'z' : 3
}

b = {
    'w' : 10,
    'x' : 11,
    'y' : 2
}
```

äÿžäžĘåŕżæĽ¿äÿďäÿłå■ŮåĚÿçŽĎçŻÿåŘŇçĆźïijŇåŘŕäżěçőĂå■ŢçŽĎåIJÍäÿďå■ŮåĚÿçŽĎ keys() æĹŰèĂĚ items() æŰźæşŢè£ŤåŻđçżŞæđIJäÿŁæĽ'ğèąŇéŻĘåŘĹæŞ■ä¡IJãĂĆæŕŤåęĆïijŽ

```
# Find keys in common
a.keys() & b.keys() # { 'x', 'y' }
# Find keys in a that are not in b
a.keys() - b.keys() # { 'z' }
# Find (key,value) pairs in common
a.items() & b.items() # { ('y', 2) }
```

è£ŹäžŻæŞ■ä¡IJäź§åŘŕäżěçŤÍäžŐä£őæŤźæĹŰèĂĚè£ĞæżďÿåŮåĚÿåĚČçťäãĂĆ æŕŤåęĆïijŇåĄĞåęĆä¡äæČşäżěçŐŕæIJĽ'å■ŮåĚÿæđĎéĂääÿĂäÿłæŐŠéŹďÿåĞääÿłæŇĞåőŽéŤőçŽĎæŰŕå■ŮåĚÿ äÿŇéÍćåĹĽçŤÍå■ŮåĚÿæŐÍåŕijæÍěåőđçŐŕè£ŹæäůçŽĎéIJĂæśĆïijŽ

```
# Make a new dictionary with certain keys removed
c = {key:a[key] for key in a.keys() - {'z', 'w'}}
# c is {'x': 1, 'y': 2}
```

èőÍèőž

äÿĂäÿłå■ŮåĚÿåŕśæŸŕäÿĂäÿłéŤőéŻĘåŘĹäÿŐåĂijéŻĘåŘĹçŽĎæŸäåŕĎåĚşçşżãĂĆ å■ŮåĚÿçŽĎ keys() æŰźæşŢè£ŤåŻđäÿĂäÿłåśŢçŐŕéŤőéŻĘåŘĹçŽĎéŤőèğĘåŻ¿åŕźèśąãĂĆ éŤőèğĘåŻ¿çŽĎäÿĂäÿłå¿ĹåŕŚèćńäžĘèğčçŽĎçĽźæĂğåŕśæŸŕåőČäžňäź§æŤŕæŇĄéŻĘåŘĹæŞ■ä¡IJïijŇæŕŤåęĆ æĽĂäżěïijŇåęĆæđIJä¡äæČşåŕźéŻĘåŘĹçŽĎéŤőæĽ'ğèąŇäÿĂäžŻæŹőéĂŽçŽĎéŻĘåŘĹæŞ■ä¡IJïijŇåŘŕäżěçŻť setãĂĆ

å■ŮåĚÿçŽĎ items() æŰźæşŢè£ŤåŻđäÿĂäÿłåŇĚåŘń (éŤőïijŇåĂij) åŕźçŽĎåĚČçť äèğĘåŻ¿åŕźèśąãĂĆ è£ŹäÿłåŕźèśąåŘŇæăůäź§æŤŕæŇĄéŻĘåŘĹæŞ■ä¡IJïijŇåźűäÿŤåŘŕäżěèćńçŤ

åř¡çőąå■ŮåĚÿçŽĎ values() æŰźæşŢäź§æŸŕçśżäijijïijŇä¡ĘæŸŕåőČåźűäÿ■æŤŕæŇĄè£ŹéĞŇäżŃçż■ç æ§Řçğ■çÍŃåžęäÿŁæŸŕåŻääÿžåĂijèğĘåŻ¿äÿ■èČ¡ä£ÌèŕĄæĽ'ĂæIJĽçŽĎåĂijäžŠäÿ■çŻÿåŘŇïijŇè£ŹæăůäijŽåŕ äÿ■è£ĞïijŇåęĆæđIJä¡äçąňèęĄåIJĺåĂijäÿŁéÍćæĽ'ğèąŇè£ŹäžŻéŻĘåŘĹæŞ■ä¡IJçŽĎèŕÍïijŇä¡äåŘŕäżěåĚĹåřĘåĂ setïijŇçĎűåŘŐåĘ■æĽ'ğèąŇéŻĘåŘĹè£ŘçőŮåŕśèąŇäžĘãĂĆ

3.10 1.10 åĹăéŹďåžŘåĹŮçŻÿåŘŇåĚČçťäåźűä£ÌæŇĄéąžåžŘ

éŮőéćŸ

æĂŐæăůåIJÍäÿĂäÿłåžŘåĹŮäÿŁéÌćä£ÌæŇĄåĚČçťäéąžåžŘçŽĎåŘŇæŮűæűĹéŹďéĞ■åď■çŽĎåĂijïij§

èğčåĘşæŰźæąĹ

åęĆæđIJåžŘåĹŮäÿŁçŽĎåĂijéČ¡æŸŕ `hashable` çśżåđŃïijŇéĆčäźĹåŘŕäżěå¿ĹçőĂå■ŢçŽĎåĹľçŤĺéŻĘ

```
def dedupe(items):
    seen = set()
    for item in items:
        if item not in seen:
            yield item
            seen.add(item)
```

äÿŃéÌćæŸŕä¡£çŤÍäÿŁè£řåĞ¡æŢřçŽĎä¿Ńå■ŘïijŽ

```
>>> a = [1, 5, 2, 1, 9, 1, 5, 10]
>>> list(dedupe(a))
[1, 5, 2, 9, 10]
>>>
```

è£ŹäÿłæŰźæşŢäżĚäżĚåIJÍåžŘåĹŮäÿ■åĚČçťääÿž `hashable` çŽĎæŮűåĂŹæĽ■çőąçŤÍãĂĆ åęĆæđIJä¡äæČşæúĹéŹďåĚČçťääÿ■åŘŕåŞĹåÿŇïijĹæŕŤåęĆ `dict` çśżåđŃïijĽçŽĎåžŘåĹŮäÿ■éĞ■åď■åĚČçťäçŽĎèŕÌïijŇä¡äéIJĂèęĄåřĘäÿŁè£řäżčçăĄçÍ■å¿őæŤźåŘŸäÿĂ

```
def dedupe(items, key=None):
    seen = set()
    for item in items:
        val = item if key is None else key(item)
        if val not in seen:
            yield item
            seen.add(val)
```

è£ŹéĞŃçŽĎkeyåŘĆæŢřæŇĞåőŽäżĘäÿĂäÿłåĞ¡æŢřïijŇåřĘåžŘåĹŮåĚČçťäè¡ňæ■ćæĹŘ `hashable` çśżåđŃãĂĆäÿŃéÌćæŸŕåőČçŽĎçŤÍæşŢçď žä¿ŃïijŽ

```
>>> a = [ {'x':1, 'y':2}, {'x':1, 'y':3}, {'x':1, 'y':2}, {'x':2, 'y
↪':4}]
>>> list(dedupe(a, key=lambda d: (d['x'],d['y'])))
[{'x': 1, 'y': 2}, {'x': 1, 'y': 3}, {'x': 2, 'y': 4}]
>>> list(dedupe(a, key=lambda d: d['x']))
[{'x': 1, 'y': 2}, {'x': 2, 'y': 4}]
>>>
```

åęĆæđIJä¡äæČşå§žäżŐå■Ţäÿłå■Ůæőţã ĂĄåśđæĂğæĹŰèĂĚæ§ŘäÿłæŻť åď ğçŽĎæŢřæ■őçżŞæđĎæÌěæűĺ

èőÍèőž

åęĆæđIJä¡ääžĚäžĚåřśæŸŕæČşæűĹéŹď'éĞ■åď'■åĚČçť'äïijŇéĂŽåÿÿåŘŕäžěçőĂå■ŢçŽĎæđĎéĂääÿĂäÿłé

```
>>> a
[1, 5, 2, 1, 9, 1, 5, 10]
>>> set(a)
{1, 2, 10, 5, 9}
>>>
```

çĎűèĂŇïijŇè£Źçğ■æŰźæşŢäÿ■èČ¡çżť'æŁď'åĚČçť'äçŽĎéąžåžŘïijŇçŤ§æĹŘçŽĎçżŞæđIJäÿ■çŽĎåĚČçť'

åIJÍæIJňèŁĆäÿ■æĹŚäžňä¡£çŤÍäžĘçŤ§æĹŘåŹÍåĞ¡æŢřèőľæĹŚäžňçŽĎåĞ¡æŢřæŻť'åŁäéĂŽçŤÍïijŇäÿ■äžł
æŕŤåęĆïijŇåęĆæđIJåęĆæđIJä¡äæČşèŕżåŘŰäÿĂäÿłæŰĞäžűïijŇæűĹéŹď'éĞ■åď'■èąŇïijŇä¡äåŘŕäžěå¿ĹåőźæŸ

```
with open(somefile,'r') as f:
for line in dedupe(f):
    ...
```

äÿŁè£řkeyåĞ¡æŢřåŘĆæŢřæÍąäž£äžĘ `sorted()` , `min()` åŠŇ `max()` ç■L'åĘĚç¡őåĞ¡æŢřçŽĎçŻÿäijijåŁ§èČ¡ãĂĆ åŘŕäžěåŘĆèĂČ 1.8 åŠŇ 1.13 åřŘèŁĆäžĘèğčæŻť'åď'ŽãĂĆ

3.11 1.11 åŚ¡åŘ■åĹĞçL'Ğ

éŮőécŸ

åęĆæđIJä¡äçŽĎçÍŃåžŘåŇĚåŘńäžĘåď'ğéĞŘæŮäæşŢçŻť'èğĘçŽĎçąňçijŰçăĄåĹĞçL'ĞïijŇåźűäÿŤä¡äæČ

èğčåĘşæŰźæąĹ

åĄĞåőŽä¡äèęĄäžŐäÿĂäÿłèőřå¡ŢïijĹæŕŤåęĆæŰĞäžűæĹŰåĚűäžŰçśżäijijæäijåijŘïijL'äÿ■çŽĎæ§ŘäžŻåŽ

```
######  ␣
↪0123456789012345678901234567890123456789012345678901234567890'
record = '....................100 .......513.25 ..........'
cost = int(record[20:23]) * float(record[31:37])
```

äÿŐåĚűéĆčæäůåĘŹïijŇäÿžäžĂäźĹäÿ■æČşè£Źæäůå Ś¡åŘ■åĹĞçL'ĞåŚćïijŽ

```
SHARES = slice(20, 23)
PRICE = slice(31, 37)
cost = int(record[SHARES]) * float(record[PRICE])
```

åIJÍè£Źäÿłçl'ĹæIJňäÿ■ïijŇä¡äéĄ£åĚ■äžĘä¡£çŤÍåď'ğéĞŘéŽ¿äžěçŘĘèğčçŽĎçąňçijŰçăĄäÿŃæäĞäĂĆè£Ź

èőÍèőž

äÿĂèĹňæİěèőšïijŇäżčçăĄäÿ■åęĆæđIJåĞžçŐřåď ğéĞŘçŽĎçąňçijŰçăĄäÿŃæăĞäijŽä¡£å¿ŮäżčçăĄçŽĎåĺ
æŕŤåęĆïijŇåęĆæđIJä¡ăåŻđè£ĞæİěçIJŃçIJŃäÿĂåźť åĽ■ä¡ăåĘźçŽĎäżčçăĄïijŇä¡ăäijŽæŚÿçİĂèĎŚèćŃæČşéĆ
è£ŹæŸŕäÿĂäÿłå¿ĹçőĂå■ŢçŽĎèğčåĘşæŰźæąĹïijŇåőČèőľä¡ăæŻť åŁăæÿĚæŹřçŽĎèąĺè¿¿äżčçăĄçŽĎçŽőçŽĎ

åĘĚç¡őçŽĎ `slice()` åĞ¡æŢřåĹŻåżžäžĘäÿĂäÿłåĹĞçĽĞåŕźèśąãĂĆæĽĂæIJĽä¡£çŤĺåĹĞçĽĞçŽĎåIJřa

```
>>> items = [0, 1, 2, 3, 4, 5, 6]
>>> a = slice(2, 4)
>>> items[2:4]
[2, 3]
>>> items[a]
[2, 3]
>>> items[a] = [10,11]
>>> items
[0, 1, 10, 11, 4, 5, 6]
>>> del items[a]
>>> items
[0, 1, 4, 5, 6]
```

åęĆæđIJä¡ăæIJĽäÿĂäÿłåĹĞçĽĞåŕźèśąaïijŇä¡ăåŘŕäżěåĹĘåĹńèřČçŤĺåőČçŽĎ `a.start` , `a.stop` , `a.step` åśđæĂğæİěèŐůåŘŰæŻť åď ŽçŽĎä£ąæĄŕãĂĆæŕŤåęĆïijŽ

```
>>> a = slice(5, 50, 2)
>>> a.start
5
>>> a.stop
50
>>> a.step
2
>>>
```

åŘęåď ŰïijŇä¡ăè£ŸåŘŕäżěéĂŽè£ĞèřČçŤĺåĹĞçĽĞçŽĎ `indices(size)` æŰźæşŢåřĘåőČæŸăåřĎåĹřäÿĂäÿłåůšç§ěåď ğåřŘçŽĎåžŘåĹŮäÿŁãĂĆ è£ŹäÿłæŰźæşŢè£ŤåŻđäÿĂäÿłäÿĽåĚČçżĎ `(start, stop, step)` ïijŇæĽĂæIJĽçŽĎåĂijéČ¡äijŽèćńçijľ åřŘïijŇçŻť åĹřéĂĆåŘĹè£Źäÿłåůšç§ěåžŘåĹŮçŽĎè¿źçŢŇäÿžæ■ćãĂĆ è£ŹæăůïijŇä¡£çŤĺçŽĎæŮűářśäÿ■äijŽåĞžçŐř `IndexError` åijĆåÿÿãĂĆæŕŤåęĆïijŽ

```
>>> s = 'HelloWorld'
>>> a.indices(len(s))
(5, 10, 2)
>>> for i in range(*a.indices(len(s))):
...     print(s[i])
...
W
r
d
>>>
```

3.12 1.12 ážŕåĹŮäÿ■åĞžçŐřæňąæŢřæIJĂåďŽçŽĎåĚČçťă

éŮőéćŸ

æĂŐæăůæĽ¿åĞžäÿĂäÿłážŕåĹŮäÿ■åĞžçŐřæňąæŢřæIJĂåďŽçŽĎåĚČçťăåŚćïij§

èğčåĘşæŰźæąĹ

`collections.Counter` çśżåřśæŸŕäÿŞéŮĺäÿžè£ŹçśżéŮőéćŸèĂŇèő¿èőąçŽĎïijŇ åőČçŤŽèĞşæIJĽäÿĂäÿłæIJĽçŤĺçŽĎ `most_common()` æŰźæşŢçŻťæŐěçżŹäžĘä¡ăç■ŤæąĹãĂĆ

äÿžäžĘæijŤçďžïijŇåĚĹåĄĞèő¿ä¡ăæIJĽäÿĂäÿłå■Ţèŕ■åĹŮèąĺåźűäÿŤæČşæĽ¿åĞžåŞłäÿłå■Ţèŕ■åĞžçŐřé

```
words = [
    'look', 'into', 'my', 'eyes', 'look', 'into', 'my', 'eyes',
    'the', 'eyes', 'the', 'eyes', 'the', 'eyes', 'not', 'around',
↪'the',
    'eyes', "don't", 'look', 'around', 'the', 'eyes', 'look', 'into
↪',
    'my', 'eyes', "you're", 'under'
]
from collections import Counter
word_counts = Counter(words)
# åĞžçŐřéćŚçŐĞæIJĂéńŸçŽĎ3äÿłå■Ţèŕ■
top_three = word_counts.most_common(3)
print(top_three)
# Outputs [('eyes', 8), ('the', 5), ('look', 4)]
```

èőĺèőž

ä¡IJäÿžè¿ŞåĚěïijŇ `Counter` åŕźèśąåŔŕäżěæŐěåŔŮäżżæĎŔçŽĎçŤśåŔŕåŞĹåÿŇïijĹhashableïijĽåĚČ åIJĺåžŢåśĆåőđçŐřäÿŁïijŇäÿĂäÿł `Counter` åŕźèśąåřśæŸŕäÿĂäÿłå■ŮåĚÿïijŇåřĘåĚČçťăæŸăåřĎåĹřåőČåĞžç

```
>>> word_counts['not']
1
>>> word_counts['eyes']
8
>>>
```

åęĆæđIJä¡ăæČşæĽŃåŁĺåćđåŁăèőąæŢřïijŇåŔŕäżěçőĂå■ŢçŽĎçŤĺåŁăæşŢïijŽ

```
>>> morewords = ['why','are','you','not','looking','in','my','eyes']
>>> for word in morewords:
...     word_counts[word] += 1
...
>>> word_counts['eyes']
9
>>>
```

æĹŰèĂĚä¡ăåŘŕäžěä¡£çŤÍ `update()` æŰźæşŢïijŽ

```
>>> word_counts.update(morewords)
>>>
```

`Counter` åőđä¿ŃäÿĂäÿłéšIJäÿžäžžç§ěçŽĎçĽźæĂğæŸŕåőČäžňåŘŕäžěå¿ĹåőźæŸŞçŽĎèů§æŢřå■ęè£Řç

```
>>> a = Counter(words)
>>> b = Counter(morewords)
>>> a
Counter({'eyes': 8, 'the': 5, 'look': 4, 'into': 3, 'my': 3, 'around
↪': 2,
"you're": 1, "don't": 1, 'under': 1, 'not': 1})
>>> b
Counter({'eyes': 1, 'looking': 1, 'are': 1, 'in': 1, 'not': 1, 'you
↪': 1,
'my': 1, 'why': 1})
>>> # Combine counts
>>> c = a + b
>>> c
Counter({'eyes': 9, 'the': 5, 'look': 4, 'my': 4, 'into': 3, 'not':␣
↪2,
'around': 2, "you're": 1, "don't": 1, 'in': 1, 'why': 1,
'looking': 1, 'are': 1, 'under': 1, 'you': 1})
>>> # Subtract counts
>>> d = a - b
>>> d
Counter({'eyes': 7, 'the': 5, 'look': 4, 'into': 3, 'my': 2, 'around
↪': 2,
"you're": 1, "don't": 1, 'under': 1})
>>>
```

æŕńæŮăçŰŚéŮőïijŃ `Counter` åŕźèśąåIJĺåĞäźŐæĽĂæIJĽéIJĂèęĄåĹűèąĺæĹŰèĂĚèőąæŢřæŢřæ■őçŽĎ
åIJĺèğčåEşè£ŹçśżéŮőécŸçŽĎæŮűåĂŹä¡ăäžŤèŕěäijŸåĚĹéĂĽæŃľåőČïijŃèĂŃäÿ■æŸŕæĽŃåŁĺçŽĎåĹľçŤĺå

3.13 1.13 éĂŽè£Ğæ§ŘäÿłåĚşéŤőå■ŮæŐŠåžŘäÿĂäÿłå■ŮåĚÿåĹŮèąĺ

éŮőécŸ

ä¡ăæIJĽäÿĂäÿłå■ŮåĚÿåĹŮèąĺïijŃä¡ăæČşæăźæ■őæ§ŘäÿłæĹŰæ§ŘåĞääÿłå■ŮåĚÿå■ŮæőţæİěæŐŠåžŘè£

èğčåEşæŰźæąĹ

éĂŽè£Ğä¡£çŤÍ `operator` æĺąåİŮçŽĎ `itemgetter`
åĞ¡æŢřïijŃåŘŕäžěéİđåÿÿåőźæŸŞçŽĎæŐŠåžŘè£ŹæăůçŽĎæŢřæ■őçżŞæđĎãĂĆ
åĄĞèő¿ä¡ääžŐæŢřæ■őåžŞäÿ■æčĂçť´ćåĞžæİěç¡ŚçńŹäijŽåŚŸä£ąæĄŕåĹŮèąĺïijŃåźűäÿŤäžěäÿŃåĹŮçŽĎæŢřæ

```
rows = [
    {'fname': 'Brian', 'lname': 'Jones', 'uid': 1003},
    {'fname': 'David', 'lname': 'Beazley', 'uid': 1002},
    {'fname': 'John', 'lname': 'Cleese', 'uid': 1001},
    {'fname': 'Big', 'lname': 'Jones', 'uid': 1004}
]
```

æăźæ■őäżżæĎŘçŽĎå■ŮåĚÿå■ŮæőţæİěæŐŠåžŘè¿ŞåĚěçżŞæđIJèąŇæŸŕå¿ĹåőźæŸŞåőđçŐřçŽĎïijŇäżč

```
from operator import itemgetter
rows_by_fname = sorted(rows, key=itemgetter('fname'))
rows_by_uid = sorted(rows, key=itemgetter('uid'))
print(rows_by_fname)
print(rows_by_uid)
```

äżčçăĄçŽĎè¿ŞåĞžåęĆäÿŃïijŽ

```
[{'fname': 'Big', 'uid': 1004, 'lname': 'Jones'},
{'fname': 'Brian', 'uid': 1003, 'lname': 'Jones'},
{'fname': 'David', 'uid': 1002, 'lname': 'Beazley'},
{'fname': 'John', 'uid': 1001, 'lname': 'Cleese'}]
[{'fname': 'John', 'uid': 1001, 'lname': 'Cleese'},
{'fname': 'David', 'uid': 1002, 'lname': 'Beazley'},
{'fname': 'Brian', 'uid': 1003, 'lname': 'Jones'},
{'fname': 'Big', 'uid': 1004, 'lname': 'Jones'}]
```

`itemgetter()` åĞ¡æŢřäź§æŤŕæŇĄåď'Žäÿł keysïijŇæŕŤåęĆäÿŃéİćçŽĎäżčçăĄ

```
rows_by_lfname = sorted(rows, key=itemgetter('lname','fname'))
print(rows_by_lfname)
```

äijŽäžğçŤ§åęĆäÿŃçŽĎè¿ŞåĞžïijŽ

```
[{'fname': 'David', 'uid': 1002, 'lname': 'Beazley'},
{'fname': 'John', 'uid': 1001, 'lname': 'Cleese'},
{'fname': 'Big', 'uid': 1004, 'lname': 'Jones'},
{'fname': 'Brian', 'uid': 1003, 'lname': 'Jones'}]
```

èőÍèőž

åIJÍäÿŁéİćä¿Ńå■Řäÿ■ïijŇ `rows` èćńäijäéĂŠçżŹæŐěåŘŮäÿĂäÿłåĚşéŤőå■ŮåŘĆæŢřçŽĎ `sorted()` åĘĚç¡őåĞ¡æŢřãĂĆ è£ŹäÿłåŘĆæŢřæŸŕ `callable` çśżåđŃïijŇåźűäÿŤäżŐ `rows` äÿ■æŐěåŘŮäÿĂäÿłå■ŢäÿĂåĚČçť'äïijŇçĎűåŘŐè£ŤåŻđèćńçŤÍæİěæŐŠåžŘçŽĎåĂijãĂĆ `itemgetter()` åĞ¡æŢřåŕśæŸŕèť'§èť'čåĹŻåżżè£Źäÿł `callable` åŕźèśąçŽĎãĂĆ

`operator.itemgetter()` åĞ¡æŢřæIJL'äÿĂäÿłèćń `rows` äÿ■çŽĎèőřå¡ŢçŤÍæİěæ§ěæL'¿åĂijçŽĎçť'ćåijŢåŘĆæŢřãĂĆåŘŕäżěæŸŕäÿĂäÿłå■ŮåĚÿéŤőåŘ■çğřïijŇäÿĂäÿłæŢť'å¡ćåĂijæĹŰèĂĚäżżä¡ŢèČ¡åď'§äijäåĚěäÿĂäÿłåŕźèśąçŽĎ `__getitem__()` æŰźæşŢçŽĎåĂijãĂĆ åęĆæđIJä¡ääijäåĚěåď'Žäÿłçť'ćåijŢåŘĆæŢřçżŹ `itemgetter()`

ïijŇåőČçŤ§æĹŘçŽĎ `callable` åŕźèśąäijŽè£ŤåŻđäÿĂäÿłåŇĚåŘńæĽĂæIJĽåĚČçť äåĂijçŽĎåĚČçżĎïijŇ åźűäÿŤ `sorted()` åĞ¡æŢřäijŽæäźæ■őè£ŹäÿłåĚČçżĎäÿ■åĚČçť äéąžåžŘåŐżæŐŠåžŘãĂĆ ä¡Ęä¡äæČşèęĄåŘŇæŮűåIJĺåĞääÿłå■ŮæőţäÿŁéĺćè£ŻèąŇæŐŠåžŘïijĹæŕŤåęĆéĂŽè£ĞåğŞåŠŇåŘ■æĺĕæŐŠåžĺ

`itemgetter()` æIJĽæŮűåĂŹäź§åŘŕäžěçŤÍ `lambda` èąÍè¿¿åijŘäžčæŻ£ïijŇæŕŤåęĆïijŽ

```
rows_by_fname = sorted(rows, key=lambda r: r['fname'])
rows_by_lfname = sorted(rows, key=lambda r: (r['lname'],r['fname']))
```

è£ŹçğĞ■æŰźæąĹäź§äÿ■éŤŹãĂĆä¡ĘæŸŕïijŇä¡£çŤÍ `itemgetter()` æŰźåijŘäijŽè£ŘèąŇçŽĎçÍ■å¿őå£ńçĆźãĂĆåŽäæ■d'ïijŇåęĆæđIJä¡ääŕźæĂğèČ¡èęĄæśĆæŕŤè¿ČéńŸçŽĎèŕÍåŕš `itemgetter()` æŰźåijŘãĂĆ

æIJĂåŘŐïijŇäÿ■èęĄå£ŸäžĘè£ŹèŁĆäÿ■åśŢçd'žçŽĎæŁĂæIJŕäź§åŘŇæăůéĂĆçŤÍäžŐ `min()` åŠŇ `max()` ç■L'åĞ¡æŢřãĂĆæŕŤåęĆïijŽ

```
>>> min(rows, key=itemgetter('uid'))
{'fname': 'John', 'lname': 'Cleese', 'uid': 1001}
>>> max(rows, key=itemgetter('uid'))
{'fname': 'Big', 'lname': 'Jones', 'uid': 1004}
>>>
```

3.14 1.14 æŐŠåžŘäÿ■æŤŕæŇĄåŐ§çŤ§æŕŤè¿ČçŽĎåŕźèśą

éŮőécŸ

ä¡äæČşæŐŠåžŘçśżåđŇçŻÿåŘŇçŽĎåŕźèśąïijŇä¡ĘæŸŕäżŰäżňäÿ■æŤŕæŇĄåŐ§çŤ§çŽĎæŕŤè¿ČæŞ■ä¡IJã

èğčåĘşæŰźæąĹ

åĘĚç¡őçŽĎ `sorted()` åĞ¡æŢřæIJĽäÿĂäÿłåĚşéŤőå■ŮåŘĆæŢř `key` ïijŇåŘŕäżěäijäåĚěäÿĂäÿł `callable` åŕźèśąçżŹåőČïijŇ è£Źäÿł `callable` åŕźèśąåŕźæŕŘäÿłäijäåĚěçŽĎåŕźèśąè£ŤåŻđäÿĂäÿłåĂijïijŇè£ŹäÿłåĂijäijŽèćń `sorted` çŤĺæĺĕæŐŠåžŘè£ŹäžŽåŕźèśąãĂĆ æŕŤåęĆïijŇåęĆđIJä¡ääIJĺåžŤçŤĺçĺŇåžŘéĞŇéĺćæIJĽäÿĂäÿł `User` åőđä¿ŃåžŘåĹŮïijŇåźűäÿŤä¡ääÿŇæIJŻéĂŽè£ĞäżŰäżňçŽĎ `user_id` åśđæĂğè£ŻèąŇæŐŠåžŘïijŇ ä¡ääŘŕäžěæŘŘä¿ŻäÿĂäÿłäžě `User` åőđä¿Ňä¡IJäÿžè¿ŞåĚěåźűè¿ŞåĞžåŕźåžŤ `user_id` åĂijçŽĎ `callable` åŕźèśąãĂĆæŕŤåęĆïijŽ

```
class User:
    def __init__(self, user_id):
        self.user_id = user_id

    def __repr__(self):
        return 'User({})'.format(self.user_id)
```

(continues on next page)

(continued from previous page)

```
def sort_notcompare():
    users = [User(23), User(3), User(99)]
    print(users)
    print(sorted(users, key=lambda u: u.user_id))
```

åŘęåď'ŰäÿĂçğ■æŰźåijŘæŸŕä¡£çŤÍ `operator.attrgetter()` æİěäžčæŻ£ lambda åĞ¡æŢřïijŽ

```
>>> from operator import attrgetter
>>> sorted(users, key=attrgetter('user_id'))
[User(3), User(23), User(99)]
>>>
```

èőÍèőž

éĂL'æŃl'ä¡£çŤÍ lambda åĞ¡æŢřæĹŰèĂĚæŸŕ `attrgetter()` åŘŕèČ¡åŘŰåĘşäžŐäÿłäžžåŰIJåě¡ãĂĆ ä¡ĘæŸŕïijŃ `attrgetter()` åĞ¡æŢřéĂŽåÿÿäijŽè£ŘèąŃçŽĎå£ńçĆźïijŃåźűäÿŤè£ŸèČ¡åŘŃæŮűåĚĄèőÿåď'Žäÿłå■Ůæőţè£ŻèąŃæŕŤè¿ČãĂ
è£ŹäÿłèůŞ `operator.itemgetter()` åĞ¡æŢřä¡IJçŤÍäžŐå■ŮåĚÿçśżåđŃå¿ĹçśżäijjïijĹåŘĆèĂČ1.13åřŘè
ä¿ŃåęĆïijŃåęĆæđIJ `User` åőđä¿Ńè£ŸæIJL'äÿĂäÿł `first_name` åŠŃ `last_name` åśđæĂğïijŃéĆčäźĹåŘŕäžěåŘŚäÿŃéİćè£ŹæăůæŐŠåžŘïijŽ

```
by_name = sorted(users, key=attrgetter('last_name', 'first_name'))
```

åŘŃæăůéIJĂèęĄæşĺæĎŘçŽĎæŸŕïijŃè£ŹäÿĂåřŘèŁĆçŤÍåĹřçŽĎæŁĂæIJŕåŘŃæăůéĂĆçŤÍäžŐåČŘ `min()` åŠŃ `max()` äźŃçśżçŽĎåĞ¡æŢřãĂĆæŕŤåęĆïijŽ

```
>>> min(users, key=attrgetter('user_id'))
User(3)
>>> max(users, key=attrgetter('user_id'))
User(99)
>>>
```

3.15 1.15 éĂŽè£Ğæ§Řäÿłå■ŮæőţåřĘèőřå¡ŢåĹĘçżĎ

éŮőécŸ

ä¡ăæIJL'äÿĂäÿłå■ŮåĚÿæĹŰèĂĚåőđä¿ŃçŽĎåžŘåĹŮïijŃçĎűåŘŐä¡ăæČşæăźæ■őæ§Řäÿłçĺ'źåőŽçŽĎå■Ů
`date` æİěåĹĘçżĎè£■äžčèő£éŮőãĂĆ

èğčåĘşæŰźæąĹ

`itertools.groupby()` åĞ¡æŢřåŕźäžŐè£ŹæăůçŽĎæŢřæ■őåĹĘçżĎæŞ■ä¡IJéİđåÿÿåőđçŤÍãĂĆ äÿžäžĘæijŤçď'žïijŃåĄĞèő¿ä¡ăåůšçżŘæIJL'äžĘäÿŃåĹŮçŽĎå■ŮåĚÿåĹŮèąĺïijŽ

```
rows = [
    {'address': '5412 N CLARK', 'date': '07/01/2012'},
    {'address': '5148 N CLARK', 'date': '07/04/2012'},
    {'address': '5800 E 58TH', 'date': '07/02/2012'},
    {'address': '2122 N CLARK', 'date': '07/03/2012'},
    {'address': '5645 N RAVENSWOOD', 'date': '07/02/2012'},
    {'address': '1060 W ADDISON', 'date': '07/02/2012'},
    {'address': '4801 N BROADWAY', 'date': '07/01/2012'},
    {'address': '1039 W GRANVILLE', 'date': '07/04/2012'},
]
```

çŐřåIJÍåĄĞèő¿ä¡ăæČşåIJÍæŇĽ date åĹĘçżĎåŘŐçŽĎæŢřæ■őåİŮäÿŁè£ŻèąŇè£■äżčãĂĆäÿžäžĘè£ŹæăůĂ date)æŐŠåžŘïijŇ çĎűåŘŐèřČçŤÍ `itertools.groupby()` åĞ¡æŢřïijŽ

```
from operator import itemgetter
from itertools import groupby

# Sort by the desired field first
rows.sort(key=itemgetter('date'))
# Iterate in groups
for date, items in groupby(rows, key=itemgetter('date')):
    print(date)
    for i in items:
        print(' ', i)
```

è£ŘèąŇçżŞæđIJïijŽ

```
07/01/2012
  {'date': '07/01/2012', 'address': '5412 N CLARK'}
  {'date': '07/01/2012', 'address': '4801 N BROADWAY'}
07/02/2012
  {'date': '07/02/2012', 'address': '5800 E 58TH'}
  {'date': '07/02/2012', 'address': '5645 N RAVENSWOOD'}
  {'date': '07/02/2012', 'address': '1060 W ADDISON'}
07/03/2012
  {'date': '07/03/2012', 'address': '2122 N CLARK'}
07/04/2012
  {'date': '07/04/2012', 'address': '5148 N CLARK'}
  {'date': '07/04/2012', 'address': '1039 W GRANVILLE'}
```

èőÍèőž

`groupby()` åĞ¡æŢřæĽńæŘŘæŢť äÿłåžŘåĹŮåźűäÿŤæ§ěæĽ¿è£đçż■çŻÿåŘŇåĂijïijĹæĹŰèĂĚæăźæ■ő key åĞ¡æŢřè£ŤåŻđåĂijçŻÿåŘŇïijĽçŽĎåĚČçť äåžŘåĹŮãĂĆ åIJÍæŕŘæňąè£■äżčçŽĎæŮűåĂŹïijŇåőČäijŽè£ŤåŻđäÿĂäÿłåĂijåŠŇäÿĂäÿłè£■äżčåŹÍåŕźèśąïijŇ è£ŹäÿłèŁ■äżčåŹÍåŕźèśąåŘŕäżěçŤ§æĹŘåĚČçť äåĂijåĚÍéČÍç■Ľäž0äÿŁéÍćéĆčäÿłåĂijçŽĎçżĎäÿ■æĽĂæIJĽ åŕ

äÿĂäÿłéÍđåÿÿéĞ■èęĄçŽĎåĞĘåď Ğæ■ěéłď æŸŕèęĄæäźæ■őæŇĞåőŽçŽĎå■Ůæőţåř ĘæŢřæ■őæŐŠåžŘãĂ åŻääÿž `groupby()` äżĚäżĚæčĂæ§ěè£đçż■çŽĎåĚČçť äïijŇåęĆæđIJäžŇåĚĹåźűæšąæIJĽæŐŠåžŘåőŇæĹŘç

åęĆæđIJä¡ăäżĚäżĚåŘłæŸŕæČşæăźæ■ő date å■ŮæőţåřĘæŢřæ■őåĹĘçżĎåĹřäÿĂäÿłåď ğçŽĎæŢřæ■őçżŞ
éĆčäźĹä¡ăæIJĂåě¡ä¡£çŤĺ defaultdict() æİěæđĎåżžäÿĂäÿłåď ŽåĂijå■ŮåĚÿïijŇåĚşäžŐåď ŽåĂijå■ŮåĚÿ
1.6 åřŘèŁĆæIJL'è£ĞèŕęçżĘçŽĎäżŃçż■ãĂĆæŕŤåęĆïijŽ

```
from collections import defaultdict
rows_by_date = defaultdict(list)
for row in rows:
    rows_by_date[row['date']].append(row)
```

è£ŹæăůçŽĎèŕĺä¡ăåŘŕäżěå¿Ĺè¡żæİ¿çŽĎåŕśèČ¡åŕźæŕŘäÿłæŇĞåőŽæŮěæIJ§èő£éŮőåŕźåžŤçŽĎèőŕå¡ŢïijŽ

```
>>> for r in rows_by_date['07/01/2012']:
... print(r)
...
{'date': '07/01/2012', 'address': '5412 N CLARK'}
{'date': '07/01/2012', 'address': '4801 N BROADWAY'}
>>>
```

åIJĺäÿŁéİćè£Źäÿłä¿Ňå■Řäÿ■ïijŇæĹŚäżňæšąæIJL'å£ĚèęĄåĚĹåřĘèőŕå¡ŢæŐŠåžŘãĂĆåŻäæ■ď'ïijŇåęĆæđ
è£Źçğ■æŰźåijŘäijŽæŕŤåĚĹæŐŠåžŘçĎűåŘŐåĘ■éĂŽè£Ğ groupby()
åĞ¡æŢřè£■äžčçŽĎæŰźåijŘè£ŘèąŇå¿Ůå£ńäÿĂäžŻãĂĆ

3.16 1.16 è£Ğæżď'åžŘåĹŮåĚČçť'ă

éŮőécŸ

ä¡ăæIJL'äÿĂäÿłæŢřæ■őåžŘåĹŮïijŇæČşåĹľçŤĺäÿĂäžŻèğĎåĹŹäżŐäÿ■æŘŘåŘŰåĞžéIJĂèęĄçŽĎåĂijæĺ

èğčåĘşæŰźæąĹ

æIJĂçőĂå■ŢçŽĎè£Ğæżď'åžŘåĹŮåĚČçť'ăçŽĎæŰźæşŢåŕśæŸŕä¡£çŤĺåĹŮèąĺæŐĺåŕijãĂĆæŕŤåęĆïijŽ

```
>>> mylist = [1, 4, -5, 10, -7, 2, 3, -1]
>>> [n for n in mylist if n > 0]
[1, 4, 10, 2, 3]
>>> [n for n in mylist if n < 0]
[-5, -7, -1]
>>>
```

ä¡£çŤĺåĹŮèąĺæŐĺåŕijçŽĎäÿĂäÿłæ¡IJåIJĺçijžéŹůåŕśæŸŕåęĆæđIJè¿ŞåĚěéİđåÿÿåď ğçŽĎæŮűåĂŹäijŽäžğç
åęĆæđIJä¡ăåŕźåĘĚå■ŸæŕŤè¿ĆæŢŘæĎ§ïijŇéĆčäźĹä¡ăåŘŕäżěä¡£çŤĺçŤ§æĹŘåŹĺèąĺè¿¿åijŘè£■äžčäžğçŤ§è£Ğ

```
>>> pos = (n for n in mylist if n > 0)
>>> pos
<generator object <genexpr> at 0x1006a0eb0>
>>> for x in pos:
... print(x)
```

(continues on next page)

(continued from previous page)

```
...
1
4
10
2
3
>>>
```

æIJL'æŮűåĂŹïijŇè£ĞæżďèğĎåĹŹæŕŤè¿Čåď■æİĆïijŇäÿ■èČ¡çőĂå■ŢçŽĎåIJĺåĹŮèąĺæŐĺåŕijæĹŰèĂĚç
æŕŤåęĆïijŇåĄĞèő¿è£ĞæżďçŽĎæŮűåĂŹéIJĂèęĄåďĎçŘĘäÿĂäžŻåijĆåÿÿæĹŰèĂĚåĚűäżŰåď■æİĆæČĚåĘ
çĎűåŘŐä¡£çŤĺåĘĚåżžçŽĎ `filter()` åĞ¡æŢřãĂĆçďžä¿ŃåęĆäÿŃïijŽ

```
values = ['1', '2', '-3', '-', '4', 'N/A', '5']
def is_int(val):
    try:
        x = int(val)
        return True
    except ValueError:
        return False
ivals = list(filter(is_int, values))
print(ivals)
# Outputs ['1', '2', '-3', '4', '5']
```

`filter()` åĞ¡æŢřåĹŻåżžäžĘäÿĂäÿłè£■äżčåŹĺïijŇåŻäæ■ď'åęĆæđIJä¡äæČşå¿ŮåĹřäÿĂäÿłåĹŮèąĺçŽĎè
`list()` åŐżè¡ňæ■ćãĂĆ

èőĺèőž

åĹŮèąĺæŐĺåŕijåŠŇçŤ§æĹŘåŹĺèąĺè¿¿åijŘéĂŽåÿÿæČĚåĘţäÿŃæŸŕè£ĞæżďæŢřæ■őæIJĂçőĂå■ŢçŽĎæŰ
åĚűåőđåőČäżňè£ŸèČ¡åIJĺè£ĞæżďçŽĎæŮűåĂŹè¡ňæ■ćæŢřæ■õãĂĆæŕŤåęĆïijŽ

```
>>> mylist = [1, 4, -5, 10, -7, 2, 3, -1]
>>> import math
>>> [math.sqrt(n) for n in mylist if n > 0]
[1.0, 2.0, 3.1622776601683795, 1.4142135623730951, 1.
↪7320508075688772]
>>>
```

è£ĞæżďæŞ■ä¡IJçŽĎäÿĂäÿłåŘŸçğ■åřśæŸŕåřĘäÿ■çňęåŘĹæĺąäżűçŽĎåĂijçŤĺæŰřçŽĎåĂijäżčæŻ£ïijŇèĂ
æŕŤåęĆïijŇåIJĺäÿĂåĹŮæŢřæ■őäÿ■ä¡ššŘŕèČ¡äÿ■äżĚæČşæL'¿åĹřæ■čæŢřïijŇèĂŇäÿŤè£ŸæČşåřĘäÿ■æŸŕæ■
éĂŽè£ĞåřĘè£ĞæżďæĺąäżűæŤ¿åĹřæĺąäżűèąĺè¿¿åijŘäÿ■åŐżïijŇåŘŕäżěå¿Ĺåőźæ ŸŞçŽĎèğčåĘşè£ŹäÿłéŮőéć

```
>>> clip_neg = [n if n > 0 else 0 for n in mylist]
>>> clip_neg
[1, 4, 0, 10, 0, 2, 3, 0]
>>> clip_pos = [n if n < 0 else 0 for n in mylist]
>>> clip_pos
```

(continues on next page)

(continued from previous page)

```
[0, 0, -5, 0, -7, 0, 0, -1]
>>>
```

åŘęåď ŰäÿĂäÿłåĂijå¿ŮåĚşæşĺçŽĎè£Ğæżď åůěåĚůåřśæŸŕ itertools.compress() ïijŇ åőČäżěäÿĂäÿł iterable åŕźèśąåŠŇäÿĂäÿłçŻÿåŕźåžŤçŽĎ Boolean éĂĽæŃľ åŹĺåžŘåĹŮä¡IJäÿžè¿ŞåĚěåŘĆæŢřãĂĆ çĎűåŘŐè¿ŞåĞž iterable åŕźèśąäÿ■åŕźåžŤéĂĽæŃľ åŹĺäÿž True çŽĎåĚČçť ăãĂĆ å¡Şä¡ăéIJĂèęĄçŤĺåŘęåď ŰäÿĂäÿłçŻÿåĚşèĄŢçŽĎåžŘåĹŮæİěè£Ğæżď æ§ŘäÿłåžŘåĹŮçŽĎæŮűåĂŹïijŇè£ŹäŕŤåęĆïijŇåĄĞåęĆçŐřåIJĺä¡ăæIJĽäÿŃéİćäÿď åĹŮæŢřæ■őïijŽ

```
addresses = [
    '5412 N CLARK',
    '5148 N CLARK',
    '5800 E 58TH',
    '2122 N CLARK',
    '5645 N RAVENSWOOD',
    '1060 W ADDISON',
    '4801 N BROADWAY',
    '1039 W GRANVILLE',
]
counts = [ 0, 3, 10, 4, 1, 7, 6, 1]
```

çŐřåIJĺä¡ăæČşåřĘéĆčäžŻåŕźåžŤ count åĂijåď ğäžŐ5çŽĎåIJřåİĂåĚĺéČĺè¿ŞåĞžïijŇéĆčäźĹä¡ăåŘŕäżěè£

```
>>> from itertools import compress
>>> more5 = [n > 5 for n in counts]
>>> more5
[False, False, True, False, False, True, True, False]
>>> list(compress(addresses, more5))
['5800 E 58TH', '1060 W ADDISON', '4801 N BROADWAY']
>>>
```

è£ŹéĞŇçŽĎåĚşéŤőçĆźåIJĺäžŐåĚĹåĹŻåżžäÿĂäÿł Boolean åžŘåĹŮïijŇæŇĞçď žåŞłäžŻåĚČçť ăçňęåŘĹæĺąäżűãĂĆ çĎűåŘŐ compress() åĞ¡æŢřæäźæ■őè£ŹäÿłåžŘåĹŮåŐżéĂĽæŃľ è¿ŞåĞžåŕźåžŤä¡■ç¡őäÿž True çŽĎåĚČçť ăãĂĆ

åŠŇ filter() åĞ¡æŢřçśżäijijïijŇ compress() äźŞæŸŕè£ŤåŻđçŽĎäÿĂäÿłè£■äżčåŹĺãĂĆåŻäæ■ď ïij éĆčäźĹä¡ăéIJĂèęĄä¡£çŤĺ list() æİěåřĘçżŞæđIJè¡ňæ■ćäÿžåĹŮèąĺçśżåđŃãĂĆ

3.17 1.17 äżŐå■ŮåĚÿäÿ■æŘŘåŘŰå■ŘéŻĘ

éŮőécŸ

ä¡ăæČşæđĎéĂääÿĂäÿłå■ŮåĚÿïijŇåőČæŸŕåŘęåď ŰäÿĂäÿłå■ŮåĚÿçŽĎå■ŘéŻĘãĂĆ

èğčåĘşæŰźæąĹ

æIJĂçőĂå■ŢçŽĎæŰźåijŔæŸŕä¡£çŤĺå■ŮåĚÿæŐĺåŕijãĂĆæŕŤåęĆïijŽ

```
prices = {
    'ACME': 45.23,
    'AAPL': 612.78,
    'IBM': 205.55,
    'HPQ': 37.20,
    'FB': 10.75
}
# Make a dictionary of all prices over 200
p1 = {key: value for key, value in prices.items() if value > 200}
# Make a dictionary of tech stocks
tech_names = {'AAPL', 'IBM', 'HPQ', 'MSFT'}
p2 = {key: value for key, value in prices.items() if key in tech_
↪names}
```

èőĺèőž

åď ğåď ŽæŢřæČĚåĘţäÿŃå■ŮåĚÿæŐĺåŕijèČ¡åĄŽåĹřçŽĎïijŇéĂŽè£ĞåĹŻåżžäÿĂäÿłåĚČçżĎåžŔåĹŮçĎűdict() åĞ¡æŢřäź§èČ¡åőđçŐřãĂĆæŕŤåęĆïijŽ

```
p1 = dict((key, value) for key, value in prices.items() if value >␣
↪200)
```

ä¡ĘæŸŕïijŇå■ŮåĚÿæŐĺåŕijæŰźåijŔèąĺæĎŔæŻť æÿĚæŹřïijŇåźűäÿŤåőđéŹĚäÿŁäź§äijŽè£ŘèąŇçŽĎæŻť ïijĹåIJĺè£Źäÿłä¿Ňå■Řäÿ■ïijŇåőđéŹĚæţŃèŕŢåĞäźŐæŕŤ dict() åĞ¡æŢřæŰźåijŔå£ńæŢť æŢť äÿĂåĂ■ïijĽãĂĆ

æIJĽæŮűåĂŹåőŇæĹŘåŘŇäÿĂäżűäžŃäijŽæIJĽåď Žçğ■æŰźåijŔãĂĆæŕŤåęĆïijŇçňňäžŇäÿłä¿Ňå■ŘçĺŃ

```
# Make a dictionary of tech stocks
tech_names = { 'AAPL', 'IBM', 'HPQ', 'MSFT' }
p2 = { key:prices[key] for key in prices.keys() & tech_names }
```

ä¡ĘæŸŕïijŇè£ŘèąŇæŮűéŮť æţŃèŕŢçżŞæđIJæŸ¿çď žè£ŹçğĺæŰźæąĹåď ğæęĆæŕŤçňňäÿĂçğĺæŰźæąĹæ 1.6 åĂ■ãĂĆ åęĆæđIJåŕźçĺŃåžŔè£ŘèąŇæĂğèČ¡èęĄæśĆæŕŤè¿ČéńŸçŽĎèŕĺïijŇéIJĂèęĄèŁśçĆźæŮűéŮť åŐżå åĚşäžŐæŻť åď Žèőąæ ŮűåŠŇæĂğèČ¡æţŃèŕŢïijŇåŔŕäżěåŔĆèĂČ 14.13 åŕŘèŁĆãĂĆ

3.18 1.18 æŸäåřĎåŘ■çğŕåĹřåžŔåĹŮåĚČçť ă

éŮőéćŸ

ä¡äæIJĽäÿĂæőţéĂŽè£ĞäÿŃæăĞèő£éŮőåĹŮèąĺæĹŰèĂĚåĚČçżĎäÿ■åĚČçť äçŽĎäżčçăĄïijŇä¡ĘæŸŕè£Ź äžŐæŸŕä¡äæČşéĂŽè£ĞåŘ■çğŕæĺěèő£éŮőåĚČçť äãĂĆ

èğčåĘşæŰźæąĹ

`collections.namedtuple()` åĞ¡æŢřéĂŽè£Ğä¡£çŤĺäÿĂäÿłæŹőéĂŽçŽĎåĚČçżĎåŕźèśąæİěåÿőä¡ă
è£ŹäÿłåĞ¡æŢřåőđéŹĚäÿŁæŸŕäÿĂäÿłè£ŤåŻđ Python äÿ■æăĞåĞĘåĚČçżĎçśżåđŃå■ŘçśżçŽĎäÿĂäÿłåůěåŐĆ
ä¡ăéIJĂèęĄäijăéĂŠäÿĂäÿłçśżåđŃåŘ■åŠŃä¡ăéIJĂèęĄçŽĎå■ŮæőţçżŹåőČïijŃçĎűåŘŐåőČåřśäijŽè£ŤåŻđäÿĂ
äżčçăĄçď žä¿ŃïijŽ

```
>>> from collections import namedtuple
>>> Subscriber = namedtuple('Subscriber', ['addr', 'joined'])
>>> sub = Subscriber('jonesy@example.com', '2012-10-19')
>>> sub
Subscriber(addr='jonesy@example.com', joined='2012-10-19')
>>> sub.addr
'jonesy@example.com'
>>> sub.joined
'2012-10-19'
>>>
```

ář¡çőą `namedtuple` çŽĎåőđä¿ŃçIJŃèţůæİěåČŘäÿĂäÿłæŹőéĂŽçŽĎçśżåőđä¿ŃïijŃä¡ĘæŸŕåőČèů§åĚČ
æŕŤåęĆïijŽ

```
>>> len(sub)
2
>>> addr, joined = sub
>>> addr
'jonesy@example.com'
>>> joined
'2012-10-19'
>>>
```

åŚ¡åŘ■åĚČçżĎçŽĎäÿĂäÿłäÿżèęĄçŤĺéĂŤæŸŕåřĘä¡ăçŽĎäżčçăĄäżŐäÿŃæăĞæŞ■ä¡IJäÿ■èğčèĎśåĞżæİě
åŻăæ■ď ïijŃåęĆæđIJä¡ääżŐæŢřæ■őåžŞèřČçŤĺäÿ■è£ŤåŻđäžĘäÿĂäÿłå¿Ĺåď ğçŽĎåĚČçżĎåĹŮèąĺïijŃéĂŽè£Ğ
å¡Şä¡ääIJĺèąĺäÿ■æůżåŁäážĘæŰřçŽĎåĹŮçŽĎæŮűåĂŹä¡ăçŽĎäżčçăĄåŘŕèČ¡ářśäijŽåĞźéŤŹäžĘãĂĆä¡ĘæŸŕåđ

äÿžäžĘèŕť æŸŐæÿĚæěŽïijŃäÿŃéİćæŸŕä¡£çŤĺæŹőéĂŽåĚČçżĎçŽĎäżčçăĄïijŽ

```
def compute_cost(records):
    total = 0.0
    for rec in records:
        total += rec[1] * rec[2]
    return total
```

äÿŃæăĞæŞ■ä¡IJéĂŽåÿÿäijŽèőľäżčçăĄèąĺæĎŘäÿ■æÿĚæŹřïijŃåźűäÿŤéİđåÿÿä¿ÌèţŰèőřå¡ŢçŽĎçżŞæđĎ
äÿŃéİćæŸŕä¡£çŤĺåŚ¡åŘ■åĚČçżĎçŽĎçĽĹæIJňïijŽ

```
from collections import namedtuple

Stock = namedtuple('Stock', ['name', 'shares', 'price'])
def compute_cost(records):
    total = 0.0
```

(continues on next page)

(continued from previous page)

```
    for rec in records:
        s = Stock(*rec)
        total += s.shares * s.price
    return total
```

èőÍèőž

åŚ¡åŘ■åĚČçżĎåŘęäÿĂäÿłçŤÍéĂŤåřśæŸŕä¡IJäÿžå■ŮåĚÿçŽĎæŻ£äżčïijŇåŻääÿžå■ŮåĚÿå■ŸåĆÍéIJĂèęĄ
åęĆæđIJä¡äéIJĂèęĄæđĎåżžäÿĂäÿłéİđåÿÿåď ğçŽĎåŇĚåŘńå■ŮåĚÿçŽĎæŢřæ■őçżŞæđĎïijŇéĆčäźĹä¡£çŤÍåŚ
ä¡ĘæŸŕéIJĂèęĄæşÍæĎŘçŽĎæŸŕïijŇäÿ■åČŘå■ŮåĚÿéĆčæăůïijŇäÿĂäÿłåŚ¡åŘ■åĚČçżĎæŸŕäÿ■åŘŕæŻťæŤźç

```
>>> s = Stock('ACME', 100, 123.45)
>>> s
Stock(name='ACME', shares=100, price=123.45)
>>> s.shares = 75
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
AttributeError: can't set attribute
>>>
```

åęĆæđIJä¡äçIJ§çŽĎéIJĂèęĄæŤźåŘŸåśđæĂğçŽĎåĂijïijŇéĆčäźĹåŘŕäżěä¡£çŤÍåŚ¡åŘ■åĚČçżĎåőđä¿ŃçŽ
_replace() æŰźæşŢïijŇ åőČäijŽåĹŻåżžäÿĂäÿłåĚÍæŰřçŽĎåŚ¡åŘ■åĚČçżĎåźűåřĘåŕźåžŤçŽĎå■Ůæőţç ŤÍæ

```
>>> s = s._replace(shares=75)
>>> s
Stock(name='ACME', shares=75, price=123.45)
>>>
```

_replace() æŰźæşŢè£ŸæIJL'äÿĂäÿłå¿ĹæIJL'çŤÍçŽĎçL'źæĂğåřśæŸŕå¡Şä¡äçŽĎåŚ¡åŘ■åĚČçżĎæŇě
åőČæŸŕäÿĂäÿłéİđåÿÿæŰźä¿£çŽĎåąńåĚĚæŢřæ■őçŽĎæŰźæşŢãĂĆ
ä¡ääŘŕäżěåĚĹåĹŻåżžäÿĂäÿłåŇĚåŘńçijžçIJĄåĂijçŽĎåŐ§åđŃåĚČçżĎïijŇçĎűåŘŐä¡£çŤÍ
_replace() æŰźæşŢåĹŻåżžæŰřçŽĎåĂijèćńæŻťæŰřè£ĞçŽĎåőđä¿ŃãĂĆæŕŤåęĆïijŽ

```
from collections import namedtuple

Stock = namedtuple('Stock', ['name', 'shares', 'price', 'date',
↪'time'])

# Create a prototype instance
stock_prototype = Stock('', 0, 0.0, None, None)

# Function to convert a dictionary to a Stock
def dict_to_stock(s):
    return stock_prototype._replace(**s)
```

äÿŃéİćæŸŕåőČçŽĎä¡£çŤÍæŰźæşŢïijŽ

```
>>> a = {'name': 'ACME', 'shares': 100, 'price': 123.45}
>>> dict_to_stock(a)
Stock(name='ACME', shares=100, price=123.45, date=None, time=None)
>>> b = {'name': 'ACME', 'shares': 100, 'price': 123.45, 'date':
↪'12/17/2012'}
>>> dict_to_stock(b)
Stock(name='ACME', shares=100, price=123.45, date='12/17/2012',␣
↪time=None)
>>>
```

æIJĂåŘŐèęĄèŕť'çŽĎæŸŕïijŇåęĆæđIJä¡ăçŽĎçŻőæăĞæŸŕåőŽäźL'äÿĂäÿłéIJĂèęĄæŻť'æŰřå¿Ĺåď'Žåőđä¿è£ŹæŮűåĂŹä¡ăåžŤèŕěèĂČèŹŚåőŽäźL'äÿĂäÿłåŇĚåŘń __slots__ æŰźæşŢçŽĎçśżïijĹåŘĆèĂČ8.4åřŘèŁĆïijL'ãĂĆ

3.19 1.19 è¡ňæ■ćåźűåŘŇæŮűèőąçőŮæŢřæ■ő

éŮőécŸ

ä¡ăéIJĂèęĄåIJĺæŢřæ■őåžŘåĹŮäÿŁæL'ğèąŇèĄŽéŻĘåĞ¡æŢřïijĹæŕŤåęĆ `sum()`,`min()`,`max()` ïijL'ïijŇ ä¡ĘæŸŕéęŰåĚĹä¡ăéIJĂèęĄåĚĹè¡ňæ■ćæĹŰèĂĚè£ĞæżďæŢřæ■ő

èğčåĘşæŰźæąĹ

äÿĂäÿłéİđåÿÿäijŸéŻĚçŽĎæŰźåijŘåŐżçżŞåŘĹæŢřæ■őèőąçőŮäÿŐè¡ňæ■ćåřśæŸŕä¡£çŤĺäÿĂäÿłçŤ§æĹŘæŕŤåęĆïijŇåęĆæđIJä¡ăæČşèőąçőŮåźşæŰźåŠŇïijŇåŘŕäżěåČŘäÿŃéİćè£ŹæăůåĄŽïijŽ

```
nums = [1, 2, 3, 4, 5]
s = sum(x * x for x in nums)
```

äÿŃéİćæŸŕæŻť'åď'ŽçŽĎä¿Ńå■ŘïijŽ

```
# Determine if any .py files exist in a directory
import os
files = os.listdir('dirname')
if any(name.endswith('.py') for name in files):
    print('There be python!')
else:
    print('Sorry, no python.')
# Output a tuple as CSV
s = ('ACME', 50, 123.45)
print(','.join(str(x) for x in s))
# Data reduction across fields of a data structure
portfolio = [
    {'name':'GOOG', 'shares': 50},
    {'name':'YHOO', 'shares': 75},
    {'name':'AOL', 'shares': 20},
    {'name':'SCOX', 'shares': 65}
```

(continues on next page)

(continued from previous page)

```
]
min_shares = min(s['shares'] for s in portfolio)
```

èőĺèőž

äÿŁéİćçŽĎçď žä¿ŃåŘŚä¡ăæijŤçď žäžĘå¡ŞçŤ§æĹŘåŹĺèąĺè¿¿åijŘä¡IJäÿžäÿĂäÿłå■ŢçŃňåŘĆæŢřäijăéĂŠç
æŕŤåęĆïijÑäÿŃéİćè£ŹäžŻèŕ■åŘěæŸŕç■L'æŢĹçŽĎïijŽ

```
s = sum((x * x for x in nums)) #␣
↪æŸ¿åijŘçŽĎäijăéĂŠäÿĂäÿłçŤŞæĹŘåŹĺèąĺè¿¿åijŘåŕźèśą
s = sum(x * x for x in nums) #␣
↪æŻt'åŁääijŸéŻĚçŽĎåőđçŐřæŰźåijŘïijÑçIJĄçŢĕäžĘąæŃňåŘů
```

ä¡£çŤĺäÿĂäÿłçŤ§æĹŘåŹĺèąĺè¿¿åijŘä¡IJäÿžåŘĆæŢřäijŽæŕŤåĚĹåĹŻåżžäÿĂäÿłäÿť æŮűåĹŮèąĺæŻť åŁăér
æŕŤåęĆïijÑåęĆæđIJä¡ääÿ■ä¡£çŤĺçŤ§æĹŘåŹĺèąĺè¿¿åijŘçŽĎèŕĺïijÑä¡ăåŘŕèČ¡äijŽèĂČèŹŚä¡£çŤĺäÿŃéİćçŽĎå

```
nums = [1, 2, 3, 4, 5]
s = sum([x * x for x in nums])
```

è£Źçğ■æŰźåijŘåŘŇæăůåŘŕäżěè¿¿åĹŕæČşèęĄçŽĎæŢĹæđIJïijÑä¡ĘæŸŕåőČäijŽåď ŽäÿĂäÿłæ■ěéłď ïijÑå
åŕźäžŐåřŘåđŃåĹŮèąĺåŘŕèČ¡æšąäżĂäźĹåĚşçşżïijÑä¡ĘæŸŕåęĆæđIJåĚČçť äæŢřéĞŘéİđåÿÿåď ğçŽĎæŮűåĂŹï
åőČäijŽåĹŻåżžäÿĂäÿłåůĺåď ğçŽĎäżĚäżĚèćńä¡£çŤĺäÿĂæňąåřśèćńäÿćåijČçŽĎäÿť æŮűæŢřæ■őçżŞæđĎãĂĆè

åIJĺä¡£çŤĺäÿĂäžŻèĄŽéŻĘåĞ¡æŢřæŕŤåęĆ min() åŠŇ max() çŽĎæŮűåĂŹä¡ăåŘŕèČ¡æŻť åŁăåĂ¿åŘŚäžŐä¡£çŤĺçŤ§æĹŘåŹĺçL'ĹæIJňïijÑ
åőČäžňæŐěåŘŮçŽĎäÿĂäÿł key åĚşéŤőå■ŮåŘĆæŢřæĹŰèőÿåŕźä¡ăå¿ĹæIJL'åÿőåŁľãĂĆ
æŕŤåęĆïijÑåIJĺäÿŁéİćçŽĎèŕĄåĹÿä¿Ńå■Řäÿ■ïijÑä¡ăåŘŕèČ¡äijŽèĂČèŹŚäÿŃéİćçŽĎåőđçŐřçL'ĹæIJňïijŽ

```
# Original: Returns 20
min_shares = min(s['shares'] for s in portfolio)
# Alternative: Returns {'name': 'AOL', 'shares': 20}
min_shares = min(portfolio, key=lambda s: s['shares'])
```

3.20 1.20 åŘĹåźűåď Žäÿłå■ŮåĚÿæĹŰæŸăåřĎ

éŮőécŸ

çŐřåIJĺæIJL'åď Žäÿłå■ŮåĚÿæĹŰèĂĚæŸăåřĎïijÑä¡ăæČşåřĘåőČäžňäżŐéĂżè¿ŚäÿŁåŘĹåźűäÿžäÿĂäÿłå■
æŕŤåęĆæ§ěæL'¿åĂijæĹŰèĂĚæčĂæ§ěæ§ŘäžŻéŤőæŸŕåŘęå■ŸåIJĺãĂĆ

èğčåĘşæŰźæąĹ

åĄĞåęĆä¡ăæIJL'åęĆäÿŃäÿď äÿłå■ŮåĚÿ:

```
a = {'x': 1, 'z': 3 }
b = {'y': 2, 'z': 4 }
```

çŐřåIJÍåĄĞèő¿ä¡åå£ĚéążåIJÍäÿď äÿłå■ŮåĚÿäÿ■æĽ ğèąŇæ§ěæĽ ¿æŞ■ä¡IJïijĹæŕŤåęĆåĚĹäżŐ a äÿ■æĽ ¿ïijŇåęĆæđIJæĽ ¿äÿ■åĹřåĘ■åIJÍ b äÿ■æĽ ¿ïijĽãĂĆ äÿĂäÿłéÍđåÿÿçőĂå■ŢçŽĎèğčåEşæŰźæąĹåřśæŸŕä¡£çŤÍ collections æÍąåİŮäÿ■çŽĎ ChainMap çśżãĂĆæŕŤåęĆïijŽ

```
from collections import ChainMap
c = ChainMap(a,b)
print(c['x']) # Outputs 1 (from a)
print(c['y']) # Outputs 2 (from b)
print(c['z']) # Outputs 3 (from a)
```

èőÍèőž

äÿĂäÿł ChainMap æŐěåŘŮåď Žäÿłå■ŮåĚÿåźűåřĘåőČäżňåIJÍéĂżè¿ŚäÿŁåŘŸäÿžäÿĂäÿłå■ŮåĚÿãĂĆ çĎűåŘŐïijŇè£ŹäžŻå■ŮåĚÿåźűäÿ■æŸŕçIJ§çŽĎåŘĹåźűåIJÍäÿĂèţůäžĘïijŇ ChainMap çśżåŘłæŸŕåIJÍåĘĚéČÍåĹŻåżžäžĘäÿĂäÿłåőźçžşè£ŹäžŻå■ŮåĚÿçŽĎåĹŮèąÍ åźűéĞ■æŰřåőŽäźĽäžĘäÿĂäžŻåÿÿèğĄçŽĎå■ŮåĚÿæŞ■ä¡IJæİěéĄ■åŐĘè£ŹäÿłåĹŮèąÍãĂĆåď ğéČÍåĹĘå■ŮåĚ

```
>>> len(c)
3
>>> list(c.keys())
['x', 'y', 'z']
>>> list(c.values())
[1, 2, 3]
>>>
```

åęĆæđIJåĞžçŐřéĞ■åď ■éŤőïijŇéĆčäźĹçňňäÿĂæňąåĞžçŐřçŽĎæŸăřĎåĂijäijŽèćńè£ŤåŻđãĂĆ åŻăæ■ď ïijŇä¿Ńå■ŘçÍŃåžŘäÿ■çŽĎ c['z'] æĂżæŸŕäijŽè£ŤåŻđå■ŮåĚÿ a äÿ■åŕźåžŤçŽĎåĂijïijŇèĂŇäÿ■æŸŕ b äÿ■åŕźåžŤçŽĎåĂijãĂĆ

åŕźäžŐå■ŮåĚÿçŽĎæŻť æŰřæĹŮåĹäéŹď æŞ■ä¡IJæĂżæŸŕå¡śåŞ■çŽĎæŸŕåĹŮèąÍäÿ■çňňäÿĂäÿłå■ŮåĚÿä

```
>>> c['z'] = 10
>>> c['w'] = 40
>>> del c['x']
>>> a
{'w': 40, 'z': 10}
>>> del c['y']
Traceback (most recent call last):
...
KeyError: "Key not found in the first mapping: 'y'"
>>>
```

ChainMap åŕźäžŐçijŰçÍŃèŕ■èÍĂäÿ■çŽĎä¡IJçŤÍèŇČåŻť åŘŸéĞŘïijĹæŕŤåęĆ globals , locals ç■Ľ ïijĽæŸŕéÍđåÿÿæIJĽçŤÍçŽĎãĂĆ äžŃåőđäÿŁïijŇæIJĽäÿĂäžŻæŰźæşŢåŘŕäžěä¡£åőČåŘŸå¿ŮçőĂå■ŢïijŽ

```
>>> values = ChainMap()
>>> values['x'] = 1
>>> # Add a new mapping
>>> values = values.new_child()
>>> values['x'] = 2
>>> # Add a new mapping
>>> values = values.new_child()
>>> values['x'] = 3
>>> values
ChainMap({'x': 3}, {'x': 2}, {'x': 1})
>>> values['x']
3
>>> # Discard last mapping
>>> values = values.parents
>>> values['x']
2
>>> # Discard last mapping
>>> values = values.parents
>>> values['x']
1
>>> values
ChainMap({'x': 1})
>>>
```

ä¡IJäÿž `ChainMap` çŽĎæŻ£äżčïijŇä¡ăåŘŕèČ¡äijŽèĂČèŹŚä¡£çŤÍ `update()` æŰźæşŢåřĘäÿď äÿłå■ŮåĚÿåŘĹåźűãĂĆæŕŤåęĆïijŽ

```
>>> a = {'x': 1, 'z': 3 }
>>> b = {'y': 2, 'z': 4 }
>>> merged = dict(b)
>>> merged.update(a)
>>> merged['x']
1
>>> merged['y']
2
>>> merged['z']
3
>>>
```

è£ŹæăůäźŞèČ¡èąŇå¿ŮéĂŽïijŇä¡ĘæŸŕåőČéIJĂèęĄä¡ăåĹŻåżžäÿĂäÿłåőŇåĚĺäÿ■åŘŇçŽĎå■ŮåĚÿåŕźèśąï
åŘŇæŮűïijŇåęĆæđIJåŐ§å■ŮåĚÿåĄŽäžĘæŻť æŰřïijŇè£ŹçğŒæŤźåŘŸäÿ■äijŽåŘ■åžŤåĹřæŰřçŽĎåŘĹåźűå■U

```
>>> a['x'] = 13
>>> merged['x']
1
```

`ChainMap` ä¡£çŤĺåŐ§æİěçŽĎå■ŮåĚÿïijŇåőČèĞłåůśäÿ■åĹŻåżžæŰřçŽĎå■ŮåĚÿãĂĆæĽĂäżěåőČåźűäÿ

```
>>> a = {'x': 1, 'z': 3 }
>>> b = {'y': 2, 'z': 4 }
```

(continues on next page)

(continued from previous page)

```
>>> merged = ChainMap(a, b)
>>> merged['x']
1
>>> a['x'] = 42
>>> merged['x'] # Notice change to merged dicts
42
>>>
```

4 çňňäžŇçńäïijŽå■ŮçňęäÿšåŠŇæŰĞæIJň

åĞääźŐæL'ĂæIJL'æIJL'çŤÍçŽĎçÍŃåžŔéČ¡äijŽæűL'åŔŁåĹřæ§ŘäžŻæŰĞæIJňåď'ĎçŘĘïijŇäÿ■çőąæŸŕèğ
è£ŹäÿĂçńäåřĘéĞ■çĆźåĚşæşĺæŰĞæIJňçŽĎæŞ■ä¡IJåď'ĎçŘĘïijŇæŕŤåęĆæŘŘåŘŰå■ŮçňęäÿšïijŇæŘIJçť'ćïijŇ
åď'ğéČĺåĹĘçŽĎéŰőéćŸéČ¡èČ¡çőÄå■ŢçŽĎèřČçŤĺå■ŮçňęäÿšçŽĎåĘĚåżžæŰźæşŢåőŇæĹŘãĂĆ
ä¡ĘæŸŕïijŇäÿĂäžŻæŻť'äÿžåď'■æİĆçŽĎæŞ■ä¡IJåŘŕèČ¡éIJĂèęĄæ■čåĹŹèąĺè¿¿åijŘæĹŰèĂĚåijžåď'ğçŽĎèğčæ
åźűäÿŤåIJÍæŞ■ä¡IJUnicodeæŮűåĂŹçćřåĹřçŽĎäÿĂäžŻæčŸæL'ŃçŽĎéŰőéćŸåIJÍè£ŹéĞŇäź§äijŽèćńæŘŘåŘł

Contents:

4.1 2.1 ä¡£çŤÍåď'Žäÿłç Ţ ŇåőŽçňęåĹĘåL'šå■Ůçňęäÿš

éŰőécŸ

ä¡ăéIJĂèęĄåřĘäÿĂäÿłå■ŮçňęäÿšåĹĘåL'šäÿžåď'Žäÿłå■ŮæőţïijŇä¡ĘæŸŕåĹĘéŽŤçňę(è£ŸæIJL'åŚÍåŻť'çŽĎ

èğčåĘşæŰźæąĹ

`string` åŕźèšąçŽĎ `split()` æŰźæşŢåŘłéĂĆåžŤäžŐéİđåÿÿçőĂå■ŢçŽĎå■ŮçňęäÿšåĹĘåL'šæČĚå¡ćïij
åőČåźűäÿ■åĚĄèőÿæIJL'åď'Žäÿłå ĹĘéŽŤçňęæĹŰèĂĚæŸŕåĹĘéŽŤçňęåŚÍåŻť'äÿ■çąőåőŽçŽĎçľžæäijãĂĆ
å¡Şä¡ăéIJĂèęĄæŻť'åŁăçĄţæť'žçŽĎåĹĞåL'šå■ŮçňęäÿšçŽĎæŮűåĂŹïijŇæIJĂåě¡ä¡£çŤÍ `re.split()` æŰźæşŢïijŽ

```
>>> line = 'asdf fjdk; afed, fjek,asdf, foo'
>>> import re
>>> re.split(r'[;,\s]\s*', line)
['asdf', 'fjdk', 'afed', 'fjek', 'asdf', 'foo']
```

èőÍèőž

åĞ¡æŢř `re.split()` æŸŕéİđåÿÿåőđçŤÍçŽĎïijŇåŻääÿžåőČåĚĄèőÿä¡ääÿžåĹĘéŽŤçňęæŇĞåőŽåď'Žäÿł
æŕŤåęĆïijŇåIJÍäÿŁéÍćçŽĎä¿Ňå■Řäÿ■ïijŇåĹĘéŽŤçňęåŘŕäžěæŸŕéĂŮåŘůïijŇåĹĘåŘůæĹŰèĂĚæŸŕçľžæäijïij
åŘłèęĄè£ŹäÿłæĺąåijŘèćńæL'¿åĹřïijŇéČčäźĹåŇźéĚ■çŽĎåĹĘéŽŤçňęäÿď'è¿źçŽĎåőđä¡ŞéČ¡äijŽèćńå¡ŞæĹŘæ
è£ŤåŻđçżŞæđIJäÿžäÿĂäÿłå■ŮæőţåĹŮèąÍïijŇè£ŹäÿłèůŞ `str.split()`
è£ŤåŻđåĂijçśżåđŇæŸŕäÿĂæăůçŽĎãĂĆ

å¡Şä¡ăä¡£çŤÍ `re.split()` åĞ¡æŢřæŮűåĂŹïijŇéIJĂèęĄçL'źåĹńæşÍæĎŘçŽĎæŸŕæ■čåĹŹèąĺè¿¿åijŘäÿ
åęĆæđIJä¡£çŤÍäžĘæ■ŢèŐůåĹĘçżĎïijŇéĆčäźĹèćńåŇźéĚ■çŽĎæŰĞæIJňäź§åřĘåĞžçŐřåIJÍçżŞæđIJåĹŮèąĺäÿ

```
>>> fields = re.split(r'(;|,|\s)\s*', line)
>>> fields
['asdf', ' ', 'fjdk', ';', 'afed', ',', 'fjek', ',', 'asdf', ',',
↪'foo']
>>>
```

èŐůåŘŰåĹĘåL'šå■Ůçňęåij Íæ§ŘäžŻæČĚåĘţäÿŇäź§æŸŕæIJL'çŤÍçŽĎãĂĆ
æŕŤåęĆïijŇä¡ăåŘŕèČ¡æČşä£ÍçŢŹåĹĘåL'šå■ŮçňęäÿšïijŇçŤÍæÍěåIJÍåŘŐéÍćéĞ■æŰřæđĎéĂääÿĂäÿłæŰřçŽĎè

```
>>> values = fields[::2]
>>> delimiters = fields[1::2] + ['']
>>> values
['asdf', 'fjdk', 'afed', 'fjek', 'asdf', 'foo']
>>> delimiters
[' ', ';', ',', ',', ',', '']
>>> # Reform the line using the same delimiters
>>> ''.join(v+d for v,d in zip(values, delimiters))
'asdf fjdk;afed,fjek,asdf,foo'
>>>
```

åęĆæđIJä¡ääÿ■æČşä£ÍçŢŹåĹĘåL'šå■ŮçňęäÿšåĹřçżŞæđIJåĹŮèąĺäÿ■åŐżïijŇä¡Ęäż■çĎűéIJĂèęĄä¡£çŤÍåĹ
çąőä£Íä¡äçŽĎåĹĘçżĎæŸŕéÍđæ■ŢèŐůåĹĘçżĎïijŇå¡ćåęĆ `(?:...)` ãĂĆæŕŤåęĆïijŽ

```
>>> re.split(r'(?:,|;|\s)\s*', line)
['asdf', 'fjdk', 'afed', 'fjek', 'asdf', 'foo']
>>>
```

4.2 2.2 å■Ůçňęäÿšåij Åď'æĹŰçżŞåř¿åŇźéĚ■

éŮőécŸ

ä¡äéIJĂèęĄéĂŽè£ĞæŇĞåőŽçŽĎæŰĞæIJňæĺąåijŘåŐżæčĂæ§ěå■ŮçňęäÿšçŽĎåijĂåď'æĹŰèĂĚçżŞåř¿
Schemeç■L'ç■L'ãĂĆ

èğčåĘşæŰźæąĹ

æčĂæ§ěå■Ůçňęäÿšåij Åď'æĹŰçżŞåř¿çŽĎäÿĂäÿłçőĂå■ŢæŰźæşŢæŸŕä¡£çŤÍ `str.startswith()` æĹŰèĂĚæŸŕ `str.endswith()` æŰźæşŢãĂĆæŕŤåęĆïijŽ

```
>>> filename = 'spam.txt'
>>> filename.endswith('.txt')
True
>>> filename.startswith('file:')
False
```

(continues on next page)

(continued from previous page)

```
>>> url = 'http://www.python.org'
>>> url.startswith('http:')
True
>>>
```

åęĆæđIJä¡ăæČşæčĂæ§ěåď'Žçğ■åŇźéĚ■åŘŕèČ¡ïijŇåŘłéIJĂèęĄåřĘæL'ĂæIJL'çŽĎåŇźéĚ■éąźæŤ¿åĚěåĹ
çĎűåŘŐäijăçżŹ `startswith()` æĹŰèĂĚ `endswith()` æŰźæşŢïijŽ

```
>>> import os
>>> filenames = os.listdir('.')
>>> filenames
[ 'Makefile', 'foo.c', 'bar.py', 'spam.c', 'spam.h' ]
>>> [name for name in filenames if name.endswith(('.c', '.h')) ]
['foo.c', 'spam.c', 'spam.h'
>>> any(name.endswith('.py') for name in filenames)
True
>>>
```

äÿŇéİćæŸŕåŘęäÿĂäÿłä¿Ňå■ŘïijŽ

```
from urllib.request import urlopen

def read_data(name):
    if name.startswith(('http:', 'https:', 'ftp:')):
        return urlopen(name).read()
    else:
        with open(name) as f:
            return f.read()
```

åěĞæĂłçŽĎæŸŕïijŇè£ŹäÿłæŰźæşŢäÿ■å£ĚéążèęĄè¿ŞåĚěäÿĂäÿłåĚČçżĎä¡IJäÿžåŘĆæŢřãĂĆ
åęĆæđIJä¡ăæĄřåůğæIJL'äÿĂäÿł `list` æĹŰèĂĚ `set` çśżåđŇçŽĎéĂL'æŃl'éąźïijŇ
èęĄçąőä£İäijăéĂŠåŘĆæŢřåL'■åĚĹèřČçŤÍ `tuple()` åřĘåĚűè¡ňæ■ćäÿžåĚČçżĎçśżåđŇãĂĆæŕŤåęĆïijŽ

```
>>> choices = ['http:', 'ftp:']
>>> url = 'http://www.python.org'
>>> url.startswith(choices)
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
TypeError: startswith first arg must be str or a tuple of str, not␣
↪list
>>> url.startswith(tuple(choices))
True
>>>
```

èőĺèőž

`startswith()` åŠŇ `endswith()` æŰźæşŢæŘŘä¿ŻäžĘäÿĂäÿłéİđåÿÿæŰźä¿£çŽĎæŰźåijŘåŐżåĄŽå
çśżäijijçŽĎæŞ■ä¡IJäź§åŘŕäżěä¡£çŤĺåĹĞçL'ĞæİěåőđçŐřïijŇä¡ĘæŸŕäżčçăĄçIJŃèţůæİěæšąæIJL'éĆčäźĹäijŸé

```
>>> filename = 'spam.txt'
>>> filename[-4:] == '.txt'
True
>>> url = 'http://www.python.org'
>>> url[:5] == 'http:' or url[:6] == 'https:' or url[:4] == 'ftp:'
True
>>>
```

ä¡ăåŘŕäžěèČ¡è£ŸæČşä¡£çŤÍæ■čåĹŹèąÍè¿¿åijŘåŐżåőđçŐřïijŇæŕŤåęĆïijŽ

```
>>> import re
>>> url = 'http://www.python.org'
>>> re.match('http:|https:|ftp:', url)
<_sre.SRE_Match object at 0x101253098>
>>>
```

è£ŹçğŒæŰźåijŘäź§èąŇå¿ŮéĂŽïijŇä¡ĘæŸŕåŕźäžŐçőĂå■ŢçŽĎåŇźéĚ■åőđåIJÍæŸŕæIJL'çĆźåřŘæİŘåď'ğ

æIJĂåŘŐæŘŘäÿĂäÿŃïijŇå¡ŞåŠŇåĚűäżŰæŞ■ä¡IJæŕŤåęĆæŹőéĂŽæŢřæ■őèĄŽåŘĹçŻÿçżŞåŘĹçŽĎæŮŭ
`startswith()` åŠŇ `endswith()` æŰźæşŢæŸŕå¿Ĺäÿ■éŤŹçŽĎãĂĆ
æŕŤåęĆïijŇäÿŃéİćè£Źäÿłèŕ■åŘěæčĂæ§ěæ§ŘäÿłæŰĞäžűåď'źäÿ■æŸŕåŘęå■ŸåIJÍæŇĞåőŽçŽĎæŰĞäžűçśžåđ

```
if any(name.endswith(('.c', '.h')) for name in listdir(dirname)):
...
```

4.3 2.3 çŤÍShelléĂŽéĚ■çňęåŇźéĚ■å■Ůçňęäÿš

éŮőécŸ

ä¡ăæČşä¡£çŤÍ **Unix Shell** äÿ■åÿÿçŤÍçŽĎéĂŽéĚ■çňę(æŕŤåęĆ `*.py`,`Dat[0-9]*.csv` ç■L')åŐżåŇźéĚ■æŰĞæIJňå■Ůçňęäÿš

èğčåĘşæŰźæąĹ

`fnmatch` æĺąåİŮæŘŘä¿ŻäžĘäÿď'äÿłåĞ¡æŢŕâĂŤâĂŤ `fnmatch()` åŠŇ `fnmatchcase()` ïijŇåŘŕäžěçŤÍæİěåőđçŐřè£ŹæăůçŽĎåŇźéĚ■ãĂĆçŤÍæşŢåęĆäÿŃïijŽ

```
>>> from fnmatch import fnmatch, fnmatchcase
>>> fnmatch('foo.txt', '*.txt')
True
>>> fnmatch('foo.txt', '?oo.txt')
True
>>> fnmatch('Dat45.csv', 'Dat[0-9]*')
True
>>> names = ['Dat1.csv', 'Dat2.csv', 'config.ini', 'foo.py']
>>> [name for name in names if fnmatch(name, 'Dat*.csv')]
```

(continues on next page)

(continued from previous page)

```
['Dat1.csv', 'Dat2.csv']
>>>
```

fnmatch() åĞ¡æŢřä¡£çŤĺåžŢåśĆæŞ■ä¡IJçżçż§çŽĎåď ğåřŘåĘŹæŢŘæĎ§èğĎåĹŹ(äÿ■åŘŇçŽĎçżçż§

```
>>> # On OS X (Mac)
>>> fnmatch('foo.txt', '*.TXT')
False
>>> # On Windows
>>> fnmatch('foo.txt', '*.TXT')
True
>>>
```

åęĆæđIJä¡ăåŕźè£ŹäÿłåŇžåĹńå¿ĹåIJĺæĎŘïijŇåŘŕäžěä¡£çŤĺ fnmatchcase() æİěäžčæŻ£ãĂĆåőČåőŇåĚĺä¡£çŤĺä¡ăçŽĎæĺąåijŘåď ğåřŘåĘŹåŇźéĚ■ãĂĆæŕŤåęĆïijŽ

```
>>> fnmatchcase('foo.txt', '*.TXT')
False
>>>
```

è£ŹäÿďäÿłåĞ¡æŢřéĂŽåÿÿäijŽèćńå£¡çŢěçŽĎäÿĂäÿłçĽźæĂğæŸŕåIJĺåď ĎçŘĘéİđæŰĞäžűåŘ■çŽĎå■Ůç æŕŤåęĆïijŇåĄĞèő¿ä¡ăæIJĽäÿĂäÿłèąŮéĄŞåIJřåİĂçŽĎåĹŮèąĺæŢřæ■őïijŽ

```
addresses = [
    '5412 N CLARK ST',
    '1060 W ADDISON ST',
    '1039 W GRANVILLE AVE',
    '2122 N CLARK ST',
    '4802 N BROADWAY',
]
```

ä¡ăåŘŕäžěåČŘè£ŹæăůåĘŹåĹŮèąĺæŐĺåŕijïijŽ

```
>>> from fnmatch import fnmatchcase
>>> [addr for addr in addresses if fnmatchcase(addr, '* ST')]
['5412 N CLARK ST', '1060 W ADDISON ST', '2122 N CLARK ST']
>>> [addr for addr in addresses if fnmatchcase(addr, '54[0-9][0-9]␣
↪*CLARK*')]
['5412 N CLARK ST']
>>>
```

èőĺèőž

fnmatch() åĞ¡æŢřåŇźéĚ■èČ¡åŁŻäżŃäžŐçőĂå■ŢçŽĎå■ŮçňęäÿšæŰźæşŢåŠŇåijžåď ğçŽĎæ■čåĹŹèą åęĆæđIJåIJĺæŢřæ■őåď ĎçŘĘæŞ■ä¡IJäÿ■åŘłéIJĂèęĄçőĂå■ŢçŽĎéĂŽéĚ■çňęåřśèČ¡åőŇæĹŘçŽĎæŮűåĂŹïijŇ

åęĆæđIJä¡ăçŽĎäžčçăĄéIJĂèęĄåĄŽæŰĞäžűåŘ■çŽĎåŇźéĚ■ïijŇæIJĂåě¡ä¡£çŤĺ glob æĺąåİŮãĂĆåŘĆèĂČ5.13åřŘèŁĆãĂĆ

4.4 2.4 å■Ůçňęäÿšå Ňźé Ě■å ŠŇæŘIJçť ć

éŮőéćŸ

ä¡ăæČşåŇźéĚ■æĹŰèĂĚæŘIJçť ćçĽźåőŽæĺąåijŘçŽĎæŰĞæIJň

èğčåĘşæŰźæąĹ

åęĆæđIJä¡ăæČşåŇźéĚ■çŽĎæŸŕå■ŮéİćÅ■ŮçňęäÿšïijŇéĆčäźĹä¡ăéĂŽåÿÿåŘłéIJĂèęĄèřČçŤĺå§žæIJňå■Ů æŕŤåęĆ `str.find()` , `str.endswith()` , `str.startswith()` æĹŰèĂĚçśżäijijçŽĎæŰźæşŢïijŽ

```
>>> text = 'yeah, but no, but yeah, but no, but yeah'
>>> # Exact match
>>> text == 'yeah'
False
>>> # Match at start or end
>>> text.startswith('yeah')
True
>>> text.endswith('no')
False
>>> # Search for the location of the first occurrence
>>> text.find('no')
10
>>>
```

åŕźäžŐåď■æİĆçŽĎåŇźéĚ■éIJĂèęĄä¡£çŤĺæ■čåĹŹèąĺè¿¿åijŘåŠŇ `re` æĺąåİŮãĂĆ äÿžäžĘèğčéĞŁæ■čåĹŹèąĺè¿¿åijŘçŽĎå§žæIJňåŐ§çŘĘïijŇåĄĞèő¿ä¡ăæČşåŇźéĚ■æŢřå■ŮæăijåijŘçŽĎæŮěæ `11/27/2012` ïijŇä¡ăåŘŕäżěè£ŹæăůåĄŽïijŽ

```
>>> text1 = '11/27/2012'
>>> text2 = 'Nov 27, 2012'
>>>
>>> import re
>>> # Simple matching: \d+ means match one or more digits
>>> if re.match(r'\d+/\d+/\d+', text1):
... print('yes')
... else:
... print('no')
...
yes
>>> if re.match(r'\d+/\d+/\d+', text2):
... print('yes')
... else:
... print('no')
...
no
>>>
```

åęĆæđIJä¡ăæČşä¡£çŤĺåŘŇäÿĂäÿłæĺąåijŘåŐżåĄŽåď Žæňąå Ňźé Ě■ïijŇä¡ăåžŤèŕěåĚĹåřĘæĺąåijŘå■Ůçňęä

```
>>> datepat = re.compile(r'\d+/\d+/\d+')
>>> if datepat.match(text1):
... print('yes')
... else:
... print('no')
...
yes
>>> if datepat.match(text2):
... print('yes')
... else:
... print('no')
...
no
>>>
```

`match()` æĂżæŸŕäżŐå■ŮçňęäÿšåijĂåğŃåŐżåŇźé Ě■ïijŇåęĆæđIJä¡ăæČşæ§ěæĽ¿å■ŮçňęäÿšäżżæĎŕé ä¡£çŤĺ `findall()` æŰźæşŢåŐżäżčæŻ£ãĂĆæŕŤåęĆïijŽ

```
>>> text = 'Today is 11/27/2012. PyCon starts 3/13/2013.'
>>> datepat.findall(text)
['11/27/2012', '3/13/2013']
>>>
```

åIJĺåőŽäźĽæ■čåĹŹåijŘçŽĎæŮűåĂŹïijŇéĂŽåÿÿäijŽåĹľçŤĺæŃňåŘůåŐżæ■ŢèŐůåĹĘçżĎãĂĆæŕŤåęĆïij

```
>>> datepat = re.compile(r'(\d+)/(\d+)/(\d+)')
>>>
```

æ■ŢèŐůåĹĘçżĎåŕŕäżěä¡£å¿ŮåŘŐéİćçŽĎåď ĎçŘĘæŻť åŁăçőĂå■ŢïijŇåŽääÿžåŕŕäżěåĹĘåĹńåřĘæŕŕäÿ

```
>>> m = datepat.match('11/27/2012')
>>> m
<_sre.SRE_Match object at 0x1005d2750>
>>> # Extract the contents of each group
>>> m.group(0)
'11/27/2012'
>>> m.group(1)
'11'
>>> m.group(2)
'27'
>>> m.group(3)
'2012'
>>> m.groups()
('11', '27', '2012')
>>> month, day, year = m.groups()
>>>
>>> # Find all matches (notice splitting into tuples)
>>> text
'Today is 11/27/2012. PyCon starts 3/13/2013.'
```

(continues on next page)

```
>>> datepat.findall(text)
[('11', '27', '2012'), ('3', '13', '2013')]
>>> for month, day, year in datepat.findall(text):
... print('{}-{}-{}'.format(year, month, day))
...
2012-11-27
2013-3-13
>>>
```

`findall()` æŰźæşŢäijŽæŘIJçť ćæŰĞæIJňåźűäżěåĹŮèąĺå¡ćåijŔè£ŤåŻđæĽ ĂæIJĽçŽĎåŇźéĚ■ãĂĆ åęĆæđIJä¡ăæČşäżěè£■äżčæŰźåijŔè£ŤåŻđåŇźéĚ■ïijŇåŘŕäżěä¡£çŤÍ `finditer()` æŰźæşŢæİěäżčæŻ£ïijŇæŕŤåęĆïijŽ

```
>>> for m in datepat.finditer(text):
... print(m.groups())
...
('11', '27', '2012')
('3', '13', '2013')
>>>
```

èőÍèőž

åĚşäžŐæ■čåĹŹèąĺè¿¿åijŔçŘĘèőžçŽĎæŢŹçĺŃåůšçżŔèűĚåĞžäžĘæIJňäźęçŽĎèŇČåŻť ãĂĆ äÿ■è£ĞïijŇè£ŹäÿĂèŁĆéŸŘè£řäžĘä¡£çŤĺreæĺąåİÜè£ŻèąŇåŇźéĚ■åŠŇæŘIJçť ćæŰĞæIJňçŽĎæIJĂå§žæIJňæ æăÿå£Čæ■ěéłď åřśæŸŕåĚĹä¡£çŤĺ `re.compile()` çijŰèŕŚæ■čåĹŹèąĺè¿¿åijŔå■ŮçňęäÿšïijŇ çĎűåŘŐä¡£çŤĺ `match()`, `findall()` æĹŰèĂĚ `finditer()` ç■Ľ æŰźæşŢãĂĆ

å¡ŞåĘŹæ■čåĹŹåijŔå■ŮçňęäÿšçŽĎæŮűåĂŹïijŇçŻÿåŕźæŹőéĄ■çŽĎåĄŽæşŢæŸŕä¡£çŤĺåŐ§åğŃå■Ůçňęä `r'(\d+)/(\d+)/(\d+)'` ãĂĆ è£Źçğ■å■ŮçňęäÿšåřĘäÿ■åŐżèğčæđŘåŔ■æŰIJæİăïijŇè£ŹåIJĺæ■čåĹŹèąĺè åęĆæđIJäÿ■è£ŹæăůåĄŽçŽĎèŕİïijŇä¡ăå£Ěéążä¡£çŤĺäÿď äÿłåŔ■æŰIJæİăïijŇçśżäijij `'(\\d+)/(\\d+)/(\\d+)'` ãĂĆ

éIJĂèęĄæşĺæĎŔçŽĎæŸŕ `match()` æŰźæşŢäżĚäżĚæčĂæ§ěå■ŮçňęäÿšçŽĎåijĂåğŃéČĺåĹĘãĂĆåőČçŽ

```
>>> m = datepat.match('11/27/2012abcdef')
>>> m
<_sre.SRE_Match object at 0x1005d27e8>
>>> m.group()
'11/27/2012'
>>>
```

åęĆæđIJä¡ăæČşçš¿çąőåŇźéĚ■ïijŇçąőä£İä¡ăçŽĎæ■čåĹŹèąĺè¿¿åijŔäżě$çżŞå¿ïijŇåřśåČŔè£ŹäźĹè£Źæă

```
>>> datepat = re.compile(r'(\d+)/(\d+)/(\d+)$')
>>> datepat.match('11/27/2012abcdef')
>>> datepat.match('11/27/2012')
<_sre.SRE_Match object at 0x1005d2750>
>>>
```

æIJĂåŘŐïijŇåęĆæđIJä¡ăäżĚäżĚæŸŕåĄŽäÿĂæňąçőĂå■ŢçŽĎæŰĞæIJňåŇźéĚ■/æŘIJçť´ćæŞ■ä¡IJçŽĎèŕİ
`re` æĺąåİŮçžğåĹńçŽĎåĞ¡æŢřãĂĆæŕŤåęĆïijŽ

```
>>> re.findall(r'(\d+)/(\d+)/(\d+)', text)
[('11', '27', '2012'), ('3', '13', '2013')]
>>>
```

ä¡ĘæŸŕéIJĂèęĄæşĺæĎŘçŽĎæŸŕïijŇåęĆæđIJä¡ăæĽ´ŞçőŮåĄŽåď´ğéĞŘçŽĎåŇźéĚ■åŠŇæŘIJçť´ćæŞ■ä¡IJ
æĺąåİŮçžğåĹńçŽĎåĞ¡æŢřäijŽåřĘæIJĂè£ŚçijŰèŕŚè£ĞçŽĎæĺąåijŘçijŞå■ŸèţůæİěïijŇåŻăæ■ď´åźűäÿ■äijŽæűĹ
ä¡ĘæŸŕåęĆæđIJä¡£çŤĺéćĎçijŰèŕŚæĺąåijŘçŽĎèŕİïijŇä¡ăăřĘäijŽåĞŘåřŚæ§ěæĽ´¿åŠŇäÿĂäżŽéćİåď´ŰçŽĎåď´Ď

4.5 2.5 å■ŮçňęäÿšæŘIJçť´ćåŠŇæŻ£æ■ć

éŮőéćŸ

ä¡ăæČşåIJĺå■Ůçňęäÿšäÿ■æŘIJçť´ćåŠŇåŇźéĚ■æŇĞåőŽçŽĎæŰĞæIJňæĺąåijŘ

èğčåĘşæŰźæąĹ

åŕźäžŐçőĂå■ŢçŽĎå■ŮéİćæĺąåijŘïijŇçŻť´æŐěä¡£çŤĺ `str.replace()` æŰźæşŢå■şåŘŕïijŇæŕŤåęĆïijŽ

```
>>> text = 'yeah, but no, but yeah, but no, but yeah'
>>> text.replace('yeah', 'yep')
'yep, but no, but yep, but no, but yep'
>>>
```

åŕźäžŐåď´■æİĆçŽĎæĺąåijŘïijŇèŕůä¡£çŤĺ `re` æĺąåİŮäÿ■çŽĎ `sub()` åĞ¡æŢřãĂĆ äÿžäžĘèŕť´æŸŐè£ŹäÿłïijŇåĄĞèő¿ä¡ăæČşåřĘå¡ćåijŘäÿž `11/27/2012` çŽĎæŮěæIJ§å■ŮçňęäÿšæŤźæĹŘ `2012-11-27` ãĂĆçď´žä¿ŃåęĆäÿŃïijŽ

```
>>> text = 'Today is 11/27/2012. PyCon starts 3/13/2013.'
>>> import re
>>> re.sub(r'(\d+)/(\d+)/(\d+)', r'\3-\1-\2', text)
'Today is 2012-11-27. PyCon starts 2013-3-13.'
>>>
```

`sub()` åĞ¡æŢřäÿ■çŽĎçňňäÿĂäÿłåŘĆæŢřæŸŕèćňåŇźéĚ■çŽĎæĺąåijŘïijŇçňňäžŇäÿłåŘĆæŢřæŸŕæŻ£æ
`\3` æŇĞåŘŚåĽ■éİćæĺąåijŘçŽĎæ■ŢèŐůçżĎåŘůãĂĆ

åęĆæđIJä¡ăæĽ´ŞçőŮçŤĺçŻÿåŘŇçŽĎæĺąåijŘåĄŽåď´ŽæňąæŻ£æ■ćïijŇèĂČèŹŚåĚĹçijŰèŕŚåőČæĺěæŘŘå

```
>>> import re
>>> datepat = re.compile(r'(\d+)/(\d+)/(\d+)')
>>> datepat.sub(r'\3-\1-\2', text)
'Today is 2012-11-27. PyCon starts 2013-3-13.'
>>>
```

åŕźäžŐæŻť´åŁåďď´■æİĆçŽĎæŻ£æ■ćïijŇåŘŕäżěäijăéĂŠäÿĂäÿłæŻ£æ■ćåŻđěřČåĞ¡æŢřæİěäžčæŻ£ïijŇæŕ

```
>>> from calendar import month_abbr
>>> def change_date(m):
... mon_name = month_abbr[int(m.group(1))]
... return '{} {} {}'.format(m.group(2), mon_name, m.group(3))
...
>>> datepat.sub(change_date, text)
'Today is 27 Nov 2012. PyCon starts 13 Mar 2013.'
>>>
```

äÿĂäÿłæŻ£æ■ćåŻđèřČåĞ¡æŢřçŽĎåŘĆæŢřæŸŕäÿĂäÿł match åŕźèśąïijŇäź§åřśæŸŕ match() æĹŰèĂĚ find() è£ŤåŻđçŽĎåŕźèśąãĂĆ ä¡£çŤÍ group() æŰźæşŢæİěæŘŘåŘŰçĽźåőŽçŽĎåŇźéĚ■éČÍåĹĘãĂĆåŻđèřČåĞ¡æŢřæIJĂåŘŐè£ŤåŻđæŻ£æ■ćå■Ůçňęäÿšã

åęĆæđIJéŹď äžĘæŻ£æ■ćåŘŐçŽĎçżŞæđIJåď ŰïijŇä¡äè£ŸæČşç§ěéĄŞæIJĽåď ŽåřŚæŻ£æ■ćåŘŚçŤ§äžĘ re.subn() æİěäžčæŻ£ãĂĆæŕŤåęĆïijŽ

```
>>> newtext, n = datepat.subn(r'\3-\1-\2', text)
>>> newtext
'Today is 2012-11-27. PyCon starts 2013-3-13.'
>>> n
2
>>>
```

èőÍèőž

åĚşäžŐæ■čåĹŹèąÍè¿¿åijŘæŘIJçť ćåŠŇæŻ£æ■ćïijŇäÿŁéİćæijŤçď žçŽĎ sub() æŰźæşŢå§žæIJňåůšçżŘæűţçŻŰäžĘæĽ ĂæIJĽãĂĆ åĚűåőđæIJĂéŽ¿çŽĎéČÍåĹĘåřśæŸŕçijŰåĘŹæ■čåĹŹèąÍè¿¿

4.6 2.6 å■Ůçňęäÿšå£¡çŢěåď ğåřŘåĘŹçŽĎæŘIJçť ćæŻ£æ■ć

éŮőécŸ

ä¡äéIJĂèęĄäżěå£¡çŢěåď ğåřŘåĘŹçŽĎæŰźåijŘæŘIJçť ćäÿŐæŻ£æ■ćæŰĞæIJňå■Ůçňęäÿš

èğčåĘşæŰźæąĹ

äÿžäžĘåIJÍæŰĞæIJňæŞ■ä¡IJæŮűå£¡çŢěåď ğåřŘåĘŹïijŇä¡äéIJĂèęĄåIJÍä¡£çŤÍ re æĺąåİŮçŽĎæŮűåĂŹçżŹè£ŹäžŻæŞ■ä¡IJæŘŘä¿Ż re.IGNORECASE äăĞå£ŮåŘĆæŢřãĂĆæŕŤåęĆïijŽ

```
>>> text = 'UPPER PYTHON, lower python, Mixed Python'
>>> re.findall('python', text, flags=re.IGNORECASE)
['PYTHON', 'python', 'Python']
>>> re.sub('python', 'snake', text, flags=re.IGNORECASE)
'UPPER snake, lower snake, Mixed snake'
>>>
```

æIJĂåŘŐçŽĎéĆčäÿłä¿Ńå■ŘæŘ■çď žäžĘäÿĂäÿłåřŘçijžéŹůïijŃæŻ£æ■ćå■Ůçňęäÿšåźűäÿ■äijŽèĞłåŁÍèů§äÿžäžĘä£őåď■è£ŹäÿłïijŃä¡ššŘŕèČ¡éIJĂèęĄäÿĂäÿłè¿ĚåŁľ åĞ¡æŢřïijŃåřšåČŘäÿŃéİćçŽĎè£ŹæăůïijŽ

```
def matchcase(word):
    def replace(m):
        text = m.group()
        if text.isupper():
            return word.upper()
        elif text.islower():
            return word.lower()
        elif text[0].isupper():
            return word.capitalize()
        else:
            return word
    return replace
```

äÿŃéİćæŸŕä¡£çŤÍäÿŁè£řåĞ¡æŢřçŽĎæŰźæşŢïijŽ

```
>>> re.sub('python', matchcase('snake'), text, flags=re.IGNORECASE)
'UPPER SNAKE, lower snake, Mixed Snake'
>>>
```

èřŚèĂĚæşÍïijŽ `matchcase('snake')` è£ŤåŻđäžĘäÿĂäÿłåŻđèřČåĞ¡æŢř(åŘĆæŢřå£ĚéążæŸŕ `match` åŕźèśą)ïijŃåĽ■éİćäÿĂèŁĆæŘŘåŁřè£ĞïijŃ `sub()` åĞ¡æŢřéŹď äžĘæŐěåŘŮæŻ£æ■ćå■Ůçňęäÿšåď ŰïijŃè£ŸèČ¡æŐěåŘŮäÿĂäÿłåŻđèřČåĞ¡æŢřãĂĆ

èőÍèőž

åŕźäžŐäÿĂèŁňçŽĎå£¡çŢěåď ğåřŘåĘŹçŽĎåŃźéĚ■æŞ■ä¡IJïijŃçőĂå■ŢçŽĎäijäéĂŠäÿĂäÿł `re.IGNORECASE` æăĞå£ŮåŘĆæŢřåřšåůšçżŘèűşåď §äžĘãĂĆ ä¡ĘæŸŕéIJĂèęĄæşÍæĎŘçŽĎæŸŕïijŃè£ŹäÿłåŕźäžŐæ§ŘäžŻéIJĂèęĄåď ğåřŘåĘŹè¡ňæ■ćçŽĎUnicodeåŃźéĚ■å åŘĆèĂČ2.10åřŘèŁĆäžĘèğčæŻť åď ŽçżĘèŁĆãĂĆ

4.7 2.7 æIJĂç§■åŃźéĚ■æÍąåijŘ

éŮőécŸ

ä¡ăæ■čåIJÍèřŢçİĂçŤÍæ■čåŁŹèąÍè¿¿åijŘåŃźéĚ■æ§ŘäÿłæŰĞæIJňæÍąåijŘïijŃä¡ĘæŸŕåőČæĽ¿åŁřçŽĎæŸ èĂŃä¡ăæČşä£őæŤźåőČåŘŸæŁŘæ§ěæĽ¿æIJĂç§■çŽĎåŘŕèČ¡åŃźéĚ■ãĂĆ

èğčåĘşæŰźæąŁ

è£ŹäÿłéŮőécŸäÿĂèŁňåĞžçŐřåIJÍéIJĂèęĄåŃźéĚ■äÿĂåŕźåŁĘéŽŤçňęäźŃéŮť çŽĎæŰĞæIJňçŽĎæŮűåĂ äÿžäžĘèŕť æŸŐæÿĚæěŽïijŃèĂČèŹŚåęĆäÿŃçŽĎä¿Ńå■ŘïijŽ

```
>>> str_pat = re.compile(r'"(.*)"')
>>> text1 = 'Computer says "no."'
>>> str_pat.findall(text1)
['no.']
>>> text2 = 'Computer says "no." Phone says "yes."'
>>> str_pat.findall(text2)
['no." Phone says "yes.']
>>>
```

åIJÍè£ŹäÿŁä¿Ńå■Řäÿ■ïijŇæÍąåijŘ `r'\"(.*)\"'` çŽĎæĎŘåŻ¿æŸŕåŇźéĚ■èćńåŘŇåijŢåŘůåŇĚåŘńçŽĎäÿ¡ĘæŸŕåIJÍæ■čåĹŹèąÍè¿¿åijŘäÿ■*æŞ■ä¡IJçňęæŸŕèťłåľłçŽĎïijŇåŻăæ■ďåŇźéĚ■æŞ■ä¡IJäijŽæ§ěæĽ¿æIJĂéŢäžŐæŸŕåIJÍçňňäžŇäÿłä¿Ńå■Řäÿ■æŘIJçťć `text2` çŽĎæŮűåĂŹè£ŤåŻđçżŞæđIJåźűäÿ■æŸŕæĹŚäžňæČşèęĄç

äÿžäžĘä£őæ■čè£ŹäÿłéŮőécŸïijŇåŘŕäżěåIJÍæÍąåijŘäÿ■çŽĎ*æŞ■ä¡IJçňęåŘŐéİćåŁääÿŁ?ä£őéěřçňęïijŇåi

```
>>> str_pat = re.compile(r'"(.*?)"')
>>> str_pat.findall(text2)
['no.', 'yes.']
>>>
```

è£Źæăůåřśä¡£å¿ŮåŇźéĚ■åŘŸæĹŘéİđèťłåľłæÍąåijŘïijŇäżŐèĂŇå¿ŮåĹřæIJĂç§■çŽĎåŇźéĚ■ïijŇäź§åřśa

èőÍèőž

è£ŹäÿĂèŁĆåśŢçďžäžĘåIJÍåĘŹåŇĚåŘńçĆź(.)å■ŮçňęçŽĎæ■čåĹŹèąÍè¿¿åijŘçŽĎæŮűåĂŹéĄĞåĹřçŽĎäÿ
åIJÍäÿĂäÿłæÍąåijŘå■Ůçňęäÿšäÿ■ïijŇçĆź(.)åŇźéĚ■éŹďäžĘæ■ćèąŇåďŰçŽĎäżżä¡Ţå■ŮçňęäĂĆ
çĎűèĂŇïijŇåęĆæđIJä¡äářĘçĆź(.)åŘůæŤ¿åIJÍåijĂåğŃäÿŐçżŞæİ§çňę(æŕŤåęĆåijŢåŘů)äźŃéŮťçŽĎæŮűåĂŹi
è£ŹæăůéĂŽåÿÿäijŽåŕijèĞťå¿Ĺåď'Žäÿ■éŮťçŽĎèćńåijĂåğŃäÿŐçżŞæİ§çňęåŇĚåŘńçŽĎæŰĞæIJňèćńå£¡çŢěa
éĂŽè£ĞåIJÍ * æĹŰèĂĚ + è£ŹæăůçŽĎæŞ■ä¡IJçňęåŘŐéİćæůżåŁääÿĂäÿł ?
åŘŕäżěåijżåĹűåŇźéĚ■çőŮæşŢæŤźæĹŘåŕżæĽ¿æIJĂç§■çŽĎåŘŕèČ¡åŇźéĚ■ãĂĆ

4.8 2.8 åďŽèąŇåŇźéĚ■æÍąåijŘ

éŮőécŸ

ä¡äæ■čåIJÍèŕŢçİĂä¡£çŤÍæ■čåĹŹèąÍè¿¿åijŘåŐżåŇźéĚ■äÿĂåď'ğåİŮçŽĎæŰĞæIJňïijŇèĂŇä¡äéIJĂèęĄèůĚ

èğčåĘşæŰźæąĹ

è£ŹäÿłéŮőécŸå¿ĹåĚÿåđŃçŽĎåĞžçŐŕåIJÍå¡Şä¡äçŤÍçĆź(.)åŐżåŇźéĚ■äżżæĎŘå■ŮçňęçŽĎæŮűåĂŹïijŇ
æŕŤåęĆïijŇåĄĞèő¿ä¡äæČşèŕŢçİĂåŐżåŇźéĚ■Cèŕ■èÍĂåĹĘåĽ'šçŽĎæşÍéĞŁïijŽ

```
>>> comment = re.compile(r'/\*(.*?)\*/')
>>> text1 = '/* this is a comment */'
>>> text2 = '''/* this is a
... multiline comment */
```

(continues on next page)

(continued from previous page)

```
... '''
>>>
>>> comment.findall(text1)
[' this is a comment ']
>>> comment.findall(text2)
[]
>>>
```

äÿžäžĘä£őæ■čè£ŹäÿłéŮőéćŸïijŇä¡ššŘŕäžěä£őæŤźæÍąššijŘå■ŮçňęäÿšïijŇšćđšŁššŕźæ■ćèąŇçŽĎæŤŕæŇ

```
>>> comment = re.compile(r'/\*((?:.|\n)*?)\*/')
>>> comment.findall(text2)
[' this is a\n multiline comment ']
>>>
```

šIJÍè£ŹäÿłæÍąššijŘäÿ■ïijŇ (?:.|\n) æŇĞšőŽäžĘäÿĂäÿłéİđæ■ŢèŐůçżĎ (äź§šŕśæŸŕšőČšőŽäźL'äžĘäÿĂäÿłäżĚäżĚçŤÍæİěšĄŽšŇźéĚ■ïijŇèĂŇäÿ■èČ¡éĂŽè£Ğš■ŢçŃňæ■ŢèŐůæŁŰèĂ

èőÍèőž

re.compile() šĞ¡æŢŕæŐěšŘŮäÿĂäÿłæăĞš£ŮšŘĆæŢŕšŘń re.DOTALL ïijŇšIJÍè£ŹéĞŇéİđšÿÿæIJL'çŤÍãĂĆ šőČšŘŕäžěèőľæ■čšŁŹèąÍè¿¿šijŘäÿ■çŽĎçĆź(.)šŇźéĚ■šŇĚæŃňæ■ćèąŇç

```
>>> comment = re.compile(r'/\*(.*?)\*/', re.DOTALL)
>>> comment.findall(text2)
[' this is a\n multiline comment ']
```

šŕźäžŐçőĂš■ŢçŽĎæČĚšĘţä¡£çŤÍ re.DOTALL æăĞèőŕšŘĆæŢŕšůěä¡IJçŽĎš¿Łšě¡ïijŇ ä¡ĘæŸŕšęĆæđIJæÍąššijŘéİđšÿÿšd'■æİĆæŁŰèĂĚæŸŕäÿžäžĘæđĎéĂššŮçňęäÿšäžd'çL'ŇèĂŇšřĘšd'Žäÿłæ
è£ŹæŮűšĂŹä¡£çŤÍè£ŹäÿłæăĞèőŕšŘĆæŢŕšřšŘŕèČ¡šĞžçŐŕäÿĂäžŹéŮőéćŸãĂĆ
šęĆæđIJèőľä¡äéĂL'æŃľçŽĎèŕİïijŇæIJĂšě¡è£ŸæŸŕšőŽäźL'èĞłšůšçŽĎæ■čšŁŹèąÍè¿¿šijŘæÍąššijŘïijŇè£Źæăů

4.9 2.9 šřĘUnicodeæŰĞælJňæăĞšĞĘšŇŰ

éŮőéćŸ

ä¡ăæ■čšIJÍšd'ĎçŘĘUnicodeš■ŮçňęäÿšïijŇéIJĂèęĄçąőä£ÍæL'ĂæIJL'š■ŮçňęäÿššIJÍšžŢšśĆæIJL'çŻÿšŘŇ

èğčšĘşæŰźæąŁ

šIJÍUnicodeäÿ■ïijŇæ§ŘäžŻš■ŮçňęèČ¡šd'§çŤÍšd'ŽäÿłšŘŁæşŢçŽĎçijŰçăĄèąÍçd'žăĂĆäÿžäžĘèŕť'æŸŐïij

```
>>> s1 = 'Spicy Jalape\u00f1o'
>>> s2 = 'Spicy Jalapen\u0303o'
>>> s1
```

(continues on next page)

(continued from previous page)

```
'Spicy JalapeÃśo'
>>> s2
'Spicy JalapeÃśo'
>>> s1 == s2
False
>>> len(s1)
14
>>> len(s2)
15
>>>
```

è£ŹéĞŇçŽĎæŰĞæIJňâĂİSpicy JalapeÃśoâĂİä¡£çŤĺäžĘäÿď´çğ■å¡ćåijŔæİěèąĺçď´žãĂĆçňňäÿĂçğ■ä¡£çŤĺæŢť´ä¡Şå■Ůçňęâ̆ĂİÃśâĂİ(U+00F1)ïijŇçňňäžŇçğ■ä¡£çŤĺæŃĽäÿĄå■Ůæŕ■âĂİnâĂİåŘŐéİćä

åIJĺéIJĂèęĄæŕŤè¿Čå■ŮçňęäÿšçŽĎçĺŃåžŔäÿ■ä¡£çŤĺå■ŮçňęçŽĎåď´Žçğ■èąĺçď´žäijŽäžğçŤ§éŰőécŸãĂĆäÿžäžĘä£őæ■čè£ŹäÿłéŰőécŸïijŇä¡åŘŕäżěä¡£çŤĺunicodedataæĺąåİŮåĚĹåřĘæŰĞæIJňæăĞåĞĘåŇŰïijŽ

```
>>> import unicodedata
>>> t1 = unicodedata.normalize('NFC', s1)
>>> t2 = unicodedata.normalize('NFC', s2)
>>> t1 == t2
True
>>> print(ascii(t1))
'Spicy Jalape\xf1o'
>>> t3 = unicodedata.normalize('NFD', s1)
>>> t4 = unicodedata.normalize('NFD', s2)
>>> t3 == t4
True
>>> print(ascii(t3))
'Spicy Jalapen\u0303o'
>>>
```

`normalize()` çňňäÿĂäÿłåŘĆæŢřæŇĞåőŽå■ŮçňęäÿšæăĞåĞĘåŇŰçŽĎæŰźåijŔãĂĆNFCèąĺçď´žå■ŮçňęåžŤèŕěæŸŕæŢť´ä¡ŞçżĎæĹŘ(æŕŤåęĆåŘŕèČ¡çŽĎèŕĺåŕśä¡£çŤĺå■ŢäÿĂçijŰçăA)ïijŇèĂŇNFI

PythonåŘŇæăůæŤŕæŇĄæĽ´ľåśŢçŽĎæăĞåĞĘåŇŰå¡ćåijŔNFKCåŠŇNFKDïijŇåőČäżňåIJĺåď´ĎçŘĘæ§Ř

```
>>> s = '\ufb01' # A single character
>>> s
'ïňĄ'
>>> unicodedata.normalize('NFD', s)
'ïňĄ'
# Notice how the combined letters are broken apart here
>>> unicodedata.normalize('NFKD', s)
'fi'
>>> unicodedata.normalize('NFKC', s)
'fi'
>>>
```

èőÍèőž

æăĞåĞĘåŇŰåŕźäžŐäżä¡ŢéIJĂèęĄäżěäÿĂèĞť'çŽĎæŰźåijŘåď'ĎçŘĘUnicodeæŰĞæIJňçŽĎçÍŃåžŘéČ¡
å¡Şåď'ĎçŘĘæÍěèĞłçŤÍæĹůè¿ŞåĚěçŽĎå■Ůçňęäÿšè ĂŇä¡åå¿ĹéŽ¿åŐżæŐğåĹűçijŰçăĄçŽĎæŰűåĂŹåřď'åĚűåę

åIJÍæÿĚçŘĘåŠŇè£Ğæżď'æŰĞæIJňçŽĎæŰűåĂŹå■ŮçňęçŽĎæăĞåĞĘåŇŰäź§æŸŕå¿ĹéĞ■èęĄçŽĎãĂĆ
æŕŤåęĆïijŇåĄĞèő¿ä¡äæČşæÿĚéŹď'æŐĹ'äÿĂäžŻæŰĞæIJňäÿŁéÍćçŽĎåŘŸé§§çňęçŽĎæŰűåĂŹ(åŘŕèČ¡æŸŕäÿ

```
>>> t1 = unicodedata.normalize('NFD', s1)
>>> ''.join(c for c in t1 if not unicodedata.combining(c))
'Spicy Jalapeno'
>>>
```

æIJĂåŘŐäÿĂäÿłä¿Ňå■ŘåśŢçď'žäžĘ `unicodedata` æĺąåÍŮçŽĎåŘęäÿĂäÿłéĞ■èęĄæŰźéÍćïijŇäź§åŕśæ
`combining()` åĞ¡æŢřåŘŕäżěæţŃèŕŢäÿĂäÿłå■ŮçňęæŸŕåŘęäÿžåŠŇé§§å■Ůçňęã ĂĆ
åIJÍè£ŹäÿłæĺąåÍŮäÿ■è£ŸæIJĹ'åĚűäžŰåĞ¡æŢřçŤÍäžŐæ§ěæĹ'¿å■Ůçňęçśżå ĹńïijŇæţŃèŕŢæŸŕåŘęäÿžæŢřå■Ůå

UnicodeæŸ¿çĎűæŸŕäÿĂäÿłå¿Ĺåď'ğçŽĎäÿżéćŸãĂĆåęĆæđIJæČşæŻť'æůśåĚěçŽĎäžĘèğčåĚşäžŐæăĞåĞ
èŕůçIJŃèĂČ UnicodeåőŸç¡Śäÿ■åĚşäžŐè£ŹéČÍåĹęçŽĎèŕť'æŸŐ
Ned BatchelderåIJÍ äżŰçŽĎç¡ŚçńŹ äÿŁåŕźPythonçŽĎUni-
codeåď'ĎçŘĘéŰőéćŸäź§æIJĹ'äÿĂäÿłå¿Ĺåě¡çŽĎäżŃçż■ãĂĆ

4.10 2.10 åIJÍæ■čåĹŹåijŘäÿ■ä¡£çŤÍUnicode

éŰőéćŸ

ä¡äæ■čåIJÍä¡£çŤÍæ■čåĹŹèąÍè¿¿åijŘåď'ĎçŘĘæŰĞæIJňïijŇä¡ĘæŸŕåĚşæşÍçŽĎæŸŕUnicodeå■Ůçňęåď'Ďç

èğčåĚşæŰźæąĹ

éżŸèőď'æČĚåĘţäÿŃ `re` æĺąåÍŮåůšçżŘåŕźäÿĂäžŻUni-
codeå■Ůçňęçśżæ IJĹ'äžĘå§žæIJňçŽĎæŤŕæŇĄãĂĆ æŕŤåęĆïijŇ `\\d`
åůšçżŘåŇźéĚ■äżżæĎŘçŽĎunicodeæŢřå■Ůå■Ůçňęäž Ęïij Ž

```
>>> import re
>>> num = re.compile('\d+')
>>> # ASCII digits
>>> num.match('123')
<_sre.SRE_Match object at 0x1007d9ed0>
>>> # Arabic digits
>>> num.match('\u0661\u0662\u0663')
<_sre.SRE_Match object at 0x101234030>
>>>
```

åęĆæđIJä¡äæČşåIJÍæĺąåijŘäÿ■åŇĚåŘńæŇĞåőŽçŽĎUnicodeå■ŮçňęïijŇä¡ääŘŕäżěä¡£çŤÍUnicodeå■Ůçň
`\uFFF` æĹŰèĂĚ `\UFFFFFFFF`)ãĂĆ æŕŤåęĆïijŇäÿŇéÍćæŸŕäÿĂäÿłåŇźéĚ■åĞääÿłäÿ■åŘŇéŸ£æŃĹ'äijŕçijŰç

```
>>> arabic = re.compile('[\u0600-\u06ff\u0750-\u077f\u08a0-\u08ff]+
↪')
>>>
```

å¡ŞæĽ'ğèąŇåŇźéĚ■åŠŇæŘIJçť'ćæŞ■ä¡IJçŽĎæŮűåĂŹïijŇæIJĂåě¡æŸŕåĚĹæăĞåĞĘåŇŰåźűäÿŤæÿĚçŘE
ä¡ĘæŸŕåŘŇæăůäź§åžŤèŕěæşĺæĎŔäÿĂäžŻçĽźæőŁæČĚåĘţïijŇæŕŤåęĆåIJĺå£¡çŢěåď'ğåřŔåĘŹåŇźéĚ■åŠŇåď'

```
>>> pat = re.compile('stra\u00dfe', re.IGNORECASE)
>>> s = 'straÃ§e'
>>> pat.match(s) # Matches
<_sre.SRE_Match object at 0x10069d370>
>>> pat.match(s.upper()) # Doesn't match
>>> s.upper() # Case folds
'STRASSE'
>>>
```

èőĺèőž

æůůåŘĹä¡£çŤĺUnicodeåŠŇæ■čåĹŹèąĺè¿¿åijŘéĂŽåÿÿäijŽèőľä¡äæŁŞçŃĆãĂĆ
åęĆæđIJä¡äçIJ§çŽĎæĽŞçőŮè£ŹæăůåĄŽçŽĎèŕÎïijŇæIJĂåě¡èĂČèŹŚäÿŇåőĽèčĚçňňäÿĽæŰźæ■čåĹŹåijŘåžŞ
åőČäžňäijŽäÿžUnicodeçŽĎåď'ğåřŔåĘŹè¡ňæ■ćåŠŇåĚűäžŰåď'ğéĞŘæIJĽèűčçĽźæĂğæŘŘä¿ŻåĚĺéİćçŽĎæŤŕ

4.11 2.11 åĹăéŹďå■Ůçňęäÿšäÿ■äÿ■éIJĂèęĄçŽĎå■Ůçňę

éŮőécŸ

ä¡äæČşåŐżæŐĽ'æŰĞæIJňå■ŮçňęäÿšåijĂåď'ťïijŇçżŞåř¿æĹŰèĂĚäÿ■éŮť'äÿ■æČşèęĄçŽĎå■ŮçňęïijŇæŕ

èğčåĘşæŰźæąĹ

`strip()` æŰźæşŢèČ¡çŤĺäžŐåĹăéŹď'åijĂåğŃæĹŰçżŞåř¿çŽĎå■ŮçňęäĂĆ
`lstrip()` åŠŇ `rstrip()` åĹĘåĹńäżŐåůęåŠŇäżŐåŘşæĽ'ğèąŇåĹăéŹď'æŞ■ä¡IJäĂĆ
éżŸèőď'æČĚåĘţäÿŃïijŇè£ŹäžŻæŰźæşŢäijŽåŐżéŹď'çľ'žçŹ¡å■ŮçňęïijŇä¡ĘæŸŕä¡ääź§åŘŕäžěæŇĞåőŽåĚűäżŰ

```
>>> # Whitespace stripping
>>> s = ' hello world \n'
>>> s.strip()
'hello world'
>>> s.lstrip()
'hello world \n'
>>> s.rstrip()
' hello world'
>>>
>>> # Character stripping
>>> t = '-----hello======'
```

(continues on next page)

(continued from previous page)

```
>>> t.lstrip('-')
'hello======'
>>> t.strip('-=')
'hello'
>>>
```

èőÍèőž

è£ŹäžŻ `strip()` æŰźæşŢåIJÍèŕżåŘŰåŠŇæÿĚçŘĘæŢřæ■őäžěåď ĞåŘŐçż■åď ĎçŘĘçŽĎæŮűåĂŹæŸŕç
æŕŤåęĆïijŇä¡ăåŘŕäžěçŤÍåőČäžňæİěåŐżæŐL'çľžæăijïijŇåijŢåŘůåŠŇåőŇæĹŘåĚűäżŰäżżåŁąãĂĆ

ä¡ĘæŸŕéIJĂèęĄæşÍæĎŘçŽĎæŸŕåŐżéŹď æŞ■ä¡IJäÿ■äijŽåŕźå■Ůçňęäÿšç ŽĎäÿ■éŮť çŽĎæŰĞæIJňäžğçŤ

```
>>> s = ' hello     world \n'
>>> s = s.strip()
>>> s
'hello     world'
>>>
```

åęĆæđIJä¡ăæČşåď ĎçŘĘäÿ■éŮť çŽĎçľžæăijïijŇéĆčäźĹä¡ăéIJĂèęĄæśĆåŁľåĚűäżŰæŁĂæIJŕãĂĆæŕŤåę
`replace()` æŰźæşŢæĹŰèĂĚæŸŕçŤÍæ■čåĹŹèąÍè¿¿åijŘæŻ£æ■ćãĂĆçď žä¿ŃåęĆäÿŃïijŽ

```
>>> s.replace(' ', '')
'helloworld'
>>> import re
>>> re.sub('\s+', ' ', s)
'hello world'
>>>
```

éĂŽåÿÿæČĚåĘţäÿŃä¡ăæČşåřĘå■Ůçňęäÿš `strip` æŞ■ä¡IJåŠŇåĚűäżŰè£■äžčæŞ■ä¡IJçŻÿçżŞåŘĹïijŇæŕ
åęĆæđIJæŸŕè£ŹæăůçŽĎèŕİïijŇéĆčäźĹçŤ§æĹŘåŹÍèąÍè¿¿åijŘåřśåŘŕäžěåď ğæŸ¿èžňæL'ŃäžĘãĂĆæŕŤåęĆïijŽ

```
with open(filename) as f:
    lines = (line.strip() for line in f)
    for line in lines:
        print(line)
```

åIJÍè£ŹéĞŇïijŇèąÍè¿¿åijŘ `lines = (line.strip() for line in f)`
æL'ğèąŇæŢřæ■ŏè¡ňæ■ćæŞ■ä¡IJãĂĆ è£ŹçğæŰźåijŘéİđåÿÿéŃŸæŢĹïijŇåŻääÿžåőČäÿ■éIJĂèęĄéćĎåĚĹèŕżåĚ
åőČäžĚäżĚåŘłæŸŕåĹŻåżžäÿĂäÿłçŤ§æĹŘåŹÍïijŇåźűäÿŤæŕŘæňąè£ŤåŽđèąŇäźŃåL'■äijŽåĚĹæL'ğèąŇ
`strip` æŞ■ä¡IJãĂĆ

åŕźäžŐæŻť'éńŸéŸűçŽĎstripïijŇä¡ăåŘŕèČ¡éIJĂèęĄä¡£çŤÍ `translate()`
æŰźæşŢãĂĆèŕůåŘĆéŸĚäÿŃäÿĂèŁĆäžĘèğčæŻť'åď ŽåĚşäžŐå■ŮçňęäÿšæÿĚçŘĘçŽĎåĘĚåőźãĂĆ

4.12 2.12 åőąæ§ěæÿĚçŘĘæŰĞæIJňå■Ůçňęäÿš

éŮőéćŸ

äÿĂäžŻæŮäěĄŁçŽĎåźijçÍŽéżŚåőćåIJÍä¡ăçŽĎç¡ŚçńŹéąţéÍćèąĺå■Ţäÿ■è¿ŞåĚěæŰĞæIJňâĂÍpÃ¡tÄěÃűÃś

èğčåĘşæŰźæąĹ

æŰĞæIJňæÿĚçŘĘéŮőéćŸäijŽæűL'åŔŁåĹřåŇĚæŃňæŰĞæIJňèğčæđŘäÿŐæŢřæ■őåď'ĎçŘĘç■L'äÿĂçşżå
åIJÍéÍđåÿÿçőĂå■ŢçŽĎæČĚå¡ćäÿŃïijŇä¡ăåŘŕèČ¡äijŽéĂL'æŃl'ä¡£çŤĺå■ŮçňęäÿšåĞ¡æŢř(æŕŤåęĆ
`str.upper()` åŠŇ `str.lower()`)åřĘæŰĞæIJňè¡ňäÿžæăĞåĞĘæăijåijŘãĂĆ ä¡£çŤĺ
`str.replace()` æĹŰèĂĚ `re.sub()` çŽĎçőĂå■ŢæŻ£æ■ćæŞ■ä¡IJèČ¡åĹăéŹď'æĹŰèĂĚæŤźåŘŸæŇĞåőŹ
ä¡ăåŘŇæăůè£ŸåŘŕäžěä¡£çŤĺ2.9åřŘèŁĆçŽĎ `unicodedata.normalize()`
åĞ¡æŢřåřĘunicodeæŰĞæIJňæăĞåĞĘåŇŰãĂĆ

çĎűåŘŐïijŇæIJL'æŮűåĂŹä¡ăåŘŕèČ¡è£ŸæČşåIJÍæÿĚçŘĘæŞ■ä¡IJäÿŁæŻť'è£ŻäÿĂæ■ěãĂĆæŕŤåęĆïijŇä¡
äÿžäžĘè£ŹæăůåĄŽïijŇä¡ăåŘŕäžěä¡£çŤĺçżŘåÿÿäijŽèćńå£¡èğĘçŽĎ `str.translate()`
æŰźæşŢãĂĆ äÿžäžĘæijŤçď'žïijŇåĄĞèő¿ä¡ăçŐřåIJÍæIJL'äÿŃéÍćè£ŹäÿłåĞŇäźśçŽĎå■ŮçňęäÿšïijŽ

```
>>> s = 'pÃ¡tÄěÃűÃś\fis\tawesome\r\n'
>>> s
'pÃ¡tÄěÃűÃś\x0cis\tawesome\r\n'
>>>
```

çňňäÿĂæ■ěæŸŕæÿĚçŘĘçl'žçŹ¡å■ŮçňęäĂĆäÿžäžĘè£ŹæăůåĄŽïijŇåĚĹåĹŻåżžäÿĂäÿłåřŘçŽĎè¡ňæ■ćèąĺ
`translate()` æŰźæşŢïijŽ

```
>>> remap = {
...     ord('\t') : ' ',
...     ord('\f') : ' ',
...     ord('\r') : None # Deleted
... }
>>> a = s.translate(remap)
>>> a
'pÃ¡tÄěÃűÃś is awesome\n'
>>>
```

æ■čåęĆä¡ăçIJŃçŽĎéĆčæăůïijŇçl'žçŹ¡å■Ůçňę `\t` åŠŇ `\f`
åůšçżŘèćńéĞ■æŰřæŸäářĎåĹřäÿĂäÿłçl'žæăijäĂĆåŻđè¡ęå■ŮçňęrçŻť'æŐěèćńåĹăéŹď'ãĂĆ

ä¡ăåŘŕäžěäžěè£Źäÿłèąĺæăijäÿžå§žçąĂè£ŻäÿĂæ■ěæđĎåżžæŻť'åď'ğçŽĎèąĺæăijäĂĆæŕŤåęĆïijŇèől'æĹŚä

```
>>> import unicodedata
>>> import sys
>>> cmb_chrs = dict.fromkeys(c for c in range(sys.maxunicode)
...                          if unicodedata.combining(chr(c)))
...
>>> b = unicodedata.normalize('NFD', a)
>>> b
```

(continues on next page)

(continued from previous page)

```
'pÃ¡tÄěÃűÃś is awesome\n'
>>> b.translate(cmb_chrs)
'python is awesome\n'
>>>
```

äÿŁéÍćä¿Ńå■Řäÿ■ïijŇéĂŽè£Ğä¡£çŤÍ dict.fromkeys() æŰźæşŢæđĎéĂääÿĂäÿłå■ŮåĚÿïijŇæŕŘäÿłUnicodeåŠŇé§şçňęä¡IJäÿžéŤőïijŇåŕźåžŤçŽĎåĂijåĚíéČÍäÿž None ãĂĆ

çĎűåŘŐä¡£çŤÍ unicodedata.normalize() åřĘåŐ§åğŃè¿ŞåĚěæăĞåĞĘåŇŰäÿžåĹĘèğčå¡ćåijŘå■ çĎűåŘŐåĘ■èřČçŤÍ translate åĞ¡æŢřåĹăéŹď æĽĂæIJĽéĞ■é§şçňęãĂĆ åŘŇæăůçŽĎæŁĂæIJŕäź§åŘŕäżěèćńçŤÍæİěåĹăéŹď åĚűäżŰçśżåđŃçŽĎå■Ůçňę(æŕŤåęĆæŐğåĹűå■Ůçňęç■Ľ)ã

ä¡IJäÿžåŘęäÿĂäÿłä¿Ńå■ŘïijŇè£ŹéĞŇæđĎéĂääÿĂäÿłåřĘæĽĂæIJĽUnicodeæŢřå■Ůå■ŮçňęæŸäåřĎåĹ

```
>>> digitmap = { c: ord('0') + unicodedata.digit(chr(c))
...         for c in range(sys.maxunicode)
...         if unicodedata.category(chr(c)) == 'Nd' }
...
>>> len(digitmap)
460
>>> # Arabic digits
>>> x = '\u0661\u0662\u0663'
>>> x.translate(digitmap)
'123'
>>>
```

åŘęäÿĂçğ■æÿĚçŘĘæŰĞæIJňçŽĎæŁĂæIJŕæűĽåŘŁåĹřI/Oèğčçă ĄäÿŐçij Ůçă ĄåĞ¡æŢřãĂĆè£ŹéĞŇçŽĎ çĎűåŘŐåĘ■çżŞåŘĹ encode() æĹŰèĂĚ decode() æŞ■ä¡IJæİěæÿĚéŹď æĹŰä£őæŤźåőČãĂĆæŕŤåęĆïijŽ

```
>>> a
'pÃ¡tÄěÃűÃś is awesome\n'
>>> b = unicodedata.normalize('NFD', a)
>>> b.encode('ascii', 'ignore').decode('ascii')
'python is awesome\n'
>>>
```

è£ŹéĞŇçŽĎæĞåĞĘåŇŰæŞ■ä¡IJåřĘåŐ§æİěçŽĎæŰĞæIJňåĹĘèğčäÿžå■ŢçŃňçŽĎåŠŇé§şçňęãĂĆæŐě å¡ŞçĎűïijŇè£Źçğ■æŰźæşŢäżĚäżĚåŘłåIJĺæIJĂåŘŐçŽĎçŻőæĂĞåŕśæŸŕèŐůåŘŰåĹřæŰĞæIJňåŕźåžŤACSIIèą

èőĺèőž

æŰĞæIJňå■ŮçňęæÿĚçŘĘäÿĂäÿłæIJĂäÿżèęĄçŽĎéŰőéćŸåžŤèŕěæŸŕè£ŘèąŇçŽĎæĂğèČ¡ãĂĆäÿĂèĹňæ åŕźäžŐçőĂå■ŢçŽĎæŻ£æ■ćæŞ■ä¡IJïijŇ str.replace() æŰźæşŢéĂŽåÿÿæŸŕæIJĂå£ńçŽĎïijŇçŤŽèĞşåIJ æŕŤåęĆïijŇäÿžäžĘæÿĚçŘĘçľžçŹ¡å■ŮçňęïijŇä¡ăåŘŕäżěè£ŹæăůåĄŽïijŽ

```
def clean_spaces(s):
    s = s.replace('\r', '')
```

(continues on next page)

(continued from previous page)

```
    s = s.replace('\t', ' ')
    s = s.replace('\f', ' ')
    return s
```

åęĆæđIJä¡ăåŐżæţŃèŕŢçŽĎèŕÍïijŇä¡ăåřśäijŽåŘŚçŐřè£Źçğ■æŰźåijŘäijŽæŕŤä¡£çŤÍ `translate()` æĹŰèĂĚæ■čåĹŹèąĺè¿¿åijŘèęĄå£ńå¿Ĺåď ŽãĂĆ

åŘęäÿĂæŰźéÍćïijŇåęĆæđIJä¡ăéIJĂèęĄæL'ğèąŇäżżä¡Ţåď■æÍĆå■Ůçňęåŕźå■ŮçňęçŽĎéĞ■æŰřæŸăåřĎæ `tanslate()` æŰźæşŢäijŽéÍđåÿÿçŽĎå£ńãĂĆ

äżŐåď ğçŽĎæŰźéÍćæÍěèőšïijŇåŕźäžŐä¡ăçŽĎåžŤçŤÍçÍŃåžŘæÍěèŕť æĂğèČ¡æŸŕä¡ăäÿ■å¿Ůäÿ■åŐżèĞłåůs äÿ■åźÿçŽĎæŸŕïijŇæĹŚäżňäÿ■åŘŕèČ¡çżŹä¡ăåżżèőőäÿĂäÿłçL'źåőŽçŽĎæŁĂæIJŕïijŇä¡£åőČèČ¡åď §éĂĆåžŤa åŻăæ■ď åőđéŹĚæČĚåĘţäÿ■éIJĂèęĄä¡ăèĞłåůšåŐżåřÍèŕŢäÿ■åŘŃçŽĎæŰźæşŢåźűèŕĎäijřåőČãĂĆ

ář¡çőąè£ŹäÿĂèŁĆéŻĘäÿ■èőÍèőžçŽĎæŸŕæŰĞæIJňïijŇä¡ĘæŸŕçśżäijijçŽĎæŁĂæIJŕäź§åŘŕäżěéĂĆçŤÍäž

4.13 2.13 å■Ůçňęäÿšåŕźé¡Ř

éŮőécŸ

ä¡ăæČşéĂŽè£Ğæ§Řçğ■åŕźé¡ŘæŰźåijŘæÍěæäijåijŘåŃŰå■Ůçňęäÿš

èğčåĘşæŰźæąĹ

åŕźäžŐå§žæIJňçŽĎå■Ůçňęäÿšåŕźé¡ŘæŞ■ä¡IJïijŇåŘŕäżěä¡£çŤÍå■ŮçňęäÿšçŽĎ `ljust()` , `rjust()` åŠŃ `center()` æŰźæşŢãĂĆæŕŤåęĆïijŽ

```
>>> text = 'Hello World'
>>> text.ljust(20)
'Hello World         '
>>> text.rjust(20)
'         Hello World'
>>> text.center(20)
'    Hello World     '
>>>
```

æL'ĂæIJL'è£ŹäžŻæŰźæşŢéČ¡èČ¡æŐěåŘŮäÿĂäÿłåŘŕéĂL'çŽĎåąńåĚĚå■ŮçňęäĂĆæŕŤåęĆïijŽ

```
>>> text.rjust(20,'=')
'=========Hello World'
>>> text.center(20,'*')
'****Hello World*****'
>>>
```

åĞ¡æŢř `format()` åŘŃæăůåŘŕäżěçŤÍæÍěå¿Ĺåőźæ ŸŞçŽĎåŕźé¡Řå■ŮçňęäÿšäĂĆ ä¡ăèęĄåĄŽçŽĎåřśæŸŕä¡£çŤÍ <, > æĹŰèĂĚ ^ å■Ůçňęå ŘŐéÍćçť ğèů§äÿĂäÿłæŃĞåőŽçŽĎåő¡åžęäĂĆæŕŤåęĆ

```
>>> format(text, '>20')
'         Hello World'
>>> format(text, '<20')
'Hello World         '
>>> format(text, '^20')
'    Hello World     '
>>>
```

åęĆæđIJä¡ăæČşæŇĞåőŽäÿĂäÿłéİđçľžæăijçŽĎåąńåĚĚå■Ůçňęïij ŇåřĘåőČåĘŹåĹřåŕźé¡Řå■ŮçňęçŽĎåĽ■

```
>>> format(text, '=>20s')
'=========Hello World'
>>> format(text, '*^20s')
'****Hello World*****'
>>>
```

å¡Şæăijåij ŕåŇŰåď Žäÿłå Ăijç ŽĎæŮűåĂŹïij Ňè£ŹäžŻæăijåij ŕäžčçăĄäź§åŕŕäżěèćńçŤĺåIJĺ format() æŰźæşŢäÿ■ãĂĆæŕŤåęĆïijŽ

```
>>> '{:>10s} {:>10s}'.format('Hello', 'World')
'     Hello      World'
>>>
```

format() åĞ¡æŢřçŽĎäÿĂäÿłåě¡åď ĎæŸŕåőČäÿ■äžĚéĂĆçŤĺäžŐå■ŮçňęäÿšãĂĆåőČåŕŕäżěçŤĺæİěæăij æŕŤåęĆïijŇä¡ăåŕŕäżěçŤĺåőČæİěæăijåij ŕåŇŰæŢřå■ŮïijŽ

```
>>> x = 1.2345
>>> format(x, '>10')
'    1.2345'
>>> format(x, '^10.2f')
'   1.23   '
>>>
```

èőĺèőž

åIJĺèĂçŽĎäžčçăĄäÿ■ïijŇä¡ăçżŕåÿÿäijŽçIJŃåĹřèćńçŤĺæİěæăijåij ŕåŇŰæŰĞæIJňçŽĎ % æŞ■ä¡IJçňęãĂĆæŕŤåęĆïijŽ

```
>>> '%-20s' % text
'Hello World         '
>>> '%20s' % text
'         Hello World'
>>>
```

ä¡ĘæŸŕïijŇåIJĺæŰřçĽĹæIJňäžčçăĄäÿ■ïijŇä¡ăåžŤèŕěäijŸåĚĹéĂĽæŃľ format() åĞ¡æŢřæĹŰèĂĚæŰźæşŢãĂĆ format() èęĄæŕŤ % æŞ■ä¡IJçňęçŽĎåŁ§èČ¡æŻť äÿžåijžåď ğãĂĆ åźűäÿŤ format() äź§æŕŤä¡£çŤĺ ljust() , rjust() æĹŰ center() æŰźæşŢæŻť éĂŽçŤĺïijŇ åŻääÿžåőČåŕŕäżěçŤĺæİěæăijåij ŕåŇŰäżżæĎŕåŕźèśąïijŇèĂŇäÿ■äžĚäżĚæŸŕå■ŮçňęäÿšãĂĆ

åęĆæđIJæČşèęĄåőŇåĚíäžĘèğč `format()` åĞ¡æŢřçŽĎæIJĽçŤíçĽźæĂğïijŇèŕůåŘĆèĂČ åIJÍçž£PythonæŰĞæąč

4.14 2.14 åŘĹåźűæŃijæŐěå■Ůçňęäÿš

éŮőécŸ

ä¡ăæČşåřĘåĞääÿłåřŘçŽĎå■ŮçňęäÿšåŘĹåźűäÿžäÿĂäÿłåď ğçŽĎå■Ůçňęäÿš

èğčåĘşæŰźæąĹ

åęĆæđIJä¡ăæČşèęĄåŘĹåźűçŽĎå■ŮçňęäÿšæŸŕåIJÍäÿĂäÿłåžŘåĹŮæĹŰèĂĚ `iterable` äÿ■ïijŇéĆčäźĹæIJĂå£ńçŽĎæŰźåijŘåřśæŸŕä¡£çŤÍ `join()` æŰźæşŢãĂĆæŕŤåęĆïijŽ

```
>>> parts = ['Is', 'Chicago', 'Not', 'Chicago?']
>>> ' '.join(parts)
'Is Chicago Not Chicago?'
>>> ','.join(parts)
'Is,Chicago,Not,Chicago?'
>>> ''.join(parts)
'IsChicagoNotChicago?'
>>>
```

åĹíçIJŃèţůæİěïijŇè£Źçğ■èŕ■æşŢçIJŃäÿŁåŐżäijŽæŕŤè¿ČæĂłïijŇä¡ĘæŸŕ `join()` èćńæŇĞåőŽäÿžå■ŮçňęäÿšçŽĎäÿĂäÿłæŰźæşŢãĂĆ è£ŹæăůåĄŽçŽĎéČĺåĹĘåŐ§åŻăæŸŕä¡ăæČşåŐżè£đæŐěçŽĎåŕźèśąåŘŕèČ¡æİěèĞłåŘĎçğ■äÿ■åŘŇçŽĎæŢřæ■ åęĆæđIJåIJÍæĽĂæIJĽè£ŹäžŽåŕźèśąäÿŁéČ¡åőŽäźĽäÿĂäÿł `join()` æŰźæşŢæŸŐæŸ¿æŸŕåĘŮä¡ŹçŽĎãĂĆ åŻăæ■ď ä¡ăåŘłéIJĂèęĄæŇĞåőŽä¡ăæČşèęĄçŽĎåĹĘåĽšå■Ůçňęäÿšåź `join()` æŰźæşŢåŐżåřĘæŰĞæIJňçĽĞæőţçżĎåŘĹèţůæİěãĂĆ

åęĆæđIJä¡äżĚäżĚåŘłæŸŕåŘĹåźűåřŚæŢřåĞääÿłå■ŮçňęäÿšïijŇä¡£çŤÍåŁäåŘů(+)éĂŽåÿÿåůšçżŘèűşåď §ä

```
>>> a = 'Is Chicago'
>>> b = 'Not Chicago?'
>>> a + ' ' + b
'Is Chicago Not Chicago?'
>>>
```

åŁäåŘů(+)æŞ■ä¡IJçňęåIJÍä¡IJäÿžäÿĂäžŽåď ■æİĆå■ŮçňęäÿšæăijåijŘåŇŰçŽĎæŻ£äžčæŰźæąĹçŽĎæŮűå

```
>>> print('{} {}'.format(a,b))
Is Chicago Not Chicago?
>>> print(a + ' ' + b)
Is Chicago Not Chicago?
>>>
```

åęĆæđIJä¡ăæČşåIJÍæžŘçăĄäÿ■åřĘäÿď äÿłå■ŮéĺĆå■ŮçňęäÿšåŘĹåźűèţůæİěïijŇä¡äåŘłéIJĂèęĄçőÄå■ŢçŽ

```
>>> a = 'Hello' 'World'
>>> a
'HelloWorld'
>>>
```

èőÍèőž

å■Ůçňęäÿšå ŘĹåźűåŘŕèČ¡çIJŃäÿŁåŐżåźűäÿ■éIJĂèęĄçŤÍäÿĂæŢť'èŁĆæÍěèőÍèőžãĂĆ
ä¡ĘæŸŕäÿ■åžŤèŕěåřŘçIJŃè£ŹäÿłéŮőéćŸïijŇçÍŃåžŘåŚŸéĂŽåÿÿåIJÍå■ŮçňęäÿšæäijåijŘåŇŰçŽĎæŮűåĂŹåŻ

æIJĂéĞ■èęĄçŽĎéIJĂèęĄåijŢèţůæşÍæĎŘçŽĎæŸŕïijŇå¡ŞæĹŚäżňä¡£çŤÍåŁååŘů(+)æŞ■ä¡IJçňęåŐżè£đæ
åŻääÿžåŁååŘůè£đæŐěäijŽåijŢèţůåĘĚå■Ÿåď■åĹűäżěåŘŁåđČåIJ¿åŻđæŤűæŞ■ä¡IJãĂĆ
çĽźåĹńçŽĎïijŇä¡äæřÿè£IJéČ¡äÿ■åžŤåČŘäÿŃéÍćè£ŹæäůåĘŹå■Ůçňęäÿšè£đæŐěäżčçäĄïijŽ

```
s = ''
for p in parts:
    s += p
```

è£Źçğ■åĘŹæşŢäijŽæŕŤä¡£çŤÍ `join()` æŰźæşŢè£ŘèąŃçŽĎèęĄæĚćäÿĂäžŻïijŇåŻääÿžæŕŘäÿĂæňąæĽ
ä¡äæIJĂåě¡æŸŕåĚĹæŤűéŻĘæĽĂæIJĽçŽĎå■Ůçňęäÿšç L'ĞæőţçĎűåŘŐåĘ■åřĘåőČäžňè£đæŐěèţůæÍěãĂĆ

äÿĂäÿłçŻÿåŕźæŕŤè¿ČèĄłæŸŐçŽĎæŁĂåůğæŸŕåĹl'çŤÍçŤ§æĹŘåŹÍèąÍè¿¿åijŘ(åŘĆèĂČ1.19åřŘèŁĆ)è¡ňa

```
>>> data = ['ACME', 50, 91.1]
>>> ','.join(str(d) for d in data)
'ACME,50,91.1'
>>>
```

åŘŇæăůè£Ÿå¿ŮæşÍæĎŘäÿ■å£ĚèęĄçŽĎå■Ůçňęäÿšè£đæŐěæŞ■ä¡IJãĂĆæIJL'æŮűåĂŹçÍŃåžŘåŚŸåIJÍæ

```
print(a + ':' + b + ':' + c) # Ugly
print(':'.join([a, b, c])) # Still ugly
print(a, b, c, sep=':') # Better
```

å¡ŞæůůåŘĹä¡£çŤÍI/OæŞ■ä¡IJåŠŇå■Ůçňęäÿšè£đæŐěæŞ■ä¡IJçŽĎæŮűåĂŹïijŇæIJL'æŮűåĂŹéIJĂèęĄäż
æŕŤåęĆïijŇèĂČèŹŚäÿŃéÍćçŽĎäÿď'çňŕäżčçäĄçL'ĞæőţïijŽ

```
# Version 1 (string concatenation)
f.write(chunk1 + chunk2)

# Version 2 (separate I/O operations)
f.write(chunk1)
f.write(chunk2)
```

åęĆæđIJäÿď'äÿłå■Ůçňęäÿšå¿ĹåřŘïijŇéĆčäźĹçňňäÿĂäÿłçL'ĹæIJňæĂğèČ¡äijŽæŻť'åě¡äžŻïijŇåŻääÿžI/Oç
åŘęåď'ŰäÿĂæŰźéÍćïijŇåęĆæđIJäÿď'äÿłå■Ůçňęäÿšå¿Ĺåď'ğïijŇéĆčäźĹçňňäžŇäÿłçL'ĹæIJňåŘŕèČ¡äijŽæŻť'åŁ
åŻääÿžåőČéĄ£åĚ■äžĘåĹŻåżžäÿĂäÿłå¿Ĺåď'ğçŽĎäÿť'æŮűçżŞæđIJåźűäÿŤèęĄåď'■åĹűåď'ğéĞŘçŽĎåĘĚå■Ÿå
è£ŸæŸŕéĆčåŘěèŕÍïijŇæIJL'æŮűåĂŹæŸŕéIJĂèęĄæäźæ■őä¡äçŽĎåžŤçŤÍçÍŃåžŘçL'źçĆźæÍěåĘşåőŽåžŤèŕěä¡£

æIJĂåŘŐèřĹäÿĂäÿŃïijŇåęĆæđIJä¡ăåĞĘåď ĞçijŰåĘŹæđĎåżžåď ğéĞŘåřŘå■ŮçňęäÿšçŽĎè¿ŞåĞžäżčçăA
ä¡ăæIJĂåě¡èĂČèŹŚäÿŃä¡£çŤĺçŤ§æĹŘåŹĺåĞ¡æŢřïijŇåĹľçŤĺyieldèŕ■åŘěäžğçŤ§è¿ŞåĞžçL'Ğæőţ̃ăĂĆæŕŤåęĆ

```
def sample():
    yield 'Is'
    yield 'Chicago'
    yield 'Not'
    yield 'Chicago?'
```

è£Źçğ■æŰźæşŢäÿĂäÿłæIJL'èűčçŽĎæŰźéĺćæŸŕåőČåźűæšąæIJL'åŕźè¿ŞåĞžçL'Ğæőţ̃åĹřåžŢèęĄæĂŐæăů
ä¿ŃåęĆïijŇä¡ăåŘŕäżěçőĂå■ŢçŽĎä¡£çŤĺ `join()` æŰźæşŢ̃åřĘè£ŹäžŻçL'Ğæőţ̃åŘĹåźűèţ̃ůæĺěïijŽ

```
text = ''.join(sample())
```

æĹŰèĂĚä¡ăäź§åŘŕäżěåřĘå■Ůçňęäÿšçl'Ğæőţ̃éĞ■åőŽåŘŚåĹřI/Oïijž

```
for part in sample():
    f.write(part)
```

åĘ■æĹŰèĂĚä¡ăè£ŸåŘŕäżěåĘŹåĞžäÿĂäžŻçżŞåŘĹI/OæŞ■ä¡IJçŽĎæůůåŘĹæŰźæąĹïijŽ

```
def combine(source, maxsize):
    parts = []
    size = 0
    for part in source:
        parts.append(part)
        size += len(part)
        if size > maxsize:
            yield ''.join(parts)
            parts = []
            size = 0
    yield ''.join(parts)

# çżŞåŘĹæŰĞäżűæŞ■ä¡IJ
with open('filename', 'w') as f:
    for part in combine(sample(), 32768):
        f.write(part)
```

è£ŹéĞŃçŽĎåĚşéŤőçĆźåIJĺäžŐåŐ§ăğŃçŽĎçŤ§æĹŘåŹĺåĞ¡æŢřäźűäÿ■éIJĂèęĄç§ěéĄŞä¡£çŤĺçżĘèŁĆïij

4.15 2.15 å■Ůçňęäÿšäÿ■æŘŞåĚěåŘŸéĞŘ

éŮőécŸ

ä¡ăæČşåĹŻäżžäÿĂäÿłåĘĚåţŇåŘŸéĞŘçŽĎå■ŮçňęäÿšïijŇåŘŸéĞŘèćńäőČçŽĎåĂijæL'Ăèąĺçď žçŽĎå■Ů

èğčåĘşæŰźæąĹ

PythonåźűæšąæIJL’åŕźåIJÍå■Ůçňęäÿšäÿ■çőĂå■ŢæŻ£æ■ćåŘŸéĞŘåĂijæŘŘä¿ŻçŻť’æŐěçŽĎæŤŕæŇĄãĂ
ä¡ĘæŸŕéĂŽè£Ğä¡£çŤÍå■Ůçňęäÿšç ŽĎ `format()` æŰźæşŢæİěèğčåĘşè£ŹäÿłéŮőéćŸãĂĆæŕŤåęĆïijŽ

```
>>> s = '{name} has {n} messages.'
>>> s.format(name='Guido', n=37)
'Guido has 37 messages.'
>>>
```

æĹŰèĂĚïijŇåęĆæđIJèęĄèćńæŻ£æ■ćçŽĎåŘŸéĞŘèČ¡åIJÍåŘŸéĞŘå§§äÿ■æL’¿åĹřïijŇ
éĆčäźĹä¡ääŘŕäżěçżŞåŘĹä¡£çŤÍ `format_map()` åŠŇ `vars()`
ãĂĆåřśåČŘäÿŃéİćè£ŹæăůïijŽ

```
>>> name = 'Guido'
>>> n = 37
>>> s.format_map(vars())
'Guido has 37 messages.'
>>>
```

`vars()` è£ŸæIJL’äÿĂäÿłæIJL’æĎŘæĂİçŽĎçL’źæĂğåřśæŸŕåőČäź§éĂĆçŤÍäžŐåŕźèśąåőđä¿ŃãĂĆæŕŤåę

```
>>> class Info:
...     def __init__(self, name, n):
...         self.name = name
...         self.n = n
...
>>> a = Info('Guido',37)
>>> s.format_map(vars(a))
'Guido has 37 messages.'
>>>
```

`format` åŠŇ `format_map()` çŽĎäÿĂäÿłçijžéŹůåřśæŸŕåőČäžňåźűäÿ■èČ¡å¿Ĺåě¡çŽĎåď’ĎçŘĘåŘŸéĞ

```
>>> s.format(name='Guido')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 'n'
>>>
```

äÿĂçğ■éĄ£åĚ■è£Źçğ■éŤŹèŕŕçŽĎæŰźæşŢæŸŕåŘęåď’ŰåőŽäźL’äÿĂäÿłåŘńæIJL’
`__missing__()` æŰźæşŢçŽĎå■ŮåĚÿåŕźèśąïijŇåřśåČŘäÿŃéİćè£ŹæăůïijŽ

```
class safesub(dict):
"""éŸšæ■ćkeyæL’¿äÿ■åĹř"""
def __missing__(self, key):
    return '{' + key + '}'
```

çŐřåIJÍä¡ääŘŕäżěåĹľçŤÍè£ŹäÿłçśżåŇĚèčĚè¿ŞåĚěåŘŐäijäéĂŠçżŹ `format_map()` ïijŽ

```
>>> del n # Make sure n is undefined
>>> s.format_map(safesub(vars()))
'Guido has {n} messages.'
>>>
```

åęĆæđIJä¡ăåŘŚçŐřèĞłåůśåIJÍäžčçăĄäÿ■éćŚçźĄçŽĎæĽ'ğèąŇè£ŹäžŻæ■ěéłď'ïijŇä¡ăåŘŕäžěåřĘåŘŸéĞŘa

```
import sys

def sub(text):
    return text.format_map(safesub(sys._getframe(1).f_locals))
```

çŐřåIJÍä¡ăåŘŕäžěåČŘäÿŃéİćè£ŹæăůåĘŹäžĘïijŽ

```
>>> name = 'Guido'
>>> n = 37
>>> print(sub('Hello {name}'))
Hello Guido
>>> print(sub('You have {n} messages.'))
You have 37 messages.
>>> print(sub('Your favorite color is {color}'))
Your favorite color is {color}
>>>
```

èőÍèőž

åď'Žåźť'äžěæİěçŤśäžŐPythonçijžäźŘåŕźåŘŸéĞŘæŻ£æ■ćçŽĎåĘĚç¡őæŤŕæŇĄèĂŇåŕijèĞť'äžĘåŘĎçğ■äÿ
ä¡IJäÿžæIJňèŁĆäÿ■åśŢçď'žçŽĎäÿĂäÿłåŘŕèČ¡çŽĎèğčåĘşæŰźæąĹïijŇä¡ăåŘŕäžěæIJL'æŮűåĂŹäijŽçIJŃåĹřåČ

```
>>> name = 'Guido'
>>> n = 37
>>> '%(name) has %(n) messages.' % vars()
'Guido has 37 messages.'
>>>
```

ä¡ăåŘŕèČ¡è£ŸäijŽçIJŃåĹřå■Ůçňęäÿšæĺąæİ£çŽĎä¡£çŤĺïijŽ

```
>>> import string
>>> s = string.Template('$name has $n messages.')
>>> s.substitute(vars())
'Guido has 37 messages.'
>>>
```

çĎűèĂŇïijŇ `format()` åŠŇ `format_map()` çŻÿæŕŤè¿ČäÿŁéİćè£ŹäžŻæŰźæąĹèĂŇåůšæŻť'åŁăåĚĹ
ä¡£çŤĺ `format()` æŰźæşŢè£ŸæIJL'äÿĂäÿłåě¡åď'ĎåŕśæŸŕä¡ăåŘŕäžěèŐůå¿Ůůŕźåĺ■ŮçňęäÿšæăijåijŘåŇŰçŽĎ
èĂŇè£ŹäžŻçL'źæĂğæŸŕä¡£çŤĺåČŘæĺąæİ£å■ŮçňęäÿšäźŇçśżçŽĎæŰźæąĹäÿ■åŘŕèČ¡èŐůå¿ŮçŽĎãĂĆ

æIJňæIJžè£ŸéČĺåĹĘäżŃçż■äžĘäÿĂäžŻéńŸçžğçL'źæĂğãĂĆæŸăářĎæĹŰèĂĚå■ŮůĚÿçśżäÿ■éšIJäÿžäžž
`__missing__()` æŰźæşŢåŘŕäžěèőľ'ä¡ăăőŽäźL'ăęĆä¡Ţåď'ĎçŘĘçijžåď'śçŽĎåĂijãĂĆ åIJÍ

SafeSub çśżäÿ■ïijŇè£ŹäÿłæŰźæşŢèćńåőŽäźL'äÿžåŕźçijžåď'śçŽĎåĂijè£ŤåŻđäÿĂäÿłå■ăä¡■çňęãĂĆ
ä¡ăåŘŕäżěåŘŚçŐřçijžåď'śçŽĎåĂijäijŽåĞžçŐřåIJÍçżŞæđIJå■Ůçňęäÿšäÿ■(åIJÍèřČèŕŢçŽĎæŮűåĂŹåŘŕèČ¡å¿Ła
KeyError åijĆåÿÿãĂĆ

sub() åĞ¡æŢřä¡£çŤÍ sys._getframe(1) è£ŤåŻđèřČçŤÍèĂĚçŽĎæăĹåÿğãĂĆåŘŕäżěäżŐäÿ■èő£éŮ
f_locals æĺěèŐůå¿ŮåśĂéČÍåŘŸéĞŘãĂĆ æŕńæŮăçŰŚéŮőçżĺåď'ğéČÍåĹĘæČĚåĘţäÿŇåIJÍäżčçăĄäÿ■åŐżç
ä¡ĘæŸŕïijŇåŕźäžŐåČŘå■ŮçňęäÿšæŻ£æ■ćåůěåĚůåĞ¡æŢřèĂŇèÍĂåőČæŸŕéĺđåÿÿæIJL'çŤÍçŽĎãĂĆ
åŘęåď'ŰïijŇåĂijå¿ŮæşÍæĎŘçŽĎæŸŕ f_locals æŸŕäÿĂäÿłåď'■åĹűèřČçŤÍåĞ¡æŢřçŽĎæIJňåIJřåŘŸéĞŘçŽ
ář¡çőąä¡ăåŘŕäżěæŤźåŘŸ f_locals çŽĎåĘĚåőźïijŇä¡ĘæŸŕè£ŹäÿłäŁőæŤźåŕźäžŐåŘŐéĺćçŽĎåŘŸéĞŘèő£é
æL'Ăäżěïij ŇèŹ¡èŕť'èő£éŮőäÿĂäÿłæăĹåÿğçIJŃäÿŁåŐżå¿ĹéĆłæĄűïijŇä¡ĘæŸŕåŕźåőČçŽĎäżżä¡ŢæŞ■ä¡IJäÿ■ä

4.16 2.16 äżěæŇĞåőŽåĹŮåő¡æăijåijŘåŇŰå■Ůçňęäÿš

éŮőécŸ

ä¡ăæIJL'äÿĂäžŻéŢ£å■Ůçňęäÿšïij ŇæČşäżěæŇĞåőŽçŽĎåĹŮåő¡åřĘåőČäżňéĞ■æŰřæăijåijŘåŇŰãĂĆ

èğčåĘşæŰźæąĹ

ä¡£çŤÍ textwrap æĺąåĺŮæĺěæăijåijŘåŇŰå■ŮçňęäÿšçŽĎè¿ŞåĞžãĂĆæŕŤåęĆïijŇåĄĞåęĆä¡ăæIJL'äÿŇå

```
s = "Look into my eyes, look into my eyes, the eyes, the eyes, \
the eyes, not around the eyes, don't look around the eyes, \
look into my eyes, you're under."
```

äÿŃéĺćæijŤçď'žä¡£çŤÍ textwrap æăijåijŘåŇŰå■ŮçňęäÿšçŽĎåď'Žçğ■æŰźåijŘïijŽ

```
>>> import textwrap
>>> print(textwrap.fill(s, 70))
Look into my eyes, look into my eyes, the eyes, the eyes, the eyes,
not around the eyes, don't look around the eyes, look into my eyes,
you're under.

>>> print(textwrap.fill(s, 40))
Look into my eyes, look into my eyes,
the eyes, the eyes, the eyes, not around
the eyes, don't look around the eyes,
look into my eyes, you're under.

>>> print(textwrap.fill(s, 40, initial_indent='    '))
    Look into my eyes, look into my
eyes, the eyes, the eyes, the eyes, not
around the eyes, don't look around the
eyes, look into my eyes, you're under.

>>> print(textwrap.fill(s, 40, subsequent_indent='    '))
Look into my eyes, look into my eyes,
    the eyes, the eyes, the eyes, not
```

(continues on next page)

(continued from previous page)

```
    around the eyes, don't look around
    the eyes, look into my eyes, you're
    under.
```

èőÍèőž

textwrap æÍąåİŮåŕźäžŐå■ŮçňęäÿšæĽŞå■řæŸŕéİđåÿÿæIJĽçŤÍçŽĎïijŇçĽźåĹńæŸŕå¡Şä¡ăåÿŇæIJŻè¿Ş
ä¡ăåŘŕäžěä¡£çŤÍ os.get_terminal_size() æŰźæşŢæİěèŐůåŘŰçżĹçńŕçŽĎåď ğåřŘåřžåŕÿãĂĆæŕŤåęĆ

```
>>> import os
>>> os.get_terminal_size().columns
80
>>>
```

fill() æŰźæşŢæŐěåŘŮäÿĂäžŻåĚűäżŰåŘŕéĂĽåŘĆæŢřæİěæŐğåĹűtabïijŇèŕ■åŘěçżŞåř¿ç■ĽãĂĆ
åŘĆéŸĚ textwrap.TextWrapperæŰĞæąč èŐůåŘŰæŻť åď ŽåĘĚåőźãĂĆ

4.17 2.17 åIJĺå■Ůçňęäÿšäÿ■åď ĎçŘĘhtmlåŠŇxml

éŮőéćŸ

ä¡ăæČşåřĘHTMLæĹŰèĂĚXMLåőđä¡ŞåęĆ &entity; æĹŰ &#code;
æŻ£æ■ćäÿžåŕźåžŤçŽĎæŰĞæIJňãĂĆ åĘ■èĂĚïijŇä¡ăéIJĂèęĄè¡ňæ■ćæŰĞæIJňäÿ■çĽźåőŽçŽĎå■Ůçňę(æŕŤåę
>, æĹŰ &)ãĂĆ

èğčåĘşæŰźæąĹ

åęĆæđIJä¡ăæČşæŻ£æ■ćæŰĞæIJňå■Ůçňęäÿšäÿ■çŽĎ âĂŸ<âĂŹ æĹŰèĂĚ âĂŸ>âĂŹ
ïijŇä¡£çŤÍ html.escape() åĞ¡æŢřåŘŕäžěå¿ĹåőźæŸ ŞçŽĎåőŇæĹŘãĂĆæŕŤåęĆïijŽ

```
>>> s = 'Elements are written as "<tag>text</tag>".'
>>> import html
>>> print(s)
Elements are written as "<tag>text</tag>".
>>> print(html.escape(s))
Elements are written as &quot;&lt;tag&gt;text&lt;/tag&gt;&quot;.

>>> # Disable escaping of quotes
>>> print(html.escape(s, quote=False))
Elements are written as "&lt;tag&gt;text&lt;/tag&gt;".
>>>
```

åęĆæđIJä¡ăæ■čåIJĺåď ĎçŘĘçŽĎæŸŕASCIIæŰĞæIJňïijŇåźűäÿŤæČşåřĘéİđASCIIæŰĞæIJňåŕźåžŤçŽĎç
åŘŕäžěçżŹæ§ŘäžŻI/OåĞ¡æŢřäijăéĂŠåŘĆæŢř errors='xmlcharrefreplace'
æİěè¿¿åĹřè£ŹäÿłçŻőãĂĆæŕŤåęĆïijŽ

```
>>> s = 'Spicy JalapeÃśo'
>>> s.encode('ascii', errors='xmlcharrefreplace')
b'Spicy Jalape&#241;o'
>>>
```

äÿžäžĘæŻ£æ■ćæŰĞæIJňäÿ■çŽĎçijŰçăĄåőđä¡ŞïijŇä¡äéIJĂèęĄä¡£çŤĺåŘęåď ŰäÿĂçğ■æŰźæşŢãĂĆ
åęĆæđIJä¡äæ■čåIJĺåď ĎçŘĘHTMLæĹŰèĂĚXMLæŰĞæIJňïijŇèŕŢçİĂåĚĹä¡£çŤĺäÿĂäÿłåŘĹéĂĆçŽĎHTML
éĂŽåÿÿæČĚåĘţäÿŃïijŇè£ŹäžŻåůěåĚůäijŽèĞłåŁĺæŻ£æ■ćè£ŹäžŻçijŰçăĄåĂijïijŇä¡äæŮäéIJĂæŃĚå£ČãĂĆ

æIJL'æŮűåĂŹïijŇåęĆæđIJä¡äæŐěæŤűåĹřäžĘäÿĂäžŻåŘńæIJL'çijŰçăĄåĂijçŽĎåŐ§åğŃæŰĞæIJňïijŇéI
éĂŽåÿÿä¡ääŘłéIJĂèęĄä¡£çŤĺHTMLæĹŰèĂĚXMLèğčæđŘåŹĺçŽĎäÿĂäžŻçŻÿåĚşåůěåĚůåĞ¡æŢř/æŰźæşŢå■

```
>>> s = 'Spicy &quot;Jalape&#241;o&quot.'
>>> from html.parser import HTMLParser
>>> p = HTMLParser()
>>> p.unescape(s)
'Spicy "JalapeÃśo".'
>>>
>>> t = 'The prompt is &gt;&gt;&gt;'
>>> from xml.sax.saxutils import unescape
>>> unescape(t)
'The prompt is >>>'
>>>
```

èőĺèőž

åIJĺçŤ§æĹŘHTMLæĹŰèĂĚXMLæŰĞæIJňçŽĎæŮűåĂŹïijŇåęĆæđIJæ■čçąőçŽĎè¡ňæ■ćçL'źæőŁæăĞèć
çL'źåĹńæŸŕå¡Şä¡ăä¡£çŤĺ `print()` åĞ¡æŢřæĹŰèĂĚåĚűäžŰå■ŮçňęäÿšæăijåijŘåŇŰæİěäžğçŤ§è¿ŞåĞžçŽĎa
ä¡£çŤĺåČŘ `html.escape()` çŽĎåůěåĚůåĞ¡æŢřåŘŕäžěå¿Ĺåőźæ ŸŞçŽĎèğčåĘşè£ŹçśżéŮőéćŸãĂĆ

åęĆæđIJä¡äæČşäžěåĚűäžŰæŰźåijŘåď ĎçŘĘæŰĞæIJňïijŇè£ŸæIJL'äÿĂäžŻåĚűäžŰçŽĎåůěåĚůåĞ¡æŢřa
`xml.sax.saxutils.unescapge()` åŘŕäžěåÿőåŁľä¡ăăĂĆ
çĎűèĂŇïijŇä¡ääžŤèŕěåĚĹèřČçăŤæÿĚæěŽæĂŐæăůä¡£çŤĺäÿĂäÿłåŘĹéĂĆçŽĎèğčæđŘåŹĺăĂĆ
æŕŤåęĆïijŇåęĆæđIJä¡ääIJĺåď ĎçŘĘHTMLæĹŰXMLæŰĞæIJňïijŇ
ä¡£çŤĺæ§ŘäÿłèğčæđŘæĺąåİŮæŕŤåęĆ `html.parse` æĹŰ `xml.etree.ElementTree`
åůšçżŘåÿőä¡äèĞłåŁĺåď ĎçŘĘäžĘçŻÿåĚşçŽĎæŻ£æ■ćçżĘèŁĆãĂĆ

4.18 2.18 å■Ůçňęäÿšäžď çL'ŇèğčæđŘ

éŮőéćŸ

ä¡äæIJL'äÿĂäÿłå■ŮçňęäÿšïijŇæČşäžŐåůęèĞşåŘşåřĘåĚűèğčæđŘäÿžäÿĂäÿłäžď çL'ŇæţĄãĂĆ

èğčåĘşæŰźæąĹ

åĄĞåęĆä¡äæIJL'äÿŃéİćè£ŹæăůäÿĂäÿłæŰĞæIJňå■ŮçňęäÿšïijŽ

```
text = 'foo = 23 + 42 * 10'
```

äÿžäžĘäżď'çĽŇåŇŰå■ŮçňęäÿšïijŇä¡ääÿ■äżĚéIJĂèęĄåŇźéĚ■æĺąåijŘïijŇè£Ÿå¿ŮæŇĞåőŽæĺąåijŘçŽĎç
æŕŤåęĆïijŇä¡ääŘŕèČ¡æČşåřĘå■ŮçňęäÿšåČŘäÿŃéİćè£Źæäůè¡ňæ■ćäÿžäžŘåĹŮåŕźïijŽ

```
tokens = [('NAME', 'foo'), ('EQ','='), ('NUM', '23'), ('PLUS','+'),
          ('NUM', '42'), ('TIMES', '*'), ('NUM', '10')]
```

äÿžäžĘæĽ'ğèąŇè£ŹæäůçŽĎåĹĞåĹĘïijŇçňňäÿĂæ■ěåŕśæŸŕåČŘäÿŃéİćè£ŹæäůåĹl'çŤĺåŚ¡åŘ■æ■ŢèŐůçż

```
import re
NAME = r'(?P<NAME>[a-zA-Z_][a-zA-Z_0-9]*)'
NUM = r'(?P<NUM>\d+)'
PLUS = r'(?P<PLUS>\+)'
TIMES = r'(?P<TIMES>\*)'
EQ = r'(?P<EQ>=)'
WS = r'(?P<WS>\s+)'

master_pat = re.compile('|'.join([NAME, NUM, PLUS, TIMES, EQ, WS]))
```

åIJĺäÿŁéİćçŽĎæĺąåijŘäÿ■ïijŇ `?P<TOKENNAME>` çŤĺäžŐçżŹäÿĂäÿłæĺąåijŘåŚ¡åŘ■ïijŇä¿ŻåŘŐéİćä¡£çŤ

äÿŃäÿĂæ■ěïijŇäÿžäžĘäżď'çĽŇåŇŰïijŇä¡£çŤĺæĺąåijŘåŕźèśąå¿ĹåŕŚèćńäžžç§ěéĄŞçŽĎ `scanner()` æŰźæşŢãĂĆ è£ŹäÿłæŰźæşŢäijŽåĹŻåžžäÿĂäÿł `scanner` åŕźèśąïijŇ åIJĺè£ŹäÿłåŕźèśąäÿŁäÿ■æŰ■çŽĎèřČçŤĺ `match()` æŰźæşŢäijŽäÿĂæ■ěæ■ěçŽĎæĽńæŘŘçŻőæăĞæŰĞæIJňïijŇæŕŘæ■ěäÿĂäÿłåŇźéĚ■ãĂĆ äÿŃéİćæŸŕæijŤçď'žäÿĂäÿł `scanner` åŕźèśąåęĆä¡Ţåůěä¡IJçŽĎäžď'äžŠåijŘä¿Ńå■ŘïijŽ

```
>>> scanner = master_pat.scanner('foo = 42')
>>> scanner.match()
<_sre.SRE_Match object at 0x100677738>
>>> _.lastgroup, _.group()
('NAME', 'foo')
>>> scanner.match()
<_sre.SRE_Match object at 0x100677738>
>>> _.lastgroup, _.group()
('WS', ' ')
>>> scanner.match()
<_sre.SRE_Match object at 0x100677738>
>>> _.lastgroup, _.group()
('EQ', '=')
>>> scanner.match()
<_sre.SRE_Match object at 0x100677738>
>>> _.lastgroup, _.group()
('WS', ' ')
>>> scanner.match()
<_sre.SRE_Match object at 0x100677738>
>>> _.lastgroup, _.group()
('NUM', '42')
>>> scanner.match()
```

(continues on next page)

(continued from previous page)

```
>>>
```

åőđéŹĚä¡£çŤÍè£Źçğ■æŁĂæIJŕçŽĎæŮűåĂŹïijŇåŘŕäžěå¿ĹåőźæŸ§çŽĎåČŘäÿŃéİćè£ŹæăůåřĘäÿŁè£řäž

```
def generate_tokens(pat, text):
    Token = namedtuple('Token', ['type', 'value'])
    scanner = pat.scanner(text)
    for m in iter(scanner.match, None):
        yield Token(m.lastgroup, m.group())

# Example use
for tok in generate_tokens(master_pat, 'foo = 42'):
    print(tok)
# Produces output
# Token(type='NAME', value='foo')
# Token(type='WS', value=' ')
# Token(type='EQ', value='=')
# Token(type='WS', value=' ')
# Token(type='NUM', value='42')
```

åęĆæđIJä¡ăæČşè£ĞæżďäżďçĽŇæţĄïijŇä¡ăåŘŕäžěåőŽäźĽæŻťďŽçŽĎçŤ§æĹŘåŹĺåĞ¡æŢřæĹŮèĂĚä¡
æŕŤåęĆïijŇäÿŃéİćæijŤçďžæĂŐæăůè£ĞæżďæĽĂæIJĽçŽĎçľžçŹ¡äżďçĽŇïijŽ

```
tokens = (tok for tok in generate_tokens(master_pat, text)
          if tok.type != 'WS')
for tok in tokens:
    print(tok)
```

èőÍèőž

éĂŽåÿÿæİěèőšäżďçĽŇåŇŮæŸŕå¿Ĺåď'ŽéńŸçžğæŮĞæIJňèğčæđŘäÿŐåď'ĎçŘĘçŽĎçňňäÿĂæ■ěãĂĆ
äÿžäžĘä¡£çŤÍäÿŁéİćçŽĎæĽ'ńæŘŘæŮźæşŢïijŇä¡ăéIJĂèęĄèőŕä¡Ŕè£ŹéĞŇäÿĂäžŻéĞ■èęĄçŽĎåĞăçĆźãĂĆ
çňňäÿĂçĆźåŕśæŸŕä¡ăå£Ěéążçąőèőď'ä¡ăä¡£çŤÍæ■čåĹŹèąÍè¿¿åijŘæŇĞåőŽäžĘæĽ'ĂæIJĽ'è¿ŞåĚěäÿ■åŘŕèČ¡åĞ
åęĆæđIJæIJĽäżżä¡Ţäÿ■åŘŕåŇźéĚ■çŽĎæŮĞæIJňåĞžçŐŕäžĘïijŇæĽńæŘŘåŕśäijŽçŻť'æŐěåĄIJæ■ćãĂĆè£Źä

äżď'çĽŇçŽĎéążåžŘäź§æŸŕæIJĽ'å¡śåŞ■çŽĎãĂĆ `re` æĺąåİŮäijŽæŇĽçĚğæŇĞåőŽåě¡çŽĎéążåžŘåŐżåĄ
åŻäæ■ď'ïijŇåęĆæđIJäÿĂäÿłæĺąåijŘæĄřåě¡æŸŕåŘęäÿĂäÿłæŻť'éŢ£æĺąåijŘçŽĎå■Řå■ŮçňęäÿšïijŇéĆčäźĹä¡ăé

```
LT = r'(?P<LT><)'
LE = r'(?P<LE><=)'
EQ = r'(?P<EQ>=)'

master_pat = re.compile('|'.join([LE, LT, EQ])) # Correct
# master_pat = re.compile('|'.join([LT, LE, EQ])) # Incorrect
```

çňňäžŇäÿłæĺąåijŘæŸŕéŤŹçŽĎïijŇåŻääÿžäőČäijŽåřĘæŮĞæIJň<=åŇźéĚ■äÿžäżď'çĽŇLTçť'ğèů§çİĂEQï

æIJĂåŘŐïijŇä¡ăéIJĂèęĄçŢŹæĎŘäÿŇå■Řå■Ůçňęäÿšå¡ćåijŘçŽĎæĺąåijŘãĂĆæŕŤåęĆïijŇåĄĞèő¿ä¡ăæIJĽ

```
PRINT = r'(?P<PRINT>print)'
NAME = r'(?P<NAME>[a-zA-Z_][a-zA-Z_0-9]*)'

master_pat = re.compile('|'.join([PRINT, NAME]))

for tok in generate_tokens(master_pat, 'printer'):
    print(tok)

# Outputs :
# Token(type='PRINT', value='print')
# Token(type='NAME', value='er')
```

åĚşäžŐæŻťénŸéŸűçŽĎäžďçĽŇåŇŰæŁĂæIJŕïijŇä¡ääŘŕèČ¡éIJĂèęĄæ§ěçIJŃ PyParsing æĹŰèĂĚ PLY åŇĚãĂĆ äÿĂäÿłèřČçŤÍPLYçŽĎä¿Ňå■ŘåIJÍäÿŃäÿĂèŁĆäijŽæIJĽæijŤçďžãĂĆ

4.19 2.19 áőđçŐřäÿĂäÿłçőĂå■ŢçŽĎéĂŠå¡ŠäÿŃéŹ■åĹĘæđŘåŹÍ

éŮőéćŸ

ä¡ăæČşæăźæ■őäÿĂçżĎèŕ■æşŢèğĎåĹŹèğčæđŘæŰĞæIJňäźűæĽğèąŇåŚ¡äžďïijŇæĹŰèĂĚæđĎéĂääÿĂä
äęĆæđIJèŕ■æşŢéİđäÿÿçőĂå■ŢïijŇä¡ääŘŕäžěäÿ■åŐżä¡£çŤÍäÿĂäžŻæąĘæđűïijŇèĂŇæŸŕèĞłäůśäĘŹè£Źäÿłèğč

èğčäĘşæŰźæąĹ

åIJÍè£ŹäÿłéŮőéćŸäÿ■ïijŇæĹŚäžňéŻĘäÿ■èőÍèőžæăźæ■őçĽźæőŁèŕ■æşŢäŐżèğčæđŘæŰĞæIJňçŽĎéŮőé
äÿžäžĘè£ŹæăůäĄŽïijŇä¡äéęŰäĚĹèęĄäžěBNFæĹŰèĂĚEBNFä¡ćäijŘæŇĞäőŽäÿĂäÿłæăĞäĞĘèŕ■æşŢäĂĆ
æŕŤäęĆïijŇäÿĂäÿłçőĂå■ŢæŢŕä■ęèąĺè¿¿äijŘèŕ■æşŢäŘŕèČ¡äČŘäÿŃéİćè£ŹæăůïijŽ

```
expr ::= expr + term
    |    expr - term
    |    term

term ::= term * factor
    |    term / factor
    |    factor

factor ::= ( expr )
    |    NUM
```

æĹŰèĂĚïijŇäžěEBNFä¡ćäijŘïijŽ

```
expr ::= term { (+|-) term }*

term ::= factor { (*|/) factor }*

factor ::= ( expr )
    |    NUM
```

åIJÍEBNFäÿ■ïijŇèćňåŇĚåŘňåIJÍ { . . . }* äÿ■çŽĎèğĎåĹŹæŸŕåŘŕéĂĽçŽĎãĂĆ*äžčèąĺ0æňąæĹŰåď'Žæ

çŐřåIJÍïijŇåęĆæđIJä¡ăåŕźBNFçŽĎåůěä¡IJæIJžåĹűè£Ÿäÿ■æŸŕå¿ĹæŸŐçŹ¡çŽĎèŕÍïijŇåřśæŁŁåőČå¡ŞåĄ
äÿĂèĹňæÍěèőšïijŇèğčæđŘçŽĎåŐ§çŘĘåřśæŸŕä¡ăåĹľçŤÍBNFåőŇæĹŘåď'ŽäÿłæŽ£æ■ćåŠŇæĽľäśŢäžěåŇźéĚ
äÿžäžĘæijŤçď'žïijŇåĄĞèő¿ä¡ăæ■čåIJÍèğčæđŘå¡ćåęĆ 3 + 4 * 5 çŽĎèąĺè¿¿åijŘãĂĆ
è£Źäÿłèąĺè¿¿åijŘåĚĹèęĄéĂŽè£Ğä¡£çŤÍ2.18èŁĆäÿ■äžŃçż■çŽĎæŁĂæIJŕåĹĘèğčäÿžäÿĂçżĎäžď'çĽŇæţĄãĂ
çżŞæđIJåŘŕèČ¡æŸŕåČŘäÿŇåĹŮè£ŹæăůçŽĎäžď'çĽŇåžŘåĹŮïijŽ

```
NUM + NUM * NUM
```

åIJÍæ■ď'å§žçąĂäÿŁïijŇ èğčæđŘåŁÍä¡IJäijŽèŕŢçİĂåŐżéĂŽè£ĞæŻ£æ■ćæŞ■ä¡IJåŇźéĚ■èŕ■æşŢåĹřè¿ŞåĚ

```
expr
expr ::= term { (+|-) term }*
expr ::= factor { (*|/) factor }* { (+|-) term }*
expr ::= NUM { (*|/) factor }* { (+|-) term }*
expr ::= NUM { (+|-) term }*
expr ::= NUM + term { (+|-) term }*
expr ::= NUM + factor { (*|/) factor }* { (+|-) term }*
expr ::= NUM + NUM { (*|/) factor}* { (+|-) term }*
expr ::= NUM + NUM * factor { (*|/) factor }* { (+|-) term }*
expr ::= NUM + NUM * NUM { (*|/) factor }* { (+|-) term }*
expr ::= NUM + NUM * NUM { (+|-) term }*
expr ::= NUM + NUM * NUM
```

äÿŃéĬćæĽĂæIJĽçŽĎèğčæđŘæ■ěéłď'åŘŕèČ¡éIJĂèęĄèŁśçĆźæŮűéŮť'åijĎæŸŐçŹ¡ïijŇä¡ĘæŸŕåőČäžňåĆ
çňňäÿĂäÿłè¿ŞåĚěäžď'çĽŇæŸŕNUMïijŇåŻäæ■ď'æŻ£æ■ćéęŰůåĚĹäijŽåŇźéĚ■éĆčäÿłéČĺåĹĘãĂĆ
äÿĂæŮęåŇźéĚ■æĹŘåŁ§ïijŇåřśäijŽè£ŻåĚěäÿŃäÿĂäÿłäžď'çĽŇ+ïijŇäžěæ■ď'çśżæŐĺãĂĆ
å¡ŞåůšçżŘçąőåőŽäÿ■èČ¡åŇźéĚ■äÿŃäÿĂäÿłäžď'çĽŇçŽĎæŮűåĂŹïijŇåŘşè¿źçŽĎéČĺåĹĘ(æŕŤåęĆ
{ (*/) factor }*)åřśäijŽèćňæÿĚçŘĘæŐĽãĂĆ åIJÍäÿĂäÿłæĹŘåŁ§çŽĎèğčæđŘäÿ■ïijŇæŢť'äÿłåŘşè¿źé

æIJĽäžĘåĽ■éĬćçŽĎç§ěèŕĘèČŇæŹŕïijŇäÿŃéĬćæĹŚäžňäÿ¿äÿĂäÿłçőĂå■Ţçď'žä¿ŃæĬěåśŢçď'žåęĆä¡ŢæđĬ

```
#!/usr/bin/env python
# -*- encoding: utf-8 -*-
"""
Topic: äÿŃéŹ■èğčæđŘåŹĺ
Desc :
"""
import re
import collections

# Token specification
NUM = r'(?P<NUM>\d+)'
PLUS = r'(?P<PLUS>\+)'
MINUS = r'(?P<MINUS>-)'
TIMES = r'(?P<TIMES>\*)'
DIVIDE = r'(?P<DIVIDE>/)'
LPAREN = r'(?P<LPAREN>\()'
RPAREN = r'(?P<RPAREN>\))'
WS = r'(?P<WS>\s+)'
```

(continues on next page)

(continued from previous page)

```
master_pat = re.compile('|'.join([NUM, PLUS, MINUS, TIMES,
                                  DIVIDE, LPAREN, RPAREN, WS]))
# Tokenizer
Token = collections.namedtuple('Token', ['type', 'value'])


def generate_tokens(text):
    scanner = master_pat.scanner(text)
    for m in iter(scanner.match, None):
        tok = Token(m.lastgroup, m.group())
        if tok.type != 'WS':
            yield tok


# Parser
class ExpressionEvaluator:
    '''
    Implementation of a recursive descent parser. Each method
    implements a single grammar rule. Use the .__accept() method
    to test and accept the current lookahead token. Use the .__
↪expect()
    method to exactly match and discard the next token on on the␣
↪input
    (or raise a SyntaxError if it doesn't match).
    '''

    def parse(self, text):
        self.tokens = generate_tokens(text)
        self.tok = None  # Last symbol consumed
        self.nexttok = None  # Next symbol tokenized
        self._advance()  # Load first lookahead token
        return self.expr()

    def _advance(self):
        'Advance one token ahead'
        self.tok, self.nexttok = self.nexttok, next(self.tokens,␣
↪None)

    def _accept(self, toktype):
        'Test and consume the next token if it matches toktype'
        if self.nexttok and self.nexttok.type == toktype:
            self._advance()
            return True
        else:
            return False

    def _expect(self, toktype):
        'Consume next token if it matches toktype or raise␣
↪SyntaxError'
```

(continues on next page)

(continued from previous page)

```
        if not self._accept(toktype):
            raise SyntaxError('Expected ' + toktype)

    # Grammar rules follow
    def expr(self):
        "expression ::= term { ('+'|'-') term }*"
        exprval = self.term()
        while self._accept('PLUS') or self._accept('MINUS'):
            op = self.tok.type
            right = self.term()
            if op == 'PLUS':
                exprval += right
            elif op == 'MINUS':
                exprval -= right
        return exprval

    def term(self):
        "term ::= factor { ('*'|'/') factor }*"
        termval = self.factor()
        while self._accept('TIMES') or self._accept('DIVIDE'):
            op = self.tok.type
            right = self.factor()
            if op == 'TIMES':
                termval *= right
            elif op == 'DIVIDE':
                termval /= right
        return termval

    def factor(self):
        "factor ::= NUM | ( expr )"
        if self._accept('NUM'):
            return int(self.tok.value)
        elif self._accept('LPAREN'):
            exprval = self.expr()
            self._expect('RPAREN')
            return exprval
        else:
            raise SyntaxError('Expected NUMBER or LPAREN')


def descent_parser():
    e = ExpressionEvaluator()
    print(e.parse('2'))
    print(e.parse('2 + 3'))
    print(e.parse('2 + 3 * 4'))
    print(e.parse('2 + (3 + 4) * 5'))
    # print(e.parse('2 + (3 + * 4)'))
    # Traceback (most recent call last):
    #    File "<stdin>", line 1, in <module>
```

(continues on next page)

(continued from previous page)

```
    #     File "exprparse.py", line 40, in parse
    #     return self.expr()
    #     File "exprparse.py", line 67, in expr
    #     right = self.term()
    #     File "exprparse.py", line 77, in term
    #     termval = self.factor()
    #     File "exprparse.py", line 93, in factor
    #     exprval = self.expr()
    #     File "exprparse.py", line 67, in expr
    #     right = self.term()
    #     File "exprparse.py", line 77, in term
    #     termval = self.factor()
    #     File "exprparse.py", line 97, in factor
    #     raise SyntaxError("Expected NUMBER or LPAREN")
    #     SyntaxError: Expected NUMBER or LPAREN


if __name__ == '__main__':
    descent_parser()
```

èőÍèőž

æŰĞæIJňèğčæđŘæŸŕäÿĂäÿłå¿Ĺåď ğçŽĎäÿżéćŸïijŇ äÿĂèĹňäijŽå■ăçŤÍå■ęçŤ§å■ęäźăçijŰèŕŚèŕ¿çÍŃæŮ
åęĆæđIJä¡ăåIJÍæĽ¿åŕżåĚşäžŐèŕ■æşŢïijŇèğčæđŘçőŮæşŢç■Ľ'çŻÿåĚşçŽĎèČŇæŹŕç§ěèŕĘçŽĎèŕÍïijŇä¡ăåžŤèŕ
å¿ĹæŸ¿çĎűïijŇåĚşäžŐè£ŹæŰźéÍćçŽĎåĘĚåőźåď łåď ŽïijŇäÿ■åŘŕèČ¡åIJÍè£ŹéĞŇåĚÍéČÍåśŢåijĂãĂĆ

ăř¡çőąåęĆæ■ď'ïijŇçijŰåĘŹäÿĂäÿłéĂŠå¡ŠäÿŃéŹ■èğčæđŘåŹÍçŽĎæŢť'ä¡ŞæĂÍèůŕæŸŕæŕŤè¿ČçőĂå■ŢçŽ
åijĂåğŃçŽĎæŮűåĂŹïijŇä¡ăåĚĹèŐůå¿ŮæĽ'ĂæIJĽ'çŽĎèŕ■æşŢèğĎåĹŹïijŇçĎűåŘŐåřĘåĚűè¡ňæ■ćäÿžäÿĂäÿ
åŻăæ■ď'åęĆæđIJä¡ăçŽĎèŕ■æşŢçśżäijijè£ŹæăůïijŽ

```
expr ::= term { ('+'|'-') term }*

term ::= factor { ('*'|'/') factor }*

factor ::= '(' expr ')'
    | NUM
```

ä¡ăåžŤèŕěéęŰåĚĹåřĘåőČäžňè¡ňæ■ćæĹŘäÿĂçżĎåČŕäÿŃéÍćè£ŹæăůçŽĎæŰźæşŢïijŽ

```
class ExpressionEvaluator:
    ...
    def expr(self):
    ...
    def term(self):
    ...
    def factor(self):
    ...
```

æŕŘäÿłæŰźæşŢèęĄåőŃæĹŘçŽĎäżżåŁąå¿ĹçőĂå■Ţ - åőČå£ĚéążäžŐåůęèĞşåŘşéĄ■åŐĘèŕ■æşŢèğĎåĹŹ

äżŐæ§Řçğ■æĎŘäźĽäÿŁèőšïijŇæŰźæşŢçŽĎçŻőçŽĎåŕśæŸŕèęĄäźĹåď'ĎçŘĘåőŇèŕ■æşŢèğĎåĹŹïijŇèęĄäźĹ
äÿžäžĘè£ŹæäůåĄŽïijŇéIJĂéĞĞçŤĺäÿŇéĺćçŽĎè£ŹäžŻåőđçŐŕæŰźæşŢïijŽ

- åęĆæđIJèğĎåĹŹäÿ■çŽĎäÿŇäÿłçňęåŘůæŸŕåŘęåď'ŰäÿĂäÿłèŕ■æşŢèğĎåĹŹçŽĎåŘ■å■Ů(æŕŤåęĆtermæ
  è£ŹåŕśæŸŕèŕěçőŮæşŢäÿ■âĂİäÿŇéŹ■âĂİçŽĎçŤśæİě -
  æŐğåĹűäÿŇéŹ■åĹŕåŘęäÿĂäÿłèŕ■æşŢèğĎåĹŹäÿ■åŐżãĂĆ
  æIJĽæŮűåĂŹèğĎåĹŹäijŽèřĆçŤĺåůšçżŘæĽğèąŇçŽĎæŰźæşŢ(æŕŤåęĆïijŇåIJĺ
  `factor ::= '('expr ')'` äÿ■åŕźexprçŽĎèřĆçŤĺ)ãĂĆ
  è£ŹåŕśæŸŕçőŮæşŢäÿ■âĂİéĂŠå¡ŠâĂİçŽĎçŤśæİěãĂĆ

- åęĆæđIJèğĎåĹŹäÿ■äÿŇäÿĂäÿłçňęåŘůæŸŕäÿłçĽźæőŁçňęåŘů(æŕŤåęĆ()ïijŇä¡åå¿Ůæ§ěæĽ¿äÿŇäÿĂäÿ
  åęĆæđIJäÿ■åŇźéĚ■ïijŇåŕśäžğçŤ§äÿĂäÿłèŕ■æşŢéŤŹèŕŕãĂĆè£ŹäÿĂèŁĆäÿ■çŽĎ
  `_expect()` æŰźæşŢåŕśæŸŕçŤĺæİěåĄŽè£ŹäÿĂæ■ěçŽĎãĂĆ

- åęĆæđIJèğĎåĹŹäÿ■äÿŇäÿĂäÿłçňęåŘůäÿžäÿĂäžŻåŘŕèČ¡çŽĎéĂĽæŃľéąź(æŕŤåęĆ +
  æĹŰ -)ïijŇ ä¡åå£ĚéążåŕźæŕŘäÿĂçğ■åŘŕèČ¡æČĚåĘţæčĂæ§ěäÿŇäÿĂäÿłäżd'çĽŇïijŇåŘłæIJĽå¡ŞåőČåŇ
  è£Źäź§æŸŕæIJňèŁĆçd'žä¿Ňäÿ■ `_accept()` æŰźæşŢçŽĎçŻőçŽĎãĂĆ
  åőČçŻÿå¡ŞäžŐ_expect()æŰźæşŢçŽĎåijśåŇŰçĽĹæIJňïijŇåŻääÿžåęĆæđIJäÿĂäÿłåŇźéĚ■æĽ¿åĹřäžĘåę
  ä¡ĘæŸŕåęĆæđIJæšąæĽ¿åĹřïijŇåőČäÿ■äijŽäžğçŤ§éŤŹèŕŕèĂŇæŸŕåŻđæżŽ(åĚĄèőÿåŘŐçż■çŽĎæčĂæ§

- åŕźäžŐæIJĽéĞ■åď'■éČĺåĹĘçŽĎèğĎåĹŹ(æŕŤåęĆåIJĺèğĎåĹŹèąĺè¿¿åijŔ `::= term {
  ('+'|'-') term }*` äÿ■)ïijŇ éĞ■åď'■åŁĺä¡IJéĂŽè£ĞäÿĂäÿłwhileå¿łçŐŕæİěåőđçŐŕãĂĆ
  å¿łçŐŕäÿżä¡ŞäijŽæŤűéŻĘæĹŰåď'ĎçŘĘæĽ'ĂæIJĽçŽĎéĞ■åď'■åĚČçť'äçŻť'åĹřæšąæIJĽåĚűäżŰåĚČçť'ä

- äÿĂæŮęæŢť'äÿłèŕ■æşŢèğĎåĹŹåď'ĎçŘĘåőŇæĹŘïijŇæŕŘäÿłæŰźæşŢäijŽè£ŤåŻđæ§Řçğ■çżŞæđIJçżŹè
  è£ŹåŕśæŸŕåIJĺèğčæđŘè£ĞçĺŇäÿ■åĂijæŸŕæĂŐæăůçť'ŕåŁäçŽĎåŐ§çŘĘãĂĆ
  æŕŤåęĆïijŇåIJĺèąĺè¿¿åijŘæśĆåĂijçĺŇåžŘäÿ■ïijŇè£ŤåŻđåĂijäžčèąĺèąĺè¿¿åijŘèğčæđŘåŘŐçŽĎéĆĺåĹĘç
  æIJĂåŘŐæĽ'ĂæIJĽåĂijäijŽåIJĺæIJĂéąűåśĆçŽĎèŕ■æşŢèğĎåĹŹæŰźæşŢäÿ■åŘĹåźűèţůæİěãĂĆ

åř¡çőąåŘŚä¡ăæijŤçď'žçŽĎæŸŕäÿĂäÿłçőĂå■ŢçŽĎä¿Ňå■ŘïijŇéĂŠå¡ŠäÿŇéŹ■èğčæđŘåŹĺåŘŕäžěçŤĺæİěå
æŕŤåęĆïijŇPythonèŕ■èĺĂæIJňèžňåŕśæŸŕéĂŽè£ĞäÿĂäÿłéĂŠå¡ŠäÿŇéŹ■èğčæđŘåŹĺåŐżèğčéĞŁçŽĎãĂĆ
åęĆæđIJä¡ååŕźæ■ď'æĎ§åĚť'èűčïijŇä¡ååŘŕäžěéĂŽè£Ğæ§ěçIJŃPythonæžŘçăĄæŰĞäžűGrammar/Grammaræİ
çIJŃåőŇä¡ääijŽåŘŚçŐŕïijŇéĂŽè£ĞæĽ'ŃåŁĺæŰźåijŘåŐżåőđçŐŕäÿĂäÿłèğčæđŘåŹĺåĚűåőđäijŽæIJĽå¿Łåď'Žç

åĚűäÿ■äÿĂäÿłåśĂéŹŘåŕśæŸŕåőČäžňäÿ■èČ¡èćńçŤĺäžŐåŇĚåŘńäżżä¡ŢåůęéĂŠå¡ŠçŽĎèŕ■æşŢèğĎåĹŹäÿ

```
items ::= items ',' item
    | item
```

äÿžäžĘè£ŹæäůåĄŽïijŇä¡åŘŕèČ¡äijŽåČŘäÿŇéĺćè£ŹæäůÄ¡£çŤĺ `items()` æŰźæşŢïijŽ

```
def items(self):
    itemsval = self.items()
    if itemsval and self._accept(','):
        itemsval.append(self.item())
    else:
        itemsval = [ self.item() ]
```

åŤŕäÿĂçŽĎéŮőéćŸæŸŕè£ŹäÿłæŰźæşŢæäźæIJňäÿ■èČ¡åůěä¡IJïijŇäžŃåőđäÿŁïijŇåőČäijŽäžğçŤ§äÿĂäÿ

åĚşäžŐèŕ■æşŢèğĎåĹŹæIJňèžňä¡åŘŕèČ¡äź§äijŽçćŕåĹŕäÿĂäžŻæčŸæĽ'ŃçŽĎéŮőéćŸãĂĆ
æŕŤåęĆïijŇä¡åŘŕèČ¡æČşç§ěéĄŞäÿŇéĺćè£ŹäÿłçőĂå■ŢæĽ'ijèŕ■æşŢæŸŕåŘęèąĺè£ŕå¿Ůå¡ŞïijŽ

```
expr ::= factor { ('+'|'-'|'*'|'/') factor }*

factor ::= '(' expression ')'
    | NUM
```

è£Źäÿłèŕ■æşŢçIJŃäÿŁåŐżæšąåŢěéŮőéćŸïijŇä¡ĘæŸŕåőČå■t'äÿ■èČ¡åŕ§èğL'åĹřæăĞåĞĘåŻŻåĹŹè£ŘçőŮ
æŕŤåęĆïijŇèąĺè¿¿åijŘ "3 + 4 * 5" äijŽå¿ŮåĹř35èĂŇäÿ■æŸŕæIJ§æIJŻçŽĎ23.
åĹĘåijĂä¡£çŤĺâĂİexprâĂİåŠŇâĂİtermâĂİèğĎåĹŹåŘŕäżěèőľåőČæ■ččąőçŽĎåůëä¡IJãĂĆ

åŕźäžŐåď■æİĆçŽĎèŕ■æşŢïijŇä¡ăæIJĂåě¡æŸŕéĂĽ'æŃľæ§ŘäÿłèğčæđŘåůěåĚůæŕŤåęĆPyParsingæĹŰèĂ
äÿŃéİćæŸŕä¡£çŤĺPLYæİěéĞ■åĘŹèąĺè¿¿åijŘæśĆåĂijçĺŃåžŘçŽĎäżčçăĄïijŽ

```
from ply.lex import lex
from ply.yacc import yacc

# Token list
tokens = [ 'NUM', 'PLUS', 'MINUS', 'TIMES', 'DIVIDE', 'LPAREN',
↪'RPAREN' ]
# Ignored characters
t_ignore = ' \t\n'
# Token specifications (as regexs)
t_PLUS = r'\+'
t_MINUS = r'-'
t_TIMES = r'\*'
t_DIVIDE = r'/'
t_LPAREN = r'\('
t_RPAREN = r'\)'

# Token processing functions
def t_NUM(t):
    r'\d+'
    t.value = int(t.value)
    return t

# Error handler
def t_error(t):
    print('Bad character: {!r}'.format(t.value[0]))
    t.skip(1)

# Build the lexer
lexer = lex()

# Grammar rules and handler functions
def p_expr(p):
    '''
    expr : expr PLUS term
         | expr MINUS term
    '''
    if p[2] == '+':
        p[0] = p[1] + p[3]
```

(continues on next page)

(continued from previous page)

```
    elif p[2] == '-':
        p[0] = p[1] - p[3]


def p_expr_term(p):
    '''
    expr : term
    '''
    p[0] = p[1]


def p_term(p):
    '''
    term : term TIMES factor
    | term DIVIDE factor
    '''
    if p[2] == '*':
        p[0] = p[1] * p[3]
    elif p[2] == '/':
        p[0] = p[1] / p[3]

def p_term_factor(p):
    '''
    term : factor
    '''
    p[0] = p[1]

def p_factor(p):
    '''
    factor : NUM
    '''
    p[0] = p[1]

def p_factor_group(p):
    '''
    factor : LPAREN expr RPAREN
    '''
    p[0] = p[2]

def p_error(p):
    print('Syntax error')

parser = yacc()
```

è£ŹäÿłçÍŃåžŘäÿ■ïijŇæL’ĂæIJL’äžčçăĄéČ¡ä¡■äžŐäÿĂäÿłæŕŤè¿ČéńŸçŽĎåśĆæňąãĂĆä¡ăåŘłéIJĂèęĄäÿ
èĂŇåőđéŹĚçŽĎè£ŘèąŇèğčæđŘåŹĺïijŇæŐěåŘŮäžd’çL’Ňç■L’ç■L’åžŢåśĆåŁĺä¡IJåůšçżŘèćńåžŞåĞ¡æŢřåőđçŐ

äÿŃéİćæŸŕäÿĂäÿłæĂŐæăůä¡£çŤĺå¿ŮåĹřçŽĎèğčæđŘåŕźèśąçŽĎä¿Ńå■ŘïijŽ

```
>>> parser.parse('2')
2
>>> parser.parse('2+3')
5
>>> parser.parse('2+(3+4)*5')
37
>>>
```

åęĆæđIJä¡ăæČşåIJÍä¡ăçŽĎçijŰçÍŃè£ĞçÍŃäÿ■æİěçĆźæŇŚæĹŸåŠŇåĹžæ£ĂïijŇçijŰåĘŹèğčæđŘåŹÍåŠŇ
åĘ■æňąïijŇäÿĂæIJňçijŰèřŚåŹÍçŽĎäźęçś■äijŽåŇĚåŘńå¿Ĺåď ŽåžŢåśĆçŽĎçŘĘèőžç§ěèřĘãĂĆäÿ■è£Ğå¿Ĺåď
PythonèĞłåůśçŽĎastæĺąåİŮäź§åĂijå¿ŮåŐżçIJŃäÿĂäÿŃãĂĆ

4.20 2.20 å■ŮèŁĆå■Ůçňęäÿšäÿ Łç ŽĎå■ŮçňęäÿšæŞ■ä¡IJ

éŮőécŸ

ä¡ăæČşåIJÍå■ŮèŁĆå■Ůçňęäÿšäÿ ŁæL'ğèąŇæŹőéĂŽçŽĎæŰĞæIJňæŞ■ä¡IJ(æřŤåęĆçğżéŹď'ïijŇæŘIJçť'ćå

èğčåĘşæŰźæąĹ

å■ŮèŁĆå■ŮçňęäÿšåŘŇæăůäź§æŤŕæŇĄåď ğéČĺåĹĘåŠŇæŰĞæIJňå■Ůçňęäÿšäÿ Ăæăůç ŽĎåĘĚç¡őæŞ■ä¡

```
>>> data = b'Hello World'
>>> data[0:5]
b'Hello'
>>> data.startswith(b'Hello')
True
>>> data.split()
[b'Hello', b'World']
>>> data.replace(b'Hello', b'Hello Cruel')
b'Hello Cruel World'
>>>
```

è£ŹäžŻæŞ■ä¡IJåŘŇæăůäź§éĂĆçŤĺäžŐå■ŮèŁĆæŢřçżĎãĂĆæřŤåęĆïijŽ

```
>>> data = bytearray(b'Hello World')
>>> data[0:5]
bytearray(b'Hello')
>>> data.startswith(b'Hello')
True
>>> data.split()
[bytearray(b'Hello'), bytearray(b'World')]
>>> data.replace(b'Hello', b'Hello Cruel')
bytearray(b'Hello Cruel World')
>>>
```

ä¡ăåŘŕäžěä¡£çŤĺæ■čåĹŹèąĺè¿¿åijŘåŇźéĚ■å■ŮèŁĆå■ŮçňęäÿšïijŇä¡ĘæŸŕæ■čåĹŹèąĺè¿¿åijŘæIJňèžňå£Ě

```
>>>
>>> data = b'FOO:BAR,SPAM'
>>> import re
>>> re.split('[:,]',data)
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
File "/usr/local/lib/python3.3/re.py", line 191, in split
return _compile(pattern, flags).split(string, maxsplit)
TypeError: can't use a string pattern on a bytes-like object
>>> re.split(b'[:,]',data) # Notice: pattern as bytes
[b'FOO', b'BAR', b'SPAM']
>>>
```

èőÍèőž

åď ğåď ŽæŢřæČĚåĘţäÿŃïijŇåIJÍæŰĞæIJňå■ŮçňęäÿšäÿŁçŽĎæŞ■ä¡IJåİĞåŔŕçŤÍäžŐå■ŮèŁĆå■Ůçňęäÿšä
çĎűèĂŇïijŇè£ŹéĞŇäź§æIJL'äÿĂäžŻéIJĂèęĄæşÍæĎŔçŽĎäÿ■åŘŇçĆźãĂĆéęŰåĚĹïijŇå■ŮèŁĆå■Ůçňęäÿšç

```
>>> a = 'Hello World' # Text string
>>> a[0]
'H'
>>> a[1]
'e'
>>> b = b'Hello World' # Byte string
>>> b[0]
72
>>> b[1]
101
>>>
```

è£Źçğ■èŕ■äźL'äÿŁçŽĎåŇžåĹńäijŽåŕźäžŐåď ĎçŘĘéİćåŘŚå■ŮèŁĆçŽĎå■ŮçňęæŢřæ■őæIJL'å¡śåŞ■ãĂĆ

çňňäžŇçĆźïijŇå■ŮèŁĆå■Ůçňęäÿšäÿ■äijŽæŔŘä¿ŻäÿĂäÿłç¿ŐèğĆçŽĎå■Ůçňęäÿšèąĺçď žïijŇäź§äÿ■èČ¡å

```
>>> s = b'Hello World'
>>> print(s)
b'Hello World' # Observe b'...'
>>> print(s.decode('ascii'))
Hello World
>>>
```

çśżäijijçŽĎïijŇäź§äÿ■å■ŸåIJÍäżżä¡ŢéĂĆçŤÍäžŐå■ŮèŁĆå■ŮçňęäÿšçŽĎæäijåijŔåŇŰæŞ■ä¡IJïijŽ

```
>>> b'%10s %10d %10.2f' % (b'ACME', 100, 490.1)
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
TypeError: unsupported operand type(s) for %: 'bytes' and 'tuple'
>>> b'{} {} {}'.format(b'ACME', 100, 490.1)
Traceback (most recent call last):
```

(continues on next page)

(continued from previous page)

```
    File "<stdin>", line 1, in <module>
AttributeError: 'bytes' object has no attribute 'format'
>>>
```

åęĆæđIJä¡ăæČşæăijåijŔåŇŰå■ŮèŁĆå■ŮçňęäÿšïijŇä¡ăå¿ŮåĚĹä¡£çŤĺæăĞåĞęçŽĎæŰĞæIJňå■Ůçňęäÿš

```
>>> '{:10s} {:10d} {:10.2f}'.format('ACME', 100, 490.1).encode(
↪'ascii')
b'ACME 100 490.10'
>>>
```

æIJĂåŘŐéIJĂèęĄæşĺæĎŔçŽĎæŸŕïijŇä¡£çŤĺå■ŮèŁĆå■ŮçňęäÿšåŕŕèČ¡äijŽæŤźåŔŸäÿĂäžŻæŞ■ä¡IJçŽĎ
æŕŤåęĆïijŇåęĆæđIJä¡ăä¡£çŤĺäÿĂäÿłçijŰçăĄäÿžå■ŮèŁĆçŽĎæŰĞäżůåŘ■ïijŇèĂŇäÿ■æŸŕäÿĂäÿłæŹőéĂŽçŽ

```
>>> # Write a UTF-8 filename
>>> with open('jalape\xf1o.txt', 'w') as f:
...     f.write('spicy')
...
>>> # Get a directory listing
>>> import os
>>> os.listdir('.') # Text string (names are decoded)
['jalapeÃśo.txt']
>>> os.listdir(b'.') # Byte string (names left as bytes)
[b'jalapen\xcc\x83o.txt']
>>>
```

æşĺæĎŔä¿Ňå■Řäÿ■çŽĎæIJĂåŘŐéČĺåĹĘçżŹçŻőå¡ŢåŘ■äijăéĂŠäÿĂäÿłå■ŮèŁĆå■ŮçňęäÿšæŸŕæĂŐæăů
åIJĺçŻőå¡Ţäÿ■çŽĎæŰĞäżůåŘ■åŇĚåŘňåŐ§åğŃçŽĎUTF-8çijŰçăĄãĂĆ
åŔĆèĂČ5.15åřŔèŁĆèŐůåŔŰæŻť'åď'ŽæŰĞäżůåŘ■çŻÿåĚşçŽĎåĘĚåőźãĂĆ

æIJĂåŘŐæŔŘäÿĂçĆźïijŇäÿĂäžŻçĺŃåžŔåŚŸäÿžäžĘæŔŘå■ĞçĺŃåžŔæĽ'ğèąŇçŽĎéĂ§åžęäijŽåĂ¿åŘŚäż
åř¡çőąæŞ■ä¡IJå■ŮèŁĆå■ŮçňęäÿšçąőåőđäijŽæŕŤæŰĞæIJňæŻť'åŁäéńŸæŢĹ(åŻääÿžåď'ĎçŘĘæŰĞæIJňåŻžæI
è£ŹæăůåĄŽéĂŽåÿÿäijŽåŕijèĞť'éİđåÿÿæİĆäźśçŽĎäžčçăĄãĂĆä¡ääijŽçżŘåÿÿåŚŚçŐřå■ŮèŁĆå■Ůçňęäÿšåźűäÿ
åźűäÿŤä¡äè£Ÿå¿ŮæĽ'ŃåŁĺåď'ĎçŘĘæĽ'ĂæIJĽçŽĎçijŰçăĄ/èğčçăĄæŞ■ä¡IJãĂĆ
åİęçŹ¡èőšïijŇåęĆæđIJä¡ăåIJĺåď'ĎçŘĘæŰĞæIJňçŽĎèŕĬïijŇářśçŻť'æŐěåIJĺçĺŃåžŔäÿ■ä¡£çŤĺæŹőéĂŽçŽĎæŰĞ

5 çňňäÿĽ'çńäïijŽæŢřå■ŮæŮěæIJ§åŠŇæŮűéŮť

åIJĺPythonäÿ■æĽ'ğèąŇæŢť'æŢřåŠŇæţőçĆźæŢřçŽĎæŢřå■ęè£ŘçőŮæŮűå¿ĹçőĂå■ŢçŽĎãĂĆ
åř¡çőąåęĆæ■ď'ïijŇåęĆæđIJä¡ăéIJĂèęĄæĽ'ğèąŇåĹĘæŢřãĂĄæŢřçżĎæĹŰèĂĚæŸŕæŮěæIJ§åŠŇæŮűéŮť'çŽĎ
æIJňçńäéŻĘäÿ■èőĺèőžçŽĎářśæŸŕè£ŹäžŻäÿżéćŸãĂĆ

Contents:

5.1 3.1 æŢřå■ŮçŽĎåŻŻèĹ■ažŤåĚě

éŮőéćŸ

ä¡ăæČşåŕźæţőçĆźæŢřæĽğèąŇæŇĞåőŽçš¿åžęçŽĎèĹ■åĚěè£ŘçőŮãĂĆ

èğčåĘşæŰźæąĹ

åŕźäžŐçőĂå■ŢçŽĎèĹ■åĚěè£ŘçőŮïijŇä¡£çŤĺåĘĚç¡őçŽĎ `round(value, ndigits)` åĞ¡æŢřå■şåŘŕãĂĆæŕŤåęĆïijŽ

```
>>> round(1.23, 1)
1.2
>>> round(1.27, 1)
1.3
>>> round(-1.27, 1)
-1.3
>>> round(1.25361,3)
1.254
>>>
```

å¡ŞäÿĂäÿłåĂijåĹŽåě¡åIJĺäÿď äÿłè¿źçŢŇçŽĎäÿ■éŮť çŽĎæŮűåĂŹïijŇ `round` åĞ¡æŢřè£ŤåŻđçężåőĆæIJĂè£ŚçŽĎåĄűæŢřãĂĆ äźŕ§åŕśæŸŕèŕť ïijŇåŕź1.5æĹŰèĂĚ2.5çŽĎèĹ■åĚěè£ŘçőŮéČ¡

äijăçżŹ `round()` åĞ¡æŢřçŽĎ `ndigits` åŘĆæŢřåŘŕäžěæŸŕèť §æŢřïijŇè£ŹçğŒæČĚåĘţäÿŃïijŇ èĹ■åĚěè£ŘçőŮäijŽä¡IJçŤĺåIJĺå■Ąä¡■ãĂĄçŹ¿ä¡■ãĂĄå■Čä¡■ç■L'äÿŁéİćãĂĆæŕŤåęĆïijŽ

```
>>> a = 1627731
>>> round(a, -1)
1627730
>>> round(a, -2)
1627700
>>> round(a, -3)
1628000
>>>
```

èőĺèőž

äÿ■èęĄåřĘèĹ■åĚěåŠŇæäijåijŘåŇŰè¿ŞåĞžæŘđæůůæůĘäžĘãĂĆ åęĆæđIJä¡äçŽĎçŻőçŽĎåŘłæŸŕçőĂå■ŢçŽĎè¿ŞåĞžäÿĂåőŽåő¡åžęçŽĎæŢřïijŇä¡ääÿ■éIJĂèęĄä¡£çŤĺ `round()` åĞ¡æŢřãĂĆ èĂŇäžĚäžĚåŘłéIJĂèęĄåIJĺæäijåijŘåŇŰçŽĎæŮűåĂŹæŇĞåőŽçš¿åžęå■şåŘŕãĂĆæŕŤ

```
>>> x = 1.23456
>>> format(x, '0.2f')
'1.23'
>>> format(x, '0.3f')
'1.235'
```

(continues on next page)

(continued from previous page)

```
>>> 'value is {:0.3f}'.format(x)
'value is 1.235'
>>>
```

åŘŇæăůïijŇäÿ■èęĄèŕŢçİĂåŐżèĹ■åĚěæţőçĆźåĂijæİěâĂİä£őæ■čâĂİèąĺéİćäÿŁçIJŃèţůæİěæ■čçąőçŽĎéŮ

```
>>> a = 2.1
>>> b = 4.2
>>> c = a + b
>>> c
6.300000000000001
>>> c = round(c, 2) # "Fix" result (???)
>>> c
6.3
>>>
```

åŕźäžŐåď'ğåď'ŽæŢřä¡£çŤĺåĹřæţőçĆźçŽĎçĺŃåžŘïijŇæšąæIJL'å£ĚèęĄäź§äÿ■æŐĺè■Řè£Źæăůå ĄŽãĂĆ
åř¡çőąåIJĺèőąçőŮçŽĎæŮűůåĂŹäijŽæIJL'äÿĂçĆźçĆźåřŘçŽĎèŕŕåůőïijŇä¡ĘæŸŕè£ŹäžŽåřŘçŽĎèŕŕåůőæŸŕèČ¡è
åęĆæđIJäÿ■èČ¡åĚĄèőÿè£ŹæăůçŽĎåřŘèŕŕåůő(æŕŤåęĆæűL'åŘŁåĹřéĞŚèđ■écĘå§§)ïijŇéĆčäźĹåřšå¿ŮèĂČèŹ
`decimal` æĺąåİŮäžĘïijŇäÿŃäÿĂèŁĆæĹŚäžňäijŽèŕęçżĘèőĺèőžãĂĆ

5.2 3.2 æL'ğèąŇçš¿çąőçŽĎæţőçĆźæŢřè£ŘçőŮ

éŮőécŸ

ä¡äéIJĂèęĄåŕźæţőçĆźæŢřæL'ğèąŇçš¿çąőçŽĎèőąçőŮæŞ■ä¡IJïijŇåźűäÿŤäÿ■åÿŇæIJŻæIJL'äżżä¡ŢåřŘèŕŕ

èğčåĘşæŰźæąĹ

æţőçĆźæŢřçŽĎäÿĂäÿłæŹőéĄ■éŮőécŸæŸŕåőČäżňåźűäÿ■èČ¡çš¿çąőçŽĎèąĺçď'žå■Ąè£ŻåĹűæŢřãĂĆ
åźűäÿŤïijŇå■şä¡£æŸŕæIJĂçőĂå■ŢçŽĎæŢřå■ęè£ŘçőŮäź§äijŽäžğçŤ§åřŘçŽĎèŕŕåůőïijŇæŕŤåęĆïijŽ

```
>>> a = 4.2
>>> b = 2.1
>>> a + b
6.300000000000001
>>> (a + b) == 6.3
False
>>>
```

è£ŹäžŽéŤŹèŕŕæŸŕçŤśåžŢåśĆCPUåŠŇIEEE 754æăĞåĞĘéĂŽè£ĞèĞłåůśçŽĎæţőçĆźå■Ţä¡■åŐżæL'ğèąŇ
çŤśäžŐPythonçŽĎæţőçĆźæŢřæ■őçśżåđŃä¡£çŤĺåžŢåśĆèąĺçď'žå■ŸåĆĺæŢřæ■őïijŇåŽäæ■ď'ä¡äæšąåŁđæşŢåŐ

åęĆæđIJä¡äæČşæŻť'åŁäçš¿çąő(åźűèČ¡åőźå£■äÿĂåőŽçŽĎæĂğèČ¡æ■§èĂŮ)ïijŇä¡ääŘŕäžěä¡£çŤĺ
`decimal` æĺąåİŮïijŽ

```
>>> from decimal import Decimal
>>> a = Decimal('4.2')
>>> b = Decimal('2.1')
>>> a + b
Decimal('6.3')
>>> print(a + b)
6.3
>>> (a + b) == Decimal('6.3')
True
```

åĹİçIJŃèţůæİěïijŃäÿŁéİćçŽĎäżčçăĄåě¡åČŕæIJĽçĆźåěĞæĂłïijŃæŕŤåęĆæĹŚäżňçŤĺå■Ůçňęäÿšæİěèąĺçç
çĎűěĂŃïijŃ `Decimal` åŕźèśąäijŽåČŕæŹőéĂŽæţőçĆźæŢřäÿĂæăůçŽĎåůěä¡IJ(æŤŕæŃĄæĽĂæIJĽçŽĎåÿÿçŤ
åęĆæđIJä¡ăæĽŞå■řáőČäżňæĹŰèĂĚåIJĺå■ŮçňęäÿšæăijåijŔåŃŰåĞ¡æŢřäÿ■ä¡£çŤĺáőČäżňïijŃçIJŃèţůæİěèů§a

`decimal` æĺąåİŮçŽĎäÿĂäÿłäÿżèęĄçĽźå¿ĄæŸŕåĚĄèőÿä¡ăæŐğåĹűèőąçőŮçŽĎæŕŔäÿĂæŰźéİćïijŃåŃ
äÿžäžĘè£ŹæăůåĄŽïijŃä¡ăåĚĹå¿ŮåĹŻåżžäÿĂäÿłæIJňåIJřäÿŁäÿŃæŰĞåźűæŻťæŤźáőČçŽĎèő¿ç¡őïijŃæŕŤåęĆ

```
>>> from decimal import localcontext
>>> a = Decimal('1.3')
>>> b = Decimal('1.7')
>>> print(a / b)
0.7647058823529411764705882353
>>> with localcontext() as ctx:
...     ctx.prec = 3
...     print(a / b)
...
0.765
>>> with localcontext() as ctx:
...     ctx.prec = 50
...     print(a / b)
...
0.76470588235294117647058823529411764705882352941176
>>>
```

èőĺèőž

`decimal` æĺąåİŮåőđçŐřäžĘIBMçŽĎâĂİéĂŽçŤĺåřŕæŢřè£ŘçőŮèğĎèŃČâĂİãĂĆäÿ■çŤĺèŕť'ïijŃæIJĽå¿

PythonæŰřæĽŃäijŽåĂ¿åŘŚäžŐä¡£çŤĺ `decimal` æĺąåİŮæİěåď'ĎçŘĘæţőçĆźæŢřçŽĎçš¿çąőè£ŘçőŮãĂ
çĎűěĂŃïijŃåĚĹçŘĘèğčä¡ăçŽĎåžŤçŤĺçĺŃåžŘçŻőçŽĎæŸŕéİđåÿÿéĞ■èęĄçŽĎãĂĆ
åęĆæđIJä¡ăæŸŕåIJĺåĄŽçğŚå■ęèőąçőŮæĹŰåůěçĺŃéćĘå§§çŽĎèőąçőŮãĂĄçŤţèĎŚçżŸåŻ¿ïijŃæĹŰèĂĚæŸŕç
éĆčäźĹä¡£çŤĺæŹőéĂŽçŽĎæţőçĆźçśżåđŃæŸŕæŕŤè¿ČæŹőéĄ■çŽĎåĄŽæşŢãĂĆ
åĚűäÿ■äÿĂäÿłåŐ§åŻăæŸŕïijŃåIJĺçIJ§åőđäÿŰçŢŃäÿ■å¿ĹåŕŚäijŽèęĄæśĆçš¿çąőåĹřæŹőéĂŽæţőçĆźæŢřèČ¡æ
åŻăæ■ď'ïijŃèőąçőŮè£ĞçĺŃäÿ■çŽĎéĆčäźĹäÿĂçĆźçĆźçŽĎèŕŕåůőæŸŕèćńåĚĄèőÿçŽĎãĂĆ
çňňäžŃçĆźåřśæŸŕïijŃåŐ§çŤ§çŽĎæţőçĆźæŢřèőąçőŮèęĄå£ńçŽĎåď'Ž-
æIJĽæŰűåĂŹä¡ăåIJĺæĽ'ğèąŃåď'ğéĞŘè£ŘçőŮçŽĎæŰűåĂŹéĂ§åžęäź§æŸŕéİđåÿÿéĞ■èęĄçŽĎãĂĆ

å■şä¿£åęĆæ■ď'ïijŃä¡ăå■ť'äÿ■èČ¡åőŃåĚĺå£¡çŢěèŕŕåůõãĂĆæŢřå■ęåőűèŁśäžĘåď'ğéĞŘæŮűéŮť'åŐżçăŤçl
ä¡ăäź§å¿ŮæşĺæĎŘäÿŃåĞŘæşŢåĹäéŹď'äžěåŘŁåď'ğæŢřåŠŃåřŘæŢřçŽĎåŁäåĹĘè£ŘçőŮæĽ'ĂåÿęæİěçŽĎå¡śå

```
>>> nums = [1.23e+18, 1, -1.23e+18]
>>> sum(nums) # Notice how 1 disappears
0.0
>>>
```

äÿŁéİćçŽĎéŤŹèŕŕåŘŕäżěåĹľçŤÍ `math.fsum()` æĽĂæŘŘä¿ŻçŽĎæŻť'çš¿çąőèőąçőŮèČ¡åŁŻæİěèğčåĘ

```
>>> import math
>>> math.fsum(nums)
1.0
>>>
```

çĎűèĂŇïijŇåŕźäžŐåĚűäżŰçŽĎçőŮæşŢïijŇä¡ššžŤèŕěäżŤçżEçăŤçľűšőČšźűçŘĘèğčšőČçŽĎèŕŕšůőäžğçŤ

æĂżçŽĎæİěèŕť'ïijŇ `decimal` æĺąšİŮäÿżèęAçŤĺšIJĺæűĽ'šŘŁšĹřéĞŚèđ■çŽĎécĘš§§ãĂĆ
šIJĺè£ŹçśżçĺŃšžŘäÿ■ïijŇšŞłæĂŢæŸŕäÿĂçĆźšřŘšřŘçŽĎèŕŕšůőšIJĺèőąçőŮè£ĞçĺŃäÿ■èŤŞšżűéČ¡æŸŕäÿ■šĚA
šŽäæ■ď'ïijŇ `decimal` æĺąšİŮäÿżèğčšEşè£ŹçśżéŮőécŸæŘŘä¿ŻäžĘæŰźæşŢãĂĆ
š¡ŞPythonšŠŇæŢŕæ■őšžŞæĽ'Şäžď'éĄŞçŽĎæŮűšĂŹäź§éĂŽšÿäijŽéĄĞšĹř `Decimal`
šŕźèśąïijŇšźűäÿŤïijŇéĂŽšÿäź§æŸŕšIJĺšď'ĎçŘĘéĞŚèđ■æŢŕæ■őçŽĎæŮűšĂŹãĂĆ

5.3 3.3 æŢŕš■ŮçŽĎæäijšijŘšŇŰè¿ŞšĞž

éŮőécŸ

ä¡äéIJĂèęĄšřĘæŢŕš■ŮæäijšijŘšŇŰšŘŐè¿ŞšĞžïijŇšźűæŐğšĹűæŢŕš■ŮçŽĎä¡■æŢŕãĂĄšŕźé¡ŘãĂĄš■Č

èğčšEşæŰźæąĹ

æäijšijŘšŇŰè¿ŞšĞžš■ŢäÿłæŢŕš■ŮçŽĎæŮűšĂŹïijŇšŘŕäżěä¡£çŤĺšĘĚç¡őçŽĎ
`format()` šĞ¡æŢŕïijŇæŕŤšęĆïijŽ

```
>>> x = 1234.56789

>>> # Two decimal places of accuracy
>>> format(x, '0.2f')
'1234.57'

>>> # Right justified in 10 chars, one-digit accuracy
>>> format(x, '>10.1f')
'    1234.6'

>>> # Left justified
>>> format(x, '<10.1f')
'1234.6    '

>>> # Centered
>>> format(x, '^10.1f')
```

(continues on next page)

(continued from previous page)

```
'  1234.6  '

>>> # Inclusion of thousands separator
>>> format(x, ',')
'1,234.56789'
>>> format(x, '0,.1f')
'1,234.6'
>>>
```

åęĆæđIJä¡ăæČşä¡£çŤĺæŇĞæŢřèőřæşŢïijŇåřĘfæŤźæĹŘeæĹŰèĂĚE(åŘŰåĘşäžŐæŇĞæŢřè¿ŞåĞžçŽĎåd

```
>>> format(x, 'e')
'1.234568e+03'
>>> format(x, '0.2E')
'1.23E+03'
>>>
```

åŘŇæŮűæŇĞåőŽåő¡åžęåŠŇçš¿åžęçŽĎäÿĂèĹňå¡ćåijŘæŸŕ
`'[<>^]?width[,]?(.digits)?'` ïijŇ åĚűäÿ■ `width` åŠŇ `digits` äÿžæŢť æŢřïijŇïij§äżčèąĺåŘŕéĂĽéČĺåĹĘãĂĆ åŘŇæăůçŽĎæăijåijŘäź§èćńçŤĺåIJĺå■ŮçňęäÿšçŽĎ `format()` æŰźæşŢäÿ■ãĂĆæŕŤåęĆïijŽ

```
>>> 'The value is {:0,.2f}'.format(x)
'The value is 1,234.57'
>>>
```

èőĺèőž

æŢřå■ŮæăijåijŘåŇŰè¿ŞåĞžéĂŽåÿÿæŸŕæŕŤè¿ČçőĂå■ŢçŽĎãĂĆäÿŁéİćæijŤçď žçŽĎæŁĂæIJŕåŘŇæŮűĺ
`decimal` æĺąåİŮäÿ■çŽĎ `Decimal` æŢřå■ŮåŕźèśąãĂĆ

å¡ŞæŇĞåőŽæŢřå■ŮçŽĎä¡■æŢřåŘŐïijŇçżŞæđIJåĂijäijŽæăźæ■ő `round()` åĞ¡æŢřåŘŇæăůçŽĎèğĎåĹŹè£ŻèąŇåŻŻèĹ■ážŤåĚěåŘŐè£ŤåŻđãĂĆæŕŤåęĆïijŽ

```
>>> x
1234.56789
>>> format(x, '0.1f')
'1234.6'
>>> format(-x, '0.1f')
'-1234.6'
>>>
```

åŇĚåŘńå■Čä¡■çňęçŽĎæăijåijŘåŇŰèů§æIJňåIJřåŇŰæšąæIJL'åĚşçşżãĂĆ
åęĆæđIJä¡ăéIJĂèęĄæăźæ■őåIJřåŇžæİěæŸ¿çď žå■Čä¡■çňęïijŇä¡ăéIJĂèęĄèĞłåůśåŐżèřČæ§ěäÿŃ
`locale` æĺąåİŮäÿ■çŽĎåĞ¡æŢřäžĘãĂĆ ä¡ăåŘŇæăůäź§åŘŕäżěä¡£çŤĺå■ŮçňęäÿšçŽĎ
`translate()` æŰźæşŢæİěäžď æ■ćå■Čä¡■çňęãĂĆæŕŤåęĆïijŽ

```
>>> swap_separators = { ord('.'):',', ord(','):'.' }
>>> format(x, ',').translate(swap_separators)
'1.234,56789'
>>>
```

åIJÍå¿Ĺåď ŽPythonäžčçăĄäÿ■äijŽçIJŃåĹřä¡£çŤÍ%æĺěæäijåijŔåŇŰæŢřå■ŮçŽĎïijŇæŕŤåęĆïijŽ

```
>>> '%0.2f' % x
'1234.57'
>>> '%10.1f' % x
'    1234.6'
>>> '%-10.1f' % x
'1234.6    '
>>>
```

è£ŹçğŒæäijåijŔåŇŰæŰźæşŢäź§æŸŕåŔŕèąŇçŽĎïijŇäÿ■è£ĞæŕŤæŻť åŁäåĚĹè£ŻçŽĎ `format()` èęĄåůőäÿĂçĆźãĂĆ æŕŤåęĆïijŇåIJĺä¡£çŤÍ%æŞ■ä¡IJçňęæäijåijŔåŇŰæŢřå■ŮçŽĎæŮűåĂŹïijŇäÿ

5.4 3.4 äžŇåĚńå■ĄåĚ■è£ŻåĹűæŢťæŢř

éŮőécŸ

ä¡äéIJĂèęĄè¡ňæ■ćæĹŰèĂĚè¿ŞåĞžä¡£çŤĺäžŇè£ŻåĹűïijŇåĚńè£ŻåĹűæĹŰå■ĄåĚ■è£ŻåĹűèąĺçď žçŽĎæŢ

èğčåĘşæŰźæąĹ

äÿžäžĘåřĘæŢť æŢřè¡ňæ■ćäÿžäžŇè£ŻåĹűãĂĄåĚńè£ŻåĹűæĹŰå■ĄåĚ■è£ŻåĹűçŽĎæŰĞæIJňäÿšïijŇ åŔŕäżěåĹĘåĹńä¡£çŤÍ `bin()`，`oct()` æĹŰ `hex()` åĞ¡æŢřïijŽ

```
>>> x = 1234
>>> bin(x)
'0b10011010010'
>>> oct(x)
'0o2322'
>>> hex(x)
'0x4d2'
>>>
```

åŔęåď ŰïijŇåęĆæđIJä¡ääÿ■æČşè¿ŞåĞž `0b`，`0o` æĹŰèĂĚ `0x` çŽĎåĽ■çijĂçŽĎèŕÏïijŇåŔŕäżěä¡£çŤÍ `format()` åĞ¡æŢřãĂĆæŕŤåęĆïijŽ

```
>>> format(x, 'b')
'10011010010'
>>> format(x, 'o')
'2322'
>>> format(x, 'x')
```

(continues on next page)

(continued from previous page)

```
'4d2'
>>>
```

æŢťæŢřæŸŕæIJL'çňęåŘůçŽĎïijŇæL'ĂäžěåęĆæđIJä¡ăåIJĺåď'ĎçŘĘèť§æŢřçŽĎèŕĺïijŇè¿ŞåĞžçżŞæđIJäij

```
>>> x = -1234
>>> format(x, 'b')
'-10011010010'
>>> format(x, 'x')
'-4d2'
>>>
```

åęĆæđIJä¡ăæČşäžğçŤ§äÿĂäÿłæŮăçňęåŘůåĂijïijŇä¡ăéIJĂèęĄåćđåŁääÿĂäÿłæŇĞçď'žæIJĂåď'ğä¡■éŢ£åž

```
>>> x = -1234
>>> format(2**32 + x, 'b')
'11111111111111111111101100101110'
>>> format(2**32 + x, 'x')
'fffffb2e'
>>>
```

äÿžäžĘäżěäÿ■åŘŇçŽĎè£ŻåĹűè¡ňæ■ćæŢť'æŢřå■ŮçňęäÿšïijŇçőĂå■ŢçŽĎä¡£çŤĺåÿęæIJL'è£ŻåĹűçŽĎ `int()` åĞ¡æŢřå■şåŘŕïijŽ

```
>>> int('4d2', 16)
1234
>>> int('10011010010', 2)
1234
>>>
```

èőĺèőž

åď'ğåď'ŽæŢřæČĚåĘţäÿŃåď'ĎçŘĘäžŇè£ŻåĹűãĂĄåĚńè£ŻåĹűåŠŇå■ĄåĚ■è£ŻåĹűæŢť'æŢřæŸŕå¿ĹçőĂåı
åŘłèęĄèőřä¡Ŕè£ŹäžŻè¡ňæ■ćåśđäžŐæŢť'æŢřåŠŇåĚűåŕźåžŤçŽĎæŰĞæIJňèąĺçď'žäźŃéŮť'çŽĎè¡ňæ■ćå■şåŘŕã

æIJĂåŘŐïijŇä¡£çŤĺåĚńè£ŻåĹűçŽĎçĺŃåžŘåŚŸæIJL'äÿĂçĆźéIJĂèęĄæşĺæĎŘäÿŃãĂĆ
PythonæŇĞåőŽåĚńè£ŻåĹűæŢřçŽĎèŕ■æşŢèů§åĚűäżŰèŕ■èĺĂçĺ■æIJL'äÿ■åŘŇãĂĆæŕŤåęĆïijŇåęĆæđIJä¡ăåČ

```
>>> import os
>>> os.chmod('script.py', 0755)
    File "<stdin>", line 1
        os.chmod('script.py', 0755)
                              ^
SyntaxError: invalid token
>>>
```

éIJĂçąőä£ĺåĚńè£ŻåĹűæŢřçŽĎåL'■çijĂæŸŕ `0o` ïijŇåŕśåČŘäÿŃéĺćè£ŹæăůïijŽ

```
>>> os.chmod('script.py', 0o755)
>>>
```

5.5 3.5 å■ŮèŁĆåĹřåďğæŢťæŢřçŽĎæĽŞåŇĚäÿŐèğčåŇĚ

éŮőécŸ

ä¡ăæIJĽäÿĂäÿłå■ŮèŁĆå■ŮçňęäÿšåźűæČşåřĘåőČèğčåŐŃæĹŘäÿĂäÿłæŢťæŢřãĂĆæĹŰèĂĚïijŇä¡ăéIJĂ

èğčåĘşæŰźæąĹ

åĄĞèő¿ä¡ăçŽĎçÍŃåžŘéIJĂèęĄåďĎçŘĘäÿĂäÿłæŃĕæIJĽ128ä¡■éŢ£çŽĎ16äÿłåĚČçťăçŽĎå■ŮèŁĆå■Ůç

```
data = b'\x00\x124V\x00x\x90\xab\x00\xcd\xef\x01\x00#\x004'
```

äÿžäžĘåřĘbytesèğčæđŘäÿžæŢťæŢřïijŇä¡£çŤÍ `int.from_bytes()` æŰźæşŢïijŇåźűåČŘäÿŃéİćè£ŹæăůæŇĞåőŽå■ŮèŁĆéąžåžŘïijŽ

```
>>> len(data)
16
>>> int.from_bytes(data, 'little')
69120565665751139577663547927094891008
>>> int.from_bytes(data, 'big')
94522842520747284487117727783387188
>>>
```

äÿžäžĘåřĘäÿĂäÿłåďğæŢťæŢřè¡ňæ■ćäÿžäÿĂäÿłå■ŮèŁĆå■ŮçňęäÿšïijŇä¡£çŤÍ `int.to_bytes()` æŰźæşŢïijŇåźűåČŘäÿŃéİćè£ŹæăůæŇĞåőŽå■ŮèŁĆæŢřåŠŇå■ŮèŁĆéąžåžŘïijŽ

```
>>> x = 94522842520747284487117727783387188
>>> x.to_bytes(16, 'big')
b'\x00\x124V\x00x\x90\xab\x00\xcd\xef\x01\x00#\x004'
>>> x.to_bytes(16, 'little')
b'4\x00#\x00\x01\xef\xcd\x00\xab\x90x\x00V4\x12\x00'
>>>
```

èőÍèőž

åďğæŢťæŢřåŠŇå■ŮèŁĆå■ŮçňęäÿšäźŃéŮťçŽĎè¡ňæ■ćæŞ■ä¡IJåźűäÿ■åÿÿèğĄãĂĆ çĎűèĂŃïijŇåIJÍäÿĂäžŻåžŤçŤÍéćĘå§§æIJĽæŮűåĂŹäź§äijŽåĞžçŐřïijŇæŕŤåęĆåŕĘçăĄå■ęæĹŰèĂĚç¡ŚçżIJã ä¿ŃåęĆïijŇIPv6ç¡ŚçżIJåIJřåİĂä¡£çŤÍäÿĂäÿł128ä¡■çŽĎæŢťæŢřèąÍçďžãĂĆ åęĆæđIJä¡äèęĄäżŐäÿĂäÿłæŢřæ■őèőřå¡Ţäÿ■æŘřåŘŰè£ŹæăůçŽĎåĂijçŽĎæŮűåĂŹïijŇä¡ăářśäijŽéİćářźè£Ź

ä¡IJäÿžäÿĂçğ■æŻ£äžčæŰźæąĹïijŇä¡ăåŘŕèČ¡æČşä¡£çŤÍ6.11ářŘèŁĆäÿ■æĽĂäżŃçż■çŽĎ `struct` æÍąåİŮæİěèğčåŐŃå■ŮèŁĆãĂĆ è£ŹæăůäźŝèąŇå¿ŮéĂŽïijŇäÿ■è£ĞåĹĺçŤÍ

struct æÍąåİŮæİěèğčåŐŃåŕźäžŐæŢťæŢřçŽĎåď ğåřŘæŸŕæIJĽéŹŘåĹűçŽĎãĂĆ
åŻäæ■ď ïijŇä¡ääŘŕèČ¡æČşèğčåŐŃåď Žäÿłå■ŮèŁĆäÿšåźűåřĘçżŞæđIJåŘĹåźűäÿžæIJĂçżĹçŽĎçżŞæđIJïijŇåŕś

```
>>> data
b'\x00\x124V\x00x\x90\xab\x00\xcd\xef\x01\x00#\x004'
>>> import struct
>>> hi, lo = struct.unpack('>QQ', data)
>>> (hi << 64) + lo
94522842520747284487117727783387188
>>>
```

å■ŮèŁĆéąžäžŘèğĎåĹŹ(littleæĹŰbig)äżĚäżĚæŇĞäőŽäžĘæđĎäżžæŢťæŢřæŮűçŽĎå■ŮèŁĆçŽĎä¡Őä¡■
æĹŚäżňäżŐäÿŃéİćçš¿å£ČæđĎéĂäçŽĎ16è£ŻåĹűæŢřçŽĎèąĺçď žäÿ■åŘŕäżěä¿ĹäőźæŸ ŞçŽĎçIJŃåĞżæİěïijŽ

```
>>> x = 0x01020304
>>> x.to_bytes(4, 'big')
b'\x01\x02\x03\x04'
>>> x.to_bytes(4, 'little')
b'\x04\x03\x02\x01'
>>>
```

åęĆæđIJä¡ăèŕŢçİĂåŕĘäÿĂäÿłæŢťæŢřæĽŞåŇĚäÿžå■ŮèŁĆå■ŮçňęäÿšïijŇéĆčäźĹäőČåŕśäÿ■åŘĹéĂĆäžĚ
åęĆæđIJéIJĂèęĄçŽĎèŕÍïijŇä¡ääŘŕäżěä¡£çŤÍ int.bit_length()
æŰźæşŢæİěåĘşäőŽéIJĂèęĄåď ŽäŕŚå■ŮèŁĆä¡■æİěå■ŸåĆÍè£ŹäÿłåĂijãĂĆ

```
>>> x = 523 ** 23
>>> x
3353813001136618751075368527140190561603556553339788490179440 67
>>> x.to_bytes(16, 'little')
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
OverflowError: int too big to convert
>>> x.bit_length()
208
>>> nbytes, rem = divmod(x.bit_length(), 8)
>>> if rem:
... nbytes += 1
...
>>>
>>> x.to_bytes(nbytes, 'little')
b'\x03X\xf1\x82iT\x96\xac\xc7c\x16\xf3\xb9\xcf...\xd0'
>>>
```

5.6 3.6 åď■æŢřçŽĎæŢřå■ęè£ŘçőŮ

éŮőécŸ

ä¡ääĘźçŽĎæIJĂæŰřçŽĎç¡ŚçżIJèőď èŕĄæŰźæąĹäżčçăĄéĄĞåĹřäžĘäÿĂäÿłéŽ¿écŸïijŇäźűäÿŤä¡ääŤŕäÿ
åĘ■æĹŰèĂĚæŸŕä¡ääżĚäżĚéIJĂèęĄä¡£çŤĺåď ■æŢřæİěæĽ ğèąŇäÿĂäžŻèőąçőŮæŞ■ä¡IJãĂĆ

èğčåĘşæŰźæąĹ

åď■æŢřåŘŕäżěçŤĺä¡£çŤĺåĞ¡æŢř `complex(real, imag)` æĹŰèĂĚæŸŕåÿęæIJĽåŘŐçijĂjçŽĎæţőçĆźæŢřæİěæŇĞåőŽãĂĆæŕŤåęĆïijŽ

```
>>> a = complex(2, 4)
>>> b = 3 - 5j
>>> a
(2+4j)
>>> b
(3-5j)
>>>
```

åŕźåžŤçŽĎåőđéČĺãĂĄèŹŽéČĺåŠŇåĚśè¡■åď■æŢřåŘŕäżěå¿Ĺåőźæ ŸŞçŽĎèŐůåŘŰãĂĆåřśåČŘäÿŃéİćè£Ź

```
>>> a.real
2.0
>>> a.imag
4.0
>>> a.conjugate()
(2-4j)
>>>
```

åŘęåď Űïij ŇæĽĂæIJĽåÿÿèğĄçŽĎæŢřå■ęè£ŘçőŮéČ¡åŕäżěåůěä¡IJïijŽ

```
>>> a + b
(5-1j)
>>> a * b
(26+2j)
>>> a / b
(-0.4117647058823529+0.6470588235294118j)
>>> abs(a)
4.47213595499958
>>>
```

åęĆæđIJèęĄæĽğèąŇåĚűäżŰçŽĎåď■æŢřåĞ¡æŢřæŕŤåęĆæ■čåijęãĂĄä¡Źåijęæ ĹŰåźşæŰźæäźïijŇä¡£çŤĺ `cmath` æĺąåİŮïijŽ

```
>>> import cmath
>>> cmath.sin(a)
(24.83130584894638-11.356612711218174j)
>>> cmath.cos(a)
(-11.36423470640106-24.814651485634187j)
>>> cmath.exp(a)
(-4.829809383269385-5.5920560936409816j)
>>>
```

èőÍèőž

Pythonäÿ■åď ğéČÍåĹĘäÿŐæŢřå■ęçŻÿåĚşçŽĎæÍąåİŮéČ¡èČ¡åď ĎçŘĘåď ■æŢřãĂĆ æŕŤåęĆåęĆæđIJä¡ăä¡£çŤÍ `numpy` ïijŇåŔŕäżěå¿ĹåőźæŸŞçŽĎæđĎéĂăäÿĂäÿłåď ■æŢřæŢřçżĎåźűåIJÍè£Źäÿłæ

```
>>> import numpy as np
>>> a = np.array([2+3j, 4+5j, 6-7j, 8+9j])
>>> a
array([ 2.+3.j, 4.+5.j, 6.-7.j, 8.+9.j])
>>> a + 2
array([ 4.+3.j, 6.+5.j, 8.-7.j, 10.+9.j])
>>> np.sin(a)
array([ 9.15449915 -4.16890696j, -56.16227422 -48.50245524j,
        -153.20827755-526.47684926j, 4008.42651446-589.49948373j])
>>>
```

PythonçŽĎæăĞåĞĘæŢřå■ęåĞ¡æŢřçąőåőđæČĚåĘţäÿŃåźűäÿ■èČ¡äžğçŤ§åď ■æŢřåĂijïijŇåŻăæ■ď ä¡ăçŽ

```
>>> import math
>>> math.sqrt(-1)
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
ValueError: math domain error
>>>
```

åęĆæđIJä¡ăæČşçŤ§æĹŘäÿĂäÿłåď ■æŢřè£ŤåŻđçżŞæđIJïijŇä¡ăå£Ěéążæ Ÿ¿çď žçŽĎä¡£çŤÍ `cmath` æÍąåİŮïijŇæĹŰèĂĚåIJÍæ§Řäÿłæ ŤŕæŇĄåď ■æŢřçŽĎåžŞäÿ■åčřæŸŐåď ■æŢřçśżåđŃçŽĎä¡£çŤÍãĂĆæ

```
>>> import cmath
>>> cmath.sqrt(-1)
1j
>>>
```

5.7 3.7 æŮăçľ ůåď ğäÿŐNaN

éŮőéćŸ

ä¡ăæČşåĹŻåżžæĹŰæţŃèŕŢæ■čæŮăçľ ůãĂĄèť §æŮăçľ ůæĹŰNaN(éİđæŢřå■Ů)çŽĎæţőçĆźæŢřãĂĆ

èğčåĘşæŰźæąĹ

PythonåźűæšąæIJĽçĽ źæőŁçŽĎèŕ■æşŢæİěèąÍçď žè£ŹäžŻçĽ źæőŁçŽĎæţőçĆźåĂijïijŇä¡ĘæŸŕåŔŕäżěä¡£ `float()` æİěåĹŻåżžåőČäżňãĂĆæŕŤåęĆïijŽ

```
>>> a = float('inf')
>>> b = float('-inf')
>>> c = float('nan')
```

(continues on next page)

(continued from previous page)

```
>>> a
inf
>>> b
-inf
>>> c
nan
>>>
```

äÿžäžĘæţŃèŕŢè£ŹäžŻåĂijçŽĎå■ŸåIJÍïijŇä¡£çŤÍ `math.isinf()` åŠŇ `math.isnan()` åĞ¡æŢřãĂĆæŕŤåęĆïijŽ

```
>>> math.isinf(a)
True
>>> math.isnan(c)
True
>>>
```

èőÍèőž

æČşäžĘèğčæŻť'åď'Žè£ŹäžŻçL'źæőŁæţőçĆźåĂijçŽĎä£ąæĄŕïijŇåŘŕäżěåŘĆèĂČIEEE 754èğĎèŇČãĂĆ çĎűèĂŇïijŇäź§æIJL'äÿĂäžŻåIJřæŰźéIJĂèęĄä¡ăçL'źåĹńæşÍæĎŘïijŇçL'źåĹńæŸŕèů§æŕŤè¿

æŮăçl'ůåď'ğæŢřåIJÍæL'ğèąŇæŢřå■ęèőąçőŮçŽĎæŮűåĂŹäijŽäijăæŠ■ïijŇæŕŤåęĆïijŽ

```
>>> a = float('inf')
>>> a + 45
inf
>>> a * 10
inf
>>> 10 / a
0.0
>>>
```

ä¡ĘæŸŕæIJL'äžŻæŞ■ä¡IJæŮűæIJłőŽäźL'çŽĎåźűäijŽè£ŤåŻđäÿĂäÿłNaNçżŞæđIJãĂĆæŕŤåęĆïijŽ

```
>>> a = float('inf')
>>> a/a
nan
>>> b = float('-inf')
>>> a + b
nan
>>>
```

NaNåĂijäijŽåIJÍæL'ĂæIJL'æŞ■ä¡IJäÿ■äijăæŠ■ïijŇèĂŇäÿ■äijŽäžğçŤ§åijĆåÿÿãĂĆæŕŤåęĆïijŽ

```
>>> c = float('nan')
>>> c + 23
nan
```

(continues on next page)

(continued from previous page)

```
>>> c / 2
nan
>>> c * 2
nan
>>> math.sqrt(c)
nan
>>>
```

NaNåĂijçŽĎäÿĂäÿłçL'źåĹńçŽĎåIJřæŰźæŮűåőČäżňäźŃéŮť'çŽĎæŕŤè¿ČæŞ■ä¡IJæĂżæŸŕè£ŤåŻđFalse

```
>>> c = float('nan')
>>> d = float('nan')
>>> c == d
False
>>> c is d
False
>>>
```

çŤśäžŐè£ŹäÿłåŐ§åŻäïijŇæţŃèŕŢäÿĂäÿłNaNåĂijå¿ŮåŤŕäÿĂåőL'åĚĺçŽĎæŰźæşŢåŕśæŸŕä¡£çŤĺ `math.isnan()` ïijŇäź§åŕśæŸŕäÿŁéİćæijŤçď'žçŽĎéĆčæăůãĂĆ

æIJL'æŮűåĂŹçĺŃåžŘåŚŸæČşæŤźåŘŸPythonéżŸèőď'èąŇäÿžïijŇåIJĺè£ŤåŻđæŮăçľ'ůåď'ğæĹŰNaNçżŞæ `fpectl` æĺąåİŮåŘŕäżěçŤĺæİěæŤźåŘŸè£Źçğ■èąŇäÿžïijŇä¡ĘæŸŕåőČåIJĺæăĞåĞĘçŽĎPythonæđĎåżžäÿ■åźűäźűäÿŤéŠĹåŕźçŽĎæŸŕäÿŞåőűçžğçĺŃåžŘåŚŸãĂĆåŘŕäżěåŘĆèĂČåIJĺçž£çŽĎPythonæŰĞæąčèŐůåŘŰæŻť'åď

5.8 3.8 åĹĘæŢřè£ŘçőŮ

éŮőécŸ

ä¡ăè£ŻåĚěæŮűéŮť'æIJžåŹĺïijŇçłĄçĎűåŘŚçŐřä¡ăæ■čåIJĺåĄŽåřŘå■ęåőűåž■ä¡IJäÿŽïijŇåźűæűL'åŘŁåĹřæĹŰèĂĚä¡ăåŘŕèČ¡éIJĂèęĄåĘŹäżčçăĄåŐżèőąçőŮåIJĺä¡ăçŽĎæIJĺåůěåůěåŐĆäÿ■çŽĎæţŃéĞŘåĂijãĂĆ

èğčåĘşæŰźæąĹ

`fractions` æĺąåİŮåŘŕäżěèćńçŤĺæİěæL'ğèąŇåŇĚåŘńåĹĘæŢřçŽĎæŢřå■ęè£ŘçőŮãĂĆæŕŤåęĆïijŽ

```
>>> from fractions import Fraction
>>> a = Fraction(5, 4)
>>> b = Fraction(7, 16)
>>> print(a + b)
27/16
>>> print(a * b)
35/64

>>> # Getting numerator/denominator
>>> c = a * b
```

(continues on next page)

(continued from previous page)

```
>>> c.numerator
35
>>> c.denominator
64

>>> # Converting to a float
>>> float(c)
0.546875

>>> # Limiting the denominator of a value
>>> print(c.limit_denominator(8))
4/7

>>> # Converting a float to a fraction
>>> x = 3.75
>>> y = Fraction(*x.as_integer_ratio())
>>> y
Fraction(15, 4)
>>>
```

èőÍèőž

åIJÍåď ğåď ŽæŢřçÍŃåžŘäÿ■äÿĂèĹňäÿ■äijŽåĞžçŐřåĹĘæŢřçŽĎèőąçőŮéŮőécŸïijŇä¡ĘæŸŕæIJL’æŮűåĂŽ
æŕŤåęĆïijŇåIJÍäÿĂäÿłåĚĄèőÿæŐěåŘŮåĹĘæŢřå¡ćåijŘçŽĎæţŃèŕŢå■Ţä¡■åźűäżěåĹĘæŢřå¡ćåijŘæL’ğèąŇè£Ř
çŻť æŐěä¡£çŤÍåĹĘæŢřåŘŕäżěåĞŘåřŚæL’ŃåŁÍè¡ňæ■ćäÿžåřŘæŢřæĹŰæţőçĆźæŢřçŽĎåůěä¡IJãĂĆ

5.9 3.9 åď ğåđŃæŢřçżĎè£ŘçőŮ

éŮőécŸ

ä¡ăéIJĂèęĄåIJÍåď ğæŢřæ■őéŻĘ(æŕŤåęĆæŢřçżĎæĹŰç¡Śæăij)äÿŁéİćæL’ğèąŇèőąçőŮãĂĆ

èğčåĘşæŰźæąĹ

æűL’åŘŁåĹřæŢřçżĎçŽĎéĞ■éĞŘçžğè£ŘçőŮæŞ■ä¡IJïijŇåŘŕäżěä¡£çŤÍ `NumPy` åžŞãĂĆ
`NumPy` çŽĎäÿĂäÿłäÿżèęĄçL’źå¿ĄæŸŕåőČäijŽçżŹPythonæŘŘä¿ŻäÿĂäÿłæŢřçżĎåŕźèśąïijŇçŻÿæŕŤæăĞåĞĘ
äÿŃéİćæŸŕäÿĂäÿłçőĂå■ŢçŽĎåřŘä¿Ńå■ŘïijŇåŘŚä¡ăåśŢçd’żæăĞåĞĘåĹŮèąĺåŕźèśąåŠŇ
`NumPy` æŢřçżĎåŕźèśąäźŃéŮt’çŽĎåůőåĹńïijŽ

```
>>> # Python lists
>>> x = [1, 2, 3, 4]
>>> y = [5, 6, 7, 8]
>>> x * 2
[1, 2, 3, 4, 1, 2, 3, 4]
```

(continues on next page)

(continued from previous page)

```
>>> x + 10
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
TypeError: can only concatenate list (not "int") to list
>>> x + y
[1, 2, 3, 4, 5, 6, 7, 8]

>>> # Numpy arrays
>>> import numpy as np
>>> ax = np.array([1, 2, 3, 4])
>>> ay = np.array([5, 6, 7, 8])
>>> ax * 2
array([2, 4, 6, 8])
>>> ax + 10
array([11, 12, 13, 14])
>>> ax + ay
array([ 6, 8, 10, 12])
>>> ax * ay
array([ 5, 12, 21, 32])
>>>
```

æ■čåęĆæĽĂèğĄïijŇäÿď'çğ■æŰźæąĹäÿ■æŢřçżĎçŽĎå§žæIJňæŢřå■ęè£ŘçŐŮçżŞæđIJåźűäÿ■çŻÿåŘŇãĂ
çĽźåĹńçŽĎïijŇ NumPy äÿ■çŽĎæăĞéĞŘè£ŘçŐŮ(æŕŤåęĆ ax
* 2 æĹŰ ax + 10)äijŽä¡IJçŤĺåIJĺæŕŘäÿĂäÿłåĚČçť'ääÿŁãĂĆ
åŘęåď'ŰïijŇå¡Şäÿď'äÿłæŞ■ä¡IJæŢřéČ¡æŸŕæŢřçżĎçŽĎæŮűåĂŹæĽ'ğèąŇåĚČçť'ääŕźç■Ľ'ä¡■ç¡ŏèŏąçŐŮïijŇåźŭ

åŕźæŢť'äÿłæŢřçżĎäÿ■æĽ'ĂæIJĽ'åĚČçť'ääŘŇæŮűæĽ'ğèąŇæŢřå■ęè£ŘçŐŮåŘŕäżěä¡£å¿Ůä¡IJçŤĺåIJĺæŢť'ä
æŕŤåęĆïijŇåęĆæđIJä¡äæČşèŏąçŐŮåď'ŽéąźåijŕçŽĎåĂijïijŇåŘŕäżěè£ŹæăůåĄŽïijŽ

```
>>> def f(x):
... return 3*x**2 - 2*x + 7
...
>>> f(ax)
array([ 8, 15, 28, 47])
>>>
```

NumPy è£ŸäÿžæŢřçżĎæŞ■ä¡IJæŘŘä¿ŻäžĘåď'ğéĞŘçŽĎéĂŽçŤĺåĞ¡æŢřïijŇè£ŹäžŻåĞ¡æŢřåŘŕäżěä¡IJäÿ
math æĺąåİŮäÿ■çśżäijijåĞ¡æŢřçŽĎæŻ£äżčãĂĆæŕŤåęĆïijŽ

```
>>> np.sqrt(ax)
array([ 1. , 1.41421356, 1.73205081, 2. ])
>>> np.cos(ax)
array([ 0.54030231, -0.41614684, -0.9899925 , -0.65364362])
>>>
```

ä¡£çŤĺè£ŹäžŻéĂŽçŤĺåĞ¡æŢřèęĄæŕŤå¿łçŐŕæŢřçżĎåźűä¡£çŤĺ
math æĺąåİŮäÿ■çŽĎåĞ¡æŢřæĽ'ğèąŇèŏąçŐŮèęĄå£ńçŽĎåď'ŽãĂĆ
åŽäæ■ď'ïijŇåŘłèęĄæIJĽ'åŘŕèČ¡çŽĎèŕĺåř¡éĞŘéĂĽ'æŃl' NumPy çŽĎæŢřçżĎæŰźæąĹãĂĆ

åžŢåśĆåőđçŐřäÿ■ïijŇ NumPy æŢřçżĎä¡£çŤĺäžĘCæĹŰèĂĚFortranèŕ■èĺĂçŽĎæIJžåĹűåĹĘéĚ■åĘĚå■Ÿã

äź§åŕśæŸŕèŕť'ïijŇåőČäžňæŸŕäÿĂäÿłéİđåÿÿåď'ğçŽĎè£đçż■çŽĎåźűçŤśåŘŇçśżåđŃæŢřæ■őçżĎæĹŘçŽĎåĘĚå
æĽĂäżěïijŇä¡ăåŘŕäžěæđĎéĂääÿĂäÿłæŕŤæŹőéĂŽPythonåĹŮèąĺåď'ğçŽĎåď'ŽçŽĎæŢřçżĎãĂĆ
æŕŤåęĆïijŇåęĆæđIJä¡ăæČşæđĎéĂääÿĂäÿł10,000*10,000çŽĎæţőçĆźæŢřäžŇçżť'ç¡ŚæăijïijŇå¿Ĺè¡żæİ¿ïijŽ

```
>>> grid = np.zeros(shape=(10000,10000), dtype=float)
>>> grid
    array([[ 0., 0., 0., ..., 0., 0., 0.],
    [ 0., 0., 0., ..., 0., 0., 0.],
    [ 0., 0., 0., ..., 0., 0., 0.],
    ...,
    [ 0., 0., 0., ..., 0., 0., 0.],
    [ 0., 0., 0., ..., 0., 0., 0.],
    [ 0., 0., 0., ..., 0., 0., 0.]])
>>>
```

æĽĂæIJĽçŽĎæŹőéĂŽæŞ■ä¡IJè£ŸæŸŕäijŽåŘŇæŮűä¡IJçŤĺåIJĺæĽĂæIJĽåĚČçť'ăäÿŁïijŽ

```
>>> grid += 10
>>> grid
array([[ 10., 10., 10., ..., 10., 10., 10.],
    [ 10., 10., 10., ..., 10., 10., 10.],
    [ 10., 10., 10., ..., 10., 10., 10.],
    ...,
    [ 10., 10., 10., ..., 10., 10., 10.],
    [ 10., 10., 10., ..., 10., 10., 10.],
    [ 10., 10., 10., ..., 10., 10., 10.]])
>>> np.sin(grid)
array([[-0.54402111, -0.54402111, -0.54402111, ..., -0.54402111,
        -0.54402111, -0.54402111],
    [-0.54402111, -0.54402111, -0.54402111, ..., -0.54402111,
        -0.54402111, -0.54402111],
    [-0.54402111, -0.54402111, -0.54402111, ..., -0.54402111,
        -0.54402111, -0.54402111],
    ...,
    [-0.54402111, -0.54402111, -0.54402111, ..., -0.54402111,
        -0.54402111, -0.54402111],
    [-0.54402111, -0.54402111, -0.54402111, ..., -0.54402111,
        -0.54402111, -0.54402111],
    [-0.54402111, -0.54402111, -0.54402111, ..., -0.54402111,
        -0.54402111, -0.54402111]])
>>>
```

åĚşäžŐ `NumPy` æIJĽäÿĂçĆźéIJĂèęĄçĽźåĹńçŽĎäÿżæĎŕïijŇéĆčåŕśæŸŕåőČæĽľåśŢPythonåĹŮèąĺçŽĎ
- çĽźåĹńæŸŕåŕźäžŐåď'Žçżť'æŢřçżĎãĂĆ äÿžäžĘèŕť'æŸŐæÿĚæěŽïijŇåĚĹæđĎéĂääÿĂäÿłçőĂå■ŢçŽĎäžŇçżť

```
>>> a = np.array([[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]])
>>> a
array([[ 1, 2, 3, 4],
[ 5, 6, 7, 8],
[ 9, 10, 11, 12]])
```

(continues on next page)

(continued from previous page)

```
>>> # Select row 1
>>> a[1]
array([5, 6, 7, 8])

>>> # Select column 1
>>> a[:,1]
array([ 2, 6, 10])

>>> # Select a subregion and change it
>>> a[1:3, 1:3]
array([[ 6, 7],
        [10, 11]])
>>> a[1:3, 1:3] += 10
>>> a
array([[ 1, 2, 3, 4],
        [ 5, 16, 17, 8],
        [ 9, 20, 21, 12]])

>>> # Broadcast a row vector across an operation on all rows
>>> a + [100, 101, 102, 103]
array([[101, 103, 105, 107],
        [105, 117, 119, 111],
        [109, 121, 123, 115]])
>>> a
array([[ 1, 2, 3, 4],
        [ 5, 16, 17, 8],
        [ 9, 20, 21, 12]])

>>> # Conditional assignment on an array
>>> np.where(a < 10, a, 10)
array([[ 1, 2, 3, 4],
        [ 5, 10, 10, 8],
        [ 9, 10, 10, 10]])
>>>
```

èőÍèőž

NumPy æŸŕPythonéćĘå§§äÿ■å¿Ĺåď ŽçğŚå■ęäÿŐåůěçÍŃåžŞçŽĎå§žçąĂïijŇåŘŇæŮűäź§æŸŕèćńåź£æşŽ å■şä¿£åęĆæ■ď ïijŇåIJÍåĹŽåijĂåğŃçŽĎæŮűåĂŹéĂŽè£ĞäÿĂäžŽçőĂå■ŢçŽĎä¿Ńå■ŘåŠŃçŐľåĚůçÍŃåžŘäź§

éĂŽåÿÿæĹŚäžňåŕijåĚě NumPy æÍąåİŮçŽĎæŮűåĂŹäijŽä¡£çŤÍèŕ■åŘě import numpy as np ãĂĆ è£ŹæăůçŽĎèŕÍä¡ăåřśäÿ■çŤÍåĘ■ä¡ăçŽĎçÍŃåžŘéĞŇéİćäÿĂéĄ■éĄ■çŽĎæŢšåĚě numpy ïijŇåŘłéIJĂèęĄè¿ŞåĚě np åřśèąŇäžĘïijŇèŁĆçIJĄäžĘäÿ■åřŚæŮűéŮťãĂĆ

åęĆæđIJæČşèŐůåŘŮæŻť åď ŽçŽĎä£ąæĄŕïijŇä¡ăå¡ŞçĎűå¿ŮåŐż NumPy åőŸç¡ŚéĂŻéĂŻäžĘïijŇç¡ŚåİĂæŸŕïijŽ http://www.numpy.org

5.10 3.10 ç§ľéŸţäÿŐçž£æĂğäżčæŢřè£ŘçőŮ

éŮőéćŸ

ä¡ăéIJĂèęĄæL'ğèąŇç§ľéŸţåŠŇçž£æĂğäżčæŢřè£ŘçőŮïijŇæŕŤåęĆç§ľéŸţäźŸæşŢãĂĄåŕżæL'¿èąŇåĹŮå

èğčåĘşæŰźæąĹ

`NumPy` åžŞæIJL'äÿĂäÿłç§ľéŸţåŕźèśąåŘŕäżěçŤĺæİěèğčåĘşè£ŹäÿłéŮőéćŸãĂĆ

ç§ľéŸţçśżäijijäžŐ3.9åřŘèŁĆäÿ■æŢřçżĎåŕźèśąïijŇä¡ĘæŸŕéĄţå¿łçž£æĂğäżčæŢřçŽĎèőąçőŮèğĎåĹŹãĂ

```
>>> import numpy as np
>>> m = np.matrix([[1,-2,3],[0,4,5],[7,8,-9]])
>>> m
matrix([[ 1, -2, 3],
        [ 0, 4, 5],
        [ 7, 8, -9]])

>>> # Return transpose
>>> m.T
matrix([[ 1, 0, 7],
        [-2, 4, 8],
        [ 3, 5, -9]])

>>> # Return inverse
>>> m.I
matrix([[ 0.33043478, -0.02608696, 0.09565217],
        [-0.15217391, 0.13043478, 0.02173913],
        [ 0.12173913, 0.09565217, -0.0173913 ]])

>>> # Create a vector and multiply
>>> v = np.matrix([[2],[3],[4]])
>>> v
matrix([[2],
        [3],
        [4]])
>>> m * v
matrix([[ 8],
        [32],
        [ 2]])
>>>
```

åŘŕäżěåIJÍ `numpy.linalg` å■ŘåŇĚäÿ■æL'¿åĹřæŻť'åď'ŽçŽĎæŞ■ä¡IJåĞ¡æŢřïijŇæŕŤåęĆïijŽ

```
>>> import numpy.linalg

>>> # Determinant
>>> numpy.linalg.det(m)
```

(continues on next page)

(continued from previous page)

```
-229.99999999999983

>>> # Eigenvalues
>>> numpy.linalg.eigvals(m)
array([-13.11474312, 2.75956154, 6.35518158])

>>> # Solve for x in mx = v
>>> x = numpy.linalg.solve(m, v)
>>> x
matrix([[ 0.96521739],
        [ 0.17391304],
        [ 0.46086957]])
>>> m * x
matrix([[ 2.],
        [ 3.],
        [ 4.]])
>>> v
matrix([[2],
        [3],
        [4]])
>>>
```

èőÍèőž

å¿ÍæŸ¿çĎűçž£æĂğäžčæŢřæŸŕäÿłéÍđåÿÿåď ğçŽĎäÿžéćŸïijŇåůšçżŘèűĚåĞžäžĘæIJňäźęèČ¡èőÍèőžçŽĎè
ä¡ĘæŸŕïijŇåęĆæđIJä¡äéIJĂèęĄæŞ■ä¡IJæŢřçżĎåŠŇåŘŚéĞŘçŽĎèŕÍïijŇ NumPy
æŸŕäÿĂäÿłäÿ■éŤŹçŽĎåĚěåŘčçĆźãĂĆ åŘŕäžěèő£éŮő NumPy åőŸç¡Ś http://www.numpy.org
èŐůåŘŰæŻť åď Žä£ąæĄŕãĂĆ

5.11 3.11 éŽŘæIJžéĂĽæŃĽ

éŮőéćŸ

ä¡äæČşäżŐäÿĂäÿłäžŘåĹŮäÿ■éŽŘæIJžæŁ¡åŘŰèŃěäźšåĚČçť äïijŇæĹŰèĂĚæČşçŤ§æĹŘåĞääÿłéŽŘæIJ

èğčåĘşæŰźæąĹ

random æÍąåÍŮæIJĽ åď ğéĞŘçŽĎåĞ¡æŢřçŤÍæÍěäžğçŤ§éŽŘæIJžæŢřåŠŇéŽŘæIJžéĂĽæŃĽ åĚČçť ääĂĆ
æŕŤåęĆïijŇèęĄæČşäżŐäÿĂäÿłäžŘåĹŮäÿ■éŽŘæIJžçŽĎæŁ¡åŘŰäÿĂäÿłåĚČçť äïijŇåŘŕäžěä¡£çŤÍ
random.choice() ïijŽ

```
>>> import random
>>> values = [1, 2, 3, 4, 5, 6]
>>> random.choice(values)
```

(continues on next page)

(continued from previous page)

```
2
>>> random.choice(values)
3
>>> random.choice(values)
1
>>> random.choice(values)
4
>>> random.choice(values)
6
>>>
```

äÿžäžĘæŔŘåŔŰåĞžNäÿłäÿ■åŘŇåĚČçť ăçŽĎæăůæIJňçŤĺæİěåĄŽè£ŻäÿĂæ■ěçŽĎæŞ■ä¡IJïijŇåŔŕäżěä¡£ `random.sample()` ïijŽ

```
>>> random.sample(values, 2)
[6, 2]
>>> random.sample(values, 2)
[4, 3]
>>> random.sample(values, 3)
[4, 3, 1]
>>> random.sample(values, 3)
[5, 4, 1]
>>>
```

åęĆæđIJä¡ääżĚäżĚåŔłæŸŕæČşæL'Şäźśäžŕ̊åĹŮäÿ■åĚČçť ăçŽĎéąžäžŔïijŇåŔŕäżěä¡£çŤĺ `random.shuffle()` ïijŽ

```
>>> random.shuffle(values)
>>> values
[2, 4, 6, 5, 3, 1]
>>> random.shuffle(values)
>>> values
[3, 5, 2, 1, 6, 4]
>>>
```

çŤ§æĹŘéŽŔæIJžæŢť æŢřïijŇèŕůä¡£çŤĺ `random.randint()` ïijŽ

```
>>> random.randint(0,10)
2
>>> random.randint(0,10)
5
>>> random.randint(0,10)
0
>>> random.randint(0,10)
7
>>> random.randint(0,10)
10
>>> random.randint(0,10)
3
>>>
```

äÿžäžĘçŤ§æĹŘ0åĹř1èŇČåŻťåĘĚåİĞåŇĂåĹĘåÿČçŽĎæţőçĆźæŢřïijŇä¡£çŤÍ `random.random()` ïijŽ

```
>>> random.random()
0.9406677561675867
>>> random.random()
0.133129581343897
>>> random.random()
0.4144991136919316
>>>
```

åęĆæđIJèęĄèŐůåŘŰNä¡■éŽŘæIJžä¡■(äžŇè£ŻåĹű)çŽĎæŢťæŢřïijŇä¡£çŤÍ `random.getrandbits()` ïijŽ

```
>>> random.getrandbits(200)
335837000776573622800628485064121869519521710558559406913275
>>>
```

èőÍèőž

`random` æĺąåİŮä¡£çŤÍ *Mersenne Twister* çőŮæşŢæİěèőąçőŮçŤ§æĹŘéŽŘæIJžæŢřãĂĆè£ŹæŸŕäÿĂäÿłçä¡ĘæŸŕä¡ääŘŕäżěéĂŽè£Ğ `random.seed()` åĞ¡æŢřä£őæŤźåĹİåğŇåŇŰçğ■å■ŘãĂĆærŤåęĆïijŽ

```
random.seed() # Seed based on system time or os.urandom()
random.seed(12345) # Seed based on integer given
random.seed(b'bytedata') # Seed based on byte data
```

éŹďäžĘäÿŁè£řäżŃçż■çŽĎåŁ§èČ¡ïijŇrandomæĺąåİŮè£ŸåŇĚåŘńå§žäžŐåİĞåŇĂåĹĘåÿČãĂĄéńŸæŰŕåærŤåęĆïijŇ `random.uniform()` èőąçőŮåİĞåŇĂåĹĘåÿČéŽŘæIJžæŢřïijŇ `random.gauss()` èőąçőŮæ■čæĂĄåĹĘåÿČéŽŘæIJžæŢřãĂĆ åŕźäžŐåĚűäżŰçŽĎåĹĘåÿČæČĚåĘţèŕůåŘĆèĂČåIJÍçž£æŰĞæąčãĂĆ

åIJÍ `random` æĺąåİŮäÿ■çŽĎåĞ¡æŢřäÿ■åžŤèŕěçŤÍåIJÍåŠŇåŕEçăĄå■ęçŻÿåĚşçŽĎçÍŇåžŘäÿ■ãĂĆ åęĆæđIJä¡ăçąőåőđéIJĂèęĄçśżäijijçŽĎåŁ§èČ¡ïijŇåŘŕäżěä¡£çŤÍsslæĺąåİŮäÿ■çŻÿåžŤçŽĎåĞ¡æŢřãĂĆ ærŤåęĆïijŇ `ssl.RAND_bytes()` åŘŕäżěçŤÍæİěçŤ§æĹŘäÿĂäÿłåőL'åĚÍçŽĎéŽŘæIJžå■ŮèŁĆåžŘåĹŮãĂĆ

5.12 3.12 å§žæIJňçŽĎæŮěæIJ§äÿŐæŮűéŮťè¡ňæ■ć

éŮőécŸ

ä¡ăéIJĂèęĄæL'ğèąŇçőĂå■ŢçŽĎæŮűéŮťè¡ňæ■ćïijŇærŤåęĆåd'l'åĹřçğŠïijŇåřŘæŮűåĹřåĹĘéŠ§ç■L'çŽĎ

èğčåĘşæŰźæąĹ

äÿžäžĘæL'ğèąŇäÿ■åŘŇæŮűéŮťå■Ţä¡■çŽĎè¡ňæ■ćåŠŇèőąçőŮïijŇèŕůä¡£çŤÍ `datetime` æĺąåİŮãĂĆ ærŤåęĆïijŇäÿžäžĘèąĺçd'žäÿĂäÿłæŮűéŮťæőţïijŇåŘŕäżěåĹŻåżžäÿĂäÿł `timedelta` åőđä¿ŃïijŇåřśåČŘäÿŇéİćè£ŹæăůïijŽ

```
>>> from datetime import timedelta
>>> a = timedelta(days=2, hours=6)
>>> b = timedelta(hours=4.5)
>>> c = a + b
>>> c.days
2
>>> c.seconds
37800
>>> c.seconds / 3600
10.5
>>> c.total_seconds() / 3600
58.5
>>>
```

åęĆæđIJä¡ăæČşèąĺçď žæŇĞåőŽçŽĎæŮěæIJ§åŠŇæŮűéŮť ïijŇåĚĺåĹŻåżžäÿĂäÿł `datetime` åőđä¿ŃçĎűåŘŐä¡£çŤĺæăĞåĞĘçŽĎæŢřå■ęè£ŘçőŮæĺěæŞ■ä¡IJåőČäżňăĂĆæŕŤåęĆïijŽ

```
>>> from datetime import datetime
>>> a = datetime(2012, 9, 23)
>>> print(a + timedelta(days=10))
2012-10-03 00:00:00
>>>
>>> b = datetime(2012, 12, 21)
>>> d = b - a
>>> d.days
89
>>> now = datetime.today()
>>> print(now)
2012-12-21 14:54:43.094063
>>> print(now + timedelta(minutes=10))
2012-12-21 15:04:43.094063
>>>
```

åIJÍèőąçőŮçŽĎæŮűåĂŹïijŇéIJĂèęĄæşĺæĎŘçŽĎæŸŕ `datetime` äijŽèĞłåŁĺåď ĎçŘĘéŮřåźť ãĂĆæŕŤåęĆïijŽ

```
>>> a = datetime(2012, 3, 1)
>>> b = datetime(2012, 2, 28)
>>> a - b
datetime.timedelta(2)
>>> (a - b).days
2
>>> c = datetime(2013, 3, 1)
>>> d = datetime(2013, 2, 28)
>>> (c - d).days
1
>>>
```

èőÍèőž

åŕźåď ğåď ŽæŢřå§žæIJňçŽĎæŮěæIJ§åŠŇæŮűéŮť åď ĎçŘĘéŮőécŸïijŇ `datetime` æÍąåİŮåůšçżŔèűşåď §äžĘãĂĆ åęĆæđIJä¡äéIJĂèęĄæL'ğèąŇæŻť åŁååď■æİĆçŽĎæŮěæIJ§æŞ■ä¡IJïijŇæŕŤåęĆ åŘŕäżěèĂČèŹŚä¡£çŤÍ dateutilæÍąåİŮ

èőÿåď Žçśżäijijç ŽĎæŮűéŮť èőąçőŮåŘŕäżěä¡£çŤÍ `dateutil.relativedelta()` åĞ¡æŢřäżčæŻ£ãĂĆ ä¡ĘæŸŕïijŇæIJL'äÿĂçĆźéIJĂèęĄæşÍæĎŘçŽĎåŕśæŸŕïijŇåőČäijŽåIJÍåď ĎçŘĘæIJÍäż¡(è

```
>>> a = datetime(2012, 9, 23)
>>> a + timedelta(months=1)
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
TypeError: 'months' is an invalid keyword argument for this function
>>>
>>> from dateutil.relativedelta import relativedelta
>>> a + relativedelta(months=+1)
datetime.datetime(2012, 10, 23, 0, 0)
>>> a + relativedelta(months=+4)
datetime.datetime(2013, 1, 23, 0, 0)
>>>
>>> # Time between two dates
>>> b = datetime(2012, 12, 21)
>>> d = b - a
>>> d
datetime.timedelta(89)
>>> d = relativedelta(b, a)
>>> d
relativedelta(months=+2, days=+28)
>>> d.months
2
>>> d.days
28
>>>
```

5.13 3.13 èőąçőŮæIJĂåŘŐäÿĂäÿłåŚÍäžŤçŽĎæŮěæIJ§

éŮőécŸ

ä¡äéIJĂèęĄæ§ěæL'¿æŸ§æIJ§äÿ■æ§ŘäÿĂåď l'æIJĂåŘŐåĞžçŐŕçŽĎæŮěæIJ§ïijŇæŕŤåęĆæŸ§æIJ§äžŤã

èğčåĘşæŰźæąĹ

PythonçŽĎ `datetime` æÍąåİŮäÿ■æIJL'åůěåĚůåĞ¡æŢřåŠŇçśżåŘŕäżěåÿőåŁl'ä¡äæL'ğèąŇè£ŹæäůçŽĎèő äÿŇéIćæŸŕåŕźçśżäijijè£ŹæäůçŽĎéŮőécŸçŽĎäÿĂäÿłéĂŽçŤÍèğčåĘşæŰźæąĹïijŽ

```
#!/usr/bin/env python
# -*- encoding: utf-8 -*-
"""
Topic: æIJĂåŘŐçŽĎåŚÍäžŤ
Desc :
"""
from datetime import datetime, timedelta

weekdays = ['Monday', 'Tuesday', 'Wednesday', 'Thursday',
            'Friday', 'Saturday', 'Sunday']


def get_previous_byday(dayname, start_date=None):
    if start_date is None:
        start_date = datetime.today()
    day_num = start_date.weekday()
    day_num_target = weekdays.index(dayname)
    days_ago = (7 + day_num - day_num_target) % 7
    if days_ago == 0:
        days_ago = 7
    target_date = start_date - timedelta(days=days_ago)
    return target_date
```

åIJÍäžď'äžŠåijŘèğčéĞŁåŹÍäÿ■ä¡£çŤÍåęĆäÿŃïijŽ

```
>>> datetime.today() # For reference
datetime.datetime(2012, 8, 28, 22, 4, 30, 263076)
>>> get_previous_byday('Monday')
datetime.datetime(2012, 8, 27, 22, 3, 57, 29045)
>>> get_previous_byday('Tuesday') # Previous week, not today
datetime.datetime(2012, 8, 21, 22, 4, 12, 629771)
>>> get_previous_byday('Friday')
datetime.datetime(2012, 8, 24, 22, 5, 9, 911393)
>>>
```

åŘŕéĂL'çŽĎ `start_date` åŘĆæŢřåŘŕäžěçŤśåŘęåď'ŰäÿĂäÿł `datetime` åőđä¿ŃæĬěæŘŘä¿ŻãĂĆæŕŤåęĆïijŽ

```
>>> get_previous_byday('Sunday', datetime(2012, 12, 21))
datetime.datetime(2012, 12, 16, 0, 0)
>>>
```

èőÍèőž

äÿŁéĬćçŽĎçőŮæşŢåŐ§çŘĘæŸŕè£ŹæăůçŽĎïijŽåĚĹåřĘåijĂåğŃæŮěæIJ§åŠŇçŻőæăĞæŮěæIJ§æŸăåřĎ
çĎűåŘŐéĂŽè£ĞæÍąè£ŘçőŮèőąçőŮåĞžçŻőæăĞæŮěæIJ§èęAçżŘè£Ğåď'ŽåřŚåď'l'æL'■èČ¡åĹřè¿¿åijĂåğŃæŮ

åęĆæđIJä¡ăèęAąåČŘè£ŹæăůæL'ğèąŇåď'ğéĞŘçŽĎæŮěæIJ§èőąçőŮçŽĎèŕĬïijŇä¡ăæIJĂåě¡åőL'èčĚçňňäÿL
`python-dateutil` æĬěäžčæŻ£ãĂĆ æŕŤåęĆïijŇäÿŃéĬćæŸŕæŸŕä¡£çŤÍ `dateutil`

æÍąåİŮäÿ■çŽĎ `relativedelta()` åĞ¡æŢřæĽ'ğèąŇåŘŇæăůçŽĎèőąçőŮïijŽ

```
>>> from datetime import datetime
>>> from dateutil.relativedelta import relativedelta
>>> from dateutil.rrule import *
>>> d = datetime.now()
>>> print(d)
2012-12-23 16:31:52.718111

>>> # Next Friday
>>> print(d + relativedelta(weekday=FR))
2012-12-28 16:31:52.718111
>>>

>>> # Last Friday
>>> print(d + relativedelta(weekday=FR(-1)))
2012-12-21 16:31:52.718111
>>>
```

5.14 3.14 èőąçőŮå¡ŞåĽ■æIJĹäż¡çŽĎæŮĕæIJ§èŇČåŻť

éŮőéćŸ

ä¡ăçŽĎäżčçăĄéIJĂèęĄåIJĺå¡ŞåĽ■æIJĹäż¡äÿ■å¿łçŐŕæŕŘäÿĂåď'l'ïijŇæČşæĽ'¿åĹřäÿĂäÿłèőąçőŮè£Źäÿła

èğčåĘşæŰźæąĹ

åIJĺè£ŹæăůçŽĎæŮĕæIJ§äÿŁå¿łçŐŕåźűéIJĂèęĄäžŇåĚĹæđĎéĂääÿĂäÿłåŇĚåŘńæĽ'ĂæIJĽ'æŮĕæIJ§çŽĎ
ä¡ăåŘŕäżěåĚĹèőąçőŮåĞžåijĂåğŃæŮĕæIJ§åŠŇçżŞæİ§æŮĕæIJ§ïijŇ
çĎűåŘŐåIJĺä¡ăæ■ěè£ŻçŽĎæŮűåĂŹä¡£çŤÍ `datetime.timedelta`
åŕźèšąéĂŠåćđè£ŹäÿłæŮĕæIJ§åŘŸéĞŘå■şåŘŕãĂĆ

äÿŃéİćæŸŕäÿĂäÿłæŐĕåŘŮäżżæĎŘ `datetime` åŕźèšąåźűè£ŤåŻđäÿĂäÿłçŤšå¡ŞåĽ■æIJĹäż¡åijĂåğŃæŮ

```
from datetime import datetime, date, timedelta
import calendar

def get_month_range(start_date=None):
    if start_date is None:
        start_date = date.today().replace(day=1)
    _, days_in_month = calendar.monthrange(start_date.year, start_
↪date.month)
    end_date = start_date + timedelta(days=days_in_month)
    return (start_date, end_date)
```

æIJĽäžĘè£ŹäÿłåŕšåŘŕäżěå¿ĹåőźæŸ ŞçŽĎåIJĺè£ŤåŻđçŽĎæŮĕæIJ§èŇČåŻť äÿŁéİćåĄŽå¿łçŐŕæŞ■ä¡IJäż

```
>>> a_day = timedelta(days=1)
>>> first_day, last_day = get_month_range()
>>> while first_day < last_day:
...     print(first_day)
...     first_day += a_day
...
2012-08-01
2012-08-02
2012-08-03
2012-08-04
2012-08-05
2012-08-06
2012-08-07
2012-08-08
2012-08-09
#... and so on...
```

èőÍèőž

äÿŁéÍćçŽĎäżčçăĄåĚÍèőąçőŮåĞžäÿĂäÿłåŕźåžŤæIJÍäż¡çňňäÿĂåď'l'çŽĎæŮěæIJ§ãĂĆ äÿĂäÿłå£ńéĂ§çŽĎæŰźæşŢåřśæŸŕä¡£çŤÍ `date` æĹŰ `datetime` åŕźèśąçŽĎ `replace()` æŰźæşŢçőĂå■ŢçŽĎåřĘ `days` åśđæĂğèő¿ç¡őæĹŘ1å■şåŘŕãĂĆ `replace()` æŰźæşŢäÿĂäÿłåě¡åď'ĎåřśæŸŕåőČäijŽåĹŻåżžåŠŇä¡ăåijĂåğŃäijăåĚěåŕźèśąçśżåđŃçŻÿåŘŇçŽĎåŕźèśąãĂĆ æĽ'Ăäżěïij Ňåę Ćæđ IJè¿ŞåĚěåŘĆæŢřæŸŕäÿĂäÿł `date` åőđä¿ŃïijŇéĆčäźĹçżŞæđIJäź§æŸŕäÿĂäÿł `date` åőđä¿ŃãĂĆ åŘŇæăůçŽĎïijŇåęĆæđIJè¿ŞåĚěæŸŕäÿĂäÿł `datetime` åőđä¿ŃïijŇéĆčäźĹä¡ăå¿ŮåĹřçŽĎåřśæŸŕäÿĂäÿł `datetime` åőđä¿ŃãĂĆ

çĎűåŘŐïijŇä¡£çŤÍ `calendar.monthrange()` åĞ¡æŢřæİěæĽ'¿åĞžèŕěæIJĹçŽĎæĂżåď'l'æŢřãĂĆ äżżä¡ŢæŮűåĂŹåŘłèęĄä¡ăæČşèŐůå¿ŮæŮěåŐĘä£ąæĄŕïijŇéĆčäźĹ `calendar` æĺąåİŮåřśéİđåÿÿæIJĽçŤÍäžĘãĂĆ `monthrange()` åĞ¡æŢřäijŽè£ŤåŻđåŇĚåŘńæŸ§æIJ§åŠŇèŕěæIJĹåď'l'æŢřçŽĎåĚČçżĎãĂĆ

äÿĂæŮęèŕěæIJĹçŽĎåď'l'æŢřåůšç§ěäžĘïijŇéĆčäźĹçżŞæİ§æŮěæIJ§åřśåŘŕäżěéĂŽè£ĞåIJÍåijĂåğŃæŮěæ æIJĽäÿłéIJĂèęĄæşÍæĎŘçŽĎæŸŕçżŞæİ§æŮěæIJ§åźűäÿ■åŇĚåŘńåIJÍè£ŹäÿłæŮěæIJ§èŇČåŻť'åĘĚ(äžŃåőđäÿ è£ŹäÿłåŠŇPythonçŽĎ `slice` äÿŐ `range` æŞ■ä¡IJèąŇäÿžä£ÍæŇĄäÿĂèĞť'ïijŇåŘŇæăůäź§äÿ■åŇĚåŘńçżŞåŇ

äÿžäžĘåIJÍæŮěæIJ§èŇČåŻť'äÿŁå¿łçŐŕïijŇèęĄä¡£çŤÍåĹřæăĞåĞĘçŽĎæŢřå■ęåŠŇæŕŤè¿ĆæŞ■ä¡IJãĂĆ æŕŤåęĆïijŇåŘŕäżěåĹl'çŤÍ `timedelta` åőđä¿ŃæĺěéĂŠåćđæŮěæIJ§ïijŇåŕŘäžŐåŘů<çŤÍæĺěæčĂæ§ěäÿĂäÿłæ

çŘĘæČşæČĚåĘţäÿŃïijŇåęĆæđIJèČ¡äÿžæŮěæIJ§è£■äżčåĹŻåżžäÿĂäÿłåŘŇåĘĚç¡őçŽĎ `range()` åĞ¡æŢřäÿĂæăůçŽĎåĞ¡æŢřåŕśåě¡äžĘãĂĆ åźÿè£ŘçŽĎæŸŕïijŇåŘŕäżěä¡£çŤÍäÿĂäÿłçŤ§æĹŘåŹÍçŽĎæĺ

```
def date_range(start, stop, step):
    while start < stop:
        yield start
        start += step
```

äÿŃéĺćæŸŕä¡£çŤÍè£ŹäÿłçŤ§æĹŘåŹÍçŽĎä¿Ńå■ŘïijŽ

```
>>> for d in date_range(datetime(2012, 9, 1), datetime(2012,10,1),
                        timedelta(hours=6)):
...     print(d)
...
2012-09-01 00:00:00
2012-09-01 06:00:00
2012-09-01 12:00:00
2012-09-01 18:00:00
2012-09-02 00:00:00
2012-09-02 06:00:00
...
>>>
```

è£Źçğ■åđçŐřäźŃæL’Ăäžěè£ŹäźĹçőĂå■ŢïijŇè£Ÿå¿Ůå¡ŠåŁ§äžŐPythonäÿ■çŽĎæŮěæIJ§åŠŇæŮűéŮť

5.15 3.15 å■Ůçňęäÿšè¡ňæ■ćäÿžæŮěæIJ§

éŮőécŸ

ä¡ăçŽĎåžŤçŤÍçÍŃåžŘæŐěåŘŮå■ŮçňęäÿšæăijåijŘçŽĎè¿ŞåĚěïijŇä¡ĘæŸŕä¡ăæČşåřĘåőČäżňè¡ňæ■ćäÿž `datetime` åŕźèśąäżěä¿£åIJÍäÿŁéİćæL’ğèąŇéİđå■ŮçňęäÿšæŞ■ä¡IJãĂĆ

èğčåĘşæŰźæąĹ

ä¡£çŤÍPythonçŽĎæăĞåĞĘæÍąåİŮ `datetime` åŘŕäżěå¿ĹåőźæŸ ŞçŽĎèğčåĘşè£ŹäÿłéŮőécŸãĂĆæŕŤåęĆ

```
>>> from datetime import datetime
>>> text = '2012-09-20'
>>> y = datetime.strptime(text, '%Y-%m-%d')
>>> z = datetime.now()
>>> diff = z - y
>>> diff
datetime.timedelta(3, 77824, 177393)
>>>
```

èőÍèőž

`datetime.strptime()` æŰźæşŢæŤŕæŇĄå¿Ĺåď’ŽçŽĎæăijåijŘåŇŰäżčçăĄïijŇ æŕŤåęĆ `%Y` äżčèąĺ4ä¡■æŢřåźť’äż¡ïijŇ `%m` äżčèąĺäÿď’ä¡■æŢřæIJĹäż¡ãĂĆ è£ŸæIJL’äÿĂçĆźåĂijå¿ŮæşÍæĎŘçŽĎæŸŕè£ŹäžŻæăijåijŘåŇŰå■ăä¡■çňęäź§åŘŕäżěåŘ■è£ĞæÍěä¡£çŤÍïijŇåřĚ

æŕŤåęĆïijŇåĄĞèő¿ä¡ăçŽĎäżčçăĄäÿ■çŤ§æĹŘäžĘäÿĂäÿł `datetime` åŕźèśąïijŇ ä¡ăæČşåřĘåőČæăijåijŘåŇŰäÿžæijĆäžőæŸŞèŕżå¡ćåijŘåŘŐæŤ¿åIJÍèĞłåŁÍçŤ§æĹŘçŽĎä£ąäżűæĹŰèĂĚæŁěåŞ

```
>>> z
datetime.datetime(2012, 9, 23, 21, 37, 4, 177393)
>>> nice_z = datetime.strftime(z, '%A %B %d, %Y')
>>> nice_z
'Sunday September 23, 2012'
>>>
```

è£ŸæIJL'äÿĂçĆźéIJĂèęĄæşĺæĎŘçŽĎæŸŕïijŇ strptime() çŽĎæĂğèČ¡èęĄæŕŤä¡ăæČşèśąäÿ■çŽĎåůőå¿Ĺåď'ŽïijŇ åŻääÿžåőČæŸŕä¡£çŤĺçžŕPythonåőđçŐřïijŇåźűäÿŤå£Ì åęĆæđIJä¡ăèęĄåIJĺäžčçăĄäÿ■éIJĂèęĄèğčæđŘåď'ğéĞŘçŽĎæŮěæIJ§åźűäÿŤåůšçżŘç§ěéĄŞäžĘæŮěæIJ§å■Ůç æŕŤåęĆïijŇåęĆæđIJä¡ăåůšçżŘç§ěéĄŞæL'Ăäžěæ̊ŮěæIJ§æăijåijŘæŸŕ YYYY-MM-DD ïijŇä¡ăåŘŕäžěåČŘäÿŃéİćè£ŹæăůåőđçŐřäÿĂäÿłèğčæđŘåĞ¡æŢřïijŽ

```
from datetime import datetime
def parse_ymd(s):
    year_s, mon_s, day_s = s.split('-')
    return datetime(int(year_s), int(mon_s), int(day_s))
```

åőđéŹĚæţŃèŕŢäÿ■ïijŇè£ŹäÿłåĞ¡æŢřæŕŤ `datetime.strptime()` å£ń7åĂ■åď'ŽãĂĆ åęĆæđIJä¡ăèęĄåď'ĎçŘĘåď'ğéĞŘçŽĎæűL'åŘŁåĹřæŮěæIJ§çŽĎæŢřæ■őçŽĎèŕİïijŇéĆčäźĹæIJĂåě¡èĂČèŹŚäÿ

5.16 3.16 çżŞåŘĹæŮűåŇžçŽĎæŮěæIJ§æŞ■ä¡IJ

éŮőécŸ

ä¡ăæIJL'äÿĂäÿłåőL'æŐŠåIJĺ2012åźť'12æIJĹ21æŮěæŮľäÿŁ9:30çŽĎçŤţèŕİäijŽèőőïijŇåIJřçĆźåIJĺèŁİåŁĂ èĂŇä¡ăçŽĎæIJŃåŘŃåIJĺå■řåžęçŽĎçŘ■åŁăç¡ŮåřŤïijŇéĆčäźĹäžŰåžŤèŕěåIJĺå¡ŞåIJřæŮűéŮť'åĞăçĆźåŘĆåŁă

èğčåĘşæŰźæąĹ

åŕźåĞääźŐæL'ĂæIJL'æűL'åŘŁåĹřæŮűåŇžçŽĎéŮőécŸïijŇä¡äéČ¡åžŤèŕěä¡£çŤĺ pytz æĺąåİŮãĂĆè£ŹäÿłåŇĚæŘŘä¿ŻäžĘOlsonæŮűåŇžæŢřæ■őåžŞïijŇ åőČæŸŕæŮűåŇžä£ąæĄŕçŽĎäžŃåőđäÿŁçŽĎæăĞåĞĘïijŇåIJĺå¿Ĺåď'Žèŕ■èĺĂåŠŇæŞ■ä¡IJçşżçż§éĞŇéİćéČ¡åŘ

pytz æĺąåİŮäÿĂäÿłäÿżèęĄçŤĺéĂŤæŸŕåřĘ datetime åžŞåĹŻåżżçŽĎçőĂå■ŢæŮěæIJ§åŕźèśąæIJňåIJŕåŇŰãĂĆ æŕŤåęĆïijŇäÿŃéİćåęĆä¡Ţèąĺçď'žäÿĂäÿłèŁİåŁäåŞěa

```
>>> from datetime import datetime
>>> from pytz import timezone
>>> d = datetime(2012, 12, 21, 9, 30, 0)
>>> print(d)
2012-12-21 09:30:00
>>>

>>> # Localize the date for Chicago
>>> central = timezone('US/Central')
>>> loc_d = central.localize(d)
```

(continues on next page)

(continued from previous page)

```
>>> print(loc_d)
2012-12-21 09:30:00-06:00
>>>
```

äÿĂæŮęæŮĕæIJ§èćńæIJňåIJřåŇŰäžĘïijŇ åőČåřśåŘŕäżĕè¡ňæ■ćäÿžåĚűäżŰæŮűåŇžçŽĎæŮűéŮť'äžĘãĂ
äÿžäžĘå¿ŮåĹřçŘ■åŁăç¡ŮåřŤåŕźåžŤçŽĎæŮűéŮť'ïijŇä¡ääŘŕäżĕè£ŹæăůåĄŽïijŽ

```
>>> # Convert to Bangalore time
>>> bang_d = loc_d.astimezone(timezone('Asia/Kolkata'))
>>> print(bang_d)
2012-12-21 21:00:00+05:30
>>>
```

åęĆæđIJä¡ăæL'ŞçőŮåIJÍæIJňåIJřåŇŰæŮĕæIJ§äÿŁæL'ğèąŇèőąçőŮïijŇä¡ăéIJĂèęĄçL'źåĹńæşÍæĎŘåď'Ŕä
æŕŤåęĆïijŇåIJÍ2013åźť'ïijŇç¿ŐåŻ¡æăĞåĞĘåď'Ŕäżď'æŮűæŮűéŮť'åijĂåğŃäžŐæIJňåIJřæŮűéŮť'3æIJĹ13æŮĕ
åęĆæđIJä¡ăæ■čåIJÍæL'ğèąŇæIJňåIJřèőąçőŮïijŇä¡ääijŽå¿ŮåĹřäÿĂäÿłéŤŹèŕŕãĂĆæŕŤåęĆïijŽ

```
>>> d = datetime(2013, 3, 10, 1, 45)
>>> loc_d = central.localize(d)
>>> print(loc_d)
2013-03-10 01:45:00-06:00
>>> later = loc_d + timedelta(minutes=30)
>>> print(later)
2013-03-10 02:15:00-06:00 # WRONG! WRONG!
>>>
```

çżŞæđIJéŤŹèŕŕæŸŕåŻääÿžåőČåźűæšąæIJL'èĂČèŹŚåIJÍæIJňåIJřæŮűéŮť'äÿ■æIJL'äÿĂåřŘæŮűçŽĎèůşèů
äÿžäžĘä£őæ■čè£ŹäÿłéŤŹèŕŕïijŇåŘŕäżĕä¡£çŤÍæŮűåŇžåŕźèšą `normalize()`
æŰźæşŢãĂĆæŕŤåęĆïijŽ

```
>>> from datetime import timedelta
>>> later = central.normalize(loc_d + timedelta(minutes=30))
>>> print(later)
2013-03-10 03:15:00-05:00
>>>
```

èőÍèőž

äÿžäžĘäÿ■èőľä¡äèćńè£ŹäžŻäÿIJäÿIJåijĎçŽĎæŹŢåď'ť'è¡ňåŘŚïijŇåď'ĎçŘĘæIJňåIJřåŇŰæŮĕæIJ§çŽĎéĂ
åźűçŤÍåőČæĬĕæL'ğèąŇæL'ĂæIJL'çŽĎäÿ■éŮť'å■ŸåĆÍåŠŇæŞ■ä¡IJãĂĆæŕŤåęĆïijŽ

```
>>> print(loc_d)
2013-03-10 01:45:00-06:00
>>> utc_d = loc_d.astimezone(pytz.utc)
>>> print(utc_d)
2013-03-10 07:45:00+00:00
>>>
```

äÿĂæŮęè¡ňæ■ćäÿžUTCïijŇä¡ăăřśäÿ■çŤĺăŐżæŃĚă£Čèů§ăď´Ŕäžď´æŮűçŻÿăĚşçŽĎéŮőéćŸäžĘãĂĆ
ăŻăæ■ď´ïijŇä¡ăăŘŕäžěèů§äźŃăĽ■äÿĂæăůæŤ¿ă£ČçŽĎæĽ´ğèąŇăÿÿèğĄçŽĎæŮěæIJ§èőąçőŮãĂĆ
ă¡Şä¡ăæČşăřĘè¿ŞăĞžăŘŸäÿžæIJňăIJřæŮűéŮť´çŽĎæŮűăĂŹïijŇä¡£çŤĺăŘĹéĂĆçŽĎæŮűăŇžăŐżè¡ňæ■ćäÿŃă

```
>>> later_utc = utc_d + timedelta(minutes=30)
>>> print(later_utc.astimezone(central))
2013-03-10 03:15:00-05:00
>>>
```

ă¡ŞæűĽ´ăŘŁăĹřæŮűăŇžæŞ■ä¡IJçŽĎæŮűăĂŹïijŇæIJĽ´äÿłéŮőéćŸăřśæŸŕæĹŚäžňăęĆä¡Ţă¿ŮăĹřæŮűăŇ
æŕŤăęĆïijŇăIJĺè£Źäÿłä¿Ňă■Řäÿ■ïijŇæĹŚäžňăęĆä¡Ţç§ěęĄŞăĂIJAsia/KolkataăĂİăřśæŸŕă■řăžęăřźăžŤçŽĎæ
äÿžäžĘæ§ěæĽ´¿ïijŇăŘŕäžěä¡£çŤĺISO 3166ăŻ¡ăőűäžčçăĄä¡IJäÿžăĚşéŤőă■ŮăŐżæ§ěéŸĚă■ŮăĚÿ
`pytz.country_timezones` ãĂĆæŕŤăęĆïijŽ

```
>>> pytz.country_timezones['IN']
['Asia/Kolkata']
>>>
```

æşĺïijŽă¡Şä¡äéŸĚèŕżăĹřè£ŹéĞŇçŽĎæŮűăĂŹïijŇæIJĽ´ăŘŕèČ¡ `pytz`
æĺąăİŮăůšçżŘäÿ■ăĘ■ăżżèőőä¡£çŤĺäžĘïijŇăŻääÿžPEP431æŘŘăĞžäžĘæŻť´ăĚĹè£ŻçŽĎæŮűăŇžæŤŕæŇĄãĂĆ
ä¡ĘæŸŕè£ŹéĞŇèřĹăĹřçŽĎă¿Ĺăď´ŽéŮőéćŸè£ŸæŸŕæIJĽ´ăŘĆèĂČäžůăĂijçŽĎ(æŕŤăęĆä¡£çŤĺUTCæŮěæIJ§ç

6 çňňăŻŻçńäïijŽè£■äžčăŹĺäÿŐçŤ§æĹŘăŹĺ

è£■äžčæŸŕPythonæIJĂăijžăď´ğçŽĎăŁ§èČ¡äźŃäÿĂãĂĆăĹĺçIJŃèţůæİěïijŇä¡ăăŘŕèČ¡äijŽçőĂă■ŢçŽĎèőć
çĎűèĂŇïijŇçżIéIđäžĚäžĚăřśæŸŕăęĆæ■ď´ïijŇè£ŸæIJĽ´ă¿Ĺăď´Žä¡ăăŘŕèČ¡äÿ■ç§ěęĄŞçŽĎïijŇ
æŕŤăęĆăĹŻăžžä¡äèĞłăůšçŽĎè£■äžčăŹĺăŕźèśąïijŇăIJĺitertoolsæĺąăİŮäÿ■ä¡£çŤĺæIJĽ´çŤĺçŽĎè£■äžčæĺąăijŘïij
è£ŹäÿĂçńäçŽőçŽĎăřśæŸŕăŘŚä¡ăăśŢçď´žèů§è£■äžčæIJĽ´ăĚşçŽĎăŘĎçğ■ăÿÿèğĄéŮőéćŸãĂĆ

Contents:

6.1 4.1 æĽŃăŁĺéĄ■ăŐĘè£■äžčăŹĺ

éŮőéćŸ

ä¡äæČşéĄ■ăŐĘäÿĂäÿłăŘŕèł■äžčăŕźèśąäÿ■çŽĎæĽ´ĂæIJĽ´ăĚČçť´äïijŇä¡ĘæŸŕă■ť´äÿ■æČşä¡£çŤĺforă¿łçČ

èğčăĘşæŰźæąĹ

äÿžäžĘæĽ´ŃăŁĺçŽĎéĄ■ăŐĘăŘŕèł■äžčăŕźèśąïijŇä¡£çŤĺ `next()`
ăĞ¡æŢřăźűăIJĺäžčçăĄäÿ■æ■ŢèŐů `StopIteration` ăijĆăÿÿăĂĆ
æŕŤăęĆïijŇäÿŃéĺćçŽĎä¿Ňă■ŘæĽ´ŃăŁĺèŕżăŘŰäÿĂäÿłæŰĞäžűäÿ■çŽĎæĽ´ĂæIJĽ´èąŇïijŽ

```
def manual_iter():
    with open('/etc/passwd') as f:
```

(continues on next page)

(continued from previous page)

```
        try:
            while True:
                line = next(f)
                print(line, end='')
        except StopIteration:
            pass
```

éĂŽåÿÿæÌěèőšïijŇ StopIteration çŤÍæÌěæŇĞçď žè£■äžčçŽĎçżŞář¿ãĂĆ çĎűèĂŇïijŇåęĆæđIJä¡ăæĽŃåŁÍä¡£çŤÍäÿŁéÌćæijŤçď žçŽĎ next() åĞ¡æŢřçŽĎèŕÌïijŇä¡ăè£ŸåŘŕäžěéĂŽè£ĞÈ£ŤåŽđäÿĂäÿłæŇĞåőŽåĂijæÌěæăĞèőřçżŞář¿ïijŇæŕŤåęĆ None ãĂĆ äÿŃéÌćæŸŕçď žä¿ŃïijŽ

```
with open('/etc/passwd') as f:
    while True:
        line = next(f, None)
        if line is None:
            break
        print(line, end='')
```

èőÍèőž

åď ğåď ŽæŢřæČĚåĘţäÿŃïijŇæĹŚäžňäijŽä¡£çŤÍ for å¿łçŐŕèŕ■åŘěçŤÍæÌěéĄ■åŐĘäÿĂäÿłåŘŕè£■äžčåŕźè
ä¡ĘæŸŕïijŇåĄűåřŤäź§éIJĂèęĄåŕźè£■äžčåĄŽæŽť åŁăçš¿çąőçŽĎæŐğåĹűïijŇè£ŹæŮűåĂŹäžĘèğčåžŢåśĆè£■ä

äÿŃéÌćçŽĎäžď äžŠçď žä¿ŃåŘŚæĹŚäžňæijŤçď žäžĘè£■äžčæIJ§éŮť æĽĂåŘŚçŤ§çŽĎå§žæIJňçżĘèŁĆïij

```
>>> items = [1, 2, 3]
>>> # Get the iterator
>>> it = iter(items) # Invokes items.__iter__()
>>> # Run the iterator
>>> next(it) # Invokes it.__next__()
1
>>> next(it)
2
>>> next(it)
3
>>> next(it)
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
StopIteration
>>>
```

æIJňçńäæŐěäÿŃæÌěåĞäåřŘèŁĆäijŽæŽť æůśåĚěçŽĎèőšèğčè£■äžčçŻÿåĚşæŁĂæIJŕïijŇåĽ■æŘŘæŸŕä¡ăä
æĽĂäžěçąőä£Ìä¡ăåůšçżŘæŁŁè£ŹçńäçŽĎåĘĚåőźçĽćçĽćèőřåIJÍå£Čäÿ■ãĂĆ

6.2 4.2 äžčçŘĘè£■äžč

éŮőécŸ

ä¡ăæđĎåžžäžĘäÿĂäÿłèĞłåőŽäźL’åőźåŹĺåŕźèśąïijŇéĞŇéİćåŇĚåŘńæIJL’åĹŮèąĺãĂĄåĚČçżĎæĹŰåĚűäžŰ
ä¡ăæČşçŻť’æŐěåIJĺä¡äçŽĎè£ŹäÿłæŰřåőźåŹĺåŕźèśąäÿŁæL’ğèąŇè£■äžčæŞ■ä¡IJãĂĆ

èğčåĘşæŰźæąĹ

åőđéŹĚäÿŁä¡ääŘłéIJĂèęĄåőŽäźL’äÿĂäÿł `__iter__()`
æŰźæşŢïijŇåřĘè£■äžčæŞ■ä¡IJäžčçŘĘåĹřåőźåŹĺåĘĚéČĺçŽĎåŕźèśąäÿŁåŐżãĂĆærŤåęĆïijŽ

```
class Node:
    def __init__(self, value):
        self._value = value
        self._children = []

    def __repr__(self):
        return 'Node({!r})'.format(self._value)

    def add_child(self, node):
        self._children.append(node)

    def __iter__(self):
        return iter(self._children)

# Example
if __name__ == '__main__':
    root = Node(0)
    child1 = Node(1)
    child2 = Node(2)
    root.add_child(child1)
    root.add_child(child2)
    # Outputs Node(1), Node(2)
    for ch in root:
        print(ch)
```

åIJĺäÿŁéİćäžčçăĄäÿ■ïijŇ `__iter__()` æŰźæşŢåŘłæŸŕçőĂå■ŢçŽĎåřĘè£■äžčèŕůæśĆäijäéĂŠçżŹåĘĚé
`_children` åśđæĂğãĂĆ

èőĺèőž

PythonçŽĎè£■äžčåŹĺå■Řèőőéİ JĂèęĄ `__iter__()` æŰźæşŢè£ŤåŻđäÿĂäÿłåőđçŐřäžĘ
`__next__()` æŰźæşŢçŽĎè£■äžčåŹĺåŕźèśąãĂĆ åęĆæđIJä¡ääŘłæŸŕè£■äžčéĄ■åŐĘåĚűäžŰåőźåŹĺçŽĎåĘĚå

è£ŹéĞŇçŽĎ `iter()` åĞ¡æŢřçŽĎä¡£çŤĺçőĂåŇŰäžĘäžčçăĄïijŇ
`iter(s)` åŘłæŸŕçőĂå■ŢçŽĎéĂŽè£ĞèřĆçŤĺ `s.__iter__()`

æŰźæşŢæİěè£ŤåŻđåŕźåžŤçŽĎè£■äžčåŹĺåŕźèšąïijŇ åřśèů§ `len(s)` äijŽèřČçŤÍ `s.__len__()` åŐ§çŘĘæŸŕäÿĂæăůçŽĎãĂĆ

6.3 4.3 ä¡£çŤÍçŤ§æĹŘåŹĺåĹŻåžžæŰřçŽĎè£■äžčæÍąåijŔ

éŮőéćŸ

ä¡ăæČşåőđçŐřäÿĂäÿłèĞłåőŽäźL'è£■äžčæÍąåijŔïijŇèů§æŹőéĂŽçŽĎåĘĚç¡őåĞ¡æŢřærŤåęĆ `range()`，`reversed()` äÿ■äÿĂæăůãĂĆ

èğčåĘşæŰźæąĹ

åęĆæđIJä¡ăæČşåőđçŐřäÿĂçğ■æŰřçŽĎè£■äžčæÍąåijŔïijŇä¡£çŤÍäÿĂäÿłçŤ§æĹŘåŹĺåĞ¡æŢřæİěåőŽäźL'å äÿŃéIćæŸŕäÿĂäÿłçŤ§äžğæ§ŘäÿłèŇČåŽť'åĘĚæţőçĆźæŢřçŽĎçŤ§æĹŘåŹĺïijŽ

```
def frange(start, stop, increment):
    x = start
    while x < stop:
        yield x
        x += increment
```

äÿžäžĘä¡£çŤÍè£ŹäÿłåĞ¡æŢřïijŇ ä¡ăåŔŕäžěçŤÍforå¿łçŐŕè£■äžčåőČæĹŰèĂĚä¡£çŤÍåĚűäžŰæŐěåŔŮäÿĂä `sum()`，`list()` ç■L')ãĂĆçd'žä¿ŃåęĆäÿŃïijŽ

```
>>> for n in frange(0, 4, 0.5):
...     print(n)
...
0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
>>> list(frange(0, 1, 0.125))
[0, 0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875]
>>>
```

èőÍèőž

äÿĂäÿłåĞ¡æŢřäÿ■éIJĂèęĄæIJL'äÿĂäÿł `yield` èŕ■åŘěå■şåŔŕåřĘåĚűè¡ňæ■ćäÿžäÿĂäÿłçŤ§æĹŘåŹĺãĂĆ èů§æŹőéĂŽåĞ¡æŢřäÿ■åŘŃçŽĎæŸŕïijŇçŤ§æĹŘåŹĺåŘłèČ¡çŤÍäžŐè£■äžčæŞ■ä¡IJãĂĆ äÿŃéIćæŸŕäÿĂäÿłåőđéłŇïijŇåŘŚä¡ăåśŢçd'žè£ŹæăůçŽĎåĞ¡æŢřåžŢåśĆåůëä¡IJæIJžåĹűïijŽ

```
>>> def countdown(n):
...     print('Starting to count from', n)
...     while n > 0:
...         yield n
...         n -= 1
...     print('Done!')
...

>>> # Create the generator, notice no output appears
>>> c = countdown(3)
>>> c
<generator object countdown at 0x1006a0af0>

>>> # Run to first yield and emit a value
>>> next(c)
Starting to count from 3
3

>>> # Run to the next yield
>>> next(c)
2

>>> # Run to next yield
>>> next(c)
1

>>> # Run to next yield (iteration stops)
>>> next(c)
Done!
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
StopIteration
>>>
```

äÿĂäÿłçŤ§æĹŘåŹĺåĞ¡æŢřäÿżèęAçL’źå¿ĄæŸŕåőČåŘłäijŽåŻđåžŤåIJĺè£■äżčäÿ■ä¡£çŤĺåĹřçŽĎ *next* æŞ■ä¡IJãĂĆ äÿĂæŮęçŤ§æĹŘåŹĺåĞ¡æŢřè£ŤåŻđéĂĂåĞžïijŇè£■äżčçżĹæ■ćãĂĆæĹŚäżňåIJĺè£■äżčäÿ■é

6.4 4.4 åőđçŐřè£■äżčåŹĺå■Ŕèőő

éŮőécŸ

ä¡ăæČşæđĎåżžäÿĂäÿłèČ¡æŤŕæŇĄè£■äżčæŞ■ä¡IJçŽĎèĞłåőŽäźL’åŕźèśąïijŇåźűåÿŇæIJŻæL’¿åĹřäÿĂäÿł

èğčåĘşæŰźæąĹ

çŻőåL’■äÿžæ■ćïijŇåIJĺäÿĂäÿłåŕźèśąäÿŁåőđçŐřè£■äżčæIJĂçőĂå■ŢçŽĎæŰźåijŔæŸŕä¡£çŤĺäÿĂäÿłçŤ§æ
åIJĺ4.2åřŔèŁĆäÿ■ïijŇä¡£çŤĺNodeçśżæİěèąĺçď’žæăŚå¡ćæŢřæ■őçżŞæđĎãĂĆä¡ăåŔŕèČ¡æČşåőđçŐřäÿĂäÿłäżěa

äÿŃéİćæŸŕäžčçăĄçď žä¿ŃïijŽ

```
class Node:
    def __init__(self, value):
        self._value = value
        self._children = []

    def __repr__(self):
        return 'Node({!r})'.format(self._value)

    def add_child(self, node):
        self._children.append(node)

    def __iter__(self):
        return iter(self._children)

    def depth_first(self):
        yield self
        for c in self:
            yield from c.depth_first()

# Example
if __name__ == '__main__':
    root = Node(0)
    child1 = Node(1)
    child2 = Node(2)
    root.add_child(child1)
    root.add_child(child2)
    child1.add_child(Node(3))
    child1.add_child(Node(4))
    child2.add_child(Node(5))

    for ch in root.depth_first():
        print(ch)
    # Outputs Node(0), Node(1), Node(3), Node(4), Node(2), Node(5)
```

åIJÍè£ŹæőţäžčçăĄäÿ■ïijŃdepth_first() æŰźæşŢçőÄå■ŢçŻť'èğĆãĂĆ åőČéęŰåĚĹè£ŤåŻđèĞłåůśæIJňèžňåźűè£■äžčæŕŘäÿĂäÿłå■ŘèŁĆçĆźåźű éĂŽè£ĞèřČçŤĺå■ŘèŁĆçĆźçŽĎ depth_first() æŰźæşŢ(ä¡£çŤĺ yield from èŕ■åŘě)è£ŤåŻđåŕźåžŤåĚČçť'äãĂĆ

èőĺèőž

PythonçŽĎè£■äžčå■ŘèőőèęĄæśĆäÿĂäÿł __iter__() æŰźæşŢè£ŤåŻđäÿĂäÿłçL'źæőŁçŽĎè£■äžčåŹĺåŕźèśąïijŃ è£Źäÿłè£■äžčåŹĺåŕźèśąåőđçŐřäžĘ __next__() æŰźæşŢåźűéĂŽè£Ğ StopIteration åijĆåÿÿæăĞèŕĘè£■äžčçŽĎåőŃæĹŘãĂĆ ä¡ĘæŸŕïijŃåőđçŐřè£ŹäžŻéĂŽåÿÿäijŽæŕŤè¿ČçźĄçŘŘãĂĆ äÿŃéİćæĹŚäžňæijŤçď žäÿŃè£Źçğ■æŰźåijŘïijŃå depth_first() æŰźæşŢïijŽ

```
class Node2:
    def __init__(self, value):
        self._value = value
        self._children = []

    def __repr__(self):
        return 'Node({!r})'.format(self._value)

    def add_child(self, node):
        self._children.append(node)

    def __iter__(self):
        return iter(self._children)

    def depth_first(self):
        return DepthFirstIterator(self)


class DepthFirstIterator(object):
    '''
    Depth-first traversal
    '''

    def __init__(self, start_node):
        self._node = start_node
        self._children_iter = None
        self._child_iter = None

    def __iter__(self):
        return self

    def __next__(self):
        # Return myself if just started; create an iterator for␣
↪children
        if self._children_iter is None:
            self._children_iter = iter(self._node)
            return self._node
        # If processing a child, return its next item
        elif self._child_iter:
            try:
                nextchild = next(self._child_iter)
                return nextchild
            except StopIteration:
                self._child_iter = None
                return next(self)
        # Advance to the next child and start its iteration
        else:
            self._child_iter = next(self._children_iter).depth_
↪first()
            return next(self)
```

`DepthFirstIterator` çśżåŠŇäÿŁéİćä¡£çŤÍçŤ§æĹŘåŹÍçŽĎçL'ĹæIJňåůëä¡IJåŐ§çŘĘçśżäijijïijŇ
ä¡ĘæŸŕåőČåĘŹèţůæĺěå¿ĹçźAçŘŘïijŇåŽääÿžè£■äżčåŹÍå£ĚéążåIJÍè£■äżčåď'ĎçŘĘè£ĞçÍŃäÿ■çżť'æŁď'åď'ğéČ
åİęçŹ¡æİěèőšïijŇæšąäžžæĎ£æĎŘåĘŹè£ŹäźĹæŹęæűľçŽĎäżčçăĄãĂĆåřĘä¡ăçŽĎè£■äżčåŹÍåőŽäźL'äÿžäÿĂäÿ

6.5 4.5 åŘ■åŘŚè£■äżč

éŮőécŸ

ä¡äæČşåŘ■æŰźåŘŚè£■äżčäÿĂäÿłäžŘåĹŮ

èğčåĘşæŰźæąĹ

ä¡£çŤÍåĘĚç¡őçŽĎ `reversed()` åĞ¡æŢřïijŇæŕŤåęĆïijŽ

```
>>> a = [1, 2, 3, 4]
>>> for x in reversed(a):
...     print(x)
...
4
3
2
1
```

åŘ■åŘŚè£■äżčäżĚäżĚå¡ŞåŕźèśąçŽĎåď'ğåřŘåŘŕécĎåĚĹçąőåőŽæĹŰèĂĚåŕźèśąåőđçŐřäžĘ
`__reversed__()` çŽĎçL'źæőŁæŰźæşŢæŮűæL'■èČ¡çŤ§æŢĹãĂĆ
åęĆæđIJäÿd'èĂĚéČ¡äÿ■çňęåŘĹïijŇéĆčä¡åå£ĚéążåĚĹåřĘåŕźèśąè¡ňæ■ćäÿžäÿĂäÿłåĹŮèąĺæL'■èąŇïijŇæŕŤåęĆ

```
# Print a file backwards
f = open('somefile')
for line in reversed(list(f)):
    print(line, end='')
```

èęĄæşĺæĎŘçŽĎæŸŕåęĆæđIJåŘŕè£■äżčåŕźèśąåĚČçť'åå¿Ĺåď'ŽçŽĎèŕÍïijŇåřĘåĚűécĎåĚĹè¡ňæ■ćäÿžäÿĂ

èőÍèőž

å¿Ĺåď'ŽçÍŃåžŘåŚŸåźűäÿ■ç§ěéĄŞåŘŕäżěéĂŽè£ĞåIJÍèĞłåőŽäźL'çśżäÿŁåőđçŐř
`__reversed__()` æŰźæşŢæÍěåőđçŐřåŘ■åŘŚè£■äżčãĂĆæŕŤåęĆïijŽ

```
class Countdown:
    def __init__(self, start):
        self.start = start

    # Forward iterator
    def __iter__(self):
        n = self.start
```

(continues on next page)

(continued from previous page)

```
        while n > 0:
            yield n
            n -= 1

    # Reverse iterator
    def __reversed__(self):
        n = 1
        while n <= self.start:
            yield n
            n += 1

for rr in reversed(Countdown(30)):
    print(rr)
for rr in Countdown(30):
    print(rr)
```

åőŽäźL'äÿĂäÿłåŔ■åŘŚè£■äżčåŹĺåŔŕäżěä¡£å¿ŮäżčçăĄéİđåÿçŽĎéńŸæŢĹïijŇ
åŻääÿžåőČäÿ■åĘ■éIJĂèęĄåřĘæŢřæ■őåąńåĚĚåĹřäÿĂäÿłåĹŮèąĺäÿ■çĎűåŘŐåĘ■åŐżåŔ■åŘŚè£■äżčè£ŹäÿłåĹ

6.6 4.6 åÿęæIJL'åď'ŰéČĺçŁűæĂĄçŽĎçŤ§æĹŘåŹĺåĞ¡æŢř

éŮőécŸ

ä¡äæČşåőŽäźL'äÿĂäÿłçŤ§æĹŘåŹĺåĞ¡æŢřïijŇä¡ĘæŸŕåőČäijŽèřČçŤĺæ§Řäÿłä¡äæČşæŽť'éIJšçżŹçŤĺæĹůä

èğčåĘşæŰźæąĹ

åęĆæđIJä¡äæČşèőľä¡äçŽĎçŤ§æĹŘåŹĺæŽť'éIJšåď'ŰéČĺçŁűæĂĄçżŹçŤĺæĹůïijŇ
åĹńå£ŸäžĘä¡äåŕäżěçőĂå■ŢçŽĎåřĘåőČåđçŐřäÿžäÿĂäÿłçśżïijŇçĎűåŘŐæŁŁçŤ§æĹŘåŹĺåĞ¡æŢřæŤ¿åĹř
`__iter__()` æŰźæşŢäÿ■è£ĞåŐżãĂĆæŕŤåęĆïijŽ

```
from collections import deque

class linehistory:
    def __init__(self, lines, histlen=3):
        self.lines = lines
        self.history = deque(maxlen=histlen)

    def __iter__(self):
        for lineno, line in enumerate(self.lines, 1):
            self.history.append((lineno, line))
            yield line

    def clear(self):
        self.history.clear()
```

äÿžäžĘä¡£çŤÍè£ŹäÿłçśżïijŇä¡ššŘŕäžěšřĘšőČš¡ŞšĄŽæŸŕäÿĂäÿłæŹőéĂŽçŽĎçŤ§æĹŘšŹÍšĞ¡æŢřãĂĆ
çĎűèĂŇïijŇçŤśäžŐšŘŕäžěšĹŻšžžäÿĂäÿłšőđä¿ŃšŕźèśąïijŇäžŐæŸŕä¡ššŘŕäžěèő£éŮőšĘĚéČÍšśđæĂğšĂijïijŇ
æŕŤšęĆ `history` śśđæĂğæĹŰèĂĚæŸŕ `clear()` æŰźæşŢãĂĆäžčçăĄçď žä¿ŃšęĆäÿŃïijŽ

```
with open('somefile.txt') as f:
    lines = linehistory(f)
    for line in lines:
        if 'python' in line:
            for lineno, hline in lines.history:
                print('{}:{}'.format(lineno, hline), end='')
```

èőÍèőž

šĚşäžŐçŤ§æĹŘšŹÍïijŇš¿Ĺšőźæ ŸŞæŐL'è£ŻšĞ¡æŢřæŮäæL'Ääÿ■èČ¡çŽĎéŹůéŸśãĂĆ
šęĆæđIJçŤ§æĹŘšŹÍšĞ¡æŢřéIJĂèęĄèů§ä¡äçŽĎçÍŃšžŘšĚűäžŰéČÍšĹĘæL'Şäžď'éĄŞçŽĎèŕİ(æŕŤšęĆæŽť'éIJšš
šŘŕèČ¡äijŽšŕijèĞť'ä¡äçŽĎäžčçăĄšijĆšÿÿçŽĎšď'■æİĆãĂĆ šęĆæđIJæŸŕè£Źçğ■æČĚšĘţçŽĎèŕİïijŇšŘŕäžěèĂĆ
šIJÍ `__iter__()` æŰźæşŢäÿ■šőŽäźL'ä¡äçŽĎçŤ§æĹŘšŹÍäÿ■äijŽæŤźšŘŸä¡ääżä¡ŢçŽĎçőŮæşŢéĂżè¿ŚãĂĆ
çŤśäžŐšőČæŸŕçśżçŽĎäÿĂéČÍšĹĘïijŇæL'ĂäžěšĚĄèőÿä¡äšőŽäźL'šŘĎçğ■šśđæĂğšŠŇæŰźæşŢæİěä¿ŻçŤÍæÍ

äÿĂäÿłéIJĂèęĄæşÍæĎŘçŽĎšřŘšIJŕæŰźæŸŕïijŇšęĆæđIJä¡äšIJÍè£■äžčæŞ■ä¡IJæŮűäÿ■ä¡£çŤÍforš¿łçŐŕè
`iter()` šĞ¡æŢřãĂĆæŕŤšęĆïijŽ

```
>>> f = open('somefile.txt')
>>> lines = linehistory(f)
>>> next(lines)
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
TypeError: 'linehistory' object is not an iterator

>>> # Call iter() first, then start iterating
>>> it = iter(lines)
>>> next(it)
'hello world\n'
>>> next(it)
'this is a test\n'
>>>
```

6.7 4.7 è£■äžčšŹÍšĹĞçL'Ğ

éŮőécŸ

ä¡äæČşš¿ŮšĹřäÿĂäÿłçŤśè£■äžčšŹÍçŤ§æĹŘçŽĎšĹĞçL'ĞšŕźèśąïijŇä¡ĘæŸŕæĂĞšĞĘšĹĞçL'ĞæŞ■ä¡IJšź

èğčšĘşæŰźæąĹ

šĞ¡æŢř `itertools.islice()` æ■čšě¡éĂĆçŤÍäžŐšIJÍè£■äžčšŹÍšŠÑçŤ§æĹŘšŹÍäÿŁšĄŽšĹĞçL'ĞæŞ

```
>>> def count(n):
...     while True:
...         yield n
...         n += 1
...
>>> c = count(0)
>>> c[10:20]
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
TypeError: 'generator' object is not subscriptable

>>> # Now using islice()
>>> import itertools
>>> for x in itertools.islice(c, 10, 20):
...     print(x)
...
10
11
12
13
14
15
16
17
18
19
>>>
```

èőÍèőž

è£■äžčåŹÍåŠŇçŤ§æĹŘåŹÍäÿ■èČ¡ä¡£çŤÍæăĞåĞĘçŽĎåĹĞçL'ĞæŞ■ä¡IJïijŇåŻääÿžåőČäžňçŽĎéŢ£åžęäžŇ
åĞ¡æŢř `islice()` è£ŤåŻđäÿĂäÿłåŘŕäžěçŤ§æĹŘæŇĞåőŽåĚČçť'äçŽĎè£■äžčåŹÍïijŇåőČéĂŽè£ĞéĄ■åŐĘå
çĎűåŘŐæL'■åijĂåğŃäÿĂäÿłäÿłçŽĎè£ŤåŻđåĚČçť'äïijŇåźűçŻť'åĹřåĹĞçL'ĞçżŞæİ§çť'ćåijŢä¡■ç¡õãĂĆ

è£ŹéĞŇèęĄçİĂéĞ■åijžèřČçŽĎäÿĂçĆźæŸŕ `islice()`
äijŽæűĹèĂŮæŐL'äijăåĚěçŽĎè£■äžčåŹÍäÿ■çŽĎæŢřæ■õãĂĆ å£Ěéążè ĂČèŹŚåĹřè£■äžčåŹÍæŸŕäÿ■åŘŕéĂĘçŽ
æL'ĂäžěåęĆæđIJä¡äéIJĂèęĄäźŃåŘŐåĘ■æňąèő£éŮőè£Źäÿłè£■äžčåŹÍçŽĎèŕÍïijŇéĆčä¡äářśå¿ŮåĚĹářĘåőČéĞ

6.8 4.8 èůşè£ĞåŘŕè£■äžčářźèśąçŽĎåijĂåğŃéČÍåĹĘ

éŮőécŸ

ä¡äæČşéĄ■åŐĘäÿĂäÿłåŘŕè£■äžčářźèśąïijŇä¡ĘæŸŕåőČåijĂåğŃçŽĎæ§ŘäžŻåĚČçť'ää¡äářűäÿ■æĎ§åĚť'è

èğčåĘşæŰźæąĹ

itertools æĺąåİŮäÿ■æIJĽäÿĂäžŻåĞ¡æŢřåŕŕäžěåőŇæĹŘè£ŹäÿłäžżåŁąãĂĆéęŰåĚĹäžŃçż■çŽĎæŸŕ itertools.dropwhile() åĞ¡æŢřãĂĆä¡£çŤĺæŮűïijŇä¡äçżŹåőČäijäéĂŠäÿĂäÿåőČäijŽè£ŤåŻđäÿĂäÿłè£■äžčåŹĺåŕźèśąïijŇäÿćåijČåŐ§æIJĽäžŘåĹŮäÿ■çŻťåĹřåĞ¡æŢřè£ŤåŻđFlaseäźŃåĽ■

äÿžäžĘæijŤçď žïijŇåĄĞåőŽä¡äåIJĺèŕżåŔŰäÿĂäÿłåijĂåğŃéČĺåĹĘæŸŕåĞäèąŇæşĺéĞŁçŽĎæžŘæŰĞäżűã

```
>>> with open('/etc/passwd') as f:
... for line in f:
...     print(line, end='')
...
##
# User Database
#
# Note that this file is consulted directly only when the system is␣
↪running
# in single-user mode. At other times, this information is provided␣
↪by
# Open Directory.
...
##
nobody:*:-2:-2:Unprivileged User:/var/empty:/usr/bin/false
root:*:0:0:System Administrator:/var/root:/bin/sh
...
>>>
```

åęĆæđIJä¡äæČşèůşè£ĞåijĂåğŃéČĺåĹĘçŽĎæşĺéĞŁèąŇçŽĎèŕĺïijŇåŕŕäžěè£ŹæăůåĄŽïijŽ

```
>>> from itertools import dropwhile
>>> with open('/etc/passwd') as f:
...     for line in dropwhile(lambda line: line.startswith('#'), f):
...         print(line, end='')
...
nobody:*:-2:-2:Unprivileged User:/var/empty:/usr/bin/false
root:*:0:0:System Administrator:/var/root:/bin/sh
...
>>>
```

è£Źäÿłä¿Ńå■ŘæŸŕå§žäžŐæäźæ■őæ§ŘäÿłæţŃèŕŢåĞ¡æŢřèůşè£ĞåijĂåğŃçŽĎåĚČçť äăĂĆ åęĆæđIJä¡äåůšçżŘæŸŐçąőç§ěéĄŞäžEèęĄèůşè£ĞçŽĎåĚČçť äçŽĎäÿłæŢřçŽĎèŕĺïijŇéĆčäźĹåŕŕäžěä¡£çŤĺ itertools.islice() æĺěäžčæŻ£ãĂĆæŕŤåęĆïijŽ

```
>>> from itertools import islice
>>> items = ['a', 'b', 'c', 1, 4, 10, 15]
>>> for x in islice(items, 3, None):
...     print(x)
...
1
4
10
```

(continues on next page)

(continued from previous page)

```
15
>>>
```

åIJÍè£Źäÿłä¿Ńå■Řäÿ■ïijŃ `islice()` åĞ¡æŢřæIJĂåŘŐéĆčäÿł `None` åŘĆæŢřæŃĞåőŽäžĘä¡äèęĄèŐůåŘŰäżŐçňň3äÿłåĹřæIJĂåŘŐçŽĎæĽĂæIJĽåĚČçť'äïijŃ åęĆæđIJ `None` åŠŃ3çŽĎä¡■ç¡őåŕźěřČïijŃæĎŘæĂİåřśæŸŕäżĚäżĚèŐůåŘŰåĽ■äÿĽäÿłåĚČçť'äæĄřæĄřçŻÿåŘ (è£Źäÿłèů§åĹĞçĽĞçŽĎçŻÿåŘ■æŞ■ä¡IJ `[3:]` åŠŃ `[:3]` åŐ§çŘĘæŸŕäÿĂæäůçŽĎ)ãĂĆ

èőÍèőž

åĞ¡æŢř `dropwhile()` åŠŃ `islice()` åĚűåőđåřśæŸŕäÿd'äÿłåÿőåŁĽåĞ¡æŢřïijŃäÿžçŽĎåřśæŸŕéĄ£å

```
with open('/etc/passwd') as f:
    # Skip over initial comments
    while True:
        line = next(f, '')
        if not line.startswith('#'):
            break

    # Process remaining lines
    while line:
        # Replace with useful processing
        print(line, end='')
        line = next(f, None)
```

èůşè£ĞäÿĂäÿłåŘŕè£■äžčåŕźèśąçŽĎåijĂåğŃéČĺåĹĘèů§éĂŽåÿÿçŽĎè£Ğæżď'æŸŕäÿ■åŘŃçŽĎãĂĆ æŕŤåęĆïijŃäÿŁè£řäžčçäĄçŽĎçňňäÿĂäÿłéČĺåĹĘåŘŕèČ¡äijŽè£ŹæäůéĞ■åĘŹïijŽ

```
with open('/etc/passwd') as f:
    lines = (line for line in f if not line.startswith('#'))
    for line in lines:
        print(line, end='')
```

è£ŹæäůåĘŹçąőåőđåŘŕäžěèůşè£ĞåijĂåğŃéČĺåĹĘçŽĎæşĺéĞŁèąŃïijŃä¡ĘæŸŕåŘŃæäůäź§äijŽèůşè£ĞæŰ æ■ćåŘěèŕÍèőšïijŃæĹŚäžňçŽĎèğčåĘşæŰźæąĹæŸŕäżĚäżĚèůşè£ĞåijĂåğŃéČĺåĹĘæżąèűşæţŃèŕŤæÍąäżűçŽĎè

æIJĂåŘŐéIJĂèęĄçÍĂéĞ■åijžèřČçŽĎäÿĂçĆźæŸŕïijŃæIJňèŁĆçŽĎæŰźæąĹéĂĆçŤĺäžŐæĽĂæIJĽåŕźè£ æŕŤåęĆçŤ§æĹŘåŹĺïijŃæŰĞäžűåŘŁåĚűçśżäijijçŽĎåŕźèśąãĂĆ

6.9 4.9 æŐŠåĹŮçżĎåŘĹçŽĎè£■äžč

éŮőéćŸ

ä¡äæČşè£■äžčéĄ■åŐĘäÿĂäÿłéŻĘåŘĹäÿ■åĚČçť'äçŽĎæĽĂæIJĽåŕźèČ¡çŽĎæŐŠåĹŮæĹŰçżĎåŘĹ

èğčåĘşæŰźæąĹ

itertoolsæĺąåİŮæŔŘä¿ŻäžĘäÿĽäÿłåĞ¡æŢřæİěèğčåĘşè£ŹçśżéŮőéćŸãĂĆ
åĚűäÿ■äÿĂäÿłæŸŕ itertools.permutations() ïijŇ
åőČæŐěåŘŮäÿĂäÿłéŻĘåŘĹåźűäžğçŤ§äÿĂäÿłåĚČçżĎåžŘåĹŮïijŇæŕŘäÿłåĚČçżĎçŤśéŻĘåŘĹäÿ■æĽĂæIJĽ
äź§åřśæŸŕèŕťéĂŽè£ĞæĽŞäźśéŻĘåŘĹäÿ■åĚČçť ăæŐŠåĹŮéąžåžŘçŤ§æĹŘäÿĂäÿłåĚČçżĎïijŇæŕŤåęĆïijŽ

```
>>> items = ['a', 'b', 'c']
>>> from itertools import permutations
>>> for p in permutations(items):
...     print(p)
...
('a', 'b', 'c')
('a', 'c', 'b')
('b', 'a', 'c')
('b', 'c', 'a')
('c', 'a', 'b')
('c', 'b', 'a')
>>>
```

åęĆæđIJä¡ăæČşå¿ŮåĹřæŇĞåőŽéŢ£åžęçŽĎæĽĂæIJĽæŐŠåĹŮïijŇä¡ăåŘŕäżěäijăéĂŠäÿĂäÿłåŘŕéĂĽçŽ

```
>>> for p in permutations(items, 2):
...     print(p)
...
('a', 'b')
('a', 'c')
('b', 'a')
('b', 'c')
('c', 'a')
('c', 'b')
>>>
```

ä¡£çŤÍ itertools.combinations() åŘŕå¿ŮåĹřè¿ŞåĚěéŻĘåŘĹäÿ■åĚČçť ăçŽĎæĽĂæIJĽçŽĎçżĎ

```
>>> from itertools import combinations
>>> for c in combinations(items, 3):
...     print(c)
...
('a', 'b', 'c')

>>> for c in combinations(items, 2):
...     print(c)
...
('a', 'b')
('a', 'c')
('b', 'c')

>>> for c in combinations(items, 1):
...     print(c)
```

(continues on next page)

(continued from previous page)

```
...
('a',)
('b',)
('c',)
>>>
```

åŕźäžŐ `combinations()` æİěèőšïijŇåĚČçť'äçŽĎéąžåžŔåůšçżŔäÿ■éĞ■èęĄäžĘãĂĆäź§åřśæŸŕèŕť'ïijŇçżĎåŘĹ `('a', 'b')` èů§ `('b', 'a')` åĚűåőđæŸŕäÿĂæăůçŽĎ(æIJĂçźĹåŔłäijŽè¿ŞåĞžåĚűäÿ■äÿĂäÿł)ãĂĆ

åIJÍèőąçőŮçżĎåŘĹçŽĎæŮűåĂŹïijŇäÿĂæŮęåĚČçť'äèćńéĂĽ'åŔŰåřśäijŽäżŐåĂŹéĂĽ'äÿ■åĽ'ŤéŹď'æŐĽ'èĂŇåĞ¡æŢř `itertools.combinations_with_replacement()` åĚĄèőÿåŘŇäÿĂäÿłåĚČçť'äèćńéĂĽ'æŃľ'åď'ŽæňąïijŇæŕŤåęĆïijŽ

```
>>> for c in combinations_with_replacement(items, 3):
...     print(c)
...
('a', 'a', 'a')
('a', 'a', 'b')
('a', 'a', 'c')
('a', 'b', 'b')
('a', 'b', 'c')
('a', 'c', 'c')
('b', 'b', 'b')
('b', 'b', 'c')
('b', 'c', 'c')
('c', 'c', 'c')
>>>
```

èőÍèőž

è£ŹäÿĂåřŔèŁĆæĹŚäžňåŘŚä¡ăåśŢçď'žçŽĎäżĚäżĚæŸŕ `itertools` æÍąåİŮçŽĎäÿĂéČÍåĹĘåŁ§èČ¡ãĂĆ åř¡çőąä¡ääź§åŔŕäžěèĞłåůśæĽ'ŃåŁÍåőđçŐřæŐŠåĹŮçżĎåŘĹçőŮæşŢïijŇä å¡ŞæĹŚäžňçćřåĹřçIJŃäÿŁåŐżæIJĽ'äžŽåď'■æİĆçŽĎè£■äžčéŮőécŸæŮűïijŇæIJĂåě¡åŔŕäžěåĚĹåŐżçIJŃçIJŃi åęĆæđIJè£ŹäÿłéŮőécŸå¿ĹæŹőéĄ■ïijŇéČčäźĹå¿ĹæIJĽ'åŔŕèČ¡äijŽåIJÍéĞŇéİćæĽ'¿åĹřèğčåĘşæŰźæąĹïijĄ

6.10 4.10 åžŔåĹŮäÿŁçť'ćåijŢåĂijè£■äžč

éŮőécŸ

ä¡äæČşåIJÍè£■äžčäÿĂäÿłåžŔåĹŮçŽĎåŘŇæŮűèů§èÿłæ■čåIJÍèćńåď'ĎçŘĘçŽĎåĚČçť'äçť'ćåijŢãĂĆ

èğčåĘşæŰźæąĹ

åĘĚç¡őçŽĎ `enumerate()` åĞ¡æŢřåŔŕäžěå¿Ĺåě¡çŽĎèğčåĘşè£ŹäÿłéŮőécŸïijŽ

```
>>> my_list = ['a', 'b', 'c']
>>> for idx, val in enumerate(my_list):
...     print(idx, val)
...
0 a
1 b
2 c
```

äÿžäžĘæŇL'äijäçż§èąŇåŘůè¿ŞåĞž(èąŇåŘůäżŐ1åijĂåğŃ)ïijŇä¡ääŘŕäžěäijäéĂŠäÿĂäÿłäijĂåğŃåŘĆæŢř

```
>>> my_list = ['a', 'b', 'c']
>>> for idx, val in enumerate(my_list, 1):
...     print(idx, val)
...
1 a
2 b
3 c
```

è£Źçğ■æČĚåĘţåIJÍä¡äéĄ■åŐĘæŰĞäżűæŮűæČşåIJÍéŤŹèŕŕæűĹæĄŕäÿ■ä¡£çŤÍèąŇåŘůäőŽä¡■æŮűäĂŹéĹ

```
def parse_data(filename):
    with open(filename, 'rt') as f:
        for lineno, line in enumerate(f, 1):
            fields = line.split()
            try:
                count = int(fields[1])
                ...
            except ValueError as e:
                print('Line {}: Parse error: {}'.format(lineno, e))
```

`enumerate()` åŕźäžŐèů§èÿłæ§ŘäžŻåĂijåIJÍåĹŮèąĺäÿ■åĞžçŐřçŽĎä¡■ç¡őæŸŕå¿ĹæIJL'çŤÍçŽĎãĂĆ
æL'ÄäżěïijŇåęĆæđIJä¡äæČşåřĘäÿĂäÿłæŰĞäżűäÿ■åĞžçŐřçŽĎå■Ţèŕ■æŸäåřĎåĹřäőČåĞžçŐřçŽĎèąŇåŘůäÿ
`enumerate()` æĺěäőŇæĹŘïijŽ

```
word_summary = defaultdict(list)

with open('myfile.txt', 'r') as f:
    lines = f.readlines()

for idx, line in enumerate(lines):
    # Create a list of words in current line
    words = [w.strip().lower() for w in line.split()]
    for word in words:
        word_summary[word].append(idx)
```

åęĆæđIJä¡äåd'ĎçŘĘäőŇæŰĞäżűåŘŐæL'Şå■ř `word_summary`
ïijŇäijŽåŘŚçŐřäőČæŸŕäÿĂäÿłå■ŮåĚÿ(åĞĘçąőæĺěèőšæŸŕäÿĂäÿł `defaultdict`
)ïijŇ åŕźäžŐæŕŘäÿłå■Ţèŕ■æIJL'äÿĂäÿł `key` ïijŇæŕŘäÿł `key`
åŕźäžŤçŽĎåĂijæŸŕäÿĂäÿłçŤśè£ŹäÿłÅ■Ţèŕ■åĞžçŐřçŽĎèąŇåŘůçżĎæĹŘçŽĎåĹŮèąĺãĂĆ
åęĆæđIJæ§Řäÿłå■Ţèŕ■åIJÍäÿĂèąŇäÿ■åĞžçŐřè£Ğäÿd'æňąïijŇéĆčäźĹè£ŹäÿłèąŇåŘůäź§äijŽåĞžçŐřäÿd'æňąï

åŘŇæŮűäź§åŔŕäżěä¡IJäÿžæŰĞæIJňçŽĎäÿĂäÿłçőĂå■Ţçż§èőąãĂĆ

èőÍèőž

å¡Şä¡äæČşéćÍåď'ŰåőŽäźĽäÿĂäÿłèőąæŢřåŘŸéĞŘçŽĎæŮűåĂŹïijŇä¡£çŤÍ `enumerate()` åĞ¡æŢřäijŽæŻť'åŁăçőĂå■ŢãĂĆä¡ăåŔŕèČ¡äijŽåČŘäÿŃéİćè£ŹæăůåĘŹäżčçăĄïijŽ

```
lineno = 1
for line in f:
    # Process line
    ...
    lineno += 1
```

ä¡ĘæŸŕåęĆæđIJä¡£çŤÍ `enumerate()` åĞ¡æŢřæİěäżčæŻ£åřśæŸ¿å¿ŮæŻť'åŁääijŸéŻĚäžĘïijŽ

```
for lineno, line in enumerate(f):
    # Process line
    ...
```

`enumerate()` åĞ¡æŢřè£ŤåŻđçŽĎæŸŕäÿĂäÿł `enumerate` åŕźèśąåőđä¿ŃïijŇ åőĆæŸŕäÿĂäÿłè£■äżčåŹÍïijŇè£ŤåŻđè£đçż■çŽĎåŇĚåŘńäÿĂäÿłèőąæŢřåŠŇäÿĂäÿłåĂijçŽĎåĚČçżĎïijŇ åĚČçżĎäÿ■çŽĎåĂijéĂŽè£ĞåIJÍäijäåĚěåžŘåĹŮäÿŁèřČçŤÍ `next()` è£ŤåŻđãĂĆ

è£ŸæIJL'äÿĂçĆźåŔŕèČ¡åźűäÿ■å¿ĹéĞ■èęĄïijŇä¡ĘæŸŕäź§åĂijå¿ŮæşÍæĎŔïijŇ æIJL'æŮűåĂŹå¡Şä¡äåIJÍäÿĂäÿłåůšçżŘèğčåŐŃåŘŐçŽĎåĚČçżĎåžŘåĹŮäÿŁä¡£çŤÍ `enumerate()` åĞ¡æŢřæŮűå¿Ĺåőźæ ŸŞèřČåĚěéŹůéŸśãĂĆ ä¡ăå¿ŮåČŘäÿŃéİćæ■čçąőçŽĎæŰźåijŘè£ŹæăůåĘŹïijŽ

```
data = [ (1, 2), (3, 4), (5, 6), (7, 8) ]

# Correct!
for n, (x, y) in enumerate(data):
    ...
# Error!
for n, x, y in enumerate(data):
    ...
```

6.11 4.11 åŘŇæŮűè£■äżčåď'ŽäÿłåžŕåĹŮ

éŮőécŸ

ä¡ăæČşåŘŇæŮűè£■äżčåď'ŽäÿłåžŕåĹŮïijŇæŕŘæňąåĹĘåĹńäżŐäÿĂäÿłåžŕåĹŮäÿ■åŔŰäÿĂäÿłåĚČçť'ăăĂ

èğčåĘşæŰźæąĹ

äÿžäžĘåŘŇæŮűè£■äżčåď'ŽäÿłåžŕåĹŮïijŇä¡£çŤÍ `zip()` åĞ¡æŢřăĂĆæŕŤåęĆïijŽ

```
>>> xpts = [1, 5, 4, 2, 10, 7]
>>> ypts = [101, 78, 37, 15, 62, 99]
>>> for x, y in zip(xpts, ypts):
...     print(x,y)
...
1 101
5 78
4 37
2 15
10 62
7 99
>>>
```

zip(a, b) äijŽçŤ§æĹŘäÿĂäÿłåŔŕè£ŤåŻđåĚČçżĎ (x, y) çŽĎè£■äżčåŹÍïijŇåĚűäÿ■xæİěèĞłaïijŇyæİěèĞłbãĂĆ äÿĂæŮęåĚűäÿ■æ§Řäÿłåžŕ̊åĹŮåĹřåžŢçżŞåř¿ïijŇè£■äż åŻĂæ■d'è£■äżčéŢ£åžęèů§åŔĆæŢřäÿ■æIJĂç§■åžŔåĹŮéŢ£åžęäÿĂèĞt'ãĂĆ

```
>>> a = [1, 2, 3]
>>> b = ['w', 'x', 'y', 'z']
>>> for i in zip(a,b):
...     print(i)
...
(1, 'w')
(2, 'x')
(3, 'y')
>>>
```

åęĆæđIJè£Źäÿłäÿ■æŸŕä¡ăæČşèęĄçŽĎæŢĹæđIJïijŇéĆčäźĹè£ŸåŔŕäżëä¡£çŤÍ itertools.zip_longest() åĞ¡æŢřæİěäżčæŻ£ãĂĆæŕŤåęĆïijŽ

```
>>> from itertools import zip_longest
>>> for i in zip_longest(a,b):
...     print(i)
...
(1, 'w')
(2, 'x')
(3, 'y')
(None, 'z')

>>> for i in zip_longest(a, b, fillvalue=0):
...     print(i)
...
(1, 'w')
(2, 'x')
(3, 'y')
(0, 'z')
>>>
```

èőÍèőž

å¡Şä¡ăæČşæĹŘåŕźåď ĎçŘĘæŢřæ■őçŽĎæŮűåĂŹ zip() åĞ¡æŢřæŸŕå¿ĹæIJĽçŤÍçŽĎãĂĆ æŕŤåęĆïijŇåĄĞèő¿ä¡ăåď ť åĹŮèąÍåŠŇäÿĂäÿłåĂijåĹŮèąÍïijŇåŕśåČŘäÿŃéİć

```
headers = ['name', 'shares', 'price']
values = ['ACME', 100, 490.1]
```

ä¡£çŤÍzip()åŘŕäżěèőľä¡ăåřĘåőČäżňæĽŞåŇĚåźűçŤ§æĹŘäÿĂäÿłå■ŮåĚÿïijŽ

```
s = dict(zip(headers,values))
```

æĹŰèĂĚä¡ăäź§åŘŕäżěåČŘäÿŃéİćè£ŹæăůäžğçŤ§è¿ŞåĞžïijŽ

```
for name, val in zip(headers, values):
    print(name, '=', val)
```

èŹ¡çĎűäÿ■åÿÿèğĄïijŇä¡ĘæŸŕ zip() åŘŕäżěæŐěåŘŮåď ŽäžŐäÿď äÿłçŽĎåžŘåĹŮçŽĎåŘĆæŢřãĂĆ è£ŹæŮűåĂŹæĽĂçŤ§æĹŘçŽĎçżŞæđIJåĚČçżĎäÿ■åĚČçť ääÿłæŢřèů§è¿ŞåĚěåžŘåĹŮäÿłæŢřäÿĂæăůãĂĆæŕŤ

```
>>> a = [1, 2, 3]
>>> b = [10, 11, 12]
>>> c = ['x','y','z']
>>> for i in zip(a, b, c):
...     print(i)
...
(1, 10, 'x')
(2, 11, 'y')
(3, 12, 'z')
>>>
```

æIJĂåŘŐåijžèřČäÿĂçĆźåŕśæŸŕïijŇ zip() äijŽåĹŻåżžäÿĂäÿłè£■äžčåŹĺæĺěä¡IJäÿžçżŞæđIJè£ŤåŻđãĂĆ åęĆæđIJä¡äéIJĂèęĄåřĘçżŞåŕźçŽĎåĂijå■ŸåĆĺåIJĺåĹŮèąĺäÿ■ïijŇèęĄä¡£çŤÍ list() åĞ¡æŢřãĂĆæŕŤåęĆïijŽ

```
>>> zip(a, b)
<zip object at 0x1007001b8>
>>> list(zip(a, b))
[(1, 10), (2, 11), (3, 12)]
>>>
```

6.12 4.12 äÿ■åŘŇéŻĘåŘĹäÿłåĚČçť äçŽĎè£■äžč

éŮőéćŸ

ä¡ăæČşåIJĺåď ŽäÿłåŕźèśąæĽ ğèąŇçŻÿåŘŇçŽĎæŞ■ä¡IJïijŇä¡ĘæŸŕè£ŹäžŻåŕźèśąåIJĺäÿ■åŘŇçŽĎåőźåŹĺäÿ

èğčåĘşæŰźæąĹ

itertools.chain() æŰźæşŢåŘŕäžěçŤĺæĺěçőĂåŇŰè£ŹäÿłäżżåŁąãĂĆ
åőČæŐěåŘŮäÿĂäÿłåŘŕè£■äżčåŕźèśąåĹŮèąĺä¡IJäÿžè¿ŞåĚěïijŇåźűè£ŤåŻđäÿĂäÿłè£■äżčåŹĺïijŇæIJĽæŢĹçŽĎ
äÿžäžĘæijŤçď žæÿĚæěŽïijŇèĂČèŹŚäÿŃéĺćè£Źäÿłä¿Ňå■ŘïijŽ

```
>>> from itertools import chain
>>> a = [1, 2, 3, 4]
>>> b = ['x', 'y', 'z']
>>> for x in chain(a, b):
... print(x)
...
1
2
3
4
x
y
z
>>>
```

ä¡£çŤĺ chain() çŽĎäÿĂäÿłåÿÿèğĄåIJžæŹŕæŸŕå¡Şä¡äæČşåŕźäÿ■åŘŇçŽĎéŻĘåŘĹäÿ■æĽĂæIJĽåĚČç

```
# Various working sets of items
active_items = set()
inactive_items = set()

# Iterate over all items
for item in chain(active_items, inactive_items):
    # Process item
```

è£Źçğ■èğčåĘşæŰźæąĹèęĄæŕŤåČŔäÿŃéĺćè£Źæăůä¡£çŤĺäÿď äÿłå■ŢçŃňçŽĎå¿łçŐŕæŻť åŁääijŸéŻĚïijŇ

```
for item in active_items:
    # Process item
    ...

for item in inactive_items:
    # Process item
    ...
```

èőĺèőž

itertools.chain() æŐěåŘŮäÿĂäÿłæĹŰåď ŽäÿłåŘŕè£■äżčåŕźèśąä¡IJäÿžè¿ŞåĚěåŘĆæŢřãĂĆ
çĎűåŘŐåĹŻåżžäÿĂäÿłè£■äżčåŹĺïijŇä¿Ìæňąè£đçż■çŽĎè£ŤåŻđæŕŘäÿłåŘŕè£■äżčåŕźèśąäÿ■çŽĎåĚČçť ããĂĆ
è£Źçğ■æŰźåijŘèęĄæŕŤåĚĹåřĘåžŘåĹŮåŘĹåźűåĘ■è£■äżčèęĄéńŸæŢĹçŽĎåď ŽãĂĆæŕŤåęĆïijŽ

```
# Inefficent
for x in a + b:
```

(continues on next page)

(continued from previous page)

```
    ...

# Better
for x in chain(a, b):
    ...
```

çňňäÿĂçğ■æŰźæąĹäÿ■ïijŇ `a + b` æŞ■ä¡IJäijŽåĹŻåżžäÿĂäÿłåĚĺæŰřçŽĎåžŕåĹŮåźűèęĄæśĆaåŠŇbçŽ `chian()` äÿ■äijŽæIJL'è£ŹäÿĂæ■ěïijŇæL'Ăäżěåę̨ĆæđIJè¿ŞåĚěåžŕåĹŮéİđåÿÿåď'ğçŽĎæŮűåĂŹäijŽå¿ĹçIJA åźűäÿŤå¡Şåŕŕŕèŕ£■äżčåŕźèśąçśżåđŇäÿ■äÿĂæăůçŽĎæŮűåĂŹ `chain()` åŘŇæăůåŕŕäżěå¿Ĺåě¡çŽĎåůěä¡IJãĂĆ

6.13 4.13 åĹŻåżžæŢřæ■őåď'ĎçŘĘçőąéĄŞ

éŮőéćŸ

ä¡ăæČşäżěæŢřæ■őçőąéĄŞ(çśżäijijUnixçőąéĄŞ)çŽĎæŰźåijŕè£■äżčåď'ĎçŘĘæŢřæ■õãĂĆ æŕŤåęĆïijŇä¡ăæIJL'äÿłåď'ğéĞŕçŽĎæŢřæ■õéIJĂèęĄåď'ĎçŘĘïijŇä¡ĘæŸŕäÿ■èČ¡åřĘåőČäżňäÿĂæňąæĂğæŤ¿

èğčåĘşæŰźæąĹ

çŤ§æĹŘåŹĺåĞ¡æŢřæŸŕäÿĂäÿłåőđçŐřçőąéĄŞæIJżåĹűçŽĎåě¡åŁđæşŢãĂĆ äÿžäżĘæijŤçď'žïijŇåĄĞåőŽä¡äèęĄåď'ĎçŘĘäÿĂäÿłéİđåÿÿåď'ğçŽĎæŮěå£ŮæŰĞäżűçŻőå¡ŢïijŽ

```
foo/
    access-log-012007.gz
    access-log-022007.gz
    access-log-032007.gz
    ...
    access-log-012008
bar/
    access-log-092007.bz2
    ...
    access-log-022008
```

åĄĞèő¿æŕŘäÿłæŮěå£ŮæŰĞäżűåŇĚåŘńè£ŹæăůçŽĎæŢřæ■õïijŽ

```
124.115.6.12 - - [10/Jul/2012:00:18:50 -0500] "GET /robots.txt ..."␣
↪200 71
210.212.209.67 - - [10/Jul/2012:00:18:51 -0500] "GET /ply/ ..." 200␣
↪11875
210.212.209.67 - - [10/Jul/2012:00:18:51 -0500] "GET /favicon.ico ..
↪." 404 369
61.135.216.105 - - [10/Jul/2012:00:20:04 -0500] "GET /blog/atom.xml␣
↪..." 304 -
...
```

äÿžäżĘåď'ĎçŘĘè£ŹäżŻæŰĞäżűïijŇä¡ăåŕŕäżěåőŽäźL'äÿĂäÿłçŤśåď'Žäÿłæ̨L'ğèąŇçL'źåőŽäżżåŁąçŃňçńŇ

```python
import os
import fnmatch
import gzip
import bz2
import re

def gen_find(filepat, top):
    '''
    Find all filenames in a directory tree that match a shell␣
↪wildcard pattern
    '''
    for path, dirlist, filelist in os.walk(top):
        for name in fnmatch.filter(filelist, filepat):
            yield os.path.join(path,name)

def gen_opener(filenames):
    '''
    Open a sequence of filenames one at a time producing a file␣
↪object.
    The file is closed immediately when proceeding to the next␣
↪iteration.
    '''
    for filename in filenames:
        if filename.endswith('.gz'):
            f = gzip.open(filename, 'rt')
        elif filename.endswith('.bz2'):
            f = bz2.open(filename, 'rt')
        else:
            f = open(filename, 'rt')
        yield f
        f.close()

def gen_concatenate(iterators):
    '''
    Chain a sequence of iterators together into a single sequence.
    '''
    for it in iterators:
        yield from it

def gen_grep(pattern, lines):
    '''
    Look for a regex pattern in a sequence of lines
    '''
    pat = re.compile(pattern)
    for line in lines:
        if pat.search(line):
            yield line
```

çŐřåIJÍä¡ăåŘŕäżěå¿Ĺåőźæ ŸŞçŽĎåřĘè£ŹäžŻåĞ¡æŢřè£đèţůæİěåĹŻåżžäÿĂäÿłåď ĎçŘĘçőąéĄŞãĂĆ
æŕŤåęĆïijÑäÿžäžĘæ§ěæĽ¿åŇĚåŘńå■Ţèŕ■pythonçŽĎæĽĂæIJĽæŮěå£ŮèąŇïijÑä¡ăåŘŕäżěè£ŹæăůåĄŽïijŽ

```
lognames = gen_find('access-log*', 'www')
files = gen_opener(lognames)
lines = gen_concatenate(files)
pylines = gen_grep('(?i)python', lines)
for line in pylines:
    print(line)
```

åęĆæđIJåřĘæİěçŽĎæŮűåĂŹä¡ăæČşæL'l'åśŢçőąéĄŞïijŇä¡ăçŤŽèĞşåŘŕäžěåIJÍçŤ§æĹŘåŹÍèąÍè¿¿åijŘäÿ■æŕŤåęĆïijŇäÿŃéÍćè£ŹäÿłçL'ĹæIJňèőąçőŮåĞžäijăè¿ŞçŽĎå■ŮèŁĆæŢřåźűèőąçőŮåĚűæĂżåŠŇãĂĆ

```
lognames = gen_find('access-log*', 'www')
files = gen_opener(lognames)
lines = gen_concatenate(files)
pylines = gen_grep('(?i)python', lines)
bytecolumn = (line.rsplit(None,1)[1] for line in pylines)
bytes = (int(x) for x in bytecolumn if x != '-')
print('Total', sum(bytes))
```

èőÍèőž

äžěçőąéĄŞæŰźåijŘåď'ĎçŘĘæŢřæ■őåŘŕäžěçŤÍæİěèğčåĘşåŘĎçśżåĚűäžŰéŮőéćŸïijŇåŇĚæŃňèğčæđŘïi

äÿžäžĘçŘĘèğčäÿŁè£řäžčçăĄïijŇéĞ■çĆźæŸŕèęĄæŸŐçŹ¡ yield èŕ■åŘëä¡IJäÿžæŢřæ■őçŽĎçŤ§äžğèĂĚèĂŇ for å¿łçŐŕèŕ■åŘëä¡IJäÿžæŢřæ■őçŽĎæűĹèť'źèĂĚãĂĆ å¡Ş è£ŹäžŻçŤ§æĹŘåŹÍèćńè£đåIJÍäÿĂèţůåŘŐïijŇæŕŘäÿł yield äijŽåřĘäÿĂäÿłå■ŢçŃňçŽĎæŢřæ■őåĚČçť'ääijäéĂŠçżŹè£■äžčåď'ĎçŘĘçőąéĄŞçŽĎäÿŃäÿĂéŸűæőţãĂĆ åIJÍä¿Ńå■ŘæIJĂåŘŐéČÍåĹĘïijŇ sum() åĞ¡æŢřæŸŕæIJĂçżĹçŽĎçÍŃåžŘél'śåŁÍèĂĚïijŇæŕŘæňąäżŐçŤ§æĹŘ

è£Źçğ■æŰźåijŘäÿĂäÿłéİđåÿÿåě¡çŽĎçL'źçĆźæŸŕæŕŘäÿłçŤ§æĹŘåŹÍåĞ¡æŢřå¿ĹåřŘåźűäÿŤéČ¡æŸŕçŃňç å¿Ĺåď'ŽæŮűåĂŹïijŇè£ŹäžŻåĞ¡æŢřåęĆæđIJæŕŤè¿ČéĂŽçŤÍçŽĎèŕÍåŘŕäžěåIJÍåĚűäžŰåIJžæŹŕéĞ■åď'■ä¡£çŤÍ åźűäÿŤæIJĂçżĹåřĘè£ŹäžŻçżĎäžűçżĎåŘÍèţůæİěçŽĎäžčçăĄçIJŃäÿŁåŐżéİđåÿÿçőĂå■ŢïijŇäź§å¿ĹåőźæŸŞç

ä¡£çŤÍè£Źçğ■æŰźåijŘçŽĎåĘĚå■ŸæŢĹçŐĞäź§äÿ■å¿Ůäÿ■æŘŘãĂĆäÿŁè£řäžčçăĄå■ş ä¿£æŸŕåIJÍäÿĂäÿ äžŃåőđäÿŁïijŇçŤśäžŐä¡£çŤÍäžĘè£■äžčæŰźåijŘåď'ĎçŘĘïijŇäžčçăĄè£ŘèąŃè£ĞçÍŃäÿ■åŘłéIJĂèęĄå¿ĹåřŘå

åIJÍèřČçŤÍ gen_concatenate() åĞ¡æŢřçŽĎæŮűåĂŹä¡ååŘŕèČ¡äijŽæIJL'äžŻäÿ■åď'łæŸŐçŹ¡ãĂĆ è£ŹäÿłåĞ¡æŢřçŽĎçŻőçŽĎæŸŕåřĘè¿ŞåĚěåžŘåĹŮæŃijæŐěæĹŘäÿĂäÿłå¿ĹéŢ£çŽĎèąŃåžŘåĹŮãĂĆ itertools.chain() åĞ¡æŢřåŘŃæăůæIJL'çśżäijijçŽĎåŁ§èČ¡ïijŇä¡ĘæŸŕåőČéIJĂèęĄåřĘæL'ĂæIJL'åŘŕè åIJÍäÿŁéİćè£ŹäÿłäźŃå■Ŕäÿ■ïijŇä¡ååŘŕèČ¡äijŽåĘŹçśżäijijè£ŹæăůçŽĎèŕ■åŘě lines = itertools.chain(*files) ïijŇ è£ŹåřĘåŕijèĞť' gen_opener() çŤ§æĹŘåŹÍèćńæŘŘåL'■åĚÍéČÍæűĹèť'źæŐL'ãĂĆ ä¡ĘçŤśäžŐ gen_opener() çŤ§æĹŘåŹÍæŕŘæňąçŤ§æĹŘäÿĂäÿłæL'ŞåijĂè£ĞçŽĎæŰĞäžűïijŇ ç■L'åĹŕäÿŃäÿĂäÿłè£■äžčæ■ěéłď'æŮűæŰĞäžűåřśåĚşéŮ■äžĘïijŇåŽăæ■ď' chain() åIJÍè£ŹéĞŃäÿ■èČ¡è£ŹæăůäźĄçŤÍãĂĆ äÿŁéİćçŽĎæŰźæąĹåŘŕäžěéĄ£åĚ■è£ŹçğæČĚåĘţãĂĆ

gen_concatenate() åĞ¡æŢřäÿ■åĞžçŐřè£Ğ yield from èŕ■åŘěïijŇåőČåřĘ yield æŞ■ä¡IJäžčçŘĘåĹřçĹűçŤ§æĹŘåŹÍäÿŁåŐżãĂĆ èŕ■åŘě yield from it çőĂå■ŢçŽĎè£ŤåŽđçŤ§æĹŘåŹÍ it æL'Ăäžğç Ť§çŽĎæL'ĂæIJL'åĂijãĂĆ åĚşäžŐè£ŹäÿłæĹŚäžňåIJÍ4.14åřŘèŁĆäijŽæIJL'æŻť'è£ŻäÿĂæ■ěçŽĎæŘŘè£řãĂĆ

æIJĂåŘŐè£ŸæIJĽäÿĂçĆźéIJĂèęĄæşÍæĎŘçŽĎæŸŕïijŇçőąéĄŞæŰźåijŘåźűäÿ■æŸŕäÿĞèČ¡çŽĎãĂĆ
æIJĽæŮűåĂŹä¡äæČşçńŇå■şåď ĎçŘĘæĽĂæIJĽæŢřæ■õãĂĆ çĎűèĂŇïijŇå■şä¿£æŸŕè£Źçğ■æČĚåĘţïijŇä¡£ç

David Beazley åIJÍäżŰçŽĎ Generator Tricks for Systems Programmers
æŢŹçÍŃäÿ■åŕźäžŐè£Źçğ■æŁĂæIJŕæIJĽéİđåÿÿæůśåĚěçŽĎèőšèğčãĂĆåŘŕäżěåŘĆèĂČè£ŹäÿłæŢŹçÍŃèŐůåŘ

6.14 4.14 åśŢåijĂåţŇåěŮçŽĎåžŘåĹŮ

éŮőéćŸ

ä¡äæČşåřĘäÿĂäÿłåď ŽåśĆåţŇåěŮçŽĎåžŘåĹŮåśŢåijĂæĹŘäÿĂäÿłå■ŢåśĆåĹŮèąĺ

èğčåĘşæŰźæąĹ

åŘŕäżěåĘŹäÿĂäÿłåŇĚåŘń `yield from` èŕ■åŘěçŽĎéĂŠå¡ŠçŤ§æĹŘåŹÍæİěè¡żæİ¿èğčåĘşè£ŹäÿłéŮőéć

```
from collections import Iterable

def flatten(items, ignore_types=(str, bytes)):
    for x in items:
        if isinstance(x, Iterable) and not isinstance(x, ignore_
↪types):
            yield from flatten(x)
        else:
            yield x

items = [1, 2, [3, 4, [5, 6], 7], 8]
# Produces 1 2 3 4 5 6 7 8
for x in flatten(items):
    print(x)
```

åIJÍäÿŁéİćäżčçăĄäÿ■ïijŇ `isinstance(x, Iterable)` æčĂæ§ěæ§ŘäÿłåĚČçť ăæŸŕåŘęæŸŕåŘŕè£■äżčçŽĎãĂĆ åęĆæđIJæŸŕçŽĎèŕİïijŇ `yield from` åŕśäijŽè£ŤåŻđæĽĂæIJĽå■Řä¿ŃçÍŃçŽĎåĂijãĂĆæIJĂçżĹè£ŤåŻđçżŞæđIJåŕśæŸŕäÿĂäÿłæšąæIJĽåţŇå

éćİåď ŰçŽĎåŘĆæŢř `ignore_types` åŠŇæčĂæţŃèŕ■åŘě `isinstance(x, ignore_types)` çŤÍæİěåřĘå■Ůçňęäÿšå ŠŇå■ŮèŁĆæŐŠéŹď åIJÍåŘŕè£■äżčåŕźèšąåď ŰïijŇéŸšæ■ćåřĘåőČä
è£ŹæäůçŽĎèŕİå■ŮçňęäÿšæŢřçżĎåŕśèČ¡æIJĂçżĹè£ŤåŻđæĹŚäżňæĽĂæIJ§æIJŻçŽĎçżŞæđIJäżĘãĂĆæŕŤåęĆ

```
>>> items = ['Dave', 'Paula', ['Thomas', 'Lewis']]
>>> for x in flatten(items):
...     print(x)
...
Dave
Paula
Thomas
Lewis
>>>
```

èőÍèőž

èŕ■åŘě `yield from` åIJÍä¡ăæČşåIJÍçŤ§æĹŘåŹÍäÿ■èřČçŤÍåĚűäżŰçŤ§æĹŘåŹÍä¡IJäÿžå■Řä¿ŃçÍŃçŽĎæ
åęĆæđIJä¡ääÿ■ä¡£çŤÍåőČçŽĎèŕÍïijŇéĆčäźĹåřśå£ĚéążåĘŹéćÍåď'ŰçŽĎ `for`
å¿łçŐŕäžĘãĂĆæŕŤåęĆïijŽ

```
def flatten(items, ignore_types=(str, bytes)):
    for x in items:
        if isinstance(x, Iterable) and not isinstance(x, ignore_
↪types):
            for i in flatten(x):
                yield i
        else:
            yield x
```

åř¡çőąåŘłæŤźäžĘäÿĂçĆźçĆźïijŇä¡ĘæŸŕ `yield from`
èŕ■åŘěçIJŃäÿŁåŐżæĎ§èğĽ'æŻť'åě¡ïijŇåźűäÿŤäź§ä¡£å¿ŮäżčçăĄæŻť'çőĂæť'ĄæÿĚçĹ¡ãĂĆ

äźŃåĽ'■æŘŘåĹřçŽĎåŕźäžŐå■ŮçňęäÿšåŠŇå■ŮèŁĆçŽĎéćÍåď'ŰæčĂæ§ěæŸŕäÿžäžĘéŸšæ■ćåřĘåőČäżňå
åęĆæđIJè£ŸæIJĽåĚűäżŰä¡ääÿ■æČşåśŢåijĂçŽĎçśżåđŃïijŇä£őæŤźåŘĆæŢř
`ignore_types` å■şåŘŕãĂĆ

æIJĂåŘŐèęĄæşÍæĎŘçŽĎäÿĂçĆźæŸŕïijŇ `yield from`
åIJÍæűĽ'åŘŁåĹřå§žäžŐå■ŘçÍŃåŠŇçŤ§æĹŘåŹÍçŽĎåźűåŘŚçijŰçÍŃäÿ■æĽ'őæijŤçİĂæŻť'åŁäéĞ■èęĄçŽĎèğŠ
åŘŕäžěåŘĆèĂČ12.12åřŘèŁĆæ§ěçIJŃåŘęåď'ŰäÿĂäÿłä¿Ńå■ŘãĂĆ

6.15 4.15 éąžåžŘè£■äžčåŘĹåźűåŘŐçŽĎæŐŠåžŘè£■äžčåŕźèśą

éŮőéćŸ

ä¡ăæIJĽäÿĂçşżåĹŮæŐŠåžŘåžŘåĹŮïijŇæČşåřĘåőČäżňåŘĹåźűåŘŐå¿ŮåĹřäÿĂäÿłæŐŠåžŘåžŘåĹŮåźűå

èğčåĘşæŰźæąĹ

`heapq.merge()` åĞ¡æŢřåŘŕäžěåÿőä¡äèğčåĘşè£ŹäÿłéŮőéćŸãĂĆæŕŤåęĆïijŽ

```
>>> import heapq
>>> a = [1, 4, 7, 10]
>>> b = [2, 5, 6, 11]
>>> for c in heapq.merge(a, b):
...     print(c)
...
1
2
4
5
6
7
```

(continues on next page)

(continued from previous page)

```
10
11
```

èőÍèőž

heapq.merge åŘŕè£■äžčçL'źæĂğæĎŘåŚşçİĂåőČäÿ■äijŽçńŃél'ňèŕżåŘŰæL'ĂæIJL'åžŘåĹŮãĂĆ
è£ŹåřśæĎŘåŚşçİĂä¡ššŘŕäžěåIJÍéİđåÿÿéŢ£çŽĎåžŘåĹŮäÿ■ä¡£çŤÍåőČïijŇèĂŇäÿ■äijŽæIJL'åď'łåď'ğçŽĎåijĂé
æŕŤåęĆïijŇäÿŃéİćæŸŕäÿĂäÿłä¿Ňå■ŘæİěæijŤçď'žåęĆä¡ŢåŘĹåźűäÿď'äÿłæŐŠåžŘæŰĞäžűïijŽ

```
with open('sorted_file_1', 'rt') as file1, \
    open('sorted_file_2', 'rt') as file2, \
    open('merged_file', 'wt') as outf:

    for line in heapq.merge(file1, file2):
        outf.write(line)
```

æIJL'äÿĂçĆźèęĄåijžèřČçŽĎæŸŕ heapq.merge() éIJĂèęĄæL'ĂæIJL'è¿ŞåĚěåžŘåĹŮå£ĚéążæŸŕæŐŠ
çL'źåĹńçŽĎïijŇåőČåźűäÿ■äijŽécĎåĚĹèŕżåŘŰæL'ĂæIJL'æŢŕæ■őåĹŕåäĘæäĹäÿ■æĹŰèĂĚécĎåĚĹæŐŠåžŘïij
åőČäžĚäżĚæŸŕæčĂæ§ěæL'ĂæIJL'åžŘåĹŮçŽĎåijĂåğŃéČĺåĹĘåźűè£ŤåŻđæIJĂåřŘçŽĎéĆčäÿłïijŇè£ŹäÿłèĚ£Č

6.16 4.16 è£■äžčåŹĺäžčæŻ£whileæŮäéŹŘå¿łçŐŕ

éŮőéćŸ

ä¡ššIJÍäžčçăĄäÿ■ä¡£çŤÍ while å¿łçŐŕæİěè£■äžčåď'ĎçŘĘæŢŕæ■őïijŇåŻääÿžåőČéIJĂèęĄèřČçŤÍæ§Řäÿÿ
èČ¡äÿ■èČ¡çŤÍè£■äžčåŹĺæİěéĞ■åĘŹè£ŹäÿłÅ¿łçŐŕåŚćïij§

èğčåĘşæŰźæąĹ

äÿĂäÿłåÿÿèğĄçŽĎIOæŞ■ä¡IJçÍŃåžŘåŘŕèČ¡äijŽæČşäÿŃéİćè£ŹæăůïijŽ

```
CHUNKSIZE = 8192

def reader(s):
    while True:
        data = s.recv(CHUNKSIZE)
        if data == b'':
            break
        process_data(data)
```

è£Źçğ■äžčçăĄéĂŽåÿÿåŘŕäžěä¡£çŤÍ iter() æİěäžčæŻ£ïijŇåęĆäÿŃæL'Ăçď'žïijŽ

```
def reader2(s):
    for chunk in iter(lambda: s.recv(CHUNKSIZE), b''):
```

(continues on next page)

(continued from previous page)

```
        pass
        # process_data(data)
```

åęĆæđIJä¡ăæĂĂçŰŚåőČåĹřåžŢèČ¡äÿ■èČ¡æ■čåÿÿåůěä¡IJïijŇåŘŕäžěèŕŢéłŇäÿŃäÿĂäÿłçőĂå■ŢçŽĎä¿Ńå

```
>>> import sys
>>> f = open('/etc/passwd')
>>> for chunk in iter(lambda: f.read(10), ''):
...     n = sys.stdout.write(chunk)
...
nobody:*:-2:-2:Unprivileged User:/var/empty:/usr/bin/false
root:*:0:0:System Administrator:/var/root:/bin/sh
daemon:*:1:1:System Services:/var/root:/usr/bin/false
_uucp:*:4:4:Unix to Unix Copy Protocol:/var/spool/uucp:/usr/sbin/
↪uucico
...
>>>
```

èőÍèőž

`iter` åĞ¡æŢřäÿĂäÿłéšIJäÿžäžžç§ěçŽĎçL'źæĂğæŸŕåőČæŐěåŘŮäÿĂäÿłåŘŕéĂL'çŽĎ `callable` åŕźèśąåŠŇäÿĂäÿłæăĞèőř(çżŞåř¿)åĂijä¡IJäÿžè¿ŞåĚěåŘĆæŢřãĂĆ å¡Ş äżěè£Źçğ■æŰźåijŘä¡£çŤÍçŽĎæŮűåĂŹïijŇåőČäijŽåĹŽåżžäÿĂäÿłè£■äžčåŹÍïijŇ è£ŹäÿłÈ£■äžčåŹÍäijŽäÿ■æŰ■èřČçŤÍ `callable` åŕźèśąçŻť'åĹřè£ŤåŻđåĂijåŠŇæăĞèőřåĂijçŻÿç■L'äÿžæ■ćãĂ

è£Źçğ■çL'źæőŁçŽĎæŰźæşŢåŕźäžŐäÿĂäžŻçL'źåőŽçŽĎäijŽèćńéĞ■åď■èřČçŤÍçŽĎåĞ¡æŢřå¿ĹæIJL'æŢÍ äÿ¿ä¿ŃæÍěèőšïijŇåęĆæđIJä¡ăæČşäżŐåěŮæŐěå■ŮæĹŰæŰĞäżűäÿ■äżěæŢřæ■őåÍŮçŽĎæŰźåijŘèŕżåŘŰæŢřæ `read()` æĹŰ `recv()` ïijŇ åźűåIJÍåŘŐéÍćçť'ğèů§äÿĂäÿłæŰĞäżűçżŞåř¿æţŃèřŢæÍěåĘşåőŽæŸŕåŘęçżĹæ■ćã `iter()` èřČçŤÍåřśåŘŕäžěåřĘäÿď'èĂĚçżŞåŘĹèţůæÍěäžĘãĂĆ åĚűäÿ■ `lambda` åĞ¡æŢřåŘĆæŢřæŸŕäÿžäžĘåĹŻåżžäÿĂäÿłæŮăåŘĆçŽĎ `callable` åŕźèśąïijŇåźűäÿž `recv` æĹŰ `read()` æŰźæşŢæŘŘä¿ŻäžĘ `size` åŘĆæŢřãĂĆ

7 çňňäžŤçńäïijŽæŰĞäżűäÿŐIO

æL'ĂæIJL'çÍŃåžŘéČ¡èęĄåď'ĎçŘĘè¿ŞåĚěåŠŇè¿ŞåĞžãĂĆ è£ŹäÿĂçńäářĘæűţçŻŰåď'ĎçŘĘäÿ■åŘŇçśżåđŃçŽĎæŰĞäżűïijŇåŇĚæŃňæŰĞæIJňåŠŇäžŇè£ŻåĹűæŰĞäżűï åŕźæŰĞäżűåŘ■åŠŇçŻőå¡ŢçŽĎæŞ■ä¡IJäź§äijŽæűL'åŘŁåĹřãĂĆ

Contents:

7.1 5.1 èŕżåĘŹæŰĞæIJňæŢřæ■ő

éŮőéćŸ

ä¡ăéIJĂèęĄèŕżåĘŹåŘĎçğ■äÿ■åŘŇçijŰçăĄçŽĎæŰĞæIJňæŢřæ■őïijŇæŕŤåęĆASCIIïijŇUTF-8æĹŰUTF-16çijŰçăĄç■L'ãĂĆ

èğčåĘşæŰźæąĹ

ä¡£çŤĺåÿęæIJL' `rt` æĺąåijŘçŽĎ `open()` åĞ¡æŢřèŕżåŘŰæŰĞæIJňæŰĞäżűãĂĆåęĆäÿŃæL'Ăçď'žïijŽ

```
# Read the entire file as a single string
with open('somefile.txt', 'rt') as f:
    data = f.read()

# Iterate over the lines of the file
with open('somefile.txt', 'rt') as f:
    for line in f:
        # process line
        ...
```

çśżäijijçŽĎïijŇäÿžäžĘåĘŹåĚěäÿĂäÿłæŰĞæIJňæŰĞäżűïijŇä¡£çŤĺåÿęæIJL' `wt` æĺąåijŘçŽĎ `open()` åĞ¡æŢřïijŇ åęĆæđIJäźŃåL'■æŰĞäżűåĘĚåőźå■ŸåIJĺåĹŹæÿĚéŹď'åźűèęĘçŻŰæŐL'ãĂĆ

```
# Write chunks of text data
with open('somefile.txt', 'wt') as f:
    f.write(text1)
    f.write(text2)
    ...

# Redirected print statement
with open('somefile.txt', 'wt') as f:
    print(line1, file=f)
    print(line2, file=f)
    ...
```

åęĆæđIJæŸŕåIJĺåůšå■ŸåIJĺæŰĞäżűäÿ■æůżåŁăåĘĚåőźïijŇä¡£çŤĺæĺąåijŘäÿž `at` çŽĎ `open()` åĞ¡æŢřãĂĆ

æŰĞäżűçŽĎèŕżåĘŹæŞ■ä¡IJéżŸèőď'ä¡£çŤĺçşżçż§çijŰçăĄïijŇåŕŕäżěéĂŽè£ĞèřČçŤĺ `sys.getdefaultencoding()` æĺěå¿ŮåĹřãĂĆ åIJĺåď'ğåď'ŽæŢřæIJžåŹĺäÿŁéĺćéČ¡æŸŕutf-8çijŰçăĄãĂĆåęĆæđIJä¡ăåůšçżŘç§ěéĄŞä¡ăèęĄèŕżåĘŹçŽĎæŰĞæIJňæŸŕåĚűäżŰçijŰçăĄæŰźåijŘïijŇéĆčäźĹåŘŕäżěéĂŽè£ĞäijăéĂŠäÿĂäÿłåŘŕéĂL'çŽĎ `encoding` åŘĆæŢřçżŹopen()åĞ¡æŢřãĂĆåęĆäÿŃæL'Ăçď'žïijŽ

```
with open('somefile.txt', 'rt', encoding='latin-1') as f:
    ...
```

PythonæŤŕæŇĄéĺđåÿÿåď ŽçŽĎæŰĞæIJňçijŰçăĄăĂĆåĞääÿłåÿÿèğĄçŽĎçijŰçăĄæŸŕascii, latin-1, utf-8åŠŇutf-16ãĂĆ åIJĺwebåžŤçŤĺçĺŃåžŔäÿ■éĂŽåÿÿéČ¡ä¡£çŤĺçŽĎæŸŕUTF-8ãĂĆ asciiåŕźåžŤäżŐU+0000åĹřU+007FèŇČåŻť åĘĚçŽĎ7ä¡■å■ŮçňęăĂĆ latin-1æŸŕå■ŮèŁĆ0-255åĹřU+0000èĞşU+00FFèŇČåŻť åĘĚUnicodeå■ŮçňęçŽĎçŻť æŐěæŸäåřĎãĂĆ å¡Ş èŕżåŔŰäÿĂäÿłæIJłç§ěçijŰçăĄçŽĎæŰĞæIJňæŮűä¡£çŤĺlatin-1çijŰçăĄæřÿè£IJäÿ■äijŽäžğçŤ§èğčçăĄéŤŹèŕŕăĂĆ ä¡£çŤĺlatin-1çijŰçăĄèŕżåŔŰäÿĂäÿłæŰĞäżűçŽĎæŮűåĂŹäź§èőÿäÿ■èČ¡äžğçŤ§åőŇåĚĺæ■čçąőçŽĎæŰĞæIJňèğčçăĄæŢŕ ä¡ĘæŸŕåőČäź§èČ¡äżŐäÿ■æŔŘåŔŰåĞžèűşåď §åď ŽçŽĎæIJL'çŤĺæŢŕæ■őăĂĆåŘŇæŮűïijŇåęĆæđIJä¡ääźŃåŘ

èőĺèőž

èŕżåĘŹæŰĞæIJňæŰĞäżűäÿĂèĹňæĺěèőšæŸŕæŕŤè¿ČçőĂå■ŢçŽĎăĂĆä¡ĘæŸŕäź§åĞäçĆźæŸŕéIJĂèęĄæş éęŰåĚĹïijŇåIJĺä¿Ňå■ŘçĺŃåžŔäÿ■çŽĎwithèŕ■åŔěçżŹèćňä¡£çŤĺåĹřçŽĎæŰĞäżűåĹŻåżžäžĘäÿĂäÿłäÿŁäÿŃæ ä¡Ę `with` æŐğåĹűåİŮçżŞæİ§æŮűïijŇæŰĞäżűäijŽèĞłåŁĺåĚşéŮ■ãĂĆä¡ääź§åŔŕäžěäÿ■ä¡£çŤĺ `with` èŕ■åŔěïijŇä¡ĘæŸŕè£ŹæŮűåĂŹä¡äåřśå£Ěéążèőřå¿ŮæĽŃåŁĺåĚşéŮ■æŰĞäżűïijŽ

```
f = open('somefile.txt', 'rt')
data = f.read()
f.close()
```

åŔęåď ŰäÿĂäÿłéŮőéćŸæŸŕåĚşäžŐæ■ćèąŇçňęçŽĎèŕĘåĹńéŮőéćŸïijŇåIJĺUnixåŠŇWindowsäÿ■æŸŕäÿ `\n` åŠŇ `\r\n`)ãĂĆ éżŸèőď æČĚåĘţäÿŃïijŇPythonäijŽäžěçż§äÿĂæĺąåijŔåď ĎçŘĘæ■ćèąŇçňęăĂĆ è£ŹçğĺæĺąåijŔäÿŃïijŇåIJĺèŕżåŔŰæŰĞæIJňçŽĎæŮűåĂŹïijŇPythonåŔŕäžěèŕĘåĹńæL'ĂæIJL'çŽĎæŹőéĂŽæ `\n` å■ŮçňęăĂĆ çśżäijijçŽĎïijŇåIJĺè¿ŞåĞžæŮűäijŽåřĘæ■ćèąŇçňę `\n` è¡ňæ■ćäÿżçşżçż§éżŸèőď çŽĎæ■ćèąŇçňęăĂĆ åęĆæđIJä¡ääÿ■åÿŇæIJŻè£ŹçğĺéżŸèőď çŽĎåď ĎçŘĘæŰźåijŔi `open()` åĞ¡æŢŕäijäåĚěåŔĆæŢŕ `newline=''` ïijŇåŕśåČŔäÿŃéĺćè£ŹæäůïijŽ

```
# Read with disabled newline translation
with open('somefile.txt', 'rt', newline='') as f:
    ...
```

äÿžäžĘèŕť æŸŐäÿď èĂĚäźŃéŮť çŽĎåůőåijĆïijŇäÿŃéĺćæĹŚåIJĺUnixæIJžåŹĺäÿŁéĺćèŕżåŔŰäÿĂäÿłWind `hello world!\r\n` ïijŽ

```
>>> # Newline translation enabled (the default)
>>> f = open('hello.txt', 'rt')
>>> f.read()
'hello world!\n'

>>> # Newline translation disabled
>>> g = open('hello.txt', 'rt', newline='')
>>> g.read()
'hello world!\r\n'
>>>
```

æIJĂåŘŐäÿĂäÿłéŮőéćŸåŕśæŸŕæŰĞæIJňæŰĞäżűäÿ■åŔŕèČ¡åĞžçŐŕçŽĎçijŰçăĄéŤŹèŕŕăĂĆ ä¡Ęä¡ăèŕżåŔŰæĹŰèĂĚåĘŹåĚěäÿĂäÿłæŰĞæIJňæŰĞäżűæŮűïijŇä¡ăåŔŕèČ¡äijŽéĄĞåĹřäÿĂäÿłçijŰçăĄæĹŰè

```
>>> f = open('sample.txt', 'rt', encoding='ascii')
>>> f.read()
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "/usr/local/lib/python3.3/encodings/ascii.py", line 26, in␣
↪decode
        return codecs.ascii_decode(input, self.errors)[0]
UnicodeDecodeError: 'ascii' codec can't decode byte 0xc3 in position
12: ordinal not in range(128)
>>>
```

åęĆæđIJåĞžçŐřè£ŹäÿłéŤŹèŕŕïijŇéĂŽåÿÿèąĺçď žä¡ăèŕżåŘŰæŰĞæIJňæŮűæŇĞåőŽçŽĎçijŰçăĄäÿ■æ■čę
ä¡ăæIJĂåě¡äżŤçżĘéŸĚèŕżèŕť æŸŐåźűçąőèőď ä¡ăçŽĎæŰĞäżűçijŰçăĄæŸŕæ■čçąőçŽĎ(æŕŤåęĆä¡£çŤĺUTF-
8èĂŇäÿ■æŸŕLatin-1çijŰçăĄæĹŰåĚűäżŰ)ãĂĆ åęĆæđIJçijŰçăĄéŤŹèŕŕè£ŸæŸŕå■ŸåIJĺçŽĎèŕĺïijŇä¡ăåŘŕäżě
open() åĞ¡æŢřäijăéĂŠäÿĂäÿłåŘŕéĂĽ'çŽĎ errors åŘĆæŢřæĺěåď ĎçŘĘè£ŹäżŻéŤŹèŕŕãĂĆ
äÿŃéĺćæŸŕäÿĂäżŻåď ĎçŘĘåÿÿèğĄéŤŹèŕŕçŽĎæŰźæşŢïijŽ

```
>>> # Replace bad chars with Unicode U+fffd replacement char
>>> f = open('sample.txt', 'rt', encoding='ascii', errors='replace')
>>> f.read()
'Spicy Jalape?o!'
>>> # Ignore bad chars entirely
>>> g = open('sample.txt', 'rt', encoding='ascii', errors='ignore')
>>> g.read()
'Spicy Jalapeo!'
>>>
```

åęĆæđIJä¡ăçżŘåÿÿä¡£çŤĺ errors åŘĆæŢřæĺěåď ĎçŘĘçijŰçăĄéŤŹèŕŕïijŇåŘŕèČ¡äijŽèőľä¡ăçŽĎçŤ§æt
åŕźäžŐæŰĞæIJňåď ĎçŘĘçŽĎéęŰèęĄåŐ§åĹŹæŸŕçąőä£ĺä¡äæĂżæŸŕä¡£çŤĺçŽĎæŸŕæ■čçąőçijŰçăĄãĂĆå¡Ş
8)ãĂĆ

7.2 5.2 æĽŞå■řè¿ŞåĞžèĞşæŰĞäżűäÿ■

éŮőécŸ

ä¡ăæČşåřĘ print() åĞ¡æŢřçŽĎè¿ŞåĞžéĞ■åőŽåŘŚåĹřäÿĂäÿłæŰĞäżűäÿ■åŐżãĂĆ

èğčåĘşæŰźæąĹ

åIJĺ print() åĞ¡æŢřäÿ■æŇĞåőŽ file åĚşéŤőå■ŮåŘĆæŢřïijŇåČŘäÿŃéĺćè£ŹæăůïijŽ

```
with open('d:/work/test.txt', 'wt') as f:
    print('Hello World!', file=f)
```

èőÍèőž

åĚşäžŐè¿ŞåĞžéĞ■åőŽåŘŚåĹřæŰĞäżűäÿ■åřśè£ŹäžŻäžĘãĂĆä¡ĘæŸŕæIJL'äÿĂçĆźèęĄæşĺæĎŘçŽĎåřśæ åęĆæđIJæŰĞäżűæŸŕäžŇè£ŻåĹűæĺąåijŘçŽĎèŕİïijŇæL'Ş å■řåřśäijŽåĞžéŤŹãĂĆ

7.3 5.3 ä¡£çŤĺåĚűäżŰåĹĘéŽŤçňęæĹŰèąŇçżĹæ■ćçňęæL'Ş å■ř

éŮőécŸ

ä¡ăæČşä¡£çŤĺ `print()` åĞ¡æŢřè¿ŞåĞžæŢřæ■őïijŇä¡ĘæŸŕæČşæŤźåŘŸéżŸèőď'çŽĎåĹĘéŽŤçňęæĹŰè

èğčåĘşæŰźæąĹ

åŘŕäżěä¡£çŤĺåIJĺ `print()` åĞ¡æŢřäÿ■ä¡£çŤĺ `sep` åŠŇ `end` åĚşéŤőå■ŮåŘĆæŢřïijŇäżěä¡ăæČşèęĄçŽĎæŰźåijŘè¿ŞåĞžãĂĆæŕŤåęĆïijŽ

```
>>> print('ACME', 50, 91.5)
ACME 50 91.5
>>> print('ACME', 50, 91.5, sep=',')
ACME,50,91.5
>>> print('ACME', 50, 91.5, sep=',', end='!!\n')
ACME,50,91.5!!
>>>
```

ä¡£çŤĺ `end` åŘĆæŢřäź§åŘŕäżěåIJĺè¿ŞåĞžäÿ■çęĄæ■ćæ■ćèąÑãĂĆæŕŤåęĆïijŽ

```
>>> for i in range(5):
...     print(i)
...
0
1
2
3
4
>>> for i in range(5):
...     print(i, end=' ')
...
0 1 2 3 4 >>>
```

èőÍèőž

å¡Şä¡ăæČşä¡£çŤĺéĐçl'žæăijåĹĘéŽŤçňęæĺěè¿ŞåĞžæŢřæ■őçŽĎæŮűåĂŹïijŇçżŹ `print()` åĞ¡æŢřäijăéĂŠäÿĂäÿł `sep` åŘĆæŢřæŸŕæIJĂçőĂå■ŢçŽĎæŰźæąĹãĂĆ æIJL'æŮűåĂŹä¡ăäijŽçIJŃåĹřäÿĂäžŽçĺŃåžŘåŚŸäijŽä¡£çŤĺ `str.join()` æĺěåőŇæĹŘåŘŇæăůçŽĎäžŃæČĚãĂĆæŕŤåęĆïijŽ

```
>>> print(','.join(('ACME','50','91.5')))
ACME,50,91.5
>>>
```

str.join() çŽĎéŮőéćŸåIJÍäžŐåőČäżĚäżĚéĂĆçŤÍäžŐå■ŮçňęäÿšäĂĆè£ŹæĎŘåŚçęİĂä¡äéĂŽåÿÿéIJ

```
>>> row = ('ACME', 50, 91.5)
>>> print(','.join(row))
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
TypeError: sequence item 1: expected str instance, int found
>>> print(','.join(str(x) for x in row))
ACME,50,91.5
>>>
```

ä¡åå¡ŞçĎůåŘŕäżěäÿ■çŤÍéĆčäźĹéžżçČęïijŇåŘłéIJĂèęĄåČŘäÿŃéİćè£ŹæăůåĘŹïijŽ

```
>>> print(*row, sep=',')
ACME,50,91.5
>>>
```

7.4 5.4 èŕżåĘŹå■ŮèŁĆæŢřæ■ő

éŮőéćŸ

ä¡äæČşèŕżåĘŹäžŇè£ŻåĹűæŰĞäżűïijŇæŕŤåęĆåŻ¿çL'ĞïijŇåčŕé§şæŰĞäżűç■L'ç■L'ãĂĆ

èğčåĘşæŰźæąĹ

ä¡£çŤÍæÍąåijŘäÿž rb æĹŰ wb çŽĎ open() åĞ¡æŢřæİěèŕżåŘŰæĹŰåĘŹåĚěäžŇè£ŻåĹűæŢřæ■őãĂĆæŕÌ

```
# Read the entire file as a single byte string
with open('somefile.bin', 'rb') as f:
    data = f.read()

# Write binary data to a file
with open('somefile.bin', 'wb') as f:
    f.write(b'Hello World')
```

åIJÍèŕżåŘŰäžŇè£ŻåĹűæŢřæ■őæŮűïijŇéIJĂèęĄæŇĞæŸŐçŽĎæŸŕæL'ĂæIJL'è£ŤåŻđçŽĎæŢřæ■őéČ¡æ
çśżäijijçŽĎïijŇåIJÍåĘŹåĚěçŽĎæŮűåĂŹïijŇå£Ěéążä£ÌèŕĄåŘĆæŢřæŸŕäżěå■ŮèŁĆå¡ćåijŘåŕźåď'ŰæŽť'éIJšæŢ

èőÍèőž

åIJÍèŕżåŘŰäžŇè£ŻåĹűæŢřæ■őçŽĎæŮűåĂŹïijŇå■ŮèŁĆå■ŮçňęäÿšåŠŇæŰĞæIJňå■ŮçňęäÿšçŽĎèŕ■äźL
çL'źåĹńéIJĂèęĄæşÍæĎŘçŽĎæŸŕïijŇçť'ćåijŢåŠŇè£■äżčåŁÍä¡IJè£ŤåŻđçŽĎæŸŕå■ŮèŁĆçŽĎåĂijèĂŇäÿ■æŸ

```
>>> # Text string
>>> t = 'Hello World'
>>> t[0]
'H'
>>> for c in t:
...     print(c)
...
H
e
l
l
o
...
>>> # Byte string
>>> b = b'Hello World'
>>> b[0]
72
>>> for c in b:
...     print(c)
...
72
101
108
108
111
...
>>>
```

åęĆæđIJä¡ăæČşäżŐäžŇè£ŻåĹűæÍąåijŔçŽĎæŰĞäżűäÿ■èŕżåŔŰæĹŰåĘŹåĚěæŰĞæIJňæŢřæ■őïijŇå£Ěéą

```
with open('somefile.bin', 'rb') as f:
    data = f.read(16)
    text = data.decode('utf-8')

with open('somefile.bin', 'wb') as f:
    text = 'Hello World'
    f.write(text.encode('utf-8'))
```

äžŇè£ŻåĹűI/Oè£ŸæIJL'äÿĂäÿłéšIJäÿžäžžç§ěçŽĎçL'źæĂğåřśæŸŕæŢřçżĎåŠŇCçżŞæđĎä¡ŞçśżåđŃèČ¡çZ

```
import array
nums = array.array('i', [1, 2, 3, 4])
with open('data.bin','wb') as f:
    f.write(nums)
```

è£ŹäÿłéĂĆçŤÍäžŐäżżä¡ŢåőđçŐřäžĘèćńçğřäźŃäÿžâĂİçijŞåĘśæŐěåŔčâĂİçŽĎåŕźèśąïijŇè£Źçğ■åŕźèśąäij
äžŇè£ŻåĹűæŢřæ■őçŽĎåĘŹåĚěåřśæŸŕè£ŹçśżæŞ■ä¡IJäźŃäÿĂãĂĆ

å¿ĹåďŽåŕźèśąè£ŸåĚĄèőÿéĂŽè£Ğä¡£çŤÍæŰĞäżűåŕźèśąçŽĎ `readinto()` æŰźæşŢçŻť'æŐěèŕżåŔŰäžŇè£ŻåĹűæŢřæ■őåĹřåĚűåžŢåśĆçŽĎåĘĚå■Ÿäÿ■åŐżãĂĆæŕŤåęĆïijŽ

```
>>> import array
>>> a = array.array('i', [0, 0, 0, 0, 0, 0, 0, 0])
>>> with open('data.bin', 'rb') as f:
...     f.readinto(a)
...
16
>>> a
array('i', [1, 2, 3, 4, 0, 0, 0, 0])
>>>
```

ä¡ĘæŸŕä¡£çŤĺè£Źçğ■æŁĂæIJŕçŽĎæŮűåĂŹéIJĂèęĄæăijåď'ŰåřŔå£ČïijŇåŻääÿžåőČéĂŽåÿÿåĚůæIJL'åź
åŔŕäżěæ§ěçIJŃ5.9åřŔèŁĆäÿ■åŘęåď'ŰäÿĂäÿłèŕżåŘŰäžŇè£ŻåĹűæŢřæ■őåĹřåŔŕä£őæŤźçijŞåĘšåŇžçŽĎä¿Ńå

7.5 5.5 æŰĞäżűäÿ■å■ŸåIJĺæL'■èČ¡åĘŹåĚě

éŮőécŸ

ä¡ăæČşåČŔäÿĂäÿłæŰĞäżűäÿ■åĘŹåĚěæŢřæ■őïijŇä¡ĘæŸŕåL'■æŘŘå£ĚéążæŸŕè£ŹäÿłæŰĞäżűåIJĺæŰĞ
äź§åřśæŸŕäÿ■åĚĄèőÿèęEçŻŰåůšå■ŸåIJĺçŽĎæŰĞäżűåĘĚåőźãĂĆ

èğčåĘşæŰźæąĹ

åŔŕäżěåIJĺ `open()` åĞ¡æŢřäÿ■ä¡£çŤĺ `x` æĺąåijŘæİěäžčæŻ£ `w` æĺąåijŘçŽĎæŰźæşŢæİěèğčåĘşè£ŹäÿłéŮőécŸãĂĆæŕŤåęĆïijŽ

```
>>> with open('somefile', 'wt') as f:
...     f.write('Hello\n')
...
>>> with open('somefile', 'xt') as f:
...     f.write('Hello\n')
...
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
FileExistsError: [Errno 17] File exists: 'somefile'
>>>
```

åęĆæđIJæŰĞäżűæŸŕäžŇè£ŻåĹűçŽĎïijŇä¡£çŤĺ `xb` æİěäžčæŻ£ `xt`

èőĺèőž

è£ŹäÿĂåřŔèŁĆæijŤçď'žäžĘåIJĺåĘŹæŰĞäżűæŮűéĂŽåÿÿäijŽéĄĞåĹřçŽĎäÿĂäÿłéŮőécŸçŽĎåőŇç¿Őèğ
äÿĂäÿłæŻ£äžčæŰźæąĹæŸŕåĚĹæţŃèŕŢè£ŹäÿłæŰĞäżűæŸŕåŘęå■ŸåIJĺïijŇåČŔäÿŃéĺćè£ŹæăůïijŽ

```
>>> import os
>>> if not os.path.exists('somefile'):
...     with open('somefile', 'wt') as f:
```

(continues on next page)

(continued from previous page)

```
...         f.write('Hello\n')
... else:
...     print('File already exists!')
...
File already exists!
>>>
```

æŸ¿èĂŇæŸŞèğĄïijŇä¡£çŤÍxæŰĞäżűæÍąåijŘæŻť åŁăçőÅå■ŢãĂĆèęĄæşÍæĎŘçŽĎæŸŕxæÍąåijŘæŸŕäÿ
`open()` åĞ¡æŢřçL'źæIJL'çŽĎæL'ľåśŢãĂĆ åIJÍPythonçŽĎæŮğçL'ĹæIJňæĹŰèĂĚæŸŕPythonåőđçŐřçŽĎåžŢ

7.6 5.6 å■Ůçňęäÿšç ŽĎI/OæŞ■ä¡IJ

éŮőéćŸ

ä¡ăæČşä¡£çŤÍæŞ■ä¡IJçśżæŰĞäżűåŕźèśąçŽĎçÍŃåžŘæİěæŞ■ä¡IJæŰĞæIJňæĹŰäžŇè£ŻåĹűå■Ůçňęäÿšã

èğčåĘşæŰźæąĹ

ä¡£çŤÍ `io.StringIO()` åŠŇ `io.BytesIO()` çśżæİěåĹŻåżżçśżæŰĞäżűåŕźèśąæŞ■ä¡IJå■ŮçňęäÿšæŢ

```
>>> s = io.StringIO()
>>> s.write('Hello World\n')
12
>>> print('This is a test', file=s)
15
>>> # Get all of the data written so far
>>> s.getvalue()
'Hello World\nThis is a test\n'
>>>

>>> # Wrap a file interface around an existing string
>>> s = io.StringIO('Hello\nWorld\n')
>>> s.read(4)
'Hell'
>>> s.read()
'o\nWorld\n'
>>>
```

`io.StringIO` åŔłèČ¡çŤÍäžŐæŰĞæIJňãĂĆåęĆæđIJä¡äèęĄæŞ■ä¡IJäžŇè£ŻåĹűæŢřæ■ŏïijŇèęĄä¡£çŤÍ
`io.BytesIO` çśżæİěäżčæŻ£ãĂĆæŕŤåęĆïijŽ

```
>>> s = io.BytesIO()
>>> s.write(b'binary data')
>>> s.getvalue()
b'binary data'
>>>
```

èőÍèőž

å¡Şä¡ăæČşæÍąæŃ§äÿĂäÿłæŹőéĂŽçŽĎæŰĞäżűçŽĎæŮűåĂŹ `StringIO` åŠŇ `BytesIO` çśżæŸŕå¿ĹæIJĽçŤÍçŽĎãĂĆ æŕŤåęĆïijŇåIJÍå■ŢåĚČæţŃèŕŢäÿ■ïijŇä¡ăåŘŕäżěä¡£çŤÍ `StringIO` æĺěåĹŽåżžäÿĂäÿłåŇĚåŘńæţŃèŕŢæŢřæ■őçŽĎçśżæŰĞäżűåŕźèśąïijŇè£ŹäÿłåŕźèśąåŘŕäżěèćńäijăçżŹæ§ŘäÿłåŘĆæŢřäÿżæŹőéĂŽæŰĞäżűåŕźèśąçŽĎåĞ¡æŢřãĂĆ

éIJĂèęĄæşĺæĎŘçŽĎæŸŕïijŇ `StringIO` åŠŇ `BytesIO` åőđä¿Ńåźűæšąæ IJĽæ■čçąőçŽĎæŢť æŢřçśżåđŃçŽĎæŰĞäżűæŘŕè£řçňęãĂĆ åŽăæ■ď ïijŇåőČäżňäÿ■èČ¡åIJĺéĆčäžŻéIJĂèęĄä¡£çŤÍçIJ§åőđçŽĎçşżçż§çžğæŰĞäżűåęĆæŰĞäżűïijŇçőąéĄŞ

7.7 5.7 èŕżåĘŹåŐŃçijľæŰĞäżű

éŮőéćŸ

ä¡ăæČşèŕżåĘŹäÿĂäÿłgzipæĹŰbz2æăijåijŘçŽĎåŐŃçijľæŰĞäżűãĂĆ

èğčåĘşæŰźæąĹ

`gzip` åŠŇ `bz2` æĺąåİŮåŘŕäżěå¿Ĺőźæ ŸŞçŽĎåď ĎçŘĘè£ŹäžŻæŰĞäżűãĂĆ äÿď äÿłæĺąåİŮéČ¡äÿž `open()` åĞ¡æŢřæŘŘä¿ŻäžĘåŘęåď ŰçŽĎåőđçŐřæİěèğčåEşè£ŹäÿłéŮőéćŸãĂĆ æŕŤåęĆïijŇäÿžäžĘäżěæŰĞæIJňå¡ćåijŘèŕżåŘŰåŐŃçijľæŰĞäżűïijŇåŘŕäżěè£ŹæăůåĄŽïijŽ

```
# gzip compression
import gzip
with gzip.open('somefile.gz', 'rt') as f:
    text = f.read()

# bz2 compression
import bz2
with bz2.open('somefile.bz2', 'rt') as f:
    text = f.read()
```

çśżäijijçŽĎïijŇäÿžäžĘåĘŹåĚěåŐŃçijľæŢřæ■őïijŇåŘŕäżěè£ŹæăůåĄŽïijŽ

```
# gzip compression
import gzip
with gzip.open('somefile.gz', 'wt') as f:
    f.write(text)

# bz2 compression
import bz2
with bz2.open('somefile.bz2', 'wt') as f:
    f.write(text)
```

åęĆäÿŁïijŇæL'ĂæIJL'çŽĎI/OæŞ■ä¡IJéČ¡ä¡£çŤÍæŰĞæIJňæĺąåijŘåźűæL'ğèąŇUnicodeçŽĎçijŰçăĄ/èğčç çśżäijijçŽĎïijŇåęĆæđIJä¡ăæČşæŞ■ä¡IJäžŇè£ŻåĹűæŢřæ■őïijŇä¡£çŤÍ `rb` æĹŰèĂĚ `wb` æŰĞäżűæĺąåijŘå■şåŘŕäĂĆ

èőÍèőž

åď ğéČÍåĹĘæČĚåĘţäÿŃèŕżåĘŹåŐŃçijľæŢřæ■őéČ¡æŸŕå¿ĹçőĂå■ŢçŽĎãĂĆä¡ĘæŸŕèęĄæşÍæĎŘçŽĎæŸ
åęĆæđIJä¡ääÿ■æŇĞåőŽæÍąåijŘïijŇéĆčäźĹéžŸèőď çŽĎåřśæŸŕäžŇè£ŻåĹűæÍąåijŘïijŇåęĆæđIJè£ŹæŮűåĂŹç
`gzip.open()` åŠŇ `bz2.open()` æŐěåŘŮèů§åĘĚç¡őçŽĎ `open()` åĞ¡æŢřäÿĂæăůçŽĎåŘĆæŢřïijŇ åŇĚæŃň `encoding`ïijŇ`errors`ïijŇ`newline` ç■L'ç■L'ãĂĆ

å¡ŞåĘŹåĚěåŐŃçijľæŢřæ■őæŮűïijŇåŘŕäžěä¡£çŤÍ `compresslevel` è£ŹäÿłåŘŕéĂL'çŽĎåĚşéŤőå■ŮåŘĆæŢřæÍěæŃĞåőŽäÿĂäÿłåŐŃçijľçžğåĹńãĂĆæŕŤåęĆïijŽ

```
with gzip.open('somefile.gz', 'wt', compresslevel=5) as f:
    f.write(text)
```

éžŸèőď çŽĎç■L'çžğæŸŕ9ïijŇäź§æŸŕæIJĂéńŸçŽĎåŐŃçijľç■L'çžğãĂĆç■L'çžğèűŁä¡ŐæĂğèČ¡èűŁåě¡ïij

æIJĂåŘŐäÿĂçĆźïijŇ `gzip.open()` åŠŇ `bz2.open()` è£ŸæIJL'äÿĂäÿłå¿ĹåřŚèćńç§ěéĄŞçŽĎçL'źæĂğïijŇ åőČäžňåŘŕäžěä¡IJçŤÍåIJÍäÿĂäÿłåůšå■ŸåIJÍåźűäžěäžŇè£

```
import gzip
f = open('somefile.gz', 'rb')
with gzip.open(f, 'rt') as g:
    text = g.read()
```

è£ŹæăůåřśåĚĄèőÿ `gzip` åŠŇ `bz2` æÍąåİŮåŘŕäžěåůěä¡IJåIJÍèőÿåď ŽçśżæŰĞäžűåŕźèśąäÿŁïijŇæŕŤåęĆåě

7.8 5.8 åŻžåőŽåď ğåřŘèőřå¡ŢçŽĎæŰĞäžűè£■äžč

éŮőécŸ

ä¡äæČşåIJÍäÿĂäÿłåŻžåőŽéŢ£åžęèőřå¡ŢæĹŰèĂĚæŢřæ■őåİŮçŽĎéŻĘåŘĹäÿŁè£■äžčïijŇèĂŇäÿ■æŸŕåIJ

èğčåĘşæŰźæąĹ

éĂŽè£ĞäÿŃéİćè£ŹäÿłåřŘæŁĂåůğä¡£çŤÍ `iter` åŠŇ `functools.partial()` åĞ¡æŢřïijŽ

```
from functools import partial

RECORD_SIZE = 32

with open('somefile.data', 'rb') as f:
    records = iter(partial(f.read, RECORD_SIZE), b'')
    for r in records:
        ...
```

è£Źäÿłä¿Ňå■Řäÿ■çŽĎ `records` åŕźèśąæŸŕäÿĂäÿłåŘŕè£■äžčåŕźèśąïijŇåőČäijŽäÿ■æŰ■çŽĎäžğçŤ§åŻž
èęĄæşÍæĎŘçŽĎæŸŕåęĆæđIJæĂżèőřå¡Ţåď ğåřŘäÿ■æŸŕåİŮåď ğåřŘçŽĎæŢť æŢřåĂ■çŽĎèŕİïijŇæIJĂåŘŐäÿĂ

èőÍèőž

`iter()` åĞ¡æŢřæIJL'äÿĂäÿłéšIJäÿžäžžç§ěçŽĎçL'źæĂğåřśæŸŕïijŇåęĆæđIJä¡ăçżŹåőČäijăéĂŠäÿĂäÿłåI
è£Źäÿłè£■äžčåŹÍäijŽäÿĂçŻť'èřČçŤÍäijăåĚěçŽĎåŘŕèřČçŤÍåŕźèśąçŻť'åĹřåőČè£ŤåŻđæăĞèőřåĂijäÿžæ■ćïijŇè£

åIJÍä¿Ňå■Řäÿ■ïijŇ `functools.partial` çŤÍæİěåĹŻåżžäÿĂäÿłæŕŘæňąèćńèřČçŤÍæŮűäżŐæŰĞäżűä
æăĞèőřåĂij `b''` åřśæŸŕå¡ŞåĹřè¿¿æŰĞäżűçżŞåř¿æŮűçŽĎè£ŤåŻđåĂijãĂĆ

æIJĂåŘŐåĘ■æŘŘäÿĂçĆźïijŇäÿŁéİćçŽĎä¿Ňå■Řäÿ■çŽĎæŰĞäżűæŮűäżěäžŇè£ŻåĹűæÍąåijŘæL'ŞåijĂç
åęĆæđIJæŸŕèŕżåŘŰåŻžåőŽåď'ğåřŘçŽĎèőřå¡ŢïijŇè£ŹéĂŽåÿÿæŸŕæIJĂæŹőéĄ■çŽĎæČĚåĘţãĂĆ
èĂŇåŕźäžŐæŰĞæIJňæŰĞäżűïijŇäÿĂèąŇäÿĂèąŇçŽĎèŕżåŘŰ(éżŸèőď'çŽĎè£■äžčèąŇäÿž)æŻť'æŹőéĄ■çĆźã

7.9 5.9 èŕżåŘŰäžŇè£ŻåĹűæŢřæ■őåĹřåŘŕåŘŸçijŞåĘšåŇžäÿ■

éŮőécŸ

ä¡ăæČşçŻť'æŐěèŕżåŘŰäžŇè£ŻåĹűæŢřæ■őåĹřäÿĂäÿłåŘŕåŘŸçijŞåĘšåŇžäÿ■ïijŇèĂŇäÿ■éIJĂèęĄåĄŽäż
æĹŰèĂĚä¡ăæČşåŐ§åIJřä£őæŤźæŢřæ■őåźűåřĘåőČåĘŹåŻđåĹřäÿĂäÿłæŰĞäżűäÿ■åŐżãĂĆ

èğčåĘşæŰźæąĹ

äÿžäžĘèŕżåŘŰæŢřæ■őåĹřäÿĂäÿłåŘŕåŘŸæŢřçżĎäÿ■ïijŇä¡£çŤÍæŰĞäżűåŕźèśąçŽĎ
`readinto()` æŰźæşŢãĂĆæŕŤåęĆïijŽ

```
import os.path

def read_into_buffer(filename):
    buf = bytearray(os.path.getsize(filename))
    with open(filename, 'rb') as f:
        f.readinto(buf)
    return buf
```

äÿŃéİćæŸŕäÿĂäÿłæijŤçď'žè£ŹäÿłåĞ¡æŢřä¡£çŤÍæŰźæşŢçŽĎä¿Ňå■ŘïijŽ

```
>>> # Write a sample file
>>> with open('sample.bin', 'wb') as f:
...     f.write(b'Hello World')
...
>>> buf = read_into_buffer('sample.bin')
>>> buf
bytearray(b'Hello World')
>>> buf[0:5] = b'Hello'
>>> buf
bytearray(b'Hello World')
>>> with open('newsample.bin', 'wb') as f:
...     f.write(buf)
...
```

(continues on next page)

(continued from previous page)

```
11
>>>
```

èőÍèőž

æŰĞäżűåŕźèšąçŽĎ readinto() æŰźæşŢèČ¡èćńçŤÍæİěäÿžéćĎåĚĹåĹĘéĚ■åĘĚå■ŸçŽĎæŢřçżĎåąńåĚ array æĺąåİŮæĹŰ numpy åžŞåĹŻåżžçŽĎæŢřçżĎãĂĆ åŠŇæŹőéĂŽ read() æŰźæşŢäÿ■åŘŇçŽĎæŸŕïijŇ readinto() åąńåĚĚåůšå■ŸåIJÍçŽĎçijŞåĘšåŇžèĂŇäÿ■æŸŕäÿžæŰřåŕźèšąéĞ åŻäæ■ď'ïijŇä¡ššŘŕäżěä¡£çŤÍåőČæİěéĄ£åĚ■åď'ğéĞŘçŽĎåĘĚå■ŸåĹĘéĚ■æŞ■ä¡IJãĂĆ æŕŤåęĆïijŇåęĆæđIJä¡ăèŕżåŘŰäÿĂäÿłçŤśçŻÿåŘŇåď'ğåřŘçŽĎèőřå¡ŢçżĎæĹŘçŽĎäžŇè£ŻåĹűæŰĞäżűæŮűïi

```
record_size = 32 # Size of each record (adjust value)

buf = bytearray(record_size)
with open('somefile', 'rb') as f:
    while True:
        n = f.readinto(buf)
        if n < record_size:
            break
        # Use the contents of buf
        ...
```

åŘęåď'ŰæIJL'äÿĂäÿłæIJL'èűčçL'źæĂğåřśæŸŕ memoryview ïijŇ åőČåŘŕäżěéĂŽè£ĞéŻűåď'■åĹűçŽĎæŰźåijŘåŕźåůšå■ŸåIJÍçŽĎçijŞåĘšåŇžæL'ğèąŇåĹĞçL'ĞæŞ■ä¡IJïijŇçŤŽ

```
>>> buf
bytearray(b'Hello World')
>>> m1 = memoryview(buf)
>>> m2 = m1[-5:]
>>> m2
<memory at 0x100681390>
>>> m2[:] = b'WORLD'
>>> buf
bytearray(b'Hello WORLD')
>>>
```

ä¡£çŤÍ f.readinto() æŮűéIJĂèęĄæşÍæĎŘçŽĎæŸŕïijŇä¡åå£ĚéążæčĂæ§ěåőČçŽĎè£ŤåŻđåĂijïijŇä

åęĆæđIJå■ŮèŁĆæŢřåřŘäžŐçijŞåĘšåŇžåď'ğåřŘïijŇèąÍæŸŐæŢřæ■őèćńæĹłæŰ■æĹŰèĂĚèćńçąť'åÍŘäžĘ

æIJĂåŘŐïijŇçŢŹå£ČèğĆåŕ§åĚűäżŰåĞ¡æŢřåžŞåŠŇæÍąåİŮäÿ■åŠŇ into çŻÿåĚşçŽĎåĞ¡æŢř(æŕŤåęĆ recv_into() ïijŇ pack_into() ç■L')ãĂĆ PythonçŽĎå¿Ĺåď'ŽåĚűäżŰéČÍåĹĘåůšçżŘèČ¡æŤŕæŇĄçŻť'æŐęçŽĎI/OæĹŰæŢřæ■őèő£éŰőæŞ■ä¡IJïijŇè£Źä

åĚşäžŐèğčæđŘäžŇè£ŻåĹűçżŞæđĎåŠŇ memoryviews ä¡£çŤÍæŰźæşŢçŽĎæŻť'énŸçžğä¿Ňå■ŘïijŇèŕůåŘĆèĂČ6.12åřŘèŁĆãĂĆ

7.10 5.10 åĘĚå■ŸæŸăåřĎçŽĎäžŇè£ŻåĹűæŰĞäżű

éŮőéćŸ

ä¡ăæČşåĘĚå■ŸæŸăåřĎäÿĂäÿłäžŇè£ŻåĹűæŰĞäżűåĹřäÿĂäÿłåŘŕåŘŸå■ŮèŁĆæŢřçżĎäÿ■ïijŇçŻőçŽĎåĽ

èğčåĘşæŰźæąĹ

ä¡£çŤÍ mmap æĺąåİŮæİěåĘĚå■ŸæŸăåřĎæŰĞäżűãĂĆäÿŇéİćæŸŕäÿĂäÿłåůěåĚůåĞ¡æŢřïijŇåŘŚä¡ăæijŤçď žäžĘåęĆä¡ŢæĽŞåijĂäÿĂäÿłæŰĞäżűåźűäžěäÿĂçğ■ä¿£æ

```
import os
import mmap

def memory_map(filename, access=mmap.ACCESS_WRITE):
    size = os.path.getsize(filename)
    fd = os.open(filename, os.O_RDWR)
    return mmap.mmap(fd, size, access=access)
```

äÿžäžĘä¡£çŤÍè£ŹäÿłåĞ¡æŢřïijŇä¡ăéIJĂèęĄæIJĽäÿĂäÿłåůšåĹŻåżžåźűäÿŤåĘĚåőźäÿ■äÿžçľžçŽĎæŰĞäż
äÿŇéİćæŸŕäÿĂäÿłä¿Ňå■ŘïijŇæŢŹä¡ăæĂŐæăůåĹİåğŇåĹŻåżžäÿĂäÿłæŰĞäżűåźűåřĘåĚűåĘĚåőźæĽľåĚĚåĹř

```
>>> size = 1000000
>>> with open('data', 'wb') as f:
...     f.seek(size-1)
...     f.write(b'\x00')
...
>>>
```

äÿŇéİćæŸŕäÿĂäÿłåĹľçŤÍ `memory_map()` åĞ¡æŢřçśżåĘĚå■ŸæŸăåřĎæŰĞäżűåĘĚåőźçŽĎä¿Ňå■ŘïijŽ

```
>>> m = memory_map('data')
>>> len(m)
1000000
>>> m[0:10]
b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
>>> m[0]
0
>>> # Reassign a slice
>>> m[0:11] = b'Hello World'
>>> m.close()

>>> # Verify that changes were made
>>> with open('data', 'rb') as f:
... print(f.read(11))
...
b'Hello World'
>>>
```

mmap() è£ŤåŻđçŽĎ mmap åŕźèśąåŘŇæăůäź§åŘŕäżěä¡IJäÿžäÿĂäÿłäÿŁäÿŃæŰĞçőąçŘĘąŹÍæİěä¡£çŤÍïij
è£ŹæŮűąĂŹąžŢąśĆçŽĎæŰĞäżűäijŽèćńèĞłąŁÍąĚşéÚ■ãĂĆæŕŤąęĆïijŽ

```
>>> with memory_map('data') as m:
...     print(len(m))
...     print(m[0:10])
...
1000000
b'Hello World'
>>> m.closed
True
>>>
```

éżŸèőď'æČĚąĘţäÿŃïijŇ memeory_map() åĞ¡æŢřæĽŞąijĂçŽĎæŰĞäżűąŘŇæŮűæŤŕæŇĄèŕżąŠŇąĘŹ
äżżä¡ŢçŽĎä£őæŤźąĘĚąőźéČ¡äijŽąď■ąĹűąŻđąŐ§æİěçŽĎæŰĞäżűäÿ■ãĂĆ
ąęĆæđIJéIJĂèęĄąŘłèŕżçŽĎèő£éŮőæÍąąijŘïijŇąŘŕäżěçżŹąŘĆæŢř access èţŃąĂijäÿž
mmap.ACCESS_READ ãĂĆæŕŤąęĆïijŽ

```
m = memory_map(filename, mmap.ACCESS_READ)
```

ąęĆæđIJä¡äæČşąIJÍæIJňąIJřä£őæŤźæŢřæ■őïijŇä¡ĘæŸŕąŘĹäÿ■æČşąřĘä£őæŤźąĘŹąŻđąĹřąŐ§ąğŃæŰĞ
mmap.ACCESS_COPY ïijŽ

```
m = memory_map(filename, mmap.ACCESS_COPY)
```

èőÍèőž

äÿžäžĘéŽŘæIJžèő£éŮőæŰĞäżűçŽĎąĘĚąőźïijŇä¡£çŤÍ mmap
ąřĘæŰĞäżűæŸäąřĎąĹřąĘĚą■ŸäÿæŸŕäÿĂäÿłéńŸæŢĹąŠŇäijŸéŻĚçŽĎæŰźæşŢãĂĆ
ä¿ŃąęĆïijŇä¡äæŮäéIJĂæĽŞąijĂäÿĂäÿłæŰĞäżűąźűæĽğèąŇąď'ğéĞŘçŽĎ seek() ïijŇ
read() ïijŇ write() èřĆçŤÍïijŇ ąŘłéIJĂèęĄçőĂą■ŢçŽĎæŸąřĎæŰĞäżűąźűä¡£çŤÍąĹĞçĽĞæŞ■ä¡IJèő£é

äÿĂèĹňæİěèőšïijŇ mmap() æĽĂæŽť'éIJšçŽĎąĘĚą■ŸçIJŃäÿŁąŐżąřśæŸŕäÿĂäÿłäžŇè£ŻąĹűæŢřçżĎąŕ
ä¡ĘæŸŕïijŇä¡äąŘŕäżěä¡£çŤÍäÿĂäÿłąĘĚą■ŸèğĘąŻ¿æİěèğčæđŘąĚűäÿ■çŽĎæŢřæ■õăĂĆæŕŤąęĆïijŽ

```
>>> m = memory_map('data')
>>> # Memoryview of unsigned integers
>>> v = memoryview(m).cast('I')
>>> v[0] = 7
>>> m[0:4]
b'\x07\x00\x00\x00'
>>> m[0:4] = b'\x07\x01\x00\x00'
>>> v[0]
263
>>>
```

éIJĂèęĄąijžèřĆçŽĎäÿĂçĆźæŸŕïijŇąĘĚą■ŸæŸąřĎäÿĂäÿłæŰĞäżűąźűäÿ■äijŽąŕijèĞť'æŢť'äÿłæŰĞäżűé
äź§ąřśæŸŕèŕť'ïijŇæŰĞäżűąźűæšąæIJĽèćńąď'■ąĹűąĹřąĘĚą■ŸçijŞą■ŸæĹŰæŢřçżĎäÿ■ãĂĆçŻÿąŘ■ïijŇæŞ■ä¡
ą¡Şä¡äèő£éŮőæŰĞäżűçŽĎäÿ■ąŘŇąŇžą§§æŮűïijŇè£ŹäžŻąŇžą§§çŽĎąĘĚąőźæĽ■æäźæ■õéIJĂèęĄèćńèŕżąŘ
èĂŇéĆčäžŻäżŐæšąèćńèő£éŮőąĹřçŽĎéČĺąĹĘè£ŸæŸŕçŢŹąIJÍçčĄçŻŸäÿŁãĂĆæĽĂæIJĽè£ŹäžŻè£ĞçÍŃæŸ

åęĆæđIJåď Žäÿł Pythonèğčé ĞŁåŹĺåĘĚå■ŸæŸăåřĎåŘŇäÿĂäÿłæŰĞäżűïijŇå¿ŮåĹřçŽĎ mmap åŕźèśąèČ¡åď§èćńçŤĺæİěåIJĺèğčé ĞŁåŹĺçŻť æŐěäžď æ■ćæŢřæ■őãĂĆ äź§åřśæŸŕèŕť ïijŇæĽĂæIJĽèğčé ĞŁåŹĺéČ¡èČ¡åŘŇæŮűèŕżåĘŹæŢřæ■őïijŇåźűäÿŤåĚűäÿ■äÿĂäÿłèğčé ĞŁåŹĺ å¿ĹæŸŐæŸ¿ïijŇè£ŹéĞŇéIJĂèęĄèĂČèŹŚåŘŇæ■ěçŽĎéŮőéćŸãĂĆä¡ĘæŸŕè£Źçğ■æŰźæşŢæIJĽæŮűåĂŹåŘ

è£ŹäÿĂåřŔèŁĆäÿ■åĞ¡æŢřåř¡éĞŔåĘŹå¿Ůå¿ĹéĂŽçŤĺïijŇåŘŇæŮűéĂĆçŤĺäžŐUnixåŠŇWindowsåźşåŔè èęĄæşĺæĎŔçŽĎæŸŕä¡£çŤĺ mmap() åĞ¡æŢřæŮűäijŽåIJĺåžŢåśĆæIJĽäÿĂäžŻåźşåŔřçŽĎåůőåijĆæĂğãĂĆ åŔęåď ŬïijŇè£ŸæIJĽäÿĂäžŽéĂĽéąźåŔŕäžěçŤĺæİěåĹŻåżžåŇ£åŘ■çŽĎåĘĚå■ŸæŸăåřĎåŇžå§§ãĂĆ åęĆæđIJä¡ăåŕźè£ŹäÿłæĎ§åĚť èűčïijŇçąőä£İä¡äżŤçżĘçăŤèŕżäžĘPythonæŰĞæąčäÿ■ è£ŹæŰźéĺćçŽĎåĘĚåőź ãĂĆ

7.11 5.11 æŰĞäżűèůŕå¿ĎåŘ■çŽĎæŞ■ä¡IJ

éŮőécŸ

ä¡äéIJĂèęĄä¡£çŤĺèůŕå¿ĎåŘ■æİěèŐůåŔŰæŰĞäżűåŘ■ïijŇçŻőå¡ŢåŘ■ïijŇçżĺåŕźèůŕå¿Ďç■L'ç■L'ãĂĆ

èğčåĘşæŰźæąĹ

ä¡£çŤĺ os.path æĺąåİŮäÿ■çŽĎåĞ¡æŢřæİěæŞ■ä¡IJèůŕå¿ĎåŘ■ãĂĆ äÿŃéĺćæŸŕäÿĂäÿłäžď äžŠåijŔä¿Ńå■ŘæİěæijŤçď žäÿĂäžŻåĚşéŤőçŽĎçĹźæĂğïijŽ

```
>>> import os
>>> path = '/Users/beazley/Data/data.csv'

>>> # Get the last component of the path
>>> os.path.basename(path)
'data.csv'

>>> # Get the directory name
>>> os.path.dirname(path)
'/Users/beazley/Data'

>>> # Join path components together
>>> os.path.join('tmp', 'data', os.path.basename(path))
'tmp/data/data.csv'

>>> # Expand the user's home directory
>>> path = '~/Data/data.csv'
>>> os.path.expanduser(path)
'/Users/beazley/Data/data.csv'

>>> # Split the file extension
>>> os.path.splitext(path)
('~/Data/data', '.csv')
>>>
```

èőÍèőž

åŕźäžŐäżä¡ŢçŽĎæŰĞäżűåŘ■çŽĎæŞ■ä¡IJïijŇä¡äéČ¡åžŤèŕěä¡£çŤÍ os.path æÍąåİŮïijŇèĂŇäÿ■æŸŕä¡£çŤÍæăĞåĞĘå■ŮçňęäÿšæŞ■ä¡IJæİěæđĎéĂäèĞłåůśçŽĎäżčçăĄãĂĆçL'źåĹńæŸŕäÿžäžĘåŘŕçğżæď■æĂğèĂČèŹŚçŽĎæŮűåĂŹæŻť'åžŤåęĆæ■ď'ïijŇ åŽääÿž os.path æÍąåİŮç§ěéĄŞUnixåŠŇWindowsçşżçż§äźŃéŮť'çŽĎåůőåijĆåźűäÿŤèČ¡åď'§åŘŕéİäåIJřåď'ĎçŘĘçśżäijij Data/data.csv åŠŇ Data\data.csv è£ŹæäůçŽĎæŰĞäżűåŘ■ãĂĆåĚűæňąïijŇä¡äçIJ§çŽĎäÿ■åžŤèŕěæţłèť'źæŮűéŮť'åŐżéĞ■åď'■éĂäè¡őå■ŘãĂĆéĂŽåÿÿæIJĂåě¡æŸŕçŻť'æŐěä¡£

èęĄæşĺæĎŘçŽĎæŸŕ os.path è£ŸæIJL'æŻť'åď'ŽçŽĎåŁ§èČ¡åIJÍè£ŹéĞŇåźűæšąæIJL'åĹŮäÿ¿åĞžæİěã åŘŕäżěæ§ěéŸĚåőŸæŰźæŰĞæąčæİěèŐůåŘŰæŻť'åď'ŽäÿŐæŰĞäżűæţŃèŕŢïijŇçňęåŘůéŞ¿æŐěç■L'çŻÿåĚşçŽ

7.12 5.12 æţŃèŕŢæŰĞäżűæŸŕåŘęå■ŸåIJÍ

éŮőéćŸ

ä¡äæČşæţŃèŕŢäÿĂäÿłæŰĞäżűæĹŰçŻőå¡ŢæŸŕåŘęå■ŸåIJÍãĂĆ

èğčåĘşæŰźæąĹ

ä¡£çŤÍ os.path æÍąåİŮæİěæţŃèŕŢäÿĂäÿłæŰĞäżűæĹŰçŻőå¡ŢæŸŕåŘęå■ŸåIJÍãĂĆæŕŤåęĆïijŽ

```
>>> import os
>>> os.path.exists('/etc/passwd')
True
>>> os.path.exists('/tmp/spam')
False
>>>
```

ä¡äè£ŸèČ¡è£ŻäÿĂæ■ěæţŃèŕŢè£ŹäÿłæŰĞäżűæŮűäżĂäźĹçśżåđŃçŽĎãĂĆ åIJÍäÿŃéİćè£ŹäžŻæţŃèŕŢäÿ■ïijŇåęĆæđIJæţŃèŕŢçŽĎæŰĞäżűäÿ■å■ŸåIJÍçŽĎæŮűåĂŹïijŇçżŞæđIJéČ¡äijŽè

```
>>> # Is a regular file
>>> os.path.isfile('/etc/passwd')
True

>>> # Is a directory
>>> os.path.isdir('/etc/passwd')
False

>>> # Is a symbolic link
>>> os.path.islink('/usr/local/bin/python3')
True

>>> # Get the file linked to
>>> os.path.realpath('/usr/local/bin/python3')
'/usr/local/bin/python3.3'
>>>
```

åęĆæđIJä¡ăè£ŸæČşèŐůåŘŰåĚČæŢřæ■ő(æŕŤåęĆæŰĞäżűåď ğåřŘæĹŰèĂĚæŸŕä£őæŤźæŮěæIJ§)ïijŇäź
os.path æĺąåİŮæİěèğčåĘşïijŽ

```
>>> os.path.getsize('/etc/passwd')
3669
>>> os.path.getmtime('/etc/passwd')
1272478234.0
>>> import time
>>> time.ctime(os.path.getmtime('/etc/passwd'))
'Wed Apr 28 13:10:34 2010'
>>>
```

èőÍèőž

ä¡£çŤÍ os.path æİěè£ŻèąŇæŰĞäżűæţŃèŕŢæŸŕå¿ĹçőĂå■ŢçŽĎãĂĆ
åIJĺåĘŹè£ŹäžŻèĎŽæIJňæŮűïijŇåŘŕèČ¡åŤŕäÿĂéIJĂèęĄæşĺæĎŘçŽĎåřśæŸŕä¡ăéIJĂèęĄèĂČèŹŚæŰĞäżűæİČ

```
>>> os.path.getsize('/Users/guido/Desktop/foo.txt')
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "/usr/local/lib/python3.3/genericpath.py", line 49, in␣
↪getsize
        return os.stat(filename).st_size
PermissionError: [Errno 13] Permission denied: '/Users/guido/
↪Desktop/foo.txt'
>>>
```

7.13 5.13 èŐůåŘŰæŰĞäżűåď źäÿ■çŽĎæŰĞäżűåĹŮèąĺ

éŮőéćŸ

ä¡ăæČşèŐůåŘŰæŰĞäżűçşżçż§äÿ■æ§ŘäÿłçŻőå¡ŢäÿŃçŽĎæĽĂæIJĽæŰĞäżűåĹŮèąĺãĂĆ

èğčåĘşæŰźæąĹ

ä¡£çŤÍ os.listdir() åĞ¡æŢřæİěèŐůåŘŰæ§ŘäÿłçŻőå¡Ţäÿ■çŽĎæŰĞäżűåĹŮèąĺïijŽ

```
import os
names = os.listdir('somedir')
```

çżŞæđIJäijŽè£ŤåŻđçŻőå¡Ţäÿ■æĽĂæIJĽæŰĞäżűåĹŮèąĺïijŇåŇĚæŃňæĽĂæIJĽæŰĞäżűïijŇå■ŘçŻőå¡
åęĆæđIJä¡ăéIJĂèęĄéĂŽè£ĞæŞŘçğ■æŰźåijŘè£Ğæżď æŢřæ■őïijŇåŘŕäżěèĂČèŹŚçżŞåŘĹ
os.path åžŞäÿ■çŽĎäÿĂäžŻåĞ¡æŢřæİěä¡£çŤĺåĹŮèąĺæŐĺåŕijãĂĆæŕŤåęĆïijŽ

```
import os.path

# Get all regular files
names = [name for name in os.listdir('somedir')
        if os.path.isfile(os.path.join('somedir', name))]

# Get all dirs
dirnames = [name for name in os.listdir('somedir')
        if os.path.isdir(os.path.join('somedir', name))]
```

å■Ůçňęäÿšç ŽĎ startswith() åŠŇ endswith() æŰźæşŢåŕźäžŐè£ĞæżďäÿĂäÿłçŻőå¡ŢçŽĎåĘĚåőźäź§æŸŕå¿ĹæIJĽçŤĺçŽĎãĂĆæŕŤåęĆïijŽ

```
pyfiles = [name for name in os.listdir('somedir')
            if name.endswith('.py')]
```

åŕźäžŐæŰĞäżűåŘ■çŽĎåŇźéĚ■ïijŇä¡ăåŘŕèČ¡äijŽèĂČèŹŚä¡£çŤÍ glob æĹŰ fnmatch æĺąåİŮãĂĆæŕŤåęĆïijŽ

```
import glob
pyfiles = glob.glob('somedir/*.py')

from fnmatch import fnmatch
pyfiles = [name for name in os.listdir('somedir')
            if fnmatch(name, '*.py')]
```

èőÍèőž

èŐůåŘŰçŻőå¡Ţäÿ■çŽĎåĹŮèąĺæŸŕå¿Ĺåőźæ ŸŞçŽĎïijŇä¡ĘæŸŕåĚűè£ŤåŻđçżŞæđIJåŘłæŸŕçŻőå¡Ţäÿ■åĉ
åęĆæđIJä¡äè£ŸæČşèŐůåŘŰåĚűäžŰçŽĎåĚČä£ąæĄŕïijŇæŕŤåęĆæŰĞäżűåď ğåřŘïijŇä£őæŤźæŮűéŮť ç■Ľç■
ä¡äæĹŰèőÿè£ŸéIJĂèęĄä¡£çŤĺåĹř os.path æĺąåİŮäÿ■çŽĎåĞ¡æŢřæĹŰçİĂ os.stat()
åĞ¡æŢřæİěæŤűéŻĘæŢřæ■őãĂĆæŕŤåęĆïijŽ

```
# Example of getting a directory listing

import os
import os.path
import glob

pyfiles = glob.glob('*.py')

# Get file sizes and modification dates
name_sz_date = [(name, os.path.getsize(name), os.path.
↪getmtime(name))
                for name in pyfiles]
for name, size, mtime in name_sz_date:
    print(name, size, mtime)
```

(continues on next page)

(continued from previous page)

```
# Alternative: Get file metadata
file_metadata = [(name, os.stat(name)) for name in pyfiles]
for name, meta in file_metadata:
    print(name, meta.st_size, meta.st_mtime)
```

æIJĂåŘŐè£ŸæIJL'äÿĂçĆźèęĄæşĺæĎŘçŽĎåřśæŸřïijŇæIJL'æŮűåĂŹåIJĺåď'ĎçŘĘæŰĞäżűåŘ■çijŰçăĄé
éĂŽåÿÿæĺěèőšïijŇåĞ¡æŢř `os.listdir()` è£ŤåŻđçŽĎåőđä¡ŞåĹŮèąĺäijŽæăźæ■őçşżçż§éżŸèőď'çŽĎæŰĞä
ä¡ĘæŸřæIJL'æŮűåĂŹäź§äijŽçćřåĹřäÿĂäžŻäÿ■èČ¡æ■čåÿÿèğčçăĄçŽĎæŰĞäżűåŘ■ãĂĆ
åĚşäžŐæŰĞäżűåŘ■çŽĎåď'ĎçŘĘéŮőécŸïijŇåIJĺ5.14åŠŇ5.15åřŘèŁĆæIJL'æŻť'èŕęçżĘçŽĎèőšèğčãĂĆ

7.14 5.14 å£¡çŢĕæŰĞäżűåŘ■çijŰçăĄ

éŮőécŸ

ä¡ăæČşä¡£çŤĺåŐ§åğŃæŰĞäżűåŘ■æL'ğèąŇæŰĞäżűçŽĎI/OæŞ■ä¡IJïijŇäź§åřśæŸřèŕť'æŰĞäżűåŘ■åźűæš

èğčåĘşæŰźæąĹ

éżŸèőď'æČĚåĘţäÿŃïijŇæL'ĂæIJL'çŽĎæŰĞäżűåŘ■éČ¡äijŽæăźæ■ő `sys.getfilesystemencoding()` è£ŤåŻđçŽĎæŰĞæIJňçijŰçăĄæĺěçijŰçăĄæĹŰèğčçăĄãĂĆæŕŤåęĆïijŽ

```
>>> sys.getfilesystemencoding()
'utf-8'
>>>
```

åęĆæđIJåŻääÿżæ§Řçğ■åŐ§åŻää¡ăæČşå£¡çŢĕè£Źçğ■çijŰçăĄïijŇåŘŕäżĕä¡£çŤĺäÿĂäÿłåŐ§åğŃå■ŮèŁĆå

```
>>> # Wrte a file using a unicode filename
>>> with open('jalape\xf1o.txt', 'w') as f:
...     f.write('Spicy!')
...
6
>>> # Directory listing (decoded)
>>> import os
>>> os.listdir('.')
['jalapeÃśo.txt']

>>> # Directory listing (raw)
>>> os.listdir(b'.') # Note: byte string
[b'jalapen\xcc\x83o.txt']

>>> # Open file with raw filename
>>> with open(b'jalapen\xcc\x83o.txt') as f:
...     print(f.read())
...
```

(continues on next page)

(continued from previous page)

```
Spicy!
>>>
```

æ■čåęĆä¡ăæĽĂèğĄïijŇåIJÍæIJĂåŘŐäÿď äÿłæŞ■ä¡IJäÿ■ïijŇå¡Şä¡ăçżŹæŰĞäżűçŻÿåĚşåĞ¡æŢřåęĆ
`open()` åŠŇ `os.listdir()` äijăéĂŠå■ŮèŁĆå■ŮçňęäÿšæŮűïijŇæŰĞäżűåŘ■çŽĎåď ĎçŘĘæŰźåijŘäijŽçĺ

èőĺèőž

éĂŽåÿÿæİěèőšïijŇä¡ääÿ■éIJĂèęĄæŃĚå£ČæŰĞäżűåŘ■çŽĎçijŰçăĄåŠŇèğčçăĄïijŇæŹőéĂŽçŽĎæŰĞäż
ä¡ĘæŸŕïijŇæIJĽäžŻæŞ■ä¡IJçşżçż§åĚĄèőÿçŤĺæĹůéĂŽè£ĞåĄűçĎűæĹŰæĄűæĎŘæŰźåijŘåŐżåĹŻåżžåŘ■å■
è£ŹäžŻæŰĞäżűåŘ■åŘŕèČ¡äijŽçěđçğŸåIJřäÿ■æŰ■éĆčäžŻéIJĂèęĄåď ĎçŘĘåď ğéĞŘæŰĞäżűçŽĎPythonçĺŃ

èŕżåŘŰçŽőå¡ŢåźűéĂŽè£ĞåŐ§åğŃæIJłèğčçăĄæŰźåijŘåď ĎçŘĘæŰĞäżűåŘ■åŘŕäżěæIJĽæŢĹçŽĎéĄ£å
ář¡çőąè£ŹæăůäijŽåÿęæİěäÿĂåőŽçŽĎçijŰçĺŃéŽ¿åžęãĂĆ

åĚşäžŐæĽŞå■řäÿ■åŘŕèğčçăĄçŽĎæŰĞäżűåŘ■ïijŇèŕůåŘĆèĂČ5.15åřŘèŁĆãĂĆ

7.15 5.15 æĽŞå■řäÿ■åŘĹæşŢçŽĎæŰĞäżűåŘ■

éŮőécŸ

ä¡ăçŽĎçĺŃåžŘèŐůåŘŰäžĘäÿĂäÿłçŻőå¡Ţäÿ■çŽĎæŰĞäżűåŘ■åĹŮèąĺïijŇä¡ĘæŸŕå¡ŞåőČèřŢçİĂåŐżæĽŞ
åĞżçŐřäžĘ `UnicodeEncodeError` åijĆåÿÿåŠŇäÿĂæİąåěĞæĂłçŽĎæűĹæĄŕâĂŤâĂŤ
`surrogates not allowed` ãĂĆ

èğčåĘşæŰźæąĹ

å¡ŞæĽŞå■řæIJłç§ěçŽĎæŰĞäżűåŘ■æŮűïijŇä¡£çŤĺäÿŃéİćçŽĎæŰźæşŢåŘŕäżěéĄ£åĚ■è£ŹæăůçŽĎéŤŹè

```
def bad_filename(filename):
    return repr(filename)[1:-1]

try:
    print(filename)
except UnicodeEncodeError:
    print(bad_filename(filename))
```

èőĺèőž

è£ŹäÿĂåřŘèŁĆèőĺèőžçŽĎæŸŕåIJĺçijŰåĘŹå£Ěéążåď ĎçŘĘæŰĞäżűçşżçż§çŽĎçĺŃåžŘæŮűäÿĂäÿłäÿ■åď
éżŸèőď æČĚåĘţäÿŃïijŇPythonåĄĞåőŽæĽĂæIJĽæŰĞäżűåŘ■éČ¡åůšçżŘæăźæ■ő
`sys.getfilesystemencoding()` çŽĎåĂijçijŰçăĄè£ĞäžĘãĂĆ
ä¡ĘæŸŕïijŇæIJĽäÿĂäžŻæŰĞäżűçşżçż§åźűæšąæIJĽåijžåĹűèęĄæśĆè£ŹæăůåĄŽïijŇåŻăæ■ď åĚĄèőÿåĹŻåžžæ
è£Źçğ■æČĚåĘţäÿ■åď łåÿÿèğĄïijŇä¡ĘæŸŕæĂżäijŽæIJĽäžŽçŤĺæĹůåĘŠéŹľè£ŹæăůåĄŽæĹŰèĂĚæŸŕæŮăæĎ

åŘŕèČ¡æŸŕåIJĺäÿĂäÿłæIJL'çijžéŹůçŽĎäžčçăĄäÿ■çżŹ `open()` åĞ¡æŢřäijăéĂŠäžĘäÿĂäÿłäÿ■åŘĹèğĎèŇČçŽĎæŰĞäžűåŘ■)ãĂĆ

å¡ŞæL'ğèąŇçśżäijij `os.listdir()` è£ŹæăůçŽĎåĞ¡æŢřæŮűïijŇè£ŹäžŻäÿ■åŘĹèğĎèŇČçŽĎæŰĞäžűäÿĂæŰźéİćïijŇåőČäÿ■èČ¡äżĚäżĚåŘłæŸŕäÿćåijČè£ŹäžŻäÿ■åŘĹæăijçŽĎåŘ■å■ŮãĂĆèĂŇåŘęäÿĂæŰźéİćïij Pythonåŕźè£ŹäÿłéŮőéćŸçŽĎèğčåĘşæŰźæąĹæŸŕäżŐæŰĞäžűåŘ■äÿ■èŐůåŘŰæIJłèğčçăĄçŽĎå■ŮèŁĆåĂijæ `\xhh` åźűåřĘåőČæŸăåřĎæĹŘUnicodeå■Ůçňę `\udchh` èąĺçď'žçŽĎæL'ĂèřŞçŽĎâĂİäżčçŘĘçijŰçăĄâĂİãĂĆ äÿŇéİćäÿĂäÿłä¿Ňå■ŘæijŤçď'žäžĘå¡ŞäÿĂäÿłäÿ■åŘĹæăijçŻőå¡ŢåĹŮèąĺäÿ■åŘńæIJL'äÿĂäÿłæŰĞäžűåŘ■äÿžb 1èĂŇäÿ■æŸŕUTF-8çijŰçăĄ)æŮűçŽĎæăůå■ŘïijŽ

```
>>> import os
>>> files = os.listdir('.')
>>> files
['spam.py', 'b\udce4d.txt', 'foo.txt']
>>>
```

åęĆæđIJä¡ăæIJL'äžčçăĄéIJĂèęĄæŞ■ä¡IJæŰĞäžűåŘ■æĹŰèĂĚåřĘæŰĞäžűåŘ■äijăéĂŠçżŹ `open()` è£ŹæăůçŽĎåĞ¡æŢřïijŇäÿĂåĹĞéČ¡èČ¡æ■čåÿÿåůëä¡IJãĂĆ åŘłæIJL'å¡Ŝä¡ăæČşèęĄè¿ŞåĞžæŰĞäžűåŘ■æŮűæL'■äijŽçćřåĹřäžŻéžżçČę(æŕŤåęĆæL'Ŝå■řè¿ŞåĞžåĹřåśŘåźŤ çL'źåĹńçŽĎïijŇå¡Ŝä¡ăæČşæL'Ŝå■řäÿŁéIćçŽĎæŰĞäžűåŘ■åĹŮèąĺæŮűïijŇä¡ăçŽĎçĺŇåžŘåřśäijŽåť'l'æžČïijŽ

```
>>> for name in files:
...     print(name)
...
spam.py
Traceback (most recent call last):
    File "<stdin>", line 2, in <module>
UnicodeEncodeError: 'utf-8' codec can't encode character '\udce4' in
position 1: surrogates not allowed
>>>
```

çĺŇåžŘåť'l'æžČçŽĎåŐ§åŻäářśæŸŕå■Ůçňę `\udce4` æŸŕäÿĂäÿłéIđæşŢçŽĎUnicodeå■ŮçňęăĂĆ åőČåĚűåőđæŸŕäÿĂäÿłèćńçğřäÿžäżčçŘĘå■ŮçňęåŕźçŽĎåŘŇå■ŮçňęçżĎåŘĹçŽĎåŘŐå■ŁéČ çŤśäžŐçijžářŚäžĘåĹ'■å■ŁéČĺåĹĘïijŇåŻäæ■ď'åőČæŸŕäÿłéIđæşŢçŽĎUnicodeăĂĆ æL'Ääżěïij ŇåŤŕäÿĂèČ¡æĹŘåŁ§è¿ŞåĞžçŽĎæŰźæşŢářśæŸŕå¡ŞéĄĞåĹřäÿ■åŘĹæşŢæŰĞäžűåŘ■æŮűéĞĞåŘŰ æŕŤåęĆåŘŕäżěářĘäÿŁè£řäżčçăĄä£őæŤźåęĆäÿŇïijŽ

```
>>> for name in files:
... try:
...     print(name)
... except UnicodeEncodeError:
...     print(bad_filename(name))
...
spam.py
b\udce4d.txt
foo.txt
>>>
```

åIJĺ `bad_filename()` åĞ¡æŢřäÿ■æĂŐæăůåď'Ďç¡őåŘŰåĘşäžŐä¡äèĞłåůśăĂĆ åŘęåď'ŰäÿĂäÿłéĂL'æŃl'ářśæŸŕéĂŽè£Ğæ§Řçğ■æŰźåijŘéĞ■æŰřçijŰçăĄïijŇçď'žä¿ŃåęĆäÿŇïijŽ

```
def bad_filename(filename):
    temp = filename.encode(sys.getfilesystemencoding(), errors=
↪'surrogateescape')
    return temp.decode('latin-1')
```

èŕŚèĂĚæşÍ:

```
surrogateescape:
è£Źçğ■æŸŕPythonåIJÍçżIåd'ğéČÍåĹĘéİćåŘŚOSçŽĎAPIäÿ■æL'Ää¡£çŤÍçŽĎéŤźèŕŕåd'ĎçŘĘåźÍïi
åőČèČ¡äżěäÿĂçğ■äijŸéŽĚçŽĎæŰźåijŔåd'ĎçŘĘçŤśæŞ■ä¡IJçşżçż§æŔŘä¿ŻçŽĎæŢřæ■őçŽĎçijŰçăĄé
åIJÍèğčçăĄåĞžéŤźæŮűäijŽåřĘåĞžéŤźå■ŮèŁĆå■ŸåĆÍåĹřäÿĂäÿłå¿ĹåřŚèćńä¡£çŤÍåĹřçŽĎUnicode
åIJÍçijŰçăĄæŮűåřĘéĆčäžźéŽŘèŮŔåĂijåŔĹè£ŸåŐ§åžđåŐ§åĚÍèğčçăĄåd'śèt'ěçŽĎå■ŮèŁĆåžŔåĹŮ
åőČäÿ■äżĚåŕźäžŐOS␣
↪APIéİđåÿÿæIJL'çŤÍïijŇäź§èČ¡å¿ĹåőźæŸŞçŽĎåd'ĎçŘĘåĚűäżŰæČĚåĘţäÿŃçŽĎçijŰçăĄéŤźèŕŕã
```

ä¡£çŤÍè£ŹäÿłçL'ĹæIJňäžğçŤ§çŽĎè¿ŞåĞžåęĆäÿŃïijŽ

```
>>> for name in files:
...     try:
...         print(name)
...     except UnicodeEncodeError:
...         print(bad_filename(name))
...
spam.py
bÃd'd.txt
foo.txt
>>>
```

è£ŹäÿĂåřŔèŁĆäÿżécŸåŔŕèČ¡äijŽèćńåd'ğéČÍåĹĘèŕżèĂĚæL'Ăå£¡çŢěãĂĆä¡ĘæŸŕåęĆæđIJä¡ăåIJÍçijŰåĘ
åŕśå£Ĕéążå¿ŮèĂČèŹŚåĹřè£ŹäÿłãĂĆåŘęåĹŹä¡ăåŔŕèČ¡äijŽåIJÍæ§ŘäÿłåŚÍæIJńèćńåŔńåĹřåŁđåĚňåőd'åŐżèřČ

7.16 5.16 åćđåŁăæĹŰæŤźåŔŸåůšæL'ŞåijĂæŰĞäżűçŽĎçijŰçăĄ

éŮőéćŸ

ä¡ăæČşåIJÍäÿ■åĚşéŮ■äÿĂäÿłåůšæL'ŞåijĂçŽĎæŰĞäżűåL'■æŔŘäÿŃåćđåŁăæĹŰæŤźåŔŸåőČçŽĎUnico

èğčåĘşæŰźæąĹ

åęĆæđIJä¡ăæČşçżŹäÿĂäÿłäżěäžŇè£ŻåĹűæÍąåijŔæL'ŞåijĂçŽĎæŰĞäżűæůżåŁăUnicodeçijŰçăĄ/èğčçăĄ
åŔŕäżěä¡£çŤÍ `io.TextIOWrapper()` åŕźèśąåŇĚèčĚåőČãĂĆæŕŤåęĆïijŽ

```
import urllib.request
import io

u = urllib.request.urlopen('http://www.python.org')
```

(continues on next page)

(continued from previous page)

```
f = io.TextIOWrapper(u, encoding='utf-8')
text = f.read()
```

åęĆæđIJä¡äæČşä£őæŤźäÿĂäÿłåůšçżŕæĽŞåijĂçŽĎæŰĞæIJňæĺąåijŕçŽĎæŰĞäżűçŽĎçijŰçăĄæŰźåijŕ detach() æŰźæşŢçğżéŹďæŐĽåůšå■ŸåIJĺçŽĎæŰĞæIJňçijŰçăĄåśĆïijŇ åźűä¡£çŤĺæŰřçŽĎçijŰçăĄæŰźåijŕäżčæŽ£ãĂĆäÿŃéĺćæŸŕäÿĂäÿłåIJĺ sys.stdout äÿŁä£őæŤźçijŰçăĄæŰźåijŕçŽĎä¿Ńå■ŘïijŽ

```
>>> import sys
>>> sys.stdout.encoding
'UTF-8'
>>> sys.stdout = io.TextIOWrapper(sys.stdout.detach(), encoding=
↪'latin-1')
>>> sys.stdout.encoding
'latin-1'
>>>
```

è£ŹæăůåĄŽåŕŕèČ¡äijŽäÿ■æŰ■ä¡äçŽĎçżĹçńŕïijŃè£ŹéĞŃäżĚäżĚæŸŕäÿżäżĘæijŤçďżèĂŃåůšãĂĆ

èőĺèőž

I/Oçşżçż§çŤśäÿĂçşżåĹŮçŽĎåśĆæňąæđĎåżżèĂŃæĹŘãĂĆä¡ääŕäżěèŕŢçĺĂè£ŘèąŃäÿŃéĺćè£ŹäÿłæŞ■ä

```
>>> f = open('sample.txt','w')
>>> f
<_io.TextIOWrapper name='sample.txt' mode='w' encoding='UTF-8'>
>>> f.buffer
<_io.BufferedWriter name='sample.txt'>
>>> f.buffer.raw
<_io.FileIO name='sample.txt' mode='wb'>
>>>
```

åIJĺè£Źäÿłä¿Ńå■Řäÿ■ïijŃio.TextIOWrapper æŸŕäÿĂäÿłçijŰçăĄåŠŃèğčçăĄUnicodeçŽĎæŰĞæIJňåďĎçŘĘäśĆïijŃ io.BufferedWriter æŸŕäÿĂäÿłåďĎçŘĘäžŃè£ŻåĹűæŢŕæ■őçŽĎåÿęçijŞåĘšçŽĎI/OåśĆïijŃ io.FileIO æŸŕäÿĂäÿłèąĺçďżæŞ■ä¡IJçşżçż§åžŢåśĆæŰĞäżűæŘŕè£řçňęçŽĎåŐ§åğŃæŰĞäżűãĂĆ åćđåŁäæĹŰæŤźåŘŸæŰĞæIJňçijŰçăĄäijŽæűĽåŘŁåćđåŁäæĹŰæŤźåŘŸæIJĂäÿŁéĺćçŽĎ io.TextIOWrapper åśĆãĂĆ

äÿĂèĹňæĺěèőšïijŃåČŘäÿŁéĺćä¿Ńå■Řè£ŹæăůéĂŽè£ĞèŐ£éŮőåśđæĂğåĂijæĺěçŻťæŐěæŞ■ä¡IJäÿ■åŘŃç ä¿ŃåęĆïijŃåęĆæđIJä¡äèŕŢçĺĂä¡£çŤĺäÿŃéĺćè£ŹæăůçŽĎæŁĂæIJŕæŤźåŘŸçijŰçăĄçIJŃçIJŃäijŽåŘŚçŤ§äżĂä

```
>>> f
<_io.TextIOWrapper name='sample.txt' mode='w' encoding='UTF-8'>
>>> f = io.TextIOWrapper(f.buffer, encoding='latin-1')
>>> f
<_io.TextIOWrapper name='sample.txt' encoding='latin-1'>
>>> f.write('Hello')
```

(continues on next page)

(continued from previous page)

```
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
ValueError: I/O operation on closed file.
>>>
```

çżŞæđIJåĞžéŤŹäžĘïijŇåŻääÿžfçŽĎåŐ§åğŃåĂijåůšçżŘèćńçăť åİŘäžĘåźűåĚşéŮ■äžĘåžŢåśÇçŽĎæŰĞäż

detach() æŰźæşŢäijŽæŰ■åijĂæŰĞäżűçŽĎæIJĂéąűåśĆåźűè£ŤåŻđçňňäžŇåśĆïijŇäźŃåŘŐæIJĂéąűå

```
>>> f = open('sample.txt', 'w')
>>> f
<_io.TextIOWrapper name='sample.txt' mode='w' encoding='UTF-8'>
>>> b = f.detach()
>>> b
<_io.BufferedWriter name='sample.txt'>
>>> f.write('hello')
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
ValueError: underlying buffer has been detached
>>>
```

äÿĂæŮęæŰ■åijĂæIJĂéąűåśĆåŘŐïijŇä¡ăåřśåŘŕäžěçżŹè£ŤåŻđçżŞæđIJæůżåŁääÿĂäÿłæŰřçŽĎæIJĂéąűå

```
>>> f = io.TextIOWrapper(b, encoding='latin-1')
>>> f
<_io.TextIOWrapper name='sample.txt' encoding='latin-1'>
>>>
```

åř¡çőąåůšçżŘåŘŚä¡ăæijŤçď žäžĘæŤźåŘŸçijŰçăĄçŽĎæŰźæşŢïijŇ
ä¡ĘæŸŕä¡ăè£ŸåŘŕäžěåĹľçŤĺè£Źçğ■æŁĂæIJŕæİěæŤźåŘŸæŰĞäžűèąŇåď ĎçŘĘãĂĄéŤŹèŕŕæIJžåĹűäžěåŘŁæ

```
>>> sys.stdout = io.TextIOWrapper(sys.stdout.detach(), encoding=
↪'ascii',
...                                errors='xmlcharrefreplace')
>>> print('Jalape\u00f1o')
Jalape&#241;o
>>>
```

æşĺæĎŘäÿŃæIJĂåŘŐè¿ŞåĞžäÿ■çŽĎéİđASCIIå■Ůçňę Ãś æŸŕåęĆä¡Ţèćń ñ åŘŰäžčçŽĎãĂĆ

7.17 5.17 åřĘå■ŮèŁĆåĘŹåĚěæŰĞæIJňæŰĞäžű

éŮőécŸ

ä¡ăæČşåIJĺæŰĞæIJňæĺąåijŘæĽŞåijĂçŽĎæŰĞäżűäÿ■åĘŹåĚěåŐ§åğŃçŽĎå■ŮèŁĆæŢřæ■õãĂĆ

èğčåĘşæŰźæąĹ

åřĘå■ŮèŁĆæŢřæ■őçŻťæŐěåĘŹåĚěæŰĞäżűçŽĎçijŞåĘšåŇžå■şåŘŕïijŇä¿ŃåęĆïijŽ

```
>>> import sys
>>> sys.stdout.write(b'Hello\n')
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
TypeError: must be str, not bytes
>>> sys.stdout.buffer.write(b'Hello\n')
Hello
5
>>>
```

çśżäijijçŽĎïijŇèČ¡åď§éĂŽè£ĞèŕżåŘŰæŰĞæIJňæŰĞäżűçŽĎ buffer åśđæĂğæĬěèŕżåŘŰäžŇè£ŻåĹűæŢřæ■őãĂĆ

èőĺèőž

I/Oçşżçż§äżěåśĆçžğçżŞæđĎçŽĎå¡ćåijŘæđĎåżžèĂŇæĹŘãĂĆ
æŰĞæIJňæŰĞäżűæŸŕéĂŽè£ĞåIJĺäÿĂäÿłæŃěæIJL'çijŞåĘšçŽĎäžŇè£ŻåĹűæĺąåijŘæŰĞäżűäÿŁåćđåŁääÿĂäÿ
buffer åśđæĂğæŇĞåŘŚåŕźåžŤçŽĎåžŢåśĆæŰĞäżűãĂĆåęĆæđIJä¡äçŻťæŐěèő£éŮőåőĆçŽĎèŕĺåřśäijŽçżŢè

æIJňåřŘèŁĆä¿Ńå■ŘåśŢçď'žçŽĎ sys.stdout åŘŕèČ¡çIJŃèţůæĬěæIJL'çĆźçL'źæőŁãĂĆ
éżŸèőď'æČĚåĘţäÿŃïijŇsys.stdout æĂżæŸŕäżěæŰĞæIJňæĺąåijŘæL'ŞåijĂçŽĎãĂĆ
ä¡ĘæŸŕåęĆæđIJä¡äåIJĺåĘŹäÿĂäÿłéIJĂèęĄæL'Şå■řäžŇè£ŻåĹűæŢřæ■őåĹřæăĞåĞĘè¿ŞåĞžçŽĎèĎŽæIJňçŽĎ

7.18 5.18 åřĘæŰĞäżűæŘŘè£řçňęåŇĚèčĚæĹŘæŰĞäżűåŕźèśą

éŮőécŸ

ä¡äæIJL'äÿĂäÿłåŕźåžŤäžŐæŞ■ä¡IJçşżçż§äÿŁäÿĂäÿłåůšæL'ŞåijĂçŽĎI/OéĂŽéĄŞ(æŕŤåęĆæŰĞäżűãĂĄç
ä¡äæČşåřĘåőČåŇĚèčĚæĹŘäÿĂäÿłæŻťénŸåśĆçŽĎPythonæŰĞäżűåŕźèśąãĂĆ

èğčåĘşæŰźæąĹ

äÿĂäÿłæŰĞäżűæŘŘè£řçňęåŠŇäÿĂäÿłæL'ŞåijĂçŽĎæŹőéĂŽæŰĞäżűæŸŕäÿ■äÿĂæăůçŽĎãĂĆ
æŰĞäżűæŘŘè£řçňęäżĚäżĚæŸŕäÿĂäÿłçŤśæŞ■ä¡IJçşżçż§æŇĞåőŽçŽĎæŢťæŢřïijŇçŤĺæĬěæŇĞäżčæ§Řäÿłçş
åęĆæđIJä¡äçćřåůğæIJL'è£ŹäźĹäÿĂäÿłæŰĞäżűæŘŘè£řçňęïijŇä¡ääŘŕäżěéĂŽè£Ğä¡£çŤĺ
open() åĞ¡æŢřæĬěåřĘåĚűåŇĚèčĚäÿžäÿĂäÿłPythonçŽĎæŰĞäżűåŕźèśąãĂĆ
ä¡ääżĚäżĚåŘłéIJĂèęĄä¡£çŤĺè£ŹäÿłæŢťæŢřåĂijçŽĎæŰĞäżűæŘŘè£řçňęä¡IJäÿžçňňäÿĂäÿłåŘĆæŢřæĬěäżčæŽ

```
# Open a low-level file descriptor
import os
fd = os.open('somefile.txt', os.O_WRONLY | os.O_CREAT)
```

(continues on next page)

(continued from previous page)

```
# Turn into a proper file
f = open(fd, 'wt')
f.write('hello world\n')
f.close()
```

å¡ŞénŸåśĆçŽĎæŰĞäżűářźèśąèćnåĚşéŮ■æĹŰèĂĚçăť åİŘçŽĎæŮűåĂŹïijŇåžŢåśĆçŽĎæŰĞäżűæŘŘè£ì åęĆæđIJè£Źäÿłåźűäÿ■æŸŕä¡ăæČşèęĄçŽĎçżŞæđIJïijŇä¡ăåŘŕäžěçżŹ `open()` åĞ¡æŢřäijăéĂŠäÿĂäÿłåŘŕéĂĽçŽĎ `colsefd=False` ãĂĆæŕŤåęĆïijŽ

```
# Create a file object, but don't close underlying fd when done
f = open(fd, 'wt', closefd=False)
...
```

èőĺèőž

åIJÍUnixçşżçż§äÿ■ïijŇè£ŹçğĞ■åŇĚèčĚæŰĞäżűæŘŘè£řçňęçŽĎæŁĂæIJŕåŘŕäžěå¿ĹæŰźä¿£çŽĎåřĘäÿĂä åęĆçőąéĄŞãĂĄåěŮæŐěå■Ůç■Ľ ãĂĆäÿ¿ä¿ŃæĺěèőšïijŇäÿŃéĺćæŸŕäÿĂäÿłæŞ■ä¡IJçőąéĄŞçŽĎä¿Ňå■ŘïijŽ

```
from socket import socket, AF_INET, SOCK_STREAM

def echo_client(client_sock, addr):
    print('Got connection from', addr)

    # Make text-mode file wrappers for socket reading/writing
    client_in = open(client_sock.fileno(), 'rt', encoding='latin-1',
                     closefd=False)

    client_out = open(client_sock.fileno(), 'wt', encoding='latin-1
↪',
                     closefd=False)

    # Echo lines back to the client using file I/O
    for line in client_in:
        client_out.write(line)
        client_out.flush()

    client_sock.close()

def echo_server(address):
    sock = socket(AF_INET, SOCK_STREAM)
    sock.bind(address)
    sock.listen(1)
    while True:
        client, addr = sock.accept()
        echo_client(client, addr)
```

éIJĂèęĄéĞ■çĆźåijžèřĚçŽĎäÿĂçĆźæŸŕïijŇäÿŁéĺćçŽĎä¿Ňå■ŘäżĚäżĚæŸŕäÿžäžĘæijŤçď žåĘĚç¡ŏçŽĎ `open()` åĞ¡æŢřçŽĎäÿĂäÿłçĽźæĂğïijŇåźűäÿŤäź§åŘłéĂĆçŤĺäžŐå§žäžŐUnixçŽĎçşżçż§ãĂĆ

åęĆæđIJä¡ăæČşåřĘäÿĂäÿłçśżæŰĞäżűæŐěåŘčä¡IJçŤĺåIJĺäÿĂäÿłåěŮæŐěå■Ůåźűåÿ̆ÑæIJŻä¡ăçŽĎäżčçăĄåŘŕä
makefile() æŰźæşŢãĂĆ ä¡ĘæŸŕåęĆæđIJäÿ■èĂČèŹŚåŘŕçğżæď■æĂğçŽĎèŕÍïijÑéĆčäÿŁéİćçŽĎèğčåĘşæ
makefile() æĂğèČ¡æŻť åě¡äÿĂçĆźãĂĆ

ä¡ääź§åŘŕäżěä¡£çŤĺè£Źçğ■æŁĂæIJŕæİěæđĎéĂääÿĂäÿłåĹńåŘ■ïijÑåĚĄèőÿäżěäÿ■åŘÑäžŐçňňäÿĂæňąą
ä¿ÑåęĆïijÑäÿÑéİćæijŤçď žåęĆä¡ŢåĹŻåżžäÿĂäÿłæŰĞäżűåŕźèśąïijÑåőČåĚĄèőÿä¡äè¿ŞåĞžäžÑè£ŻåĹűæŢŕæ■

```
import sys
# Create a binary-mode file for stdout
bstdout = open(sys.stdout.fileno(), 'wb', closefd=False)
bstdout.write(b'Hello World\n')
bstdout.flush()
```

ář¡çőąåŘŕäżěářĘäÿĂäÿłåůšå■ŸåIJĺçŽĎæŰĞäżűæŘŕè£řçňęåÑĚèčĚæĹŘäÿĂäÿłæ■čåÿÿçŽĎæŰĞäżűåŕźè
ä¡ĘæŸŕèęĄæşĺæĎŘçŽĎæŸŕåźűäÿ■æŸŕæĽĂæIJĽçŽĎæŰĞäżűæĺąåijŘéČ¡èćńæŤŕæÑĄïijÑåźűäÿŤæ§ŘäžŻçś
(çĽźåĹńæŸŕæűĽåŘŁåĹřéŤŹèŕŕåď ĎçŘĘãĂĄæŰĞäżűçżŞář¿æİąäżűç■Ľç■Ľ çŽĎæŮűåĂŹ)ãĂĆ
åIJĺäÿ■åŘÑçŽĎæŞ■ä¡IJçşżçż§äÿŁè£Źçğ■èąÑäÿžäź§æŸŕäÿ■äÿĂæăůïijÑçĽźåĹńçŽĎïijÑäÿŁéİćçŽĎä¿Ńå■Ře
æĹŚèŕť äžĘè£ŹäźĹåď ŽïijÑæĎŘæĂİåřśæŸŕèőľ ä¡ääĚĚåĹĘæţÑèřŢèĞłåůśçŽĎåőđçŐŕäżčçăĄïijÑçąőä£İåőČèČ

7.19 5.19 åĹŻåżžäÿť æŮűæŰĞäżűåŠÑæŰĞäżűåď ź

éŮőécŸ

ä¡ăéIJĂèęĄåIJĺçÍÑåžŘæĽ ğèąÑæŮűåĹŻåżžäÿĂäÿłäÿť æŮűæŰĞäżűæĹŰçŻőå¡ŢïijÑåźűåÿÑæIJŻä¡£çŤĺå

èğčåĘşæŰźæąĹ

tempfile æĺąåİŮäÿ■æIJĽ å¿Ĺåď ŽçŽĎåĞ¡æŢŕåŘŕäżěäőÑæĹŘè£ŹäżżåŁąãĂĆ
äÿżäžĘåĹŻåżžäÿĂäÿłåÑ£åŘ■çŽĎäÿť æŮűæŰĞäżűïijÑåŘŕäżěä¡£çŤĺ tempfile.
TemporaryFile ïijŽ

```
from tempfile import TemporaryFile

with TemporaryFile('w+t') as f:
    # Read/write to the file
    f.write('Hello World\n')
    f.write('Testing\n')

    # Seek back to beginning and read the data
    f.seek(0)
    data = f.read()

# Temporary file is destroyed
```

æĹŰèĂĚïijÑåęĆæđIJä¡ăåŰIJæňćïijÑä¡ăè£ŸåŘŕäżěåČŘè£Źæăůä¡£çŤĺäÿť æŮűæŰĞäżűïijŽ

```
f = TemporaryFile('w+t')
# Use the temporary file
...
f.close()
# File is destroyed
```

TemporaryFile() çŽĎçňňäÿĂäÿłåŘĆæŢřæŸŕæŰĞäżűæĺąåijŘïijŇéĂŽåÿÿæİěèőšæŰĞæIJňæĺąåijŘä
w+t ïijŇäžŇè£ŻåĹűæĺąåijŘä¡£çŤĺ w+b ãĂĆ è£ŹäÿłæĺąåijŘåŘŇæŮűæŤŕæŇĄèŕżåŠŇåĘŹæŞ■ä¡IJïijŇåIJĺè£Ź
TemporaryFile() åŘęåď Ůè£ŸæŤŕæŇĄèů§åĘĚç¡őçŽĎ open()
åĞ¡æŢřäÿĂæăůçŽĎåŘĆæŢřãĂĆæŕŤåęĆïijŽ

```
with TemporaryFile('w+t', encoding='utf-8', errors='ignore') as f:
    ...
```

åIJĺåď ğåď ŽæŢřUnixçşżçż§äÿŁïijŇéĂŽè£Ğ TemporaryFile()
åĹŻåżžçŽĎæŰĞäżűéČ¡æŸŕåŇ£åŘ■çŽĎïijŇçŤŽèĞşè£đçŻőå¡ŢéČ¡æšąæIJL'ãĂĆ
åęĆæđIJä¡ăæČşæL'Şçăť è£ŹäÿłéŹŘåĹűïijŇåŘŕäżěä¡£çŤĺ NamedTemporaryFile()
æİěäžčæŻ£ãĂĆæŕŤåęĆïijŽ

```
from tempfile import NamedTemporaryFile

with NamedTemporaryFile('w+t') as f:
    print('filename is:', f.name)
    ...

# File automatically destroyed
```

è£ŹéĞŇïijŇèćńæL'ŞåijĂæŰĞäżűçŽĎ f.name åšđæĂğåŇĚåŘńäžĘèŕěäÿť æŮűæŰĞäżűçŽĎæŰĞäżűåŘ
å¡Şä¡ăéIJĂèęĄåřĘæŰĞäżűåŘ■äijăéĂŠçżŹåĚűäżŰäżčçăĄæİěæL'ŞåijĂè£ŹäÿłæŰĞäżűçŽĎæŮűåĂŹïijŇè£Źä
åŠŇ TemporaryFile() äÿĂæăůïijŇçżŞæđIJæŰĞäżűåĚşéŮ■æŮűäijŽèćńèĞłåŁĺåĹăéŹď æŐL'ãĂĆ
åęĆæđIJä¡ăäÿ■æČşè£ŹäźĹåĄŽïijŇåŘŕäżěäijăéĂŠäÿĂäÿłåĚşéŤőå■ŮåŘĆæŢř
delete=False å■şåŘŕãĂĆæŕŤåęĆïijŽ

```
with NamedTemporaryFile('w+t', delete=False) as f:
    print('filename is:', f.name)
    ...
```

äÿžäžĘåĹŻåżžäÿĂäÿłäÿť æŮűçŻőå¡ŢïijŇåŘŕäżěä¡£çŤĺ tempfile.
TemporaryDirectory() ãĂĆæŕŤåęĆïijŽ

```
from tempfile import TemporaryDirectory

with TemporaryDirectory() as dirname:
    print('dirname is:', dirname)
    # Use the directory
    ...
# Directory and all contents destroyed
```

èőÍèőž

TemporaryFile() ãĂĄNamedTemporaryFile() åŠŇ TemporaryDirectory() åĞ¡æŢř åžŤèŕěæŸŕåď ĎçŘĘäÿť æŮűæŰĞäžűçŻőå¡ŢçŽĎæIJĂçőĂå■ŢçŽĎæŰź åIJÍäÿĂäÿłæŻť ä¡ŐçŽĎçžğåĹńïijŇä¡ääŘŕäžěä¡£çŤÍ mkstemp() åŠŇ mkdtemp() æÍěåĹŻåżžäÿť æŮűæŰĞäžűåŠŇçŻőå¡ŢãĂĆæŕŤåęĆïijŽ

```
>>> import tempfile
>>> tempfile.mkstemp()
(3, '/var/folders/7W/7WZl5sfZEF0pljrEB1UMWE+++TI/-Tmp-/tmp7fefhv')
>>> tempfile.mkdtemp()
'/var/folders/7W/7WZl5sfZEF0pljrEB1UMWE+++TI/-Tmp-/tmp5wvcv6'
>>>
```

ä¡ĘæŸŕïijŇè£ŹäžŻåĞ¡æŢřåźűäÿ■äijŽåĄŽè£ŻäÿĂæ■ěçŽĎçőąçŘĘäžĘãĂĆ ä¿ŃåęĆïijŇåĞ¡æŢř mkstemp() äżĚäżĚåřśè£ŤåŻđäÿĂäÿłåŐ§åğŃçŽĎOSæŰĞäžűæŘŔè£řçňęïijŇä¡äéIJĂèę åŘŇæăůä¡äè£ŸéIJĂèęĄèĞłåůśæÿĚçŘĘè£ŹäžŻæŰĞäžűãĂĆ

éĂŽåÿÿæÍěèőšïijŇäÿť æŮűæŰĞäžűåIJÍçşżçż§éżŸèőď çŽĎä¡■ç¡őèćńåĹŻåżžïijŇæŕŤåęĆ /var/tmp æĹŰçśżäijijçŽĎåIJřæŰźãĂĆ äÿžäžĘèŐůåŘŰçIJ§åőđçŽĎä¡■ç¡őïijŇåŘŕäžěä¡£çŤÍ tempfile.gettempdir() åĞ¡æŢřãĂĆæŕŤåęĆïijŽ

```
>>> tempfile.gettempdir()
'/var/folders/7W/7WZl5sfZEF0pljrEB1UMWE+++TI/-Tmp-'
>>>
```

æL'ĂæIJL'åŠŇäÿť æŮűæŰĞäžűçŻÿåĚşçŽĎåĞ¡æŢřéČ¡åĚĄèőÿä¡äéĂŽè£Ğä¡£çŤÍåĚşéŤőå■ŮåŘĆæŢř prefix ãĂĄsuffix åŠŇ dir æÍěèĞłåőŽäźL'çŽőå¡Ţäžěå ŘŁåŚ¡åŘ■èğĎåĹźãĂĆæŕŤåęĆïijŽ

```
>>> f = NamedTemporaryFile(prefix='mytemp', suffix='.txt', dir='/tmp
↪')
>>> f.name
'/tmp/mytemp8ee899.txt'
>>>
```

æIJĂåŘŐè£ŸæIJL'äÿĂçĆźïijŇåř¡åŘŕèČ¡äžěæIJĂåőL'åĚÍçŽĎæŰźåijŘä¡£çŤÍ tempfile æÍąåÍŮæÍěåĹŻåżžäÿť æŮűæŰĞäžűãĂĆ åŇĚæŃňäžĚçżŹå¡ŞåL'■çŤÍæĹůæŐĹæÍČèő£éŮőäžěåŘŁåIJÍæŰĞäžű èęĄæşÍæĎŘçŽĎæŸŕäÿ■åŘŇçŽĎåźşåŘřåŘŕèČ¡äijŽäÿ■äÿĂæăůãĂĆåŽäæ■ď ä¡äæIJĂåě¡éŸĚèŕż åőŸæŰźæŰĞæąč æÍěäžĘèğčæŻť åď ŽçŽĎçżĘèŁĆãĂĆ

7.20 5.20 äÿŐäÿšèąŇçńŕåŘčçŽĎæŢřæ■őéĂŽä£ą

éŮőécŸ

ä¡äæČşéĂŽè£ĞäÿšèąŇçńŕåŘčèŕżåĘŹæŢřæ■őïijŇåĚÿåđŃåIJžæŹŕåřśæŸŕåŠŇäÿĂäžŻçąňäžűèő¿åď ĞæL'

èğčåĘşæŰźæąĹ

åř¡çőąä¡ääŘŕäžěéĂŽè£Ğä¡£çŤÍPythonåĘĚç¡őçŽĎI/OæÍąåİŮæİěåőŇæĹŘè£ŹäÿłäżżåŁąïijŇä¡Ęåŕźäž

pySerialåŇĚ ãĂĆ è£ŹäÿłåŇĚçŽĎä¡£çŤÍéİđåÿÿçőĂå■ŢïijŇåĚĹåőL'èčĚpySerialïijŇä¡£çŤÍçśżäijijäÿŃéİćè£Ź

```
import serial
ser = serial.Serial('/dev/tty.usbmodem641', # Device name varies
                    baudrate=9600,
                    bytesize=8,
                    parity='N',
                    stopbits=1)
```

èő¿åď ĞåŘ■åŕźäžŐäÿ■åŘŇçŽĎèő¿åď ĞåŠŇæŞ■ä¡IJçşżçż§æŸŕäÿ■äÿĂæăůçŽĎãĂĆ
æŕŤåęĆïijŇåIJÍWindowsçşżçż§äÿŁïijŇä¡ääŘŕäžěä¡£çŤÍ0, 1ç■L'èąÍçď žçŽĎäÿĂäÿłèő¿åď ĞæİěæL'ŞåijĂéĂŽäł
äÿĂæŮęçńŕåŘčæL'ŞåijĂïijŇéĆčåŕśåŘŕäžěä¡£çŤÍ `read()`ïijŇ`readline()` åŠŇ `write()`
åĞ¡æŢřèŕżåĘŹæŢřæ■őäžĘãĂĆä¿ŃåęĆïijŽ

```
ser.write(b'G1 X50 Y50\r\n')
resp = ser.readline()
```

åď ğåď ŽæŢřæČĚåĘţäÿŃïijŇçőĂå■ŢçŽĎäÿšåŘčéĂŽä£ąäżŐæ■ď åŘŸå¿Ůå■ĄåĹĘçőĂå■ŢãĂĆ

èőÍèőž

åř¡çőąèąÍéİćäÿŁçIJŃèţůæİěå¿ĹçőĂå■ŢïijŇåĚűåőđäÿšåŘčéĂŽä£ąæIJL'æŮűåĂŹäź§æŸŕæŇžéżżçČęçŽĎ
æŐÍè■Řä¡ää¡£çŤÍçňňäÿL'æŰźåŇĚåęĆ `pySerial` çŽĎäÿĂäÿłåŐ§åŽäæŸŕåőČæŘŘä¿ŻäžĘåŕźéńŸçžğçL'źæĂ
(æŕŤåęĆèűĚæŮűïijŇæŐğåĹűæţĄïijŇçijŞåĘšåŇžåĹůæŰřïijŇæŘąæL'Ňå■Řèőőç■L'ç■L')ãĂĆäÿ¿äÿłä¿Ňå■Řïij
`RTS-CTS` æŘąæL'Ňå■Řèőőïij Ň ä¡ääŘłéIJĂèęĄçżŹ `Serial()` äijäéĂŠäÿĂäÿł
`rtscts=True` çŽĎåŘĆæŢřå■şåŘŕãĂĆ åĚűåőŸæŰźæŰĞæąčéİđåÿÿåőŇåŰĎïijŇåŽäæ■ď æĹŚåIJÍè£ŹéĞŇa

æŮűåĹżèőřä¡ŘæL'ĂæIJL'æűL'åŘŁåĹřäÿšåŘčçŽĎI/OéČ¡æŸŕäžŇè£ŻåĹűæÍąåijŘçŽĎãĂĆåŽäæ■ď ïijŇçą
(æĹŰæIJL'æŮűåĂŹæL'ğèąŇæŰĞæIJňçŽĎçijŰçăĄ/èğčçăĄæŞ■ä¡IJ)ãĂĆ
åŘęåď Ůå¡Şä¡äéIJĂèęĄåĹŻåżżäžŇè£ŻåĹűçijŰçăĄçŽĎæŇĞäżď æĹŰæŢřæ■őåŇĚçŽĎæŮűåĂŹïijŇ`struct`
æÍąåİŮäź§æŸŕéİđåÿÿæIJL'çŤÍçŽĎãĂĆ

7.21 5.21 åžŘåĹŮåŇŰPythonåŕźèśą

éŮőéćŸ

ä¡äéIJĂèęĄåřĘäÿĂäÿłPythonåŕźèśąåžŘåĹŮåŇŰäÿžäÿĂäÿłå■ŮèŁĆæţĄïijŇäżěä¿£åřĘåőČä£İå■ŸåĹřäÿĂ

èğčåĘşæŰźæąĹ

åŕźäžŐåžŘåĹŮåŇŰæIJĂæŹőéĄ■çŽĎåĄŽæşŢåŕśæŸŕä¡£çŤÍ `pickle`
æÍąåİŮãĂĆäÿžäžĘåřĘäÿĂäÿłåŕźèśąä£İå■ŸåĹřäÿĂäÿłæŰĞäżűäÿ■ïijŇåŘŕäżěè£ŹæăůåĄŽïijŽ

```
import pickle

data = ... # Some Python object
f = open('somefile', 'wb')
pickle.dump(data, f)
```

äÿžäžĘåřĘäÿĂäÿłåŕźèśąè¡ňåĆÍäÿžäÿĂäÿłå■ŮçňęäÿšïijŇåŘŕäżěä¡£çŤÍ `pickle.dumps()` ïijŽ

```
s = pickle.dumps(data)
```

äÿžäžĘäżŐå■ŮèŁĆæţĄäÿ■æĄćåď■äÿĂäÿłåŕźèśąïijŇä¡£çŤÍ `pickle.load()` æĹŰ `pickle.loads()` åĞ¡æŢřãĂĆæŕŤåęĆïijŽ

```
# Restore from a file
f = open('somefile', 'rb')
data = pickle.load(f)

# Restore from a string
data = pickle.loads(s)
```

èőÍèőž

åŕźäžŐåď ğåď ŽæŢřåžŤçŤÍçÍŃåžŘæÌěèőšïijŇdump() åŠŇ load() åĞ¡æŢřçŽĎä¡£çŤÍåřśæŸŕä¡ăæIJL'æŢĹä¡£çŤÍ `pickle` æÍąåÌŮæL'ĂéIJĂçŽĎåĚÍéČÍäžĘãĂĆ åőČåŘŕéĂĆçŤÍäžŐçżÌåď ğéČÍåĹĘPythonæŢřæ■őçśżåđŃåŠŇçŤÍæĹůèĞłåőŽäźL'çśżçŽĎåŕźèśąåőđä¿ŃãĂĆ åęĆæđIJä¡äçćŕåĹřæ§ŘäÿłåžŞåŘŕäżěèől'ä¡äåIJÍæŢřæ■őåžŞäÿ■ä£Ìå■Ÿ/æĄćåď■PythonåŕźèśąæĹŰèĂĚæŸŕéĂŽ éČčäźĹå¿ĹæIJL'åŘŕèČ¡è£ŹäÿłåžŞçŽĎåžŢåśĆåŕśä¡£çŤÍäžĘ `pickle` æÍąåÌŮãĂĆ

`pickle` æŸŕäÿĂçğ■PythonçL'źæIJL'çŽĎèĞłæŘŘè£řçŽĎæŢřæ■őçijŰçăĄãĂĆ éĂŽè£ĞèĞłæŘŘè£řïijŇèćňåžŘåĹŮåŇŰåŘŐçŽĎæŢřæ■őåŇĚåŘńæŕŘäÿłåŕźèśąåijĂåğŃåŠŇçżŞæÌ§äżěåŘŁåč åŻäæ■ď'ïijŇä¡äæŮäéIJĂæŃĚå£Čåŕźèśąèőŕå¡ŢçŽĎåőŽäźL'ïijŇåőČæĂżæŸŕèČ¡åůëä¡IJãĂĆ äÿ¿äÿłä¿Ňå■ŘïijŇåęĆæđIJèęĄåď ĎçŘĘåď ŽäÿłåŕźèśąïijŇä¡äåŘŕäżěè£ŹæăůåĄŽïijŽ

```
>>> import pickle
>>> f = open('somedata', 'wb')
>>> pickle.dump([1, 2, 3, 4], f)
>>> pickle.dump('hello', f)
>>> pickle.dump({'Apple', 'Pear', 'Banana'}, f)
>>> f.close()
>>> f = open('somedata', 'rb')
>>> pickle.load(f)
[1, 2, 3, 4]
>>> pickle.load(f)
'hello'
>>> pickle.load(f)
{'Apple', 'Pear', 'Banana'}
>>>
```

ä¡ãè£ŸèČ¡åžŘåĹŮåŇŰåĞ¡æŢřijŇçśżïijŇè£ŸæIJĽæŐěåŘčïijŇä¡ĘæŸŕçżŞæđIJæŢřæ■őäżĚäżĚářĘåőČä

```
>>> import math
>>> import pickle.
>>> pickle.dumps(math.cos)
b'\x80\x03cmath\ncos\nq\x00.'
>>>
```

å¡ŞæŢřæ■őåŘ■åžŘåĹŮåŇŰåŻđæİěçŽĎæŮűåĂŹïijŇäijŽåĚĹåĄĞåőŽæĽĂæIJĽçŽĎæžŘæŢřæ■őæŮűåŘ
æĺąåİŮãĂĄçśżåŠŇåĞ¡æŢřäijŽèĞłåŁĺæŇĽéIJĂåŕijåĚěè£ŻæİěãĂĆåŕźäžŐPythonæŢřæ■őèćńäÿ■åŘŇæIJžåŹĺ
æŢřæ■őçŽĎä£Íå■ŸåŘŕèČ¡äijŽæIJĽéŮőećŸïijŇåŻääÿżæĽĂæIJĽçŽĎæIJžåŹĺéČ¡å£Ěéążèő£éŮőåŘŇäÿĂäÿł

æşÍ

```
å■ČäÿĞäÿ■èęĄåŕźäÿ■ä£ąäżżçŽĎæŢřæ■őä¡£çŤÍpickle.load()ãĂĆ
pickleåIJÍåŁäè¡¡æŮűæIJĽäÿĂäÿłåĽŕä¡IJçŤÍåřśæŸŕåőČäijŽèĞłåŁÍåŁäè¡¡çŻÿäžŤæÍąåİŮåź
ä¡ĘæŸŕæ§ŘäÿłåİŔäžžåęĆæđIJç§ěéĄŞpickleçŽĎåůěä¡IJåŐ§çŘĘïijŇ
äżŰåřśåŔŕäżěåĹżåżžäÿĂäÿłæĄűæĎŔçŽĎæŢřæ■őåŕijèĞt'PythonæĽ'ğèąŇéŽŔæĎŔæŇĞåőŽçŽĎçşżçż
åŽäæ■d'ïijŇäÿĂåőŽèęĄä£ÍèŕĄpickleåŔłåIJÍçŻÿäžŠäźŇéŮt'åŔŕäżěèőd'èŕĄåŕźæŰźçŽĎèğčæđŘ
```

æIJĽäžŻçśżåđŃçŽĎåŕźèśąæŸŕäÿ■èČ¡èćńåžŘåĹŮåŇŰçŽĎãĂĆè£ŹäžŻéĂŽåÿÿæŸŕéĆčäžŻä¿ÍèţŰåď'Űé
æŕŤåęĆæĽŞåijĂçŽĎæŰĞäżűïijŇç¡ŚçżIJè£đæŐěïijŇçž£çÍŃïijŇè£ŻçÍŃïijŇæäĹåÿğç■Ľç■ĽãĂĆ
çŤÍæĹůèĞłåőŽäźĽçśżåŘŕäżěéĂŽè£ĞæŘŘä¿Ż `__getstate__()`
åŠŇ `__setstate__()` æŰźæşŢæİěçżŢè£Ğè£ŹäžŻéŹŘåĹűãĂĆ
åęĆæđIJåőŽäźĽäžĘè£Źäÿď'äÿłæŰźæşŢïijŇ`pickle.dump()`
åŕśäijŽèřČçŤÍ `__getstate__()` èŐůåŘŰåžŘåĹŮåŇŰçŽĎåŕźèśąãĂĆ
çśżäijijçŽĎïijŇ`__setstate__()` åIJÍåŘ■åžŘåĹŮåŇŰæŮűèćńèřČçŤÍãĂĆäÿžäžĘæijŤçď'žè£Źäÿłåůěä¡IJåĆ
äÿŃéİćæŸŕäÿĂäÿłåIJÍåĘĚéČÍåőŽäźĽäžĘäÿĂäÿłçž£çÍŃä¡Ęäż■çĎůåŘŕäżěåžŘåĹŮåŇŰåŠŇåŘ■åžŘåĹŮåŇŰç

```
# countdown.py
import time
import threading

class Countdown:
    def __init__(self, n):
        self.n = n
        self.thr = threading.Thread(target=self.run)
        self.thr.daemon = True
        self.thr.start()

    def run(self):
        while self.n > 0:
            print('T-minus', self.n)
            self.n -= 1
            time.sleep(5)

    def __getstate__(self):
        return self.n

    def __setstate__(self, n):
        self.__init__(n)
```

èŕŢçİĂè£ŘèąŇäÿŇéİćçŽĎážŘåĹŮåŇŰèŕŢéłŇäžčçăĄïijŽ

```
>>> import countdown
>>> c = countdown.Countdown(30)
>>> T-minus 30
T-minus 29
T-minus 28
...

>>> # After a few moments
>>> f = open('cstate.p', 'wb')
>>> import pickle
>>> pickle.dump(c, f)
>>> f.close()
```

çĎűåŘŐéĂĂåĞžPythonèğčæđŘåŹĺåźűéĞ■åŘŕåŘŐåĘ■èŕŢéłŇäÿŃïijŽ

```
>>> f = open('cstate.p', 'rb')
>>> pickle.load(f)
countdown.Countdown object at 0x10069e2d0>
T-minus 19
T-minus 18
...
```

ä¡ăåŘŕäžěçIJŃåĹřçž£çĺŃåŘĹåěĞè£źèĹňçŽĎéĞ■çŤ§äžĘïijŇäžŐä¡ăçňňäÿĂæňąåžŘåĹŮåŇŰåőČçŽĎåIJ

`pickle` åŕźäžŐåď ğåđŃçŽĎæŢřæ■őçżŞæđĎæŕŤåęĆä¡£çŤĺ `array` æĹŰ `numpy` æĺąåĺŮåĹŻåżžçŽĎäžŇè£ŻåĹűæŢřçżĎæŢĹçŐĞåźűäÿ■æŸŕäÿĂäÿłéńŸæŢĹçŽĎçijŰçăĄæŰźåijŘãĂĆ åęĆæđIJä¡ăéIJĂèęĄçğżåŁĺåď ğéĞŘçŽĎæŢřçżĎæŢřæ■őïijŇä¡ăæIJĂåě¡æŸŕåĚĹåIJĺäÿĂäÿłæŰĞäźűäÿ■åřĘåĚ (éIJĂèęĄçňňäÿL'æŰźåžŞçŽĎæŤŕæŇĄ)ãĂĆ

çŤśäžŐ `pickle` æŸŕPythonçL'źæIJL'çŽĎåźűäÿŤéŹĎçİĂåIJĺæžŘçăĄäÿŁïijŇæL'ĂæIJL'åęĆæđIJéIJĂèę ä¿ŃåęĆïijŇåęĆæđIJæžŘçăĄåŘŸåŁĺäžĘïijŇä¡ăæL'ĂæIJL'çŽĎå■ŸåĆĺæŢřæ■őåŘŕèČ¡äijŽèćńçăť'åĺŘåźűäÿŤåŘ åĺęçŹ¡æĺěèőšïijŇåŕźäžŐåIJĺæŢřæ■őåžŞåŠŇå■ŸæąčæŰĞäźűäÿ■å■ŸåĆĺæŢřæ■őæŮűïijŇä¡ăæIJĂåě¡ä¡£çŤĺæŽ è£ŹäžŻçijŰçăĄæăijåijŘæŽť'æăĞåĞĘïijŇåŘŕäżěèćńäÿ■åŘŇçŽĎèŕ■èĺĂæŤŕæŇĄïijŇåźűäÿŤäź§èČ¡å¿Ĺåě¡çŽĎ

æIJĂåŘŐäÿĂçĆźèęĄæşĺæĎŘçŽĎæŸŕ `pickle` æIJL'åď ğéĞŘçŽĎéĚ■ç¡őéĂL'éąźåŠŇäÿĂäžŻæčŸæL'Ń åŕźäžŐæIJĂåÿÿèğĄçŽĎä¡£çŤĺåIJżæŹŕïijŇä¡ääÿ■éIJĂèęĄåŐżæŃĚå£Čè£ŹäÿłïijŇä¡ĘæŸŕåęĆæđIJä¡äèęĄåIJĺ æIJĂåě¡åŐżæ§ěéŸĚäÿĂäÿŃ åőŸæŰźæŰĞæąč ãĂĆ

8 çňňåĚ■çńäïijŽæŢřæ■őçijŰçăĄåŠŇåď ĎçŘĘ

è£ŹäÿĂçńääÿżèęĄèőĺèőžä¡£çŤĺPythonåď ĎçŘĘåŘĎçğ■äÿ■åŘŇæŰźåijŘçijŰçăĄçŽĎæŢřæ■őïijŇæŕŤåę åŠŇæŢřæ■őçżŞæđĎéĆčäÿĂçńääÿ■åŘŇçŽĎæŸŕïijŇè£Źçńääÿ■äijŽèőĺèőžçL'źæőŁçŽĎçőŮæşŢéŮőéćŸïijŇè

Contents:

8.1 6.1 èŕżåĘŹCSVæŢřæ■ő

éŮőéćŸ

ä¡ăæČşèŕżåĘŹäÿĂäÿłCSVæăijåijŔçŽĎæŰĞäżűãĂĆ

èğčåĘşæŰźæąĹ

åŕźäžŐåď ğåď ŽæŢřçŽĎCSVæăijåijŔçŽĎæŢřæ■őèŕżåĘŹéŮőéćŸïijŇéČ¡åŔŕäžěä¡£çŤÍ `csv` åžŞãĂĆ ä¿ŃåęĆïijŽåĄĞèő¿ä¡ăåIJÍäÿĂäÿłåŘ■åŔństocks.csvæŰĞäżűäÿ■æIJĽäÿĂäžŻèĆąçĚÍåÿĆåIJžæŢ

```
Symbol,Price,Date,Time,Change,Volume
"AA",39.48,"6/11/2007","9:36am",-0.18,181800
"AIG",71.38,"6/11/2007","9:36am",-0.15,195500
"AXP",62.58,"6/11/2007","9:36am",-0.46,935000
"BA",98.31,"6/11/2007","9:36am",+0.12,104800
"C",53.08,"6/11/2007","9:36am",-0.25,360900
"CAT",78.29,"6/11/2007","9:36am",-0.23,225400
```

äÿŃéİćåŘŚä¡ååśŢçď žåęĆä¡ŢåřĘè£ŹäžŻæŢřæ■őèŕżåŔŰäÿžäÿĂäÿłåĚČçżĎçŽĎåžŔåĹŮïijŽ

```
import csv
with open('stocks.csv') as f:
    f_csv = csv.reader(f)
    headers = next(f_csv)
    for row in f_csv:
        # Process row
        ...
```

åIJÍäÿŁéİćçŽĎäžčçăĄäÿ■ïijŇ `row` äijŽæŸŕäÿĂäÿłåĹŮèąÍãĂĆåŻäæ■ď ïijŇäÿžäžĘèő£éŮőæ§Řäÿłå■Ůæő `row[0]` èő£éŮőSymbolïijŇ `row[4]` èő£éŮőChangeãĂĆ

çŤśäžŐè£Źçğ■äÿŃæăĞèő£éŮőéĂŽåÿÿäijŽåijŢèţůæůůæůĘïijŇä¡åŔŕäžěèĂČèŹŚä¡£çŤÍåŚ¡åŘ■åĚČçżĎ

```
from collections import namedtuple
with open('stock.csv') as f:
    f_csv = csv.reader(f)
    headings = next(f_csv)
    Row = namedtuple('Row', headings)
    for r in f_csv:
        row = Row(*r)
        # Process row
        ...
```

åőČåĚĄèőÿä¡ää¡£çŤÍåĹŮåŘ■åęĆ `row.Symbol` åŠŇ `row.Change` äžčæŻ£äÿŃæăĞèő£éŮőãĂĆ éIJĂèęĄæşÍæĎŔçŽĎæŸŕè£ŹäÿłåŔłæIJĽåIJÍåĹŮåŘ■æŸŕåŘĹæşŢçŽĎPythonæ ä¡åŔŕèČ¡éIJĂèęĄä£őæŤźäÿŃåŐ§åğŃçŽĎåĹŮåŘ■(åęĆåřĘéİđæăĞèŕĘçňęå■ŮçňęæŻ£æ■ćæĹŘäÿŃåĹŠçž£äź

åŔęåď ŰäÿĂäÿłéĂĽæŃľ åŕśæŸŕåřĘæŢřæ■őèŕżåŔŰåĹřäÿĂäÿłå■ŮåĚÿåžŔåĹŮäÿ■åŐżãĂĆåŔŕäžěè£Źæă

```
import csv
with open('stocks.csv') as f:
    f_csv = csv.DictReader(f)
    for row in f_csv:
        # process row
        ...
```

åIJÍè£ŹäÿłçL'ĹæIJňäÿ■ïijŇä¡ăåŘŕäżěä¡£çŤĺåĹŮåŘ■åŐżèő£éŮőæŕŘäÿĂèąŇçŽĎæŢřæ■őäžĘãĂĆæŕŤåęĆ
æĹŰèĂĚ `row['Change']`

äÿžäžĘåĘŹåĚěCSVæŢřæ■őïijŇä¡ääż■çĎűåŘŕäżěä¡£çŤĺcsvæĺąåİŮïijŇäÿ■è£ĞèŹæŮűåĂŹåĚĹåĹŻåżžäÿ
`writer` åŕźèśąãĂĆä¿ŃåęĆ:

```
headers = ['Symbol','Price','Date','Time','Change','Volume']
rows = [('AA', 39.48, '6/11/2007', '9:36am', -0.18, 181800),
         ('AIG', 71.38, '6/11/2007', '9:36am', -0.15, 195500),
         ('AXP', 62.58, '6/11/2007', '9:36am', -0.46, 935000),
       ]

with open('stocks.csv','w') as f:
    f_csv = csv.writer(f)
    f_csv.writerow(headers)
    f_csv.writerows(rows)
```

åęĆæđIJä¡ăæIJL'äÿĂäÿłå■ŮåĚÿåžŘåĹŮçŽĎæŢřæ■őïijŇåŘŕäżěåČŘè£ŹæăůåĄŽïijŽ

```
headers = ['Symbol', 'Price', 'Date', 'Time', 'Change', 'Volume']
rows = [{'Symbol':'AA', 'Price':39.48, 'Date':'6/11/2007',
        'Time':'9:36am', 'Change':-0.18, 'Volume':181800},
        {'Symbol':'AIG', 'Price': 71.38, 'Date':'6/11/2007',
        'Time':'9:36am', 'Change':-0.15, 'Volume': 195500},
        {'Symbol':'AXP', 'Price': 62.58, 'Date':'6/11/2007',
        'Time':'9:36am', 'Change':-0.46, 'Volume': 935000},
        ]

with open('stocks.csv','w') as f:
    f_csv = csv.DictWriter(f, headers)
    f_csv.writeheader()
    f_csv.writerows(rows)
```

èőĺèőž

ä¡ăážŤèřěæĂżæŸŕäijŸåĚĹéĂL'æŃl'csvæĺąåİŮåĹĘåL'šæĹŰèğčæđŘCSVæŢřæ■õäĂĆä¿ŃåęĆïijŇä¡ăåŘŕ

```
with open('stocks.csv') as f:
for line in f:
    row = line.split(',')
    # process row
    ...
```

ä¡£çŤÍè£Źçğ■æŰźåijŘçŽĎäÿĂäÿłçijžçĆźåřśæŸŕä¡ääż■çĎűéIJĂèęĄåŐżåď ĎçŘĘäÿĂäžŻæčŸæĽŃçŽĎç
æŕŤåęĆïijŇåęĆæđIJæ§ŘäžŻå■Ůæőţ̌åĂijèćńåijŢåŘůåŇĚåŻť ïijŇä¡ääÿ■å¿Ůäÿ■åŐżéŹď è£ŹäžŻåijŢåŘůãĂĆ
åŘęåď ŰïijŇåęĆæđIJäÿĂäÿłèćńåijŢåŘůåŇĚåŻť çŽĎå■ŮæőţçćřåůgåŘńæIJĽäÿĂäÿłéĂŮůåŘůïijŇéĆčäźĹçÍŃåž

éżŸèőď æČĚåĘţäÿŃïijŇcsv åžŞåŘŕèŕĘåĹńMicrosoft Excelæ Ľ Ää¡£çŤÍçŽĎCSVçijŰçăĄèğĎåĹŹãĂĆ è£ŹæĹŰèőÿäź§æŸŕæIJĂåÿÿèğĄçŽĎå¡ćåijŘïijŇåźűäÿŤäź§äijŽ
çĎűèĂŇïijŇåęĆæđIJä¡äæ§ěçIJŃcsvçŽĎæŰĞæąčïijŇåřśäijŽåŘŚçŐřæIJĽ å¿Ĺåď Žçğ■æŰźæşŢåřĘåőČåžŤçŤÍ
ä¿ŃåęĆïijŇåęĆæđIJä¡äæČşèŕżåŘŰäžětabåĹĘåĽšçŽĎæŢřæ■őïijŇåŘŕäžěè£ŹæăůåĄŽïijŽ

```
# Example of reading tab-separated values
with open('stock.tsv') as f:
    f_tsv = csv.reader(f, delimiter='\t')
    for row in f_tsv:
        # Process row
        ...
```

åęĆæđIJä¡äæ■čåIJÍèŕżåŘŰCSVæŢřæ■őáźűåřĘåőČäżňè¡ňæ■ćäÿžåŚ¡åŘ■åĚČçżĎïijŇéIJĂèęĄæşÍæĎŘåŕź
ä¿ŃåęĆïijŇäÿĂäÿłCSVæăijåijŘæŰĞäżűæIJĽäÿĂäÿłåŇĚåŘńéİđæşŢæăĞèŕĘçňęçŽĎåĹŮåď ťèąŃïijŇçśżäijij

```
StreetÂăAddress,Num-Premises,Latitude,Longitude 5412ÂăNÂăCLARK,10,
↪41.980262,-87.668452
```

è£ŹæăůæIJĂçżĹäijŽåŕijèĞť åIJÍåĹŻåżžäÿĂäÿłåŚ¡åŘ■åĚČçżĎæŮűäžğçŤ§äÿĂäÿł
ValueError åijĆåÿÿèĂŇåď śèť ěãĂĆ äÿžäžĘèğčåĘşè£ŹéŮőécŸïijŇä¡ääŘŕèČ¡äÿ■å¿Ůäÿ■åĚĹåŐżä£őæ■čå
ä¿ŃåęĆïijŇåŘŕäžěåČŘäÿŃéİćè£ŹæăůåIJÍéİđæşŢæăĞèŕĘçňęäÿŁä¡£çŤÍäÿĂäÿłæ■čåĹŹèąÍè¿¿åijŘæŻ£æ■ćïijŽ

```
import re
with open('stock.csv') as f:
    f_csv = csv.reader(f)
    headers = [ re.sub('[^a-zA-Z_]', '_', h) for h in next(f_csv) ]
    Row = namedtuple('Row', headers)
    for r in f_csv:
        row = Row(*r)
        # Process row
        ...
```

è£ŸæIJĽéĞ■èęĄçŽĎäÿĂçĆźéIJĂèęĄåijžèřĆçŽĎæŸŕïijŇcsväžğçŤ§çŽĎæŢřæ■őéČ¡æŸŕå■Ůçňęäÿšçśżå
åęĆæđIJä¡äéIJĂèęĄåĄŽè£ŹæăůçŽĎçśżåđŃè¡ňæ■ćïijŇä¡ääĚéążèĞłåůśæĽŃåŁĺåŐżåőđçŐřăĂĆ
äÿŃéİćæŸŕäÿĂäÿłåIJÍCSVæŢřæ■őäÿŁæĽ ğèąŃåĚűäžŰçśżåđŃè¡ňæ■ćçŽĎä¿Ńå■ŘïijŽ

```
col_types = [str, float, str, str, float, int]
with open('stocks.csv') as f:
    f_csv = csv.reader(f)
    headers = next(f_csv)
    for row in f_csv:
        # Apply conversions to the row items
        row = tuple(convert(value) for convert, value in zip(col_
↪types, row))
        ...
```

åŘęåď ŰïijŇäÿŃéİćæŸŕäÿĂäÿłè¡ňæ■ćå■ŮåĚÿäÿ■çĽźåőŽå■Ůæőţ̌çŽĎä¿Ńå■ŘïijŽ

```
print('Reading as dicts with type conversion')
field_types = [ ('Price', float),
                ('Change', float),
                ('Volume', int) ]

with open('stocks.csv') as f:
    for row in csv.DictReader(f):
        row.update((key, conversion(row[key]))
                for key, conversion in field_types)
        print(row)
```

éĂŽåÿÿæİěèőšïijŇä¡ăåŕŕèČ¡åźűäÿ■æČşè£Ğåď ŽåŐżèĂČèŹŚè£ŹäžŻè¡ňæ■ćéŮőéćŸãĂĆ
åIJĺåőđéŹĚæČĚåĘţäÿ■ïijŇCSVæŰĞäžűéČ¡æĹŰåď ŽæĹŰåřŚæIJL'äžŻçijžåď śçŽĎæŢřæ■őïijŇèćńçăť åİŕçŽĎ
åŻăæ■ď ïijŇéŹď éİđä¡äçŽĎæŢřæ■őçąőåőđæIJL'ä£İéŽIJæŸŕåĞĘçąőæŮăèŕŕçŽĎïijŇåŘęåĹŹä¡ăå£ĚéążèĂČèŹ

æIJĂåŘŐïijŇåęĆæđIJä¡ăèŕżåŘŰCSVæŢřæ■őçŽĎçŻőçŽĎæŸŕåĄŽæŢřæ■őåĹĘæđŘåŠŇçż§èőąçŽĎèŕİïij
ä¡ăåŕŕèČ¡éIJĂèęĄçIJŃäÿĂçIJŃ Pandas åŇĚãĂĆPandas
åŇĚåŘńäžĘäÿĂäÿłéİđåÿÿæŰźä¿£çŽĎåĞ¡æŢřåŘń pandas.read_csv()
ïijŇ åőČåŕŕäżěåŁäè¡¡CSVæŢřæ■őåĹřäÿĂäÿł DataFrame åŕźèśąäÿ■åŐżãĂĆ
çĎűåŘŐåĹl'çŤĺè£Źäÿłåŕźèśąä¡ăåŕśåŕŕäżěçŤ§æĹŘåŘĎçğ■å¡ćåijŕçŽĎçż§èőąãĂĄè£ĞæżďæŢřæ■őäżěåŘŁæI
åIJĺ6.13åřŕèŁĆäÿ■äijŽæIJL'è£ŹæăůäÿĂäÿłä¿Ňå■ŘãĂĆ

8.2 6.2 èŕżåĘŹJSONæŢřæ■ő

éŮőéćŸ

ä¡ăæČşèŕżåĘŹJSON(JavaScript Object Notation)çijŰçăĄæäijåijŕçŽĎæŢřæ■őãĂĆ

èğčåĘşæŰźæąĹ

json æĺąåİŮæŘŘä¿ŻäžĘäÿĂçğ■å¿ĹçőĂå■ŢçŽĎæŰźåijŕæİěçijŰçăĄåŠŇèğčçăĄJSONæŢřæ■őãĂĆ
åĚűäÿ■äÿďäÿłäÿżèęĄçŽĎåĞ¡æŢřæŸŕ json.dumps() åŠŇ json.loads()
ïijŇ èęĄæŕŤåĚűäżŰåžŘåĹŮåŇŰåĞ¡æŢřåžŞåęĆpickleçŽĎæŐěåŘčåřŚå¿Ůåď ŽãĂĆ
äÿŃéİćæijŤçď žåęĆä¡ŢåřĘäÿĂäÿłPythonæŢřæ■őçżŞæđĎè¡ňæ■ćäÿžJSONïijŽ

```
import json

data = {
    'name' : 'ACME',
    'shares' : 100,
    'price' : 542.23
}

json_str = json.dumps(data)
```

äÿŃéİćæijŤçď žåęĆä¡ŢåřĘäÿĂäÿłJSONçijŰçăĄçŽĎå■Ůçňęäÿšè¡ňæ■ćåŻđäÿĂäÿłPythonæŢřæ■őçżŞæđĎ

```
data = json.loads(json_str)
```

åęĆæđIJä¡ăèęĄåď ĎçŘĘçŽĎæŸŕæŰĞäźűèĂŇäÿ■æŸŕå■ŮçňęäÿšïijŇä¡ăåŘŕäžěä¡£çŤÍ
`json.dump()` åŠŇ `json.load()` æİěçijŰçăĄåŠŇèğčçăĄJSONæŢřæ■őãĂĆä¿ŃåęĆïijŽ

```
# Writing JSON data
with open('data.json', 'w') as f:
    json.dump(data, f)

# Reading data back
with open('data.json', 'r') as f:
    data = json.load(f)
```

èőĺèőž

JSONçijŰçăĄæŤŕæŇĄçŽĎå§žæIJňæŢřæ■őçśżåđŃäÿž `None` ïijŇ `bool` ïijŇ `int` ïijŇ
`float` åŠŇ `str` ïijŇ äžěåŘŁåŇĚåŘńè£ŹäžŻçśżåđŃæŢřæ■őçŽĎlistsïijŇtuplesåŠŇdictionariesãĂĆ
åŕźäžŐdictionariesïijŇkeyséIJĂèęĄæŸŕå■ŮçňęäÿšçśżåđŃ(å■ŮåĚÿäÿ■äżżä¡ŢéİđåĄ■ŮçňęäÿšçśżåđŃçŽĎkeyåI
äÿžäžĘéĄţå¿ŁJSONèğĎèŇČïijŇä¡ăåžŤèŕěåŘłçijŰçăĄPythonçŽĎlistsåŠŇdictionariesãĂĆ
èĂŇäÿŤïijŇåIJÍwebåžŤçŤÍçÍŃåžŘäÿ■ïijŇéąűåśĆåŕźèśąèćńçijŰçăĄäÿžäÿĂäÿłå■ŮåĚÿæŸŕäÿĂäÿłæăĞåĞĘåA

JSONçijŰçăĄçŽĎæăijåijŘåŕźäžŐPythonèŕ■æşŢèĂŇåůšåĞääźŐæŸŕåőŇåĚÍäÿĂæăůçŽĎïijŇéŹď äžĘäÿĂ
æŕŤåęĆïijŇTrueäijŽèćńæŸăåřĎäÿžtrueïijŇFalseèćńæŸăåřĎäÿž-
falseïijŇèĂŇNoneäijŽèćńæŸăåřĎäÿžnullãĂĆ äÿŃéİćæŸŕäÿĂäÿłä¿Ńå■ŘïijŇæijŤçď žäžĘçijŰçăĄåŘŐçŽĎå■

```
>>> json.dumps(False)
'false'
>>> d = {'a': True,
...      'b': 'Hello',
...      'c': None}
>>> json.dumps(d)
'{"b": "Hello", "c": null, "a": true}'
>>>
```

åęĆæđIJä¡ăèŕŢçİĂåŐżæčĂæ§ěJSONèğčçăĄåŘŐçŽĎæŢřæ■őïijŇä¡ăéĂŽåÿÿå¿ĹéŽ¿éĂŽè£ĞçőĂå■ŢçŽĎ
çL'źåĹńæŸŕå¡ŞæŢřæ■őçŽĎåţŇåěŮçżŞæđĎåśĆæňąå¿ŁæůśæĹŰèĂĚåŇĚåŘńåď ğéĞŘçŽĎå■Ůæőţæ Ůűã ĂĆ
äÿžäžĘèğčåĘşè£ŹäÿłéŮőéćŸïijŇåŘŕäžěèĂČèŹŚä¡£çŤÍpprintæÍąåİŮçŽĎ
`pprint()` åĞ¡æŢřæİěäžčæŻ£æŹőéĂŽçŽĎ `print()` åĞ¡æŢřãĂĆ
åőČäijŽæŇL'çĚğkeyçŽĎå■Ůæŕ■éąžäžŘåźűäžěäÿĂçğ■æŻť åŁăç¿ŐèğĆçŽĎæŰźåijŘè¿ŞåĞžãĂĆ
äÿŃéİćæŸŕäÿĂäÿłæijŤçď žäęĆä¡ŢæijĆäžőçŽĎæL'Ŝå■řè¿ŞåĞžTwitteräÿŁæŘIJçť ćçżŞæđIJçŽĎä¿Ńå■ŘïijŽ

```
>>> from urllib.request import urlopen
>>> import json
>>> u = urlopen('http://search.twitter.com/search.json?q=python&
↪rpp=5')
>>> resp = json.loads(u.read().decode('utf-8'))
>>> from pprint import pprint
>>> pprint(resp)
```

(continues on next page)

(continued from previous page)

```
{'completed_in': 0.074,
'max_id': 264043230692245504,
'max_id_str': '264043230692245504',
'next_page': '?page=2&max_id=264043230692245504&q=python&rpp=5',
'page': 1,
'query': 'python',
'refresh_url': '?since_id=264043230692245504&q=python',
'results': [{'created_at': 'Thu, 01 Nov 2012 16:36:26 +0000',
            'from_user': ...
            },
            {'created_at': 'Thu, 01 Nov 2012 16:36:14 +0000',
            'from_user': ...
            },
            {'created_at': 'Thu, 01 Nov 2012 16:36:13 +0000',
            'from_user': ...
            },
            {'created_at': 'Thu, 01 Nov 2012 16:36:07 +0000',
            'from_user': ...
            }
            {'created_at': 'Thu, 01 Nov 2012 16:36:04 +0000',
            'from_user': ...
            }],
'results_per_page': 5,
'since_id': 0,
'since_id_str': '0'}
>>>
```

äÿĂèĹňæĺĕèőšïijŇJSONèğčçăĄäijŽæăźæ■őæŔŘä¿ŻçŽĎæŢřæ■őåĹŻåżždictsæĹŰlistsãĂĆ åęĆæđIJä¡ăæČşèęĄåĹŻåżžåĚűäżŰçśżåđŃçŽĎåŕźèśąïijŇåŔŕäżěçżŹ `json.loads()` äijăéĂŠobject_pairs_hookæĹŰobject_hookåŔĆæŢřãĂĆ ä¿ŃåęĆïijŇäÿŃéĺćæŸŕæijŤçď žåęĆä¡ŢèğčçăĄ JSONæŢřæ■őåźűåIJĺäÿĂäÿłOrderedDictäÿ■ä£ĺçŢŹåĚűéąžåžŔ

```
>>> s = '{"name": "ACME", "shares": 50, "price": 490.1}'
>>> from collections import OrderedDict
>>> data = json.loads(s, object_pairs_hook=OrderedDict)
>>> data
OrderedDict([('name', 'ACME'), ('shares', 50), ('price', 490.1)])
>>>
```

äÿŃéĺćæŸŕåęĆä¡ŢåřĘäÿĂäÿłJSONå■ŮåĚÿè¡ňæ■ćäÿžäÿĂäÿłPythonåŕźèśąä¿Ńå■ŘïijŽ

```
>>> class JSONObject:
...     def __init__(self, d):
...         self.__dict__ = d
...
>>>
>>> data = json.loads(s, object_hook=JSONObject)
>>> data.name
'ACME'
```

(continues on next page)

(continued from previous page)

```
>>> data.shares
50
>>> data.price
490.1
>>>
```

æIJĂåŘŐäÿĂäÿłä¿Ńå■Řäÿ■ïijŇJSONèğčçăĄåŘŐçŽĎå■ŮåĚÿä¡IJäÿžäÿĂäÿłå■ŢäÿłåŘĆæŢřäijăéĂŠçżŹ
`__init__()` ãĂĆ çĎűåŘŐïijŇä¡ăářšåŘŕäżěéŽŘå£ČæĽĂæňšçŽĎä¡£çŤĺåőČäžĘïijŇæŕŤåęĆä¡IJäÿžäÿĂäÿłå

åIJĺçijŰçăĄJSONçŽĎæŮűåĂŹïijŇè£ŸæIJĽäÿĂäžŻéĂĽéąźå¿ĹæIJĽçŤĺãĂĆ
åęĆæđIJä¡ăæČşèŐůå¿ŮæijĆäžőçŽĎæăijåijŘåŇŰå■ŮçňęäÿšåŘŐè¿ŞåĞžïijŇåŘŕäżěä¡£çŤĺ
`json.dumps()` çŽĎindentåŘĆæŢřãĂĆ åőČäijŽä¡£å¿Ůè¿ŞåĞžåŠŇpprint()åĞ¡æŢřæŢĹæđIJçśżäijijãĂĆæŕŤ

```
>>> print(json.dumps(data))
{"price": 542.23, "name": "ACME", "shares": 100}
>>> print(json.dumps(data, indent=4))
{
    "price": 542.23,
    "name": "ACME",
    "shares": 100
}
>>>
```

åŕźèśąåőđä¿ŃéĂŽåÿÿåźűäÿ■æŸŕJSONåŘŕåžŘåĹŮåŇŰçŽĎãĂĆä¿ŃåęĆïijŽ

```
>>> class Point:
...     def __init__(self, x, y):
...         self.x = x
...         self.y = y
...
>>> p = Point(2, 3)
>>> json.dumps(p)
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "/usr/local/lib/python3.3/json/__init__.py", line 226, in␣
↪dumps
        return _default_encoder.encode(obj)
    File "/usr/local/lib/python3.3/json/encoder.py", line 187, in␣
↪encode
        chunks = self.iterencode(o, _one_shot=True)
    File "/usr/local/lib/python3.3/json/encoder.py", line 245, in␣
↪iterencode
        return _iterencode(o, 0)
    File "/usr/local/lib/python3.3/json/encoder.py", line 169, in␣
↪default
        raise TypeError(repr(o) + " is not JSON serializable")
TypeError: <__main__.Point object at 0x1006f2650> is not JSON␣
↪serializable
>>>
```

åęĆæđIJä¡ăæČşåžŘåĹŮåŇŰåŕźèśąåőđä¿ŃïijŇä¡ăåŘŕäżěæŘŘä¿ŻäÿĂäÿłåĞ¡æŢřïijŇåőČçŽĎè¿ŞåĚěæŸŕ

```
def serialize_instance(obj):
    d = { '__classname__' : type(obj).__name__ }
    d.update(vars(obj))
    return d
```

åęĆæđIJä¡ăæČşåŘ■è£ĞæİěèŐůåŘŰè£Źäÿłåőđä¿ŃïijŇåŘŕäżěè£ŹæăůåĄŽïijŽ

```
# Dictionary mapping names to known classes
classes = {
    'Point' : Point
}

def unserialize_object(d):
    clsname = d.pop('__classname__', None)
    if clsname:
        cls = classes[clsname]
        obj = cls.__new__(cls) # Make instance without calling __
↪init__
        for key, value in d.items():
            setattr(obj, key, value)
        return obj
    else:
        return d
```

äÿŃéİćæŸŕåęĆä¡Ţä¡£çŤĺè£ŹäžŻåĞ¡æŢřçŽĎä¿Ńå■ŘïijŽ

```
>>> p = Point(2,3)
>>> s = json.dumps(p, default=serialize_instance)
>>> s
'{"__classname__": "Point", "y": 3, "x": 2}'
>>> a = json.loads(s, object_hook=unserialize_object)
>>> a
<__main__.Point object at 0x1017577d0>
>>> a.x
2
>>> a.y
3
>>>
```

`json` æĺąåİŮè£ŸæIJĽå¿Ĺåď ŽåĚűäżŰéĂĽéąźæİěæŐğåĹűæŻť ä¡ŐçžğåĹńçŽĎæŢřå■ŮãĂĄçĽźæőŁåĂi
åŘŕäżěåŘĆèĂČåőŸæŰźæŰĞæąčèŐůåŘŰæŻť åď ŽçżĘèŁĆãĂĆ

8.3 6.3 èğčæđŘçőĂå■ŢçŽĎXMLæŢřæ■ő

éŮőécŸ

ä¡ăæČşäżŐäÿĂäÿłçőĂå■ŢçŽĎXMLæŰĞæąčäÿ■æŘŘåŘŰæŢřæ■õãĂĆ

èğčåĘşæŰźæąĹ

åŘŕäžěä¡£çŤÍ xml.etree.ElementTree æĺąåİŮäžŐçőĂå■ŢçŽĎXMLæŰĞæąčäÿ■æŘŘåŘŰæŢřæ■őãĂĆ äÿžäžĘæijŤçď žïijŇåĄĞèő¿ä¡ăæČşèğčæđŘPlanet PythonäÿŁçŽĎRSSæžŘãĂĆäÿŃéİćæŸŕçŻÿåžŤçŽĎäžčçăĄïijŽ

```
from urllib.request import urlopen
from xml.etree.ElementTree import parse

# Download the RSS feed and parse it
u = urlopen('http://planet.python.org/rss20.xml')
doc = parse(u)

# Extract and output tags of interest
for item in doc.iterfind('channel/item'):
    title = item.findtext('title')
    date = item.findtext('pubDate')
    link = item.findtext('link')

    print(title)
    print(date)
    print(link)
    print()
```

è£ŘèąŇäÿŁéİćçŽĎäžčçăĄïijŇè¿ŞåĞžçżŞæđIJçśżäijijè£ŹæăůïijŽ

```
Steve Holden: Python for Data Analysis
Mon, 19 Nov 2012 02:13:51 +0000
http://holdenweb.blogspot.com/2012/11/python-for-data-analysis.html

Vasudev Ram: The Python Data model (for v2 and v3)
Sun, 18 Nov 2012 22:06:47 +0000
http://jugad2.blogspot.com/2012/11/the-python-data-model.html

Python Diary: Been playing around with Object Databases
Sun, 18 Nov 2012 20:40:29 +0000
http://www.pythondiary.com/blog/Nov.18,2012/been-...-object-
↪databases.html

Vasudev Ram: Wakari, Scientific Python in the cloud
Sun, 18 Nov 2012 20:19:41 +0000
http://jugad2.blogspot.com/2012/11/wakari-scientific-python-in-
↪cloud.html

Jesse Jiryu Davis: Toro: synchronization primitives for Tornado␣
↪coroutines
Sun, 18 Nov 2012 20:17:49 +0000
http://feedproxy.google.com/~r/EmptysquarePython/~3/_DOZT2Kd0hQ/
```

å¿ĹæŸ¿çĎűïijŇåęĆæđIJä¡ăæČşåĄŽè£ŻäÿĂæ■ěçŽĎåď ĎçŘĘïijŇä¡ăéIJĂèęĄæŻ£æ■ć `print()` èŕ■åŘěæİěåőŇæĹŘåĚűäžŰæIJL'èűčçŽĎäžŃãĂĆ

èőÍèőž

åIJÍå¿Ĺåď ŽåžŤçŤÍçÍŃåžŔäÿ■åď ĎçŘĘXMLçijŰçăĄæăijåijŔçŽĎæŢřæ■őæŸŕå¿ĹåÿÿèğĄçŽĎãĂĆäÿ■äżĚåŻääÿžXMLåIJÍInternetäÿŁéİćåůšçżŘèćńåź£æşŽåžŤçŤÍäžŐæŢřæ■őäžď æ■ćïijŇåŘŇæŮűåőČäź§æŸŕäÿĂçğ■å■ŸåĆÍåžŤçŤÍçÍŃåžŘæŢřæ■őçŽĎåÿÿçŤÍæăijåijŔ(æŕŤåęĆå■ŮåďĎçŘĘïijŇé§§äæŐěäÿŃæİěçŽĎèőÍèőžäijŽåĚĹåĄĞåőŽèŕżèĂĚåůšçżŘåŕźXMLå§žçąĂæŕŤè¿ČçĘ§æĆĽäžĘãĂĆ

åIJÍå¿Ĺåď ŽæČĚåĘţäÿŃïijŇå¡Şä¡£çŤÍXMLæİěäżĚäżĚå■ŸåĆÍæŢřæ■őçŽĎæŮűåĂŹïijŇåŕźåžŤçŽĎæŰĞä¿ŃåęĆïijŇäÿŁéİćä¿Ńå■Řäÿ■çŽĎRSSèőćéŸĚæžŘçśżäijijäžŐäÿŃéİćçŽĎæäijåijŘïijŽ

```
<?xml version="1.0"?>
<rss version="2.0" xmlns:dc="http://purl.org/dc/elements/1.1/">
    <channel>
        <title>Planet Python</title>
        <link>http://planet.python.org/</link>
        <language>en</language>
        <description>Planet Python - http://planet.python.org/</
↪description>
        <item>
            <title>Steve Holden: Python for Data Analysis</title>
            <guid>http://holdenweb.blogspot.com/...-data-analysis.
↪html</guid>
            <link>http://holdenweb.blogspot.com/...-data-analysis.
↪html</link>
            <description>...</description>
            <pubDate>Mon, 19 Nov 2012 02:13:51 +0000</pubDate>
        </item>
        <item>
            <title>Vasudev Ram: The Python Data model (for v2 and␣
↪v3)</title>
            <guid>http://jugad2.blogspot.com/...-data-model.html</
↪guid>
            <link>http://jugad2.blogspot.com/...-data-model.html</
↪link>
            <description>...</description>
            <pubDate>Sun, 18 Nov 2012 22:06:47 +0000</pubDate>
        </item>
        <item>
            <title>Python Diary: Been playing around with Object␣
↪Databases</title>
            <guid>http://www.pythondiary.com/...-object-databases.
↪html</guid>
            <link>http://www.pythondiary.com/...-object-databases.
↪html</link>
            <description>...</description>
            <pubDate>Sun, 18 Nov 2012 20:40:29 +0000</pubDate>
        </item>
        ...
    </channel>
</rss>
```

xml.etree.ElementTree.parse() åĞ¡æŢřèğčæđŘæŢť äÿłXMLæŰĞæąčåźűåřĘåĚűè¡ňæ■ćæĹŘ
çĎűåŘŐïijŇä¡ăåřśèČ¡ä¡£çŤÍ find() ãĂĄiterfind() åŠŇ
findtext() ç■L'æŰźæşŢæÌěæŘIJçť ćçL'źåőŽçŽĎXMLåĚČçť äážĘãĂĆ
è£ŹäžŻåĞ¡æŢřçŽĎåŘĆæŢřåřśæŸŕæ§ŘäÿłæŇĞåőŽçŽĎæăĞç■¿åŘ■ïijŇä¿ŃåęĆ channel/
item æĹŰ title ãĂĆ

æŕŘæňąæŇĞåőŽæ§ŘäÿłæăĞç■¿æŮűïijŇä¡ăéIJĂèęĄéĄ■åŐĘæŢť äÿłæŰĞæąčçżŞæđĎãĂĆæŕŘæňąæŘIJ
åŘŇæăůïijŇæŕŘæňąæŞ■ä¡IJæL'ĂæŇĞåőŽçŽĎæăĞç■¿åŘ■äź§æŸŕèţůåğŃåĚČçť äçŽĎçŻÿåŕźèůŕå¿ĎãĂĆ
ä¿ŃåęĆïijŇæL'ğèąŇ doc.iterfind('channel/item')
æÌěæŘIJçť ćæL'ĂæIJL'åIJÍ channel åĚČçť ääÿŃéÌćçŽĎ item åĚČçť äãĂĆ
doc äžčèąÍæŰĞæąčçŽĎæIJĂéąűåśĆ(äź§åřśæŸŕçňňäÿĂçžğçŽĎ rss åĚČçť ä)ãĂĆ
çĎűåŘŐæŐěäÿŃæÌěçŽĎèřČçŤÍ item.findtext() äijŽäžŐåůšæL'¿åĹřçŽĎ item
åĚČçť ää¡■ç¡őåijĂåğŃæŘIJçť ćãĂĆ

ElementTree æÍąåÌŮäÿ■çŽĎæŕŘäÿłåĚČçť ăæIJL'äÿĂäžŻéĞ■èęĄçŽĎåśđæĂğåŠŇæŰźæşŢïijŇåIJÍèğč
tag åśđæĂğåŇĚåŘńäžĘæăĞç■¿çŽĎåŘ■å■ŮïijŇtext åśđæĂğåŇĚåŘńäžĘąĘéČÍçŽĎæŰĞæIJňïijŇèĂŇ
get() æŰźæşŢèČ¡èŐůåŘŰåśđæĂğåĂijãĂĆä¿ŃåęĆïijŽ

```
>>> doc
<xml.etree.ElementTree.ElementTree object at 0x101339510>
>>> e = doc.find('channel/title')
>>> e
<Element 'title' at 0x10135b310>
>>> e.tag
'title'
>>> e.text
'Planet Python'
>>> e.get('some_attribute')
>>>
```

æIJL'äÿĂçĆźèęĄåijžèřČçŽĎæŸŕ xml.etree.ElementTree åźűäÿ■æŸŕXM-
LèğčæđŘçŽĎåŤŕäÿĂæŰźæşŢãĂĆ åŕźäžŐæŻť énŸçžğçŽĎåžŤçŤÍçÍŃåžŘïijŇä¡ăéIJĂèęĄèĂČèŹŚä¡£çŤÍ
lxml ãĂĆ åőČä¡£çŤÍäžĘåŠŇElementTreeåŘŇæăůçŽĎçijŰçÍŃæŐěåŘčïijŇåŽäæ■ď äÿŁéÌćçŽĎä¿Ńå■ŘåŘŇa
ä¡ăåŘłéIJĂèęĄåřĘåĹŽåijĂåğŃçŽĎimportèŕ■åŘěæ■ćæĹŘ from lxml.etree import
parse åřśèąŇäžĘãĂĆ lxml åőŇåĚÍéĄţå¿łXMLæăĞåĞĘïijŇåźűäÿŤéĂ§åžęäź§éÌđåÿÿå£ńïijŇåŘŇæŮűè£Ÿa

8.4 6.4 åćđéĞŘåijŘèğčæđŘåď ğåđŃXMLæŰĞäžű

éŮőécŸ

ä¡ăæČşä¡£çŤÍåř¡åŘŕèČ¡åřŚçŽĎåĘĚå■ŸäżŐäÿĂäÿłèűĚåď ğçŽĎXMLæŰĞæąčäÿ■æŘŘåŘŰæŢřæ■őãĂĆ

èğčåĘşæŰźæąĹ

äżżä¡ŢæŮűåĂŹåŘłèęĄä¡ăéĄĞåĹřåćđéĞŘåijŘçŽĎæŢřæ■őåď ĎçŘĘæŮűïijŇçňňäÿĂæŮűéŮť åřśåžŤèŕěa
äÿŃéÌćæŸŕäÿĂäÿłå¿ĹçőĂå■ŢçŽĎåĞ¡æŢřïijŇåŘłä¡£çŤÍå¿ĹåřŚçŽĎåĘĚå■ŸåřśèČ¡åćđéĞŘåijŘçŽĎåď ĎçŘĘä

```
from xml.etree.ElementTree import iterparse

def parse_and_remove(filename, path):
    path_parts = path.split('/')
    doc = iterparse(filename, ('start', 'end'))
    # Skip the root element
    next(doc)

    tag_stack = []
    elem_stack = []
    for event, elem in doc:
        if event == 'start':
            tag_stack.append(elem.tag)
            elem_stack.append(elem)
        elif event == 'end':
            if tag_stack == path_parts:
                yield elem
                elem_stack[-2].remove(elem)
            try:
                tag_stack.pop()
                elem_stack.pop()
            except IndexError:
                pass
```

äÿžäžĘæţŃèŕŢè£ŹäÿłåĞ¡æŢřïjŇä¡äéIJĂèęĄåĚĹæIJL'äÿĂäÿłåď ğåđŃçŽĎXMLæŰĞäżűãĂĆ
éĂŽåÿÿä¡äåŘŕäżěåIJĺæŤ£åžIJç¡ŚçńŹæĹŰåĚňåĚśæŢřæ■őç¡ŚçńŹäÿŁæL'¿åĹřè£ŹæăůçŽĎæŰĞäżűãĂĆ
ä¿ŃåęĆïijŇä¡äåŘŕäżěäÿŃè¡¡XMLæăijåijŘçŽĎèŁİåŁäåŞěå§ŐåÿĆéĄŞèůŕåİŚæť'ijæŢřæ■őåžŞãĂĆ
åIJĺåĘŹè£ŹæIJňäźęçŽĎæŮűåĂŹïijŇäÿŃè¡¡æŰĞäżűåůšçżŘåŇĚåŘńèűĚè£Ğ100,000èąŇæŢřæ■őïijŇçijŰçăĄ

```
<response>
    <row>
        <row ...>
            <creation_date>2012-11-18T00:00:00</creation_date>
            <status>Completed</status>
            <completion_date>2012-11-18T00:00:00</completion_date>
            <service_request_number>12-01906549</service_request_
↪number>
            <type_of_service_request>Pot Hole in Street</type_of_
↪service_request>
            <current_activity>Final Outcome</current_activity>
            <most_recent_action>CDOT Street Cut ... Outcome</most_
↪recent_action>
            <street_address>4714 S TALMAN AVE</street_address>
            <zip>60632</zip>
            <x_coordinate>1159494.68618856</x_coordinate>
            <y_coordinate>1873313.83503384</y_coordinate>
            <ward>14</ward>
            <police_district>9</police_district>
            <community_area>58</community_area>
            <latitude>41.808090232127896</latitude>
```

(continues on next page)

(continued from previous page)

```
            <longitude>-87.69053684711305</longitude>
            <location latitude="41.808090232127896"
            longitude="-87.69053684711305" />
        </row>
        <row ...>
            <creation_date>2012-11-18T00:00:00</creation_date>
            <status>Completed</status>
            <completion_date>2012-11-18T00:00:00</completion_date>
            <service_request_number>12-01906695</service_request_
↪number>
            <type_of_service_request>Pot Hole in Street</type_of_
↪service_request>
            <current_activity>Final Outcome</current_activity>
            <most_recent_action>CDOT Street Cut ... Outcome</most_
↪recent_action>
            <street_address>3510 W NORTH AVE</street_address>
            <zip>60647</zip>
            <x_coordinate>1152732.14127696</x_coordinate>
            <y_coordinate>1910409.38979075</y_coordinate>
            <ward>26</ward>
            <police_district>14</police_district>
            <community_area>23</community_area>
            <latitude>41.91002084292946</latitude>
            <longitude>-87.71435952353961</longitude>
            <location latitude="41.91002084292946"
            longitude="-87.71435952353961" />
        </row>
    </row>
</response>
```

åĄĞèő¿ä¡ăæČşåĘŹäÿĂäÿłèĎŽæIJňæİĕæŇL'çĚğåİŚæť'ijæŁĕåŚŁæŢřéĞŘæŐŠåĹŮéĆőçijŰåŘůçăĄãĂĆä

```
from xml.etree.ElementTree import parse
from collections import Counter

potholes_by_zip = Counter()

doc = parse('potholes.xml')
for pothole in doc.iterfind('row/row'):
    potholes_by_zip[pothole.findtext('zip')] += 1
for zipcode, num in potholes_by_zip.most_common():
    print(zipcode, num)
```

è£ŹäÿłèĎŽæIJňåŤřäÿĂçŽĎéŮőécŸæŸřåőČäijŽåĚĹåřĘæŢť'äÿłXMLæŰĞäżűåŁäè¡¡åĹřåĘĚå■Ÿäÿ■çĎűa
åIJĺæĹŚçŽĎæIJžåŹĺäÿŁïijŇäÿžäžĘè£ŘèąŇè£ŹäÿłçĺŃåžŘéIJĂèęĄçŤĺåĹř450MBåůęåŘşçŽĎåĘĚå■Ÿçľ'žéŮť'ä
åęĆæđIJä¡£çŤĺåęĆäÿŃäżčçăĄïijŇçĺŃåžŘåŘłéIJĂèęĄä£őæŤźäÿĂçĆźçĆźïijŽ

```
from collections import Counter
```

(continues on next page)

(continued from previous page)

```
potholes_by_zip = Counter()

data = parse_and_remove('potholes.xml', 'row/row')
for pothole in data:
    potholes_by_zip[pothole.findtext('zip')] += 1
for zipcode, num in potholes_by_zip.most_common():
    print(zipcode, num)
```

çżŞæđIJæŸŕïijŽè£ŹäÿłçL'ŁæIJňçŽĎäžčçăĄè£ŘèąŇæŮűåŘłéIJĂèęĄ7MBçŽĎåĘĚå■Ÿ–åď ğåď ğèŁĆçžę

èőĺèőž

è£ŹäÿĂèŁĆçŽĎæŁĂæIJŕäijŽä¿ÌèţŰ `ElementTree` æĺąåİŮäÿ■çŽĎäÿď äÿłæăÿå£ČåŁ§èČ¡ãĂĆ çňňäÿĂïijŇ`iterparse()` æŰźæşŢåĚĄèőÿåŕźXMLæŰĞæąčè£ŻèąŇåćđéĞŘæŞ■ä¡IJãĂĆ ä¡£çŤĺæŮűïijŇä¡äéIJĂèęĄæŘŘä¿ŻæŰĞäžűåŘ■åŠŇäÿĂäÿłåŇĚåŘńäÿŇéİćäÿĂçğ■æĹŰåď Žçğ■çśżåđŇçŽĎä `start`, `end`, `start-ns` åŠŇ `end-ns` ãĂĆ çŤś `iterparse()` åĹŻåžžçŽĎè£■äžčåŹĺäijŽäžğçŤ§å¡ćåęĆ `(event, elem)` çŽĎåĚČçżĎïijŇ åĚűäÿ■ `event` æŸŕäÿŁè£řäžŇäžűåĹŮèąĺäÿ■çŽĎæ§ŘäÿĂäÿłïijŇèĂŇ `elem` æŸŕçŻÿåžŤçŽĎXMLåĚČçť'äãĂĆä¿ŇåęĆïijŽ

```
>>> data = iterparse('potholes.xml',('start','end'))
>>> next(data)
('start', <Element 'response' at 0x100771d60>)
>>> next(data)
('start', <Element 'row' at 0x100771e68>)
>>> next(data)
('start', <Element 'row' at 0x100771fc8>)
>>> next(data)
('start', <Element 'creation_date' at 0x100771f18>)
>>> next(data)
('end', <Element 'creation_date' at 0x100771f18>)
>>> next(data)
('start', <Element 'status' at 0x1006a7f18>)
>>> next(data)
('end', <Element 'status' at 0x1006a7f18>)
>>>
```

`start` äžŇäžűåIJĺæ§ŘäÿłåĚČçť'äçňňäÿĂæňąèćňåĹŻåžžåźűäÿŤè£ŸæšąæIJL'èćňæŘŠåĚěåĚűäžŰæŢřæ■ èĂŇ `end` äžŇäžűåIJĺæ§ŘäÿłåĚČçť'äåůšçżŘåőŇæĹŘæŮűèćňåĹŻåžžãĂĆ åř¡çőąæšąæIJL'åIJĺä¿Ňå■ Räÿ■æijŤçď'žïijŇ `start-ns` åŠŇ `end-ns` äžŇäžűèćňçŤĺæĺěåď'ĎçŘĘXMLæŰĞæąčåŚ¡åŘ■çľ'žéŮť'çŽĎåčřæŸŐãĂĆ

è£ŹæIJňèŁĆä¿Ňå■ Räÿ■ïijŇ `start` åŠŇ `end` äžŇäžűèćňçŤĺæĺěçőąçŘĘåĚČçť'äåŠŇæăĞç■¿æăĹãĂĆ æăĹäžčèąĺäžĘæŰĞæąčèćňèğčæđŘæŮűçŽĎåśĆæňąçżŞæđĎïijŇ è£ŸèćňçŤĺæĺěåĹď'æŰ■æ§ŘäÿłåĚČçť'äæŸŕåŘęåŇźéĚ■äijäçżŹåĞ¡æŢř `parse_and_remove()` çŽĎèůŕå¿ĎãĂĆ åęĆæđIJåŇźéĚ■ïijŇåřśåĹľçŤĺ `yield` èŕ■åŘěåŘŚèřĆçŤĺèĂĚè£ŤåŻđè£ŹäÿłåĚČçť'äãĂĆ

åIJĺ `yield` äźŇåŘŐçŽĎäÿŇéİćè£Źäÿłèŕ■åŘěæĹ'■æŸŕä¡£å¿ŮçĺŇåžŘå■äçŤĺæđĄåřŚåĘĚå■ŸçŽĎElemen

```
elem_stack[-2].remove(elem)
```

è£Źäÿłèŕ■åŘëä¡£å¿ŮäźŃåĽ■çŤś `yield` äžğçŤ§çŽĎåĚČçť ääżŐåőČçŽĎçĹűèŁĆçĆźäÿ■åĹäéŹď æŐĽ'ãĹ
åĄĞèő¿åůšçżŘæšąæIJĽ'åĚűåőČçŽĎåIJřæŰźåijŢçŤÍè£ŹäÿłåĚČçť ääžĘïijŇéĆčäźĹè£ŹäÿłåĚČçť äářšèćńéŤĂæ

åŕźèŁĆçĆźçŽĎè£■äžčåijŘèğčæđŘåŠŇåĹäéŹď çŽĎæIJĂçżĹæŢĹæđIJářśæŸŕäÿĂäÿłåIJÍæŰĞæąčäÿŁéńŸ
æŰĞæąčæăŚçżŞæđĎäżŐåğŃèĞłçżĹæšąèćńåőŇæŢť çŽĎåĹŻåžžè£ĞãĂĆåř¡çőąåęĆæ■ď'ïijŇè£ŸæŸŕèČ¡éĂŽ

è£ŹçğğæŰźæąĹçŽĎäÿżèęĄçijžéŹůářśæŸŕåőČçŽĎè£ŘèąŇæĂğèČ¡äžĘãĂĆ
æĹŚèĞłåůśæţŃèŕŢçŽĎçżŞæđIJæŸŕïijŇèŕżåŘŰæŢť äÿłæŰĞæąčåĹřåĘĚå■ŸäÿçŽĎçĽĹæIJňçŽĎè£ŘèąŇéĂ§
ä¡ĘæŸŕåőČå■ť ä¡£çŤÍäžĘèűĚè£ĞåŘŐèĂĚ60åĂ■çŽĎåĘĚå■ŸãĂĆ
åŽäæ■ď'ïijŇåęĆæđIJä¡äæŽť åĚşå£ČåĘĚå■Ÿä¡£çŤÍéĞŘçŽĎèŕÍïijŇéĆčäźĹåćđéĞŘåijŘçŽĎçĽĹæIJňåőŇèČIJä

8.5 6.5 ářĘå■ŮåĚÿè¡ňæ■ćäÿžXML

éŮőécŸ

ä¡äæČşä¡£çŤÍäÿĂäÿłPythonå■ŮåĚÿå■ŸåĆÍæŢřæ■őïijŇåźűářĘåőČè¡ňæ■ćæĹŘXMLæăijåijŘãĂĆ

èğčåĘşæŰźæąĹ

ář¡çőą `xml.etree.ElementTree` åžŞéĂŽåÿÿçŤÍæÍěåĄŽèğčæđŘåůëä¡IJïijŇåĚűåőđåőČäź§åŘŕäżěåĹ
ä¿ŃåęĆïijŇèĂČèŹŚåęĆäÿŃè£ŹäÿłåĞ¡æŢřïijŽ

```
from xml.etree.ElementTree import Element

def dict_to_xml(tag, d):
'''
Turn a simple dict of key/value pairs into XML
'''
elem = Element(tag)
for key, val in d.items():
    child = Element(key)
    child.text = str(val)
    elem.append(child)
return elem
```

äÿŃéÍćæŸŕäÿĂäÿłä¡£çŤÍä¿Ńå■ŘïijŽ

```
>>> s = { 'name': 'GOOG', 'shares': 100, 'price':490.1 }
>>> e = dict_to_xml('stock', s)
>>> e
<Element 'stock' at 0x1004b64c8>
>>>
```

è¡ňæ■ćçżŞæđIJæŸŕäÿĂäÿł `Element` åőđä¿ŃãĂĆåŕźäžŐI/OæŞ■ä¡IJïijŇä¡£çŤÍ `xml.etree.ElementTree` äÿ■çŽĎ `tostring()` åĞ¡æŢřå¿ĹåőźæŸŞåřśèČ¡åřĘåőČè¡ňæ■ćæĹŘäÿĂäÿłå■ŮèŁ

```
>>> from xml.etree.ElementTree import tostring
>>> tostring(e)
b'<stock><price>490.1</price><shares>100</shares><name>GOOG</name></
 ↪stock>'
>>>
```

åęĆæđIJä¡ăæČşçżŹæ§ŘäÿłåĚČçť'ăæůżåŁăåśđæĂğåĂijïijŇåŔŕäżěä¡£çŤÍ `set()` æŰźæşŢïijŽ

```
>>> e.set('_id','1234')
>>> tostring(e)
b'<stock _id="1234"><price>490.1</price><shares>100</shares><name>
 ↪GOOG</name>
</stock>'
>>>
```

åęĆæđIJä¡ăè£ŸæČşä£ÍæŇĄåĚČçť'ăçŽĎéąžåžŘïijŇåŔŕäżěèĂČèŹŚæđĎéĂääÿĂäÿł `OrderedDict` æÍěäżčæŻ£äÿĂäÿłæŹőéĂŽçŽĎå■ŮåĚÿãĂĆèŕůåŔĆèĂČ1.7åřŘèŁĆãĂĆ

èőÍèőž

å¡ŞåĹŻåżžXMLçŽĎæŮűåĂŹïijŇä¡ăèćńéŹŘåĹűåŔłèČ¡æđĎéĂäå■ŮçňęäÿšçśżåđŃçŽĎåĂijãĂĆä¿ŃåęĆï

```
def dict_to_xml_str(tag, d):
    '''
    Turn a simple dict of key/value pairs into XML
    '''
    parts = ['<{}>'.format(tag)]
    for key, val in d.items():
        parts.append('<{0}>{1}</{0}>'.format(key,val))
    parts.append('</{}>'.format(tag))
    return ''.join(parts)
```

éŮőécŸæŸŕåęĆæđIJä¡ăæL'ŃåŁÍçŽĎåŐżæđĎéĂäçŽĎæŮűåĂŹåŔŕèČ¡äijŽçćŕåĹŕäÿĂäžŻéžżçČęãĂĆä¿Ń

```
>>> d = { 'name' : '<spam>' }

>>> # String creation
>>> dict_to_xml_str('item',d)
'<item><name><spam></name></item>'

>>> # Proper XML creation
>>> e = dict_to_xml('item',d)
>>> tostring(e)
b'<item><name>&lt;spam&gt;</name></item>'
>>>
```

æşÍæĎŔåĹŕçÍŃåžŔçŽĎåŘŐéÍćéĆčäÿłä¿Ńå■Řäÿ■ïijŇå■Ůçňę âĂŸ<âĂŹ åŠŇ âĂŸ>âĂŹ èćńæŻ£æ■ćæĹŘäžĘ `<` åŠŇ `>`

äÿŃéİćäżĚä¿ŻåŘĆèĂČïijŃåęĆæđIJä¡ăéIJĂèęĄæĽŃåŁĺåŐżè¡ňæ■ćè£ŹäžŻå■ŮçňęïijŃ åŘŕäżĕä¡£çŤĺ `xml.sax.saxutils` äÿ■çŽĎ `escape()` åŠŃ `unescape()` åĞ¡æŢřãĂĆä¿ŃåęĆïijŽ

```
>>> from xml.sax.saxutils import escape, unescape
>>> escape('<spam>')
'&lt;spam&gt;'
>>> unescape(_)
'<spam>'
>>>
```

éŹďäžĘèČ¡åĹŻåżžæ■čçąőçŽĎè¿ŞåĞžåď ŰïijŃè£ŸæIJĽåŘęåď ŰäÿĂäÿłåŐ§åŻăæŐĺè■Řä¡ăåĹŻåżž `Element` åőđä¿ŃèĂŃäÿ■æŸŕå■ŮçňęäÿšïijŃ éĆčåřśæŸŕä¡£çŤĺå■ŮçňęäÿšçżĎåŘĹæđĎéĂääÿĂäÿłæŻť åď ğç èĂŃ `Element` åőđä¿ŃåŘŕäżĕäÿ■çŤĺèĂČèŹŚèğčæđŘXMLæŰĞæIJňçŽĎæČĚåĘţäÿŃéĂŽè£Ğåď Žçğ■æŰźäi äź§åřśæŸŕèŕť ïijŃä¡ăåŘŕäżĕåIJĺäÿĂäÿłéńŸçžğæŢřæ■őçżŞæđĎäÿŁåőŃæĹŘä¡ăæĽĂæIJĽçŽĎæŞ■ä¡IJïijŃåźű

8.6 6.6 èğčæđŘåŠŃä£őæŤźXML

éŮőécŸ

ä¡ăæČşèŕżåŘŰäÿĂäÿłXMLæŰĞæąčïijŃåŕźåőČæIJĂäÿĂäžŻä£őæŤźïijŃçĎűåŘŐåřEçżŞæđIJåĘŹåŻđXM

èğčåĘşæŰźæąĹ

ä¡£çŤĺ `xml.etree.ElementTree` æĺąåİŮåŘŕäżĕå¿ĹåőźæŸŞçŽĎåď ĎçŘEè£ŹäžŻäżżåŁąãĂĆ çňňäÿĂæ■ĕæŸŕäżĕéĂŽåÿÿçŽĎæŰźåijŘæİĕèğčæđŘè£ŹäÿłæŰĞæąčãĂĆä¿ŃåęĆïijŃåĄĞèő¿ä¡ăæIJĽäÿĂäÿłå `pred.xml` çŽĎæŰĞæąčïijŃçśżäijijäÿŃéİćè£ŹæăůïijŽ

```
<?xml version="1.0"?>
<stop>
    <id>14791</id>
    <nm>Clark &amp; Balmoral</nm>
    <sri>
        <rt>22</rt>
        <d>North Bound</d>
        <dd>North Bound</dd>
    </sri>
    <cr>22</cr>
    <pre>
        <pt>5 MIN</pt>
        <fd>Howard</fd>
        <v>1378</v>
        <rn>22</rn>
    </pre>
    <pre>
        <pt>15 MIN</pt>
        <fd>Howard</fd>
```

(continues on next page)

(continued from previous page)

```
        <v>1867</v>
        <rn>22</rn>
    </pre>
</stop>
```

äÿŃéĮćæŸŕäÿĂäÿłåĹľçŤĺ `ElementTree` æĮĕèŕżåŘŰè£ŹäÿłæŰĞæąčåźűåŕźåőČåĄŽäÿĂäžŻä£őæŤźçŽ

```
>>> from xml.etree.ElementTree import parse, Element
>>> doc = parse('pred.xml')
>>> root = doc.getroot()
>>> root
<Element 'stop' at 0x100770cb0>

>>> # Remove a few elements
>>> root.remove(root.find('sri'))
>>> root.remove(root.find('cr'))
>>> # Insert a new element after <nm>...</nm>
>>> root.getchildren().index(root.find('nm'))
1
>>> e = Element('spam')
>>> e.text = 'This is a test'
>>> root.insert(2, e)

>>> # Write back to a file
>>> doc.write('newpred.xml', xml_declaration=True)
>>>
```

åď ĎçŘEçżŞæđIJæŸŕäÿĂäÿłåČŘäÿŃéĮćè£ŹæăůæŰřçŽĎXMLæŰĞäžűïijŽ

```
<?xml version='1.0' encoding='us-ascii'?>
<stop>
    <id>14791</id>
    <nm>Clark &amp; Balmoral</nm>
    <spam>This is a test</spam>
    <pre>
        <pt>5 MIN</pt>
        <fd>Howard</fd>
        <v>1378</v>
        <rn>22</rn>
    </pre>
    <pre>
        <pt>15 MIN</pt>
        <fd>Howard</fd>
        <v>1867</v>
        <rn>22</rn>
    </pre>
</stop>
```

èőÍèőž

ä£őæŤźäÿĂäÿłXMLæŰĞæąčçżŞæđĎæŸŕå¿ĹåőźæŸ ŞçŽĎïijŇä¡ĘæŸŕä¡ăå£ĚéążçĽćèőřçŽĎæŸŕæĽĂæł
åřĘåőČä¡IJäÿžäÿĂäÿłåĹŮèąÍæİěåď ĎçŘĘãĂĆä¿ŃåęĆïijŇåęĆæđIJä¡ăåĹăéŹď æ§ŘäÿłåĚČçť ăïijŇéĂŽè£ĞèřČ
`remove()` æŰźæşŢäżŐåőČçŽĎçŻť æŐěçĹűèŁĆçĆźäÿ■åĹăéŹď ãĂĆ
åęĆæđIJä¡ăæŘŠåĚěæĹŰåćđåŁăæŰřçŽĎåĚČçť ăïijŇä¡ăåŘŇæăůä¡£çŤĺçĹűèŁĆçĆźåĚČçť ăçŽĎ
`insert()` åŠŇ `append()` æŰźæşŢãĂĆ è£ŸèČ¡åŕźåĚČçť ăä¡£çŤĺçť ćåijŢåŠŇåĹĞçĽĞæŞ■ä¡IJïijŇæŕŤåęĆ
`element[i]` æĹŰ `element[i:j]`

åęĆæđIJä¡ăéIJĂèęĄåĹŻåżžæŰřçŽĎåĚČçť ăïijŇåŘŕäżěä¡£çŤĺæIJňèŁĆæŰźæąĹäÿ■æijŤçď žçŽĎ
`Element` çśżãĂĆæĹŚäżňåIJĺ6.5åřŘèŁĆåůšçżŘèŕęçżĘèőĺèőžè£ĞäžĘãĂĆ

8.7 6.7 åĹľçŤĺåŚ¡åŘ■çľžéŮťèğčæđŘXMLæŰĞæąč

éŮőécŸ

ä¡ăæČşèğčæđŘæ§ŘäÿłXMLæŰĞæąčïijŇæŰĞæąčäÿ■ä¡£çŤĺäžĘXMLåŚ¡åŘ■çľžéŮťãĂĆ

èğčåĘşæŰźæąĹ

èĂČèŹŚäÿŃéİćè£Źäÿłä¡£çŤĺäžĘåŚ¡åŘ■çľžéŮťçŽĎæŰĞæąčïijŽ

```
<?xml version="1.0" encoding="utf-8"?>
<top>
    <author>David Beazley</author>
    <content>
        <html xmlns="http://www.w3.org/1999/xhtml">
            <head>
                <title>Hello World</title>
            </head>
            <body>
                <h1>Hello World!</h1>
            </body>
        </html>
    </content>
</top>
```

åęĆæđIJä¡ăèğčæđŘè£ŹäÿłæŰĞæąčåźűæĽğèąŇæŹőéĂŽçŽĎæ§ěèŕćïijŇä¡ăäijŽåŘŚçŐřè£Źäÿłåźűäÿ■æŸ

```
>>> # Some queries that work
>>> doc.findtext('author')
'David Beazley'
>>> doc.find('content')
<Element 'content' at 0x100776ec0>
>>> # A query involving a namespace (doesn't work)
>>> doc.find('content/html')
>>> # Works if fully qualified
```

(continues on next page)

(continued from previous page)

```
>>> doc.find('content/{http://www.w3.org/1999/xhtml}html')
<Element '{http://www.w3.org/1999/xhtml}html' at 0x1007767e0>
>>> # Doesn't work
>>> doc.findtext('content/{http://www.w3.org/1999/xhtml}html/head/
↪title')
>>> # Fully qualified
>>> doc.findtext('content/{http://www.w3.org/1999/xhtml}html/'
... '{http://www.w3.org/1999/xhtml}head/{http://www.w3.org/1999/
↪xhtml}title')
'Hello World'
>>>
```

ä¡ăăŘŕäžěéĂŽè£ĞăřĘăŚ¡ăŘ■çľžéŮťăďĎçŘĘéĂżè¿ŚăŇĚèčĚäÿžäÿĂäÿłăůěăĚůçśżæĺěçőĂăŇŰè£Źäÿłèŕ

```
class XMLNamespaces:
    def __init__(self, **kwargs):
        self.namespaces = {}
        for name, uri in kwargs.items():
            self.register(name, uri)
    def register(self, name, uri):
        self.namespaces[name] = '{'+uri+'}'
    def __call__(self, path):
        return path.format_map(self.namespaces)
```

éĂŽè£ĞäÿŃéĺćçŽĎæŰźăijŘä¡£çŤĺè£ŹäÿłçśżïijŽ

```
>>> ns = XMLNamespaces(html='http://www.w3.org/1999/xhtml')
>>> doc.find(ns('content/{html}html'))
<Element '{http://www.w3.org/1999/xhtml}html' at 0x1007767e0>
>>> doc.findtext(ns('content/{html}html/{html}head/{html}title'))
'Hello World'
>>>
```

èőĺèőž

èğčæđŘăŘńæIJL'ăŚ¡ăŘ■çľžéŮťçŽĎXMLæŰĞæąčäijŽæŕŤè¿ČçźĄçŘŘãĂĆ äÿŁéĺćçŽĎ `XMLNamespaces` äžĚäžĚæŸŕăĚĄèőÿä¡ăä¡£çŤĺçijľçŢěăŘ■äžčæŻ£ăőŇæŢťçŽĎURIăřĘăĚűăŘŸă¿ŮçĺŒă¿őç

ă¿Ĺäÿ■ăźÿçŽĎæŸŕŕijŇăIJĺă§žæIJňçŽĎ `ElementTree` èğčæđŘäÿ■æšąæIJL'äżżä¡ŢéĂŤă¿ĎèŐůăŘŰăŚ¡ăŘ■çľžéŮťçŽĎä£ąæĄŕãĂĆ ä¡ĘæŸŕŕijŇăęĆæđIJä¡ăä¡£çŤĺ `iterparse()` ăĞ¡æŢřçŽĎèŕĺăřśăŘŕäžěèŐůăŘŰæŻťăďŽăĚşäžŐăŚ¡ăŘ■çľžé

```
>>> from xml.etree.ElementTree import iterparse
>>> for evt, elem in iterparse('ns2.xml', ('end', 'start-ns', 'end-
↪ns')):
... print(evt, elem)
...
end <Element 'author' at 0x10110de10>
```

(continues on next page)

(continued from previous page)

```
start-ns ('', 'http://www.w3.org/1999/xhtml')
end <Element '{http://www.w3.org/1999/xhtml}title' at 0x1011131b0>
end <Element '{http://www.w3.org/1999/xhtml}head' at 0x1011130a8>
end <Element '{http://www.w3.org/1999/xhtml}h1' at 0x101113310>
end <Element '{http://www.w3.org/1999/xhtml}body' at 0x101113260>
end <Element '{http://www.w3.org/1999/xhtml}html' at 0x10110df70>
end-ns None
end <Element 'content' at 0x10110de68>
end <Element 'top' at 0x10110dd60>
>>> elem # This is the topmost element
<Element 'top' at 0x10110dd60>
>>>
```

æIJĂåŘŐäÿĂçĆźïijŇåęĆæđIJä¡ăèęĄåď ĎçŘEçŽĎXMLæŰĞæIJňéŹď äžĘèęĄä¡£çŤĺåĹřåĚűäżŰéńŸçžğ
åżžèőőä¡ăæIJĂåě¡æŸŕä¡£çŤĺ `lxml` åĞ¡æŢřåžŞæİěäžčæŻ£ `ElementTree` ãĂĆ
ä¿ŃåęĆïijŇ`lxml` åŕźåĹľçŤĺDTDéłŇèŕĄæŰĞæąčãĂĄæŻť åě¡çŽĎXPathæŤŕæŇĄåŠŇäÿĂäžŻåĚűäżŰéńŸçžğ
è£ŹäÿĂåřŘèŁĆåĚűåőđåŘłæŸŕæŢŹä¡ääęĆä¡Ţèőľ XMLèğčæđŘçĺ■å¿őçőĂå■ŢäÿĂçĆźãĂĆ

8.8 6.8 äÿŐåĚşçşżåđŃæŢřæ■őåžŞçŽĎäžď äžŠ

éŮőécŸ

ä¡ăæČşåIJĺåĚşçşżåđŃæŢřæ■őåžŞäÿ■æ§ěèŕćãĂĄåćđåŁăæĹŰåĹăéŹď èőřå¡ŢãĂĆ

èğčåEşæŰźæąĹ

Pythonäÿ■èąĺçď žåď ŽèąŇæŢřæ■őçŽĎæăĞåĞEąæŰźåijŘæŸŕäÿĂäÿłçŤśåĚČçżĎæđĎæĹŘçŽĎåžŘåĹŮãĂ

```
stocks = [
    ('GOOG', 100, 490.1),
    ('AAPL', 50, 545.75),
    ('FB', 150, 7.45),
    ('HPQ', 75, 33.2),
]
```

ä¿İæ■őPEP249ïijŇéĂŽè£Ğè£Źçğ■å¡ćåijŘæŘŘä¿ŻæŢřæ■őïijŇ
åŘŕäżěå¿Ĺåőźæ ŸŞçŽĎä¡£çŤĺPythonæăĞåĞEąæŢřæ■őåžŞAPIåŠŇåĚşçşżåđŃæŢřæ■őåžŞè£ŻèąŇäžď äžŠãĂĆ
æĽĂæIJĽæŢřæ■őåžŞäÿŁçŽĎæŞ■ä¡IJéČ¡éĂŽè£ĞSQLæ§ěèŕćèŕ■åŘěæİěåőŃæĹŘãĂĆæŕŘäÿĂèąŇè¿ŞåĚěè¿Ş

äÿžäžĘæijŤçď žèŕť æŸŐïijŇä¡ääŘŕäżěä¡£çŤĺPythonæăĞåĞEąžŞäÿ■çŽĎ `sqlite3`
æĺąåİŮãĂĆ åęĆæđIJä¡ää¡£çŤĺçŽĎæŸŕäÿĂäÿłäÿ■åŘŇçŽĎæŢřæ■őåžŞ(æŕŤåęĆMySqlãĂĄPostgresqlæĹŰèĂ
è£Ÿå¿ŮåőĽèčĚçŻÿåžŤçŽĎçňňäÿĽæŰźæĺąåİŮæİěæŘŘä¿ŻæŤŕæŇĄãĂĆ
äÿ■è£ĞçŻÿåžŤçŽĎçijŰçĺŃæŐěåŘčåĞäźŐéČ¡æŸŕäÿĂæăůçŽĎïijŇéŹď äžĘäÿĂçĆźçĆźçżĘå¿őåůőåĹńåď Ůã

çňňäÿĂæ■ěæŸŕè£đæŐěåĹřæŢřæ■őåžŞãĂĆéĂŽåÿÿä¡ăèęĄæĽğèąŇ `connect()`
åĞ¡æŢřïijŇ çżŹåőČæŘŘä¿ŻäÿĂäžŻæŢřæ■őåžŞåŘ■ãĂĄäÿżæIJžãĂĄçŤĺæĹůåŘ■ãĂĄåŕĘçăĄåŠŇåĚűäżŰå£Ě

```
>>> import sqlite3
>>> db = sqlite3.connect('database.db')
>>>
```

äÿžäžĘåď ĎçŘĘæŢřæ■őïijŇäÿŃäÿĂæ■ěä¡äéIJĂèęĄåĹŻåżžäÿĂäÿłæÿÿæăĞãĂĆ äÿĂæŮęä¡äæIJĽäžĘæÿÿæăĞïijŇéĆčäźĹä¡ääřśåŘŕäżěæĽ ğèąŇSQLæ§ěèŕćèŕ■åŘěäžĘãĂĆæŕŤåęĆïijŽ

```
>>> c = db.cursor()
>>> c.execute('create table portfolio (symbol text, shares integer,␣
↪price real)')
<sqlite3.Cursor object at 0x10067a730>
>>> db.commit()
>>>
```

äÿžäžĘåŘŚæŢřæ■őåžŞèąĺäÿ■æŘŚåĚěåď Žæĺąèőŕå¡ŢïijŇä¡£çŤĺçśżäijijäÿŃéĺćè£ŹæăůçŽĎèŕ■åŘěïijŽ

```
>>> c.executemany('insert into portfolio values (?,?,?)', stocks)
<sqlite3.Cursor object at 0x10067a730>
>>> db.commit()
>>>
```

äÿžäžĘæĽ ğèąŇæ§Řäÿłæ§ěèŕćïijŇä¡£çŤĺåČŘäÿŃéĺćè£ŹæăůçŽĎèŕ■åŘěïijŽ

```
>>> for row in db.execute('select * from portfolio'):
...     print(row)
...
('GOOG', 100, 490.1)
('AAPL', 50, 545.75)
('FB', 150, 7.45)
('HPQ', 75, 33.2)
>>>
```

åęĆæđIJä¡äæČşæŐěåŘŮçŤĺæĹůè¿ŞåĚěä¡IJäÿžåŘĆæŢřæĺěæĽ ğèąŇæ§ěèŕćæŞ■ä¡IJïijŇå£Ěéążçąőä£Ìä¡ää

```
>>> min_price = 100
>>> for row in db.execute('select * from portfolio where price >= ?
↪',
                          (min_price,)):
...     print(row)
...
('GOOG', 100, 490.1)
('AAPL', 50, 545.75)
>>>
```

èőĺèőž

åIJĺæŕŤè¿Čä¡ŐçŽĎçžğåĹńäÿŁåŠŇæŢřæ■őåžŞäžď äžŠæŸŕéĺđåÿÿçőĂå■ŢçŽĎãĂĆ ä¡ääŘłéIJĂæŘŘä¿ŻSQLèŕ■åŘěåźűěřČçŤĺçŻÿåžŤçŽĎæĺąåĺŮåřśåŘŕäżěæŻť æŰřæĹŰæŘŘåŘŰæŢřæ■őäžĘãĂĆ èŹ¡èŕť åęĆæ■ď ïijŇè£ŸæŸŕæIJĽäÿĂäžŻæŕŤè¿ČæčŸæĽ ŃçŽĎçżĘèŁĆéŮőécŸéIJĂèęĄä¡äéĂŘäÿłåĹŮåĞžåĆ

äÿĂäÿłéŽ¿çĆźæŸŕæŢŕæ■ŏåžŞäÿ■çŽĎæŢŕæ■ŏåŠŇPythonçśżåđŃçŻť æŐěçŽĎæŸăåřĎãĂĆ
åŕźäžŐæŮěæIJ§çśżåđŃïijŇéĂŽåÿÿåŔŕäżěä¡£çŤÍ `datetime` æĺąåİŮäÿ■çŽĎ `datetime`
åŏđä¿ŃïijŇ æĹŰèĂĚåŔŕèČ¡æŸŕ `time` æĺąåİŮäÿ■çŽĎçşżçż§æŮűéŮť æĹşãĂĆ
åŕźäžŐæŢŕå■ŮçśżåđŃïijŇçĽźåĹńæŸŕä¡£çŤĺåĹřåřŔæŢřçŽĎéĞŚèđ■æŢŕæ■ŏïijŇåŔŕäżěçŤÍ
`decimal` æĺąåİŮäÿ■çŽĎ `Decimal` åŏđä¿Ńæİěèąĺçď žãĂĆ
äÿ■åźÿçŽĎæŸŕïijŇåŕźäžŐäÿ■åŘŇçŽĎæŢŕæ■ŏåžŞèĂŇèÍĂåĚůä¡ŞæŸăåřĎèğĎåĹŹæŸŕäÿ■äÿĂæăůçŽĎïijŇä¡

åŔęåď ŰäÿĂäÿłæŻť åŁăåď■æİĆçŽĎéŮŏéćŸåřśæŸŕSQLèŕ■åŔěå■ŮçňęäÿšçŽĎæđĎéĂăãĂĆ
ä¡ăå■ČäÿĞäÿ■èęĄä¡£çŤÍPythonå■ŮçňęäÿšæăijåijŔåŇŰæŞ■ä¡IJçňę(åęĆ%)æĹŰèĂĚ
`.format()` æŰźæşŢæİěåĹŻåżžè£ŹæăůçŽĎå■Ůçňęäÿšã ĂĆ
åęĆæđIJäijăéĂŠçżŹè£ŹäžŻæăijåijŔåŇŰæŞ■ä¡IJçňęçŽĎåĂijæİěèĞłäžŐçŤÍæĹůçŽĎè¿ŞåĚěïijŇéĆčäźĹä¡ăçŽ
http://xkcd.com/327)ãĂĆ æ§èŕćèŕ■åŔěäÿ■çŽĎéĂŽéĚ■çňę ?
æŇĞçď žåŘŐåŔřæŢŕæ■ŏåžŞä¡£çŤÍåőČèĞłåůśçŽĎå■ŮçňęäÿšæŻ£æ■ćæIJžåĹűïijŇè£ŹæăůæŻť åŁăçŽĎåőĽ åĺ

äÿ■åźÿçŽĎæŸŕïijŇäÿ■åŘŇçŽĎæŢŕæ■ŏåžŞåŘŐåŔřåŕźäžŐéĂŽéĚ■çňęçŽĎä¡£çŤÍæŸŕäÿ■äÿĂæăůçŽĎã
? æĹŰ `%s` ïijŇ è£ŸæIJĽåĚűäżŰäÿĂäžŻä¡£çŤÍäžĘäÿ■åŘŇçŽĎçňęåŘůïijŇæŕŤåęĆ:0æĹŰ:1æİěæŇĞçď žåŘĆ
åŘŇæăůçŽĎïijŇä¡äè£ŸæŸŕå¿ŮåŐżåŘĆèĂČä¡ää¡£çŤÍçŽĎæŢŕæ■ŏåžŞæĺąåİŮçŻÿåžŤçŽĎæŰĞæąčãĂĆ
äÿĂäÿłæŢŕæ■ŏåžŞæĺąåİŮçŽĎ `paramstyle` åśđæĂğåŇĚåŘńäžĘåŘĆæŢřäijŢçŤÍéčŐæäijçŽĎä£ąæĄŕãĂĆ

åŕźäžŐçőĂå■ŢçŽĎæŢŕæ■ŏåžŞæŢŕæ■ŏçŽĎèŕżåĘźéŮŏéćŸïijŇä¡£çŤÍæŢŕæ■ŏåžŞAPIéĂŽåÿÿéİđåÿÿçőĂ
åęĆæđIJä¡äèęĄåď ĎçŘĘæŻť åŁăåď■æİĆçŽĎéŮŏéćŸïijŇåżżèőőä¡ää¡£çŤÍæŻť åŁăéńŸçžğçŽĎæŐěåŘčïijŇæŕ
çśżäijij `SQLAlchemy` è£ŹæăůçŽĎåžŞåĚĄèőÿä¡ää¡£çŤÍPythonçśżæİěèąĺçď žäÿĂäÿłæŢŕæ■ŏåžŞèąĺïijŇ
åźűäÿŤèČ¡åIJĺéŽŘèŮŔåžŢåśĆSQLçŽĎæČĚåĘţäÿŃåŏđçŐřåŘĎçğ■æŢŕæ■ŏåžŞçŽĎæŞ■ä¡IJãĂĆ

8.9 6.9 çijŰçăĄåŠŇèğčçăĄå■ĄåĚ■è£ŻåĹűæŢř

éŮŏéćŸ

ä¡ăæČşåřĘäÿĂäÿłå■ĄåĚ■è£ŻåĹűå■ŮçňęäÿšèğčçăĄæĹŘäÿĂäÿłå■ŮèŁĆå■ŮçňęäÿšæĹŰèĂĚåřĘäÿĂäÿłå

èğčåĘşæŰźæąĹ

åęĆæđIJä¡ăåŔłæŸŕçőĂå■ŢçŽĎèğčçăĄæĹŰçijŰçăĄäÿĂäÿłå■ĄåĚ■è£ŻåĹűçŽĎåŐ§åğŃå■ŮçňęäÿšïijŇå
æĺąåİŮãĂĆä¿ŃåęĆïijŽ

```
>>> # Initial byte string
>>> s = b'hello'
>>> # Encode as hex
>>> import binascii
>>> h = binascii.b2a_hex(s)
>>> h
b'68656c6c6f'
>>> # Decode back to bytes
>>> binascii.a2b_hex(h)
b'hello'
>>>
```

çśżäijijçŽĎåŁ§èČ¡åŘŇæăůåŔŕäżěåIJÍ `base64` æĺąåİŮäÿ■æĽ¿åĹřãĂĆä¿ŃåęĆïijŽ

```
>>> import base64
>>> h = base64.b16encode(s)
>>> h
b'68656C6C6F'
>>> base64.b16decode(h)
b'hello'
>>>
```

èőÍèőž

åď ğéČÍåĹĘæČĚåĘţäÿŃïijŇéĂŽè£Ğä¡£çŤÍäÿŁè£řçŽĎåĞ¡æŢřæİěè¡ňæ■ćå■ĄåĚ■è£ŻåĹűæŸŕå¿ĹçőĂå■Ţ
äÿŁéİćäÿď çğ■æŁĂæIJŕçŽĎäÿżèęĄäÿ■åŘŇåIJÍäžŐåď ğåřŘåĘźçŽĎåď ĎçŘĘãĂĆ
åĞ¡æŢř `base64.b16decode()` åŠŇ `base64.b16encode()`
åŘłèČ¡æŞ■ä¡IJåď ğåĘźå¡ćåijŘçŽĎå■ĄåĚ■è£ŻåĹűå■Ůæŕ■ïijŇ èĂŇ `binascii`
æÍąåİŮäÿ■çŽĎåĞ¡æŢřåď ğåřŘåĘźéČ¡èČ¡åď ĎçŘĘãĂĆ

è£ŸæIJL'äÿĂçĆźéIJĂèęĄæşÍæĎŘçŽĎæŸŕçijŰçăĄåĞ¡æŢřæL'Ăäžğç Ť§çŽĎè¿ŞåĞžæĂżæŸŕäÿĂäÿłå■Ů
åęĆæđIJæČşåijžåĹűäžěUnicodeå¡ćåijŘè¿ŞåĞžïijŇä¡ăéIJĂèęĄåćđåŁääÿĂäÿłéćİåď ŰçŽĎçŢŇéİćæ■ěéłď ãĂĆ

```
>>> h = base64.b16encode(s)
>>> print(h)
b'68656C6C6F'
>>> print(h.decode('ascii'))
68656C6C6F
>>>
```

åIJÍèğčçăĄå■ĄåĚ■è£ŻåĹűæŢřæŮűïijŇåĞ¡æŢř `b16decode()`
åŠŇ `a2b_hex()` åŘŕäžěæŐěåŘŮå■ŮèŁĆæĹŰunicodeå■Ůçňęäÿšã ĂĆ
ä¡ĘæŸŕïijŇunicodeå■Ůçňęäÿšå£ĚéążäžĚäžĚåŘłåŇĚåŘńASCIIçijŰçăĄçŽĎå■ĄåĚ■è£ŻåĹűæŢřã ĂĆ

8.10 6.10 çijŰçăĄèğčçăĄBase64æŢřæ■ő

éŮőécŸ

ä¡ăéIJĂèęĄä¡£çŤÍBase64æăijåijŘèğčçăĄæĹŰçijŰçăĄäžŇè£ŻåĹűæŢřæ■őã ĂĆ

èğčåĘşæŰźæąĹ

`base64` æÍąåİŮäÿ■æIJL'äÿď äÿłåĞ¡æŢř `b64encode()` and `b64decode()`
åŘŕäžěåÿőä¡äèğčåĘşè£ŹäÿłéŮőécŸã ĂĆä¿ŃåęĆ;

```
>>> # Some byte data
>>> s = b'hello'
>>> import base64
```

(continues on next page)

(continued from previous page)

```
>>> # Encode as Base64
>>> a = base64.b64encode(s)
>>> a
b'aGVsbG8='

>>> # Decode from Base64
>>> base64.b64decode(a)
b'hello'
>>>
```

èőÍèőž

Base64çijŰçăĄäżĚäżĚçŤÍäžŐéÍćåŘŚå■ŮèŁÇçŽĎæŢřæ■őæŕŤåęĆå■ŮèŁĆå■ŮçňęäÿšåŠŇå■ŮèŁĆæŢřçż
æ■ď'åď'ŰïijŇçijŰçăĄåď'ĎçŘĘçŽĎè¿ŞåĞžçżŞæđIJæĂżæŸŕäÿĂäÿłå■ŮèŁĆå■ŮçňęäÿšãĂĆ
åęĆæđIJä¡äæČşæůůåŘĹä¡£çŤÍBase64çijŰçăĄçŽĎæŢřæ■őåŠŇUnicodeæŰĞæIJňïijŇä¡åå£ĚéążæůżåŁääÿĂä

```
>>> a = base64.b64encode(s).decode('ascii')
>>> a
'aGVsbG8='
>>>
```

å¡ŜèğčçăĄ Base64çŽĎæŮűåĂŹïijŇå■ŮèŁĆå■ŮçňęäÿšåŠŇUnicodeæŰĞæIJňéČ¡åŘŕäżěä¡IJäÿžåŘĆæŢřä
ä¡ĘæŸŕïijŇUnicodeå■ŮçňęäÿšåŘłèČ¡åŇĚåŘńASCIIå■ŮçňęäĂĆ

8.11 6.11 èŕżåĘŹäžŇè£ŻåĹűæŢřçżĎæŢřæ■ő

éŮőéćŸ

ä¡äæČşèŕżåĘŹäÿĂäÿłäžŇè£ŻåĹűæŢřçżĎçŽĎçżŞæđĎåŇŰæŢřæ■őåĹřPythonåĚČçżĎäÿ■ãĂĆ

èğčåĘşæŰźæąĹ

åŘŕäżěä¡£çŤÍ struct æÍąåÍŮåď'ĎçŘĘäžŇè£ŻåĹűæŢřæ■õãĂĆ
äÿŇéÍćæŸŕäÿĂæőţçď'žä¿ŃäżčçăĄåřĘäÿĂäÿłPythonåĚČçżĎåĹŮèąĺåĘŹåĚěäÿĂäÿłäžŇè£ŻåĹűæŰĞäżűïijŇåż
struct åřĘæŕŘäÿłåĚČçżĎçijŰçăĄäÿžäÿĂäÿłçżŞæđĎä¡ŞãĂĆ

```
from struct import Struct
def write_records(records, format, f):
    '''
    Write a sequence of tuples to a binary file of structures.
    '''
    record_struct = Struct(format)
    for r in records:
        f.write(record_struct.pack(*r))
```

(continues on next page)

(continued from previous page)

```
# Example
if __name__ == '__main__':
    records = [ (1, 2.3, 4.5),
                (6, 7.8, 9.0),
                (12, 13.4, 56.7) ]
    with open('data.b', 'wb') as f:
        write_records(records, '<idd', f)
```

æIJL'å¿Ĺåď'Žçğ■æŰźæşŢæİěèŕżåŔŰè£ŹäÿłæŰĞäżűåźűè£ŤåŻđäÿĂäÿłåĚČçżĎåĹŮèąĺãĂĆ
éęŰåĚĹïijŇåęĆæđIJä¡ăæL'ŞçőŮäżěåİŮçŽĎå¡ćåijŘåćđéĞŘèŕżåŔŰæŰĞäżűïijŇä¡ăåŘŕäżěè£ŹæăůåĄŽïijŽ

```
from struct import Struct

def read_records(format, f):
    record_struct = Struct(format)
    chunks = iter(lambda: f.read(record_struct.size), b'')
    return (record_struct.unpack(chunk) for chunk in chunks)

# Example
if __name__ == '__main__':
    with open('data.b','rb') as f:
        for rec in read_records('<idd', f):
            # Process rec
            ...
```

åęĆæđIJä¡ăæČşåřĘæŢť'äÿłæŰĞäżűäÿĂæňąæĂğèŕżåŔŰåĹřäÿĂäÿłå■ŮèŁĆå■Ůçňęäÿšäÿ■ïijŇçĎűåŘŐåĹ

```
from struct import Struct

def unpack_records(format, data):
    record_struct = Struct(format)
    return (record_struct.unpack_from(data, offset)
            for offset in range(0, len(data), record_struct.size))

# Example
if __name__ == '__main__':
    with open('data.b', 'rb') as f:
        data = f.read()
    for rec in unpack_records('<idd', data):
        # Process rec
        ...
```

äÿď'çğ■æČĚåĘţäÿŃçŽĎçżŞæđIJéČ¡æŸŕäÿĂäÿłåŔŕè£ŤåŻđçŤĺæİěåĹŻåżżèŕěæŰĞäżűçŽĎåŐ§åğŃåĚČçż

èőĺèőž

åŕźäžŐéIJĂèęĄçijŰçăĄåŠŇèğčçăĄäžŇè£ŻåĹűæŢŕæ■őçŽĎçĺŃåžŘèĂŇèĺĂïijŇéĂŽåÿÿäijŽä¡£çŤĺ
`struct` æĺąåİŮãĂĆ äÿžäžĘåčŕæŸŐäÿĂäÿłæŰřçŽĎçżŞæđĎä¡ŞïijŇåŘłéIJĂèęĄåČŘè£ŹæăůåĹŻåżžäÿĂäÿł

Struct åðđä¿Ňå■şåŘřijŽ

```
# Little endian 32-bit integer, two double precision floats
record_struct = Struct('<idd')
```

çżŞæđĎä¡ŞéĂŽåÿÿäijŽä¡£çŤÍäÿĂäžŻçżŞæđĎçăĄåĂiji, d, fç■L' [åŘĆèĂČ PythonæŰĞæąč]ãĂĆ è£ŹäžŻäżčçăĄåĹĘåĹńäżčèąĺæ§Řäÿłç L'źåŐŽçŽĎäžŇè£ŻåĹűæŢřæ■őçśżåđŇåęĆ32ä¡■a çňňäÿĂäÿłå■Ůçňę < æŇĞåŐŽäžĘå■ŮèŁĆéążåžŘãĂĆåIJÍè£ŹäÿłäĺŇå■Řäÿ■ïijŇåőČèąĺçď'žâĂİä¡Őä¡■åIJÍåL'■ æŽť'æŤźè£ŹäÿłäĺŇå■Ůçňęäÿž > èąĺçď'žéńŸä¡■åIJÍåL'■ïijŇæĹŰèĂĚæŸŕ ! èąĺçď'žç¡ŚçżIJå■ŮèŁĆéążåžŘãĂĆ

äžğçŤ§çŽĎ Struct åðđä¿ŃæIJL'å¿Ĺåď'ŽåśđæĂğåŠŇæŰźæşŢçŤÍæİěæŞ■ä¡IJçŻÿåžŤçśżåđŃçŽĎçżŞæđ size åśđæĂğåŇĚåŘńäžĘçżŞæđĎçŽĎå■ŮèŁĆæŢřïijŇè£ŹåIJÍI/OæŞ■ä¡IJæŮűéİđåÿÿæIJL'çŤÍãĂĆ pack() åŠŇ unpack() æŰźæşŢèćńçŤÍæİěæL'ŞåŇĚåŠŇèğčåŇĚæŢřæ■őãĂĆæŕŤåęĆïijŽ

```
>>> from struct import Struct
>>> record_struct = Struct('<idd')
>>> record_struct.size
20
>>> record_struct.pack(1, 2.0, 3.0)
b
↪'\x01\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00@\x00\x00\x00\x00\x00\x00\x08@
↪'
>>> record_struct.unpack(_)
(1, 2.0, 3.0)
>>>
```

æIJL'æŮűåĂŹä¡ăè£ŸäijŽçIJŃåĹř pack() åŠŇ unpack() æŞ■ä¡IJäżěæĺąåİŮçžğåĹńåĞ¡æŢřèćńèřČçŤÍïijŇçśżäijijäÿŃéİćè£ŹæăůïijŽ

```
>>> import struct
>>> struct.pack('<idd', 1, 2.0, 3.0)
b
↪'\x01\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00@\x00\x00\x00\x00\x00\x00\x08@
↪'
>>> struct.unpack('<idd', _)
(1, 2.0, 3.0)
>>>
```

è£ŹæăůåŘŕäżěåůëä¡IJïijŇä¡ĘæŸŕæĐ§èğL'æšąæIJL'åðđä¿ŃæŰźæşŢéĆčäźĹäijŸéŻĚïijŇçL'źåĹńæŸŕåIJÍä éĂŽè£ĞåĹŻåżžäÿĂäÿł Struct åðđä¿ŃïijŇæăijåijŘäżčçăĄåŘłäijŽæŃĞåŐŽäÿĂæňąåźűäÿŤæL'ĂæIJL'çŽĎæ è£ŹæăůäÿĂæİěäżčçăĄçżť'æŁď'åŕśåŘŸå¿ŮæŽť'åŁăçőĂå■ŢäžĘ(åŽääÿžä¡ăåŘłéIJĂèęĄæŤźåŘŸäÿĂåď'Ďäżčçă

èŕżåŘŰäžŇè£ŻåĹűçżŞæđĎçŽĎäżčçăĄèęĄçŤÍåĹřäÿĂäžŻéİđåÿÿæIJL'èűčèĂŃäijŸç¿ŐçŽĎçijŰçÍŃæŁĂ åIJÍåĞ¡æŢřãĂĂread_records äÿ■ïijŇiter() èćńçŤÍæİěåĹŽåżžäÿĂäÿłè£ŤåŽđåŽžåőŽåď'ğåŕŘæŢřæ■őå è£ŹäÿłèŁ■äżčåŹÍäijŽäÿ■æŰ■çŽĎèřČçŤÍäÿĂäÿłçŤÍæĹůæŘŘä¿ŻçŽĎåŘŕèřČçŤÍåŕźèśą(æŕŤåęĆ lambda: f.read(record_struct.size))ïijŇ çŽť'åĹřőČè£ŤåŽđäÿĂäÿłçL'źæőŁçŽĎåĂij(åęĆbâ

```
>>> f = open('data.b', 'rb')
>>> chunks = iter(lambda: f.read(20), b'')
>>> chunks
```

(continues on next page)

(continued from previous page)

```
<callable_iterator object at 0x10069e6d0>
>>> for chk in chunks:
... print(chk)
...
b'\x01\x00\x00\x00ffffff\x02@\x00\x00\x00\x00\x00\x00\x12@'
b'\x06\x00\x00\x00333333\x1f@\x00\x00\x00\x00\x00\x00"@'
b'\x0c\x00\x00\x00\xcd\xcc\xcc\xcc\xcc\xcc*@\x9a\x99\x99\x99\x99YL@'
>>>
```

åęĆä¡ăæL'ĂèğĄïijŇåĹŻåżžäÿĂäÿłåŕŕè£■äżčåŕźèśąçŽĎäÿĂäÿłåŐ§åŻăæŸŕåőČèČ¡åĚĄèőÿä¡£çŤĺäÿĂäÿ
åęĆæđIJä¡ääÿ■ä¡£çŤĺè£Źçğ■æŁĂæIJŕïijŇéĆčäźĹäżčçăĄåŕŕèČ¡äijŽåČŕäÿŃéİćè£Źæăůïijž

```
def read_records(format, f):
    record_struct = Struct(format)
    while True:
        chk = f.read(record_struct.size)
        if chk == b'':
            break
        yield record_struct.unpack(chk)
```

åIJĺåĞ¡æŢř `unpack_records()` äÿ■ä¡£çŤĺäžĘåŕęåď'ŰäÿĂçğ■æŰźæşŢ `unpack_from()` ãĂĆ `unpack_from()` åŕźäžŐäżŐäÿĂäÿłåď'ğåđŃäžŇè£ŻåĹűæŢřçżĎäÿ■æŔŘåŔŰäžŇè
åŻääÿžåőČäÿ■äijŽäžğçŤ§äżżä¡ŢçŽĎäÿť'æŮűåŕźèśąæĹŰèĂĚè£ŻèąŇåĘĚå■Ÿåď'■åĹűæŞ■ä¡IJãĂĆ
ä¡ăåŔłéIJĂèęĄçżŹåőČäÿĂäÿłå■ŮèŁĆå■Ůçňęäÿš(æĹŰæŢřçżĎ)åŠŇäÿĂäÿłå■ŮèŁĆåĄŕçğżéĞŔïijŇåőČäijŽä

åęĆæđIJä¡ää¡£çŤĺ `unpack()` æİěäžčæŻ£ `unpack_from()` ïijŇ
ä¡äéIJĂèęĄä£őæŤźäżčçăĄæİěæđĎéĂäåď'ğéĞŔçŽĎåřŔçŽĎåĹĞçL'ĞäżěåŔŁè£ŻèąŇåĄŕçğżéĞŔçŽĎèőąçőŮ

```
def unpack_records(format, data):
    record_struct = Struct(format)
    return (record_struct.unpack(data[offset:offset + record_struct.
↪size])
            for offset in range(0, len(data), record_struct.size))
```

è£Źçğ■æŰźæąĹéŹď'äžĘäżčçăĄçIJŃäÿŁåŐżå¿Ĺåď'■æİĆåď'ŰïijŇè£Ÿå¿ŮåĄŽå¿Ĺåď'Žéćİåď'ŰçŽĎåůěä¡
åď'■åĹűæŢřæ■őäżěåŔŁæđĎéĂäåřŔçŽĎåĹĞçL'ĞåŕźèśąăĂĆ åęĆæđIJä¡äåĞĘåď'ĞäżŐèŕżåŔŰåĹřçŽĎäÿĂäÿłå
äijŽèąĺçŐřçŽĎæŻť'åĞžèL'šăĂĆ

åIJĺèğčåŇĚçŽĎæŮűåĂŹïijŇcollections æĺąåİŮäÿ■çŽĎåŚ¡åŘ■åĚČçżĎåŕźèśąæĹŰèőÿæŸŕä¡ăæČşè
åőČåŔŕäżěèőľ'ä¡äçżŹè£ŤåŻđåĚČçżĎèő¿ç¡őåśđæĂğåŘ■çğřăĂĆä¿ŃåęĆïijž

```
from collections import namedtuple

Record = namedtuple('Record', ['kind','x','y'])

with open('data.p', 'rb') as f:
    records = (Record(*r) for r in read_records('<idd', f))

for r in records:
    print(r.kind, r.x, r.y)
```

åęĆæđIJä¡ăçŽĎçÍŃåžŘéIJĂèęĄåď ĎçŘĘåď ğéĞŘçŽĎäžŇè£ŻåĹűæŢřæ■őïijŇä¡ăæIJĂåě¡ä¡£çŤÍ numpy æĺąåİŮãĂĆ ä¿ŃåęĆïijŇä¡ăåŘŕäżěåřĘäÿĂäÿłäžŇè£ŻåĹűæŢřæ■őèŕżåŘŰåĹřäÿĂäÿłçżŞæđĎåŇŰæŢřçż

```
>>> import numpy as np
>>> f = open('data.b', 'rb')
>>> records = np.fromfile(f, dtype='<i,<d,<d')
>>> records
array([(1, 2.3, 4.5), (6, 7.8, 9.0), (12, 13.4, 56.7)],
dtype=[('f0', '<i4'), ('f1', '<f8'), ('f2', '<f8')])
>>> records[0]
(1, 2.3, 4.5)
>>> records[1]
(6, 7.8, 9.0)
>>>
```

æIJĂåŘŐæŘŘäÿĂçĆźïijŇåęĆæđIJä¡ăéIJĂèęĄäżŐåůšç§ěçŽĎæŰĞäżűæăijåijŘ(åęĆåŻ¿çL'ĞæăijåijŘïijŇ åĚĹæčĂæ§ěçIJŃçIJŃPythonæŸŕäÿ■æŸŕåůšçżŘæŘŘä¿ŻäžEçŐŕå■ŸçŽĎæĺąåİŮãĂĆåŻääÿžäÿ■åĹřäÿĞäÿ■å¿

8.12 6.12 èŕżåŘŰåţŇåěŮåŠŇåŘŕåŘŸéŢ£äžŇè£ŻåĹűæŢřæ■ő

éŮőéćŸ

ä¡ăéIJĂèęĄèŕżåŘŰåŇĚåŘŕåţŇåěŮæĹŰèĂĚåŘŕåŘŸéŢ£èőŕå¡ŢéŻęåŘĹçŽĎåď■æİĆäžŇè£ŻåĹűæăijåijŘ

èğčåEşæŰźæąĹ

struct æĺąåİŮåŘŕèćńçŤĺæİěçijŰçăĄ/èğčçăĄąåĞääźŐæL'ĂæIJL'çśżåđŃçŽĎäžŇè£ŻåĹűçŽĎæŢřæ■őçż æİěèąĺçď žäÿĂäÿłçżĎæĹŘäÿĂçşżåĹŮåď Žè¿źå¡ćçŽĎçĆźçŽĎéŻęåŘĹïijŽ

```
polys = [
    [ (1.0, 2.5), (3.5, 4.0), (2.5, 1.5) ],
    [ (7.0, 1.2), (5.1, 3.0), (0.5, 7.5), (0.8, 9.0) ],
    [ (3.4, 6.3), (1.2, 0.5), (4.6, 9.2) ],
]
```

çŐřåIJĺåĄĞèő¿è£ŹäÿłæŢřæ■őèćńçijŰçăĄąåĹřäÿĂäÿłäżěäÿŃåĹŮåď ťéČĺåijĂąğŃçŽĎäžŇè£ŻåĹűæŰĞäż

```
+------+--------+-------------------------------------+
|Byte  | Type   |  Description                        |
+======+========+=====================================+
|0     | int    |  æŰĞäżűäżčçăĄïijĹ0x1234ïijŇåřŘçńŕïijL'          |
+------+--------+-------------------------------------+
|4     | double |  x çŽĎæIJĂåřŘåĂijïijĹåřŘçńŕïijL'                |
+------+--------+-------------------------------------+
|12    | double |  y çŽĎæIJĂåřŘåĂijïijĹåřŘçńŕïijL'                |
+------+--------+-------------------------------------+
|20    | double |  x çŽĎæIJĂåď'ğåĂijïijĹåřŘçńŕïijL'                |
```

(continues on next page)

(continued from previous page)

```
+------+--------+-------------------------------------+
|28    | double |  y çŽĎæIJĂåd'ğåĂijïijĹåřŘçńŕïijL'                |
+------+--------+-------------------------------------+
|36    | int    |  äÿL'èğŠå¡ćæŢřéĞŘïijĹåřŘçńŕïijL'                |
+------+--------+-------------------------------------+
```

çť'ğèů§çİĂåď'ť'éČĺæŸŕäÿĂçşżåĹŮçŽĎåď'Žè¿źå¡ćèőřå¡ŢïijŇçijŰçăĄæăijåijŘåęĆäÿŃïijŽ

```
+------+--------+-------------------------------------------+
|Byte  | Type   |  Description                              |
+======+========+===========================================+
|0     | int    |  èőřå¡ŢéŢ£åžęïijĹNå■ŮèŁĆïijL'                 ␣
↪  |
+------+--------+-------------------------------------------+
|4-N   | Points |  (X,Y) åİŘæăĞïijŇäžěæţőçĆźæŢřèąĺçd'ž        ␣
↪   |
+------+--------+-------------------------------------------+
```

äÿžäžĘåĘŹè£ŹæăůçŽĎæŰĞäżűïijŇä¡ăåŘŕäžěä¡£çŤĺåęĆäÿŃçŽĎPythonäžčçăĄïijŽ

```
import struct
import itertools

def write_polys(filename, polys):
    # Determine bounding box
    flattened = list(itertools.chain(*polys))
    min_x = min(x for x, y in flattened)
    max_x = max(x for x, y in flattened)
    min_y = min(y for x, y in flattened)
    max_y = max(y for x, y in flattened)
    with open(filename, 'wb') as f:
        f.write(struct.pack('<iddddi', 0x1234,
                            min_x, min_y,
                            max_x, max_y,
                            len(polys)))
        for poly in polys:
            size = len(poly) * struct.calcsize('<dd')
            f.write(struct.pack('<i', size + 4))
            for pt in poly:
                f.write(struct.pack('<dd', *pt))
```

ářĘæŢřæ■őèŕżåŘŰåŽđæĺěçŽĎæŮűåĂŹïijŇåŘŕäžěåĹľçŤĺåĞ¡æŢř `struct.unpack()` ïijŇäžčçăĄå¿ĹçŽÿäijijïijŇå§žæIJňářśæŸŕäÿŁéĺćåĘŹæŞ■ä¡IJçŽĎéĂĘåžŘãĂĆåęĆäÿŃïijŽ

```
def read_polys(filename):
    with open(filename, 'rb') as f:
        # Read the header
        header = f.read(40)
        file_code, min_x, min_y, max_x, max_y, num_polys = \
```

(continues on next page)

(continued from previous page)

```
            struct.unpack('<iddddi', header)
        polys = []
        for n in range(num_polys):
            pbytes, = struct.unpack('<i', f.read(4))
            poly = []
            for m in range(pbytes // 16):
                pt = struct.unpack('<dd', f.read(16))
                poly.append(pt)
            polys.append(poly)
    return polys
```

ář¡çőąè£ŹäÿłäžčçăĄåŘŕäžěåůěä¡IJïijŇä¡ĘæŸŕéĞŇéİćæůůæİĆäžĘą¿Ĺåď Žèŕżå ŘŰãĂĄèğčåŇĚæŢřæ■őçž
éĆčæIJłåĚ■äź§åď łçźĄæİĆäžĘçĆźãĂĆåŻäæ■ď å¿ĹæŸ¿çĎűäžŤèŕěæIJĽ åŘęäÿĂçğ■èğčåĘşæŰźæşŢåŘŕäžěçő

åIJÍæIJňářŘèŁĆæŐěäÿŃæİěçŽĎéČÍåĹĘïijŇæĹŚäijŽéĂŘæ■ěæijŤçď žäÿĂäÿłæŻť åŁääijŸçğĂçŽĎèğčæć
çŻőæăĞæŸŕåŘŕäžěçżŹçÍŃåžŘåŚŸæŘŘä¿ŻäÿĂäÿłéńŸçžğçŽĎæŰĞäżűæăijåijŘåŇŰæŰźæşŢïijŇåźűçőĂåŇŰ
æIJňářŘèŁĆæŐěäÿŃæİěçŽĎéČÍåĹĘäžčçăĄåžŤèŕěæŸŕæŢť æIJňäźęäÿ■æIJĂåď■æİĆæIJĂéńŸçžğçŽĎä¿Ńå■
äÿĂåőŽèęĄäżŤçżĘçŽĎéŸĚèŕżæĹŚäžňçŽĎèőÍèőžéČÍåĹĘïijŇåŘęåď Űäź§èęĄåŘĆèĂČäÿŃåĚűäżŰçńäèŁĆåĘ

éęŰåĚĹïijŇå¡ŞèŕżåŘŰå■ŮèŁĆæŢřæ■őçŽĎæŮűåĂŹïijŇéĂŽåÿÿåIJÍæŰĞäżűåijĂåğŃéČÍåĹĘäijŽåŇĚåŘ
ář¡çőąstructæÍąåİŮåŘŕäžěèğčåŇĚè£ŹäžŻæŢřæ■őåĹřäÿĂäÿłåĚČçżĎäÿ■åŐżïijŇåŘęåď ŰäÿĂçğ■èąÍçď žè£Źçğ
ářśåČŘäÿŃéİćè£ŹæăůïijŽ

```
import struct

class StructField:
    '''
    Descriptor representing a simple structure field
    '''
    def __init__(self, format, offset):
        self.format = format
        self.offset = offset
    def __get__(self, instance, cls):
        if instance is None:
            return self
        else:
            r = struct.unpack_from(self.format, instance._buffer,␣
↪self.offset)
            return r[0] if len(r) == 1 else r

class Structure:
    def __init__(self, bytedata):
        self._buffer = memoryview(bytedata)
```

è£ŹéĞŇæĹŚäžňä¡£çŤÍäžĘäÿĂäÿłæŘŘè£řåŹÍæİěèąÍçď žæŕŘäÿłçżŞæđĎå■ŮæőţïijŇæŕŘäÿłæŘŘè£řåŹÍåŇ
å■ŸåĆÍåIJÍåĘĚéČÍçŽĎåĘĚå■ŸçijŞåĘšäÿ■ãĂĆåIJÍ `__get__()` æŰźæşŢäÿ■ïijŇ`struct.unpack_from()` åĞ¡æŢřèćńçŤÍæİěäżŐçijŞåĘšäÿ■èğčåŇĚäÿĂäÿłåĂijïijŇçIJĄåŐżäžĘéćİåď ŰçŽĎåĹĘçĹ'Č

`Structure` çśżåřśæŸŕäÿĂäÿłå§žçąĂçśżïijŇæŐěåŘŮå■ŮèŁĆæŢřæ■őåźűå■ŸåĆÍåIJÍåĘĚéČÍçŽĎåĘĚå■
`StructField` æŘŘè£řåŹÍä¡£çŤÍãĂĆ è£ŹéĞŇä¡£çŤÍäžĘ `memoryview()`

ïijŇæĹŚäžňäijŽåIJĺåŘŐéİćèŕęçżĘèőšèğčåőČæŸŕçŤĺæİěåźšåŸŻçŽĎãĂĆ

ä¡£çŤĺè£Źäÿłäžčçă Ąïij Ňä¡ăçŐŕåIJĺåřśèČ¡åőŽäźĽäÿĂäÿłéńŸåśĆæňąçŽĎçżŞæđĎåŕźèśąæİěèąĹçď žäÿŁéİ

```
class PolyHeader(Structure):
    file_code = StructField('<i', 0)
    min_x = StructField('<d', 4)
    min_y = StructField('<d', 12)
    max_x = StructField('<d', 20)
    max_y = StructField('<d', 28)
    num_polys = StructField('<i', 36)
```

äÿŃéİćçŽĎä¿Ńå■ŘåĹľçŤĺè£ŹäÿłçśżæİěèŕżåŘŰäźŃåĽ■æĹŚäžňåĘŹåĚěçŽĎåď Žè¿źå¡ćæŢřæ■őçŽĎåď ť

```
>>> f = open('polys.bin', 'rb')
>>> phead = PolyHeader(f.read(40))
>>> phead.file_code == 0x1234
True
>>> phead.min_x
0.5
>>> phead.min_y
0.5
>>> phead.max_x
7.0
>>> phead.max_y
9.2
>>> phead.num_polys
3
>>>
```

è£Źäÿłå¿ĹæIJĽèűčïijŇäÿ■è£Ğè£Źçğ■æŰźåijŔè£ŸæŸŕæIJĽäÿĂäžŻçČęäžžçŽĎåIJřæŰźãĂĆéęŰåĚĹïij
ä¡ĘæŸŕè£Źäÿłäžčçă Ąè£ŸæŸŕæIJĽçĆźèĞČèĆ£ïijŇè£ŸéIJĂèęĄä¡£çŤĺèĂĚæŇĞåőŽå¿Ĺåď ŽåžŢåśĆçŽĎçżĘ
`StructField` ïijŇæŇĞåőŽåĄŔçğżéĞŔç■Ľ)ãĂĆ åŔęåď ŰïijŇè£ŤåŻđçŽĎçżŞæđIJçśżåŘŇæăůçąőåőđäÿĂä

äżżä¡ŢæŮűåĂŹåŔłèęĄä¡ăéĄĞåĹřäžĘåČŔè£ŹæăůåĘŮä¡ŹçŽĎçśżåőŽäźĽïijŇä¡ăåžŤèŕěèĂČèŹŚäÿŃä¡£ç
åĚČçśżæIJĽäÿĂäÿłçĽźæĂğåřśæŸŕåőČèČ¡åď §èćńçŤĺæİěåąńåĚĚèőÿåď Žä¡ŐåśĆçŽĎåőđçŐřçżĘèŁĆïijŇäżŐ
äÿŃéİćæĹŚæİěäÿ¿äÿłä¿Ńå■ŘïijŇä¡£çŤĺåĚČçśżçĺ■å¿őæŤźéĂääÿŃæĹŚäžňçŽĎ `Structure`
çśżïijŽ

```
class StructureMeta(type):
    '''
    Metaclass that automatically creates StructField descriptors
    '''
    def __init__(self, clsname, bases, clsdict):
        fields = getattr(self, '_fields_', [])
        byte_order = ''
        offset = 0
        for format, fieldname in fields:
            if format.startswith(('<','>','!','@')):
                byte_order = format[0]
                format = format[1:]
```

(continues on next page)

(continued from previous page)

```
            format = byte_order + format
            setattr(self, fieldname, StructField(format, offset))
            offset += struct.calcsize(format)
        setattr(self, 'struct_size', offset)

class Structure(metaclass=StructureMeta):
    def __init__(self, bytedata):
        self._buffer = bytedata

    @classmethod
    def from_file(cls, f):
        return cls(f.read(cls.struct_size))
```

ä¡£çŤÍæŰřçŽĎ `Structure` çśżïijŇä¡ăåŘŕäžěåČŘäÿŃéİćè£ŹæăůåőŽäźĽäÿĂäÿłçżŞæđĎïijŽ

```
class PolyHeader(Structure):
    _fields_ = [
        ('<i', 'file_code'),
        ('d', 'min_x'),
        ('d', 'min_y'),
        ('d', 'max_x'),
        ('d', 'max_y'),
        ('i', 'num_polys')
    ]
```

æ■čåęĆä¡äæĽĂèğĄïijŇè£ŹæăůåĘŹåřśçőĂå■Ţåď ŽäžĘãĂĆæĹŚäžňæůżåŁăçŽĎçśżæŰźæşŢ `from_file()` èőľæĹŚäžňåIJÍäÿ■éIJĂèęĄç§ěéĄŞäżżä¡ŢæŢřæ■őçŽĎåď ğåřŘåŠŇçżŞæđĎçŽĎæČĚåĘţäÿŃ

```
>>> f = open('polys.bin', 'rb')
>>> phead = PolyHeader.from_file(f)
>>> phead.file_code == 0x1234
True
>>> phead.min_x
0.5
>>> phead.min_y
0.5
>>> phead.max_x
7.0
>>> phead.max_y
9.2
>>> phead.num_polys
3
>>>
```

äÿĂæŮęä¡ăåijĂåğŃä¡£çŤÍäžĘåĚČçśżïijŇä¡ăåřśåŘŕäžěèőľ åőČåŘŸå¿ŮæŻť åŁăæŹżèČ¡ãĂĆä¿ŃåęĆïijŇå äÿŃéİćæŸŕåŕźåĽ■éİćåĚČçśżçŽĎäÿĂäÿłåřŘçŽĎæŤźè£ŻïijŇæŘŘä¿ŻäžĘäÿĂäÿłæŰřçŽĎè¿ĚåŁľæŘŘè£řåŹĺa

```
class NestedStruct:
    '''
```

(continues on next page)

(continued from previous page)

```
    Descriptor representing a nested structure
    '''
    def __init__(self, name, struct_type, offset):
        self.name = name
        self.struct_type = struct_type
        self.offset = offset

    def __get__(self, instance, cls):
        if instance is None:
            return self
        else:
            data = instance._buffer[self.offset:
                                    self.offset+self.struct_type.struct_
↪size]
            result = self.struct_type(data)
            # Save resulting structure back on instance to avoid
            # further recomputation of this step
            setattr(instance, self.name, result)
            return result

class StructureMeta(type):
    '''
    Metaclass that automatically creates StructField descriptors
    '''
    def __init__(self, clsname, bases, clsdict):
        fields = getattr(self, '_fields_', [])
        byte_order = ''
        offset = 0
        for format, fieldname in fields:
            if isinstance(format, StructureMeta):
                setattr(self, fieldname,
                        NestedStruct(fieldname, format, offset))
                offset += format.struct_size
            else:
                if format.startswith(('<','>','!','@')):
                    byte_order = format[0]
                    format = format[1:]
                format = byte_order + format
                setattr(self, fieldname, StructField(format,␣
↪offset))
                offset += struct.calcsize(format)
        setattr(self, 'struct_size', offset)
```

åIJÍè£ŹæőţäžčçăĄäÿ■ïijŇNestedStruct æŘŘè£řåŹÍèćńçŤÍæÌěåŘäåŁäåŘęåď ŰäÿĂäÿłåőŽäźĽåIJÍæ§
åőČéĂŽè£ĞåřĘåŐ§åğŃåĘĚå■ŸçijŞåĘšè£ŻèąŇåĹĞçĽĞæŞ■ä¡IJåŘŐåőđä¿ŃåŇŰçżŹåőŽçŽĎçżŞæđĎçśżåđŃ
æĽĂäżěè£Źçğ■åĹĞçĽĞæŞ■ä¡IJäÿ■äijŽåijŢåŘŚäżżä¡ŢçŽĎéćÌåď ŰçŽĎåĘĚå■Ÿÿÿåď ■åĹűãĂĆçŻÿåŘ■ïijŇåőČ
åŘęåď ŰïijŇäÿžäžĘéŸšæ■ćéĞ■åď ■åőđä¿ŃåŇŰïijŇéĂŽè£Ğä¡£çŤÍåŠŇ8.10åřŘèŁĆåŘŇæăůçŽĎæŁĂæIJŕïijŇ

ä¡£çŤÍè£ŹäÿłæŰřçŽĎä£őæ■čçĽĹïijŇä¡ăářśåŘŕäżěåČŘäÿŃéÌćè£ŹæăůçijŰåĘŹïijŽ

```
class Point(Structure):
    _fields_ = [
        ('<d', 'x'),
        ('d', 'y')
    ]

class PolyHeader(Structure):
    _fields_ = [
        ('<i', 'file_code'),
        (Point, 'min'), # nested struct
        (Point, 'max'), # nested struct
        ('i', 'num_polys')
    ]
```

äžd’äžžæČŁèőűçŽĎæŸŕïijŇåőČäź§èČ¡æŇL’çĚğéćĎæIJ§çŽĎæ■čåÿÿåůëä¡IJïijŇæĹŚäžňåőđéŹĚæŞ■ä¡IJ

```
>>> f = open('polys.bin', 'rb')
>>> phead = PolyHeader.from_file(f)
>>> phead.file_code == 0x1234
True
>>> phead.min # Nested structure
<__main__.Point object at 0x1006a48d0>
>>> phead.min.x
0.5
>>> phead.min.y
0.5
>>> phead.max.x
7.0
>>> phead.max.y
9.2
>>> phead.num_polys
3
>>>
```

åĹřçŻőåL’■äÿžæ■ćïijŇäÿĂäÿłåd’ĎçŘĘåőŽéŢ£èőřå¡ŢçŽĎæąĘæđűåůšçżŘåĘŹåě¡äžĘãĂĆä¡ĘæŸŕåęĆæđI
æŕŤåęĆïijŇåd’Žè¿źå¡ćæŰĞäžűåŇĚåŘńåŘŸéŢ£çŽĎéČĺåĹĘãĂĆ

äÿĂçğ■æŰźæąĹæŸŕåĘŹäÿĂäÿłçśżæİěèąĺçd’žå■ŮèŁĆæŢřæ■őïijŇåŘŇæŮűåĘŹäÿĂäÿłåůěåĚůåĞ¡æŢřæl

```
class SizedRecord:
    def __init__(self, bytedata):
        self._buffer = memoryview(bytedata)

    @classmethod
    def from_file(cls, f, size_fmt, includes_size=True):
        sz_nbytes = struct.calcsize(size_fmt)
        sz_bytes = f.read(sz_nbytes)
        sz, = struct.unpack(size_fmt, sz_bytes)
        buf = f.read(sz - includes_size * sz_nbytes)
        return cls(buf)
```

(continues on next page)

(continued from previous page)

```
    def iter_as(self, code):
        if isinstance(code, str):
            s = struct.Struct(code)
            for off in range(0, len(self._buffer), s.size):
                yield s.unpack_from(self._buffer, off)
        elif isinstance(code, StructureMeta):
            size = code.struct_size
            for off in range(0, len(self._buffer), size):
                data = self._buffer[off:off+size]
                yield code(data)
```

çśżæŰźæşŢ `SizedRecord.from_file()` æŸŕäÿĂäÿłåůěåĚůïijŇçŤĺæİěäżŐäÿĂäÿłæŰĞäżűäÿ■èŕżå
è£Źäź§æŸŕå¿Ĺåď ŽæŰĞäżűæăijåijŘåÿÿçŤĺçŽĎæŰźåijŘãĂĆä¡IJäÿžè¿ŞåĚěïijŇåőČæŐěåŘŮäÿĂäÿłåŇĚåŘńå
åŘŕéĂĽçŽĎ `includes_size` åŘĆæŢřæŇĞåőŽäžĘå■ŮèŁĆæŢřæŸŕåŘęåŇĚåŘńåď ť éČĺåď ğåřŘãĂĆ
äÿŃéĺćæŸŕäÿĂäÿłä¿Ńå■ŘæŢŹä¡äæĂŐæăůä¡£çŤĺäżŐåď Žè¿źå¡ćæŰĞäżűäÿ■èŕżåŘŰå■ŢçŃňçŽĎåď Žè¿źå¡ćæ

```
>>> f = open('polys.bin', 'rb')
>>> phead = PolyHeader.from_file(f)
>>> phead.num_polys
3
>>> polydata = [ SizedRecord.from_file(f, '<i')
...              for n in range(phead.num_polys) ]
>>> polydata
[<__main__.SizedRecord object at 0x1006a4d50>,
<__main__.SizedRecord object at 0x1006a4f50>,
<__main__.SizedRecord object at 0x10070da90>]
>>>
```

åŘŕäżěçIJŃåĞžïijŇ`SizedRecord` åőđä¿ŃçŽĎåĘĚåőźè£ŸæšąæIJĽèćńèğčæđŘåĞžæİěãĂĆ
åŘŕäżěä¡£çŤĺ `iter_as()` æŰźæşŢæİěè¿¿åĹřçŽőçŽĎïijŇè£ŹäÿłæŰźæşŢæŐěåŘŮäÿĂäÿłçżŞæđĎæăijåijŘåĺ
`Structure` çśżä¡IJäÿžè¿ŞåĚěãĂĆ è£Źæăůå■ŘåŘŕäżěå¿ĹçĄţæť żçŽĎåŐżèğčæđŘæŢřæ■őïijŇä¿ŃåęĆïijŽ

```
>>> for n, poly in enumerate(polydata):
...     print('Polygon', n)
...     for p in poly.iter_as('<dd'):
...         print(p)
...
Polygon 0
(1.0, 2.5)
(3.5, 4.0)
(2.5, 1.5)
Polygon 1
(7.0, 1.2)
(5.1, 3.0)
(0.5, 7.5)
(0.8, 9.0)
Polygon 2
(3.4, 6.3)
```

(continues on next page)

(continued from previous page)

```
(1.2, 0.5)
(4.6, 9.2)
>>>

>>> for n, poly in enumerate(polydata):
...     print('Polygon', n)
...     for p in poly.iter_as(Point):
...         print(p.x, p.y)
...
Polygon 0
1.0 2.5
3.5 4.0
2.5 1.5
Polygon 1
7.0 1.2
5.1 3.0
0.5 7.5
0.8 9.0
Polygon 2
3.4 6.3
1.2 0.5
4.6 9.2
>>>
```

åřĘæL'ĂæIJL'è£ŹäžŻçżŞåŘĹèţůæİěïijŇäÿŃéİćæŸŕäÿĂäÿł `read_polys()` åĞ¡æŢřçŽĎåŘęåď'ŰäÿĂäÿłä£őæ■čçL'ĹïijŽ

```
class Point(Structure):
    _fields_ = [
        ('<d', 'x'),
        ('d', 'y')
    ]

class PolyHeader(Structure):
    _fields_ = [
        ('<i', 'file_code'),
        (Point, 'min'),
        (Point, 'max'),
        ('i', 'num_polys')
    ]

def read_polys(filename):
    polys = []
    with open(filename, 'rb') as f:
        phead = PolyHeader.from_file(f)
        for n in range(phead.num_polys):
            rec = SizedRecord.from_file(f, '<i')
            poly = [ (p.x, p.y) for p in rec.iter_as(Point) ]
            polys.append(poly)
```

(continues on next page)

(continued from previous page)

```
    return polys
```

èőÍèőž

è£ŹäÿĂèŁĆåŘŚä¡ăåśŢçď žäžĘèőÿåď ŽéńŸçžğçŽĎçijŰçÍŃæŁĂæIJŕïijŇåŇĚæŃňæŘŘè£řåŹÍïijŇåżűè£§è
çĎűèĂŇïijŇåőČäžňéČ¡äÿžäžĘåŘŇäÿĂäÿłçĽźåőŽçŽĎçŽőæăĞæIJ■åŁąãĂĆ

äÿŁéİćçŽĎåőđçŐřçŽĎäÿĂäÿłäÿżèęĄçĽźå¿ĄæŸŕåőČæŸŕå§žäžŐæĞŠèğčåŇĚçŽĎæĂİæČşãĂĆå¡ŞäÿĂä
`Structure` åőđä¿ŃèćńåĹŻåżžæŮűïijŇ `__init__()` äžĚäžĚåŘłæŸŕåĹŻåżžäÿĂäÿłå■ŮèŁĆæŢŕæ■őçŽĎå
çĽźåĹńçŽĎïijŇè£ŹæŮűåĂŹåźűæšąæIJĽäżżä¡ŢçŽĎèğčåŇĚæĹŰèĂĚåĚűäżŰäÿŐçżŞæđĎçŻÿåĚşçŽĎæŞ■ä¡
è£ŹæăůåĄŽçŽĎäÿĂäÿłåŁĺæIJżæŸŕä¡ăåŘŕèČ¡äżĚäżĚåŘłåŕźäÿĂäÿłå■ŮèŁĆèőŕå¡ŢçŽĎæ§ŘäÿĂåŕŘéČĺåĹĘæ

äÿžäžĘåőđçŐřæĞŠèğčåŇĚåŠŇæĽŞåŇĚïijŇéIJĂèęĄä¡£çŤÍ
`StructField` æŘŘè£řåŹÍçśżãĂĆ çŤÍæĹůåIJÍ `_fields_`
äÿ■åĹŮåĞżæİèçŽĎæŕŘäÿłåśđæĂğéČ¡äijŽèćńè¡ňåŇŰæĹŘäÿĂäÿł `StructField`
æŘŘè£řåŹÍïijŇ åőČåŕĘçŻÿåĚşçżŞæđĎæăijåijŘçăĄåŠŇåĄŘçğżåĂijä£İå■ŸåĹřå■ŸåĆÍçijŞå■Ÿäÿ■ãĂĆåĚČçś
`StructureMeta` åIJÍåď ŽäÿłçżŞæđĎçśżèćńåőŽäźĽæŮűèĞłåŁÍåĹŻåżžäžĘè£ŹäžŽæŘŘè£řåŹÍãĂĆ
æĹŚäżňä¡£çŤÍåĚČçśżçŽĎäÿĂäÿłäÿżèęĄåŐ§åŻăæŸŕåőČä¡£å¿ŮçŤÍæĹůéİđåÿÿæŰźä¿£çŽĎéĂŽè£ĞäÿĂäÿłér

`StructureMeta` çŽĎäÿĂäÿłå¿Ĺå¿őåęŹçŽĎåIJřæŰźåŕśæŸŕåőČäijŽåŻžåőŽå■ŮèŁĆæŢŕæ■őéąžäžŘãĂ
äź§åŕśæŸŕèŕť ïijŇåęĆæđIJäżżæĎŘçŽĎåśđæĂğæŇĞåőŽäžĘäÿĂäÿłå■ŮèŁĆéąžäžŘ(<èąÍçď žä¡Őä¡■äijŸåĚĹ
æĹŰèĂĚ >èąÍçď žéńŸä¡■äijŸåĚĹ)ïijŇ éĆčåŘŐéİćæĽĂæIJĽå■ŮæőţçŽĎéąžäžŘéČ¡äżěè£ŹäÿłéąžäžŘäÿžåĞE
æŕŤåęĆïijŇä¡ăåŘŕèČ¡æIJĽäÿĂäžŻæŕŤè¿Čåď ■æİĆçŽĎçżŞæđĎïijŇåŕśåČŘäÿŇéİćè£Źæăůïijž

```
class ShapeFile(Structure):
    _fields_ = [ ('>i', 'file_code'), # Big endian
        ('20s', 'unused'),
        ('i', 'file_length'),
        ('<i', 'version'), # Little endian
        ('i', 'shape_type'),
        ('d', 'min_x'),
        ('d', 'min_y'),
        ('d', 'max_x'),
        ('d', 'max_y'),
        ('d', 'min_z'),
        ('d', 'max_z'),
        ('d', 'min_m'),
        ('d', 'max_m') ]
```

äźŃåĽ■æĹŚäżňæŘŘåĹřè£ĞïijŇmemoryview() çŽĎä¡£çŤÍåŘŕäżěåÿőåŁľæĹŚäżňéĄ£åĚ■åĘĚå■ŸçŽĎ
å¡ŞçżŞæđĎå■ŸåIJÍåţŇåěŮçŽĎæŰűåĂŹïijŇmemoryviews åŘŕäżěåŘäåŁäåŘŇäÿĂåĘĚå■ŸåŇżå§§äÿŁåőŽä
è£ŹäÿłçĽźæĂğæŕŤè¿Čå¿őåęŹïijŇä¡ĘæŸŕåőČåĚşæşÍçŽĎæŸŕåĘĚå■ŸèğĘåŻ¿äÿŐæŹőéĂŽå■ŮèŁĆæŢŕçżĎçŽ
åęĆæđIJä¡ăåIJÍäÿĂäÿłå■ŮèŁĆå■ŮçňęäÿšæĹŰå■ŮèŁĆæŢŕçżĎäÿŁæĽğèąŇåĹĞçĽĞæŞ■ä¡IJïijŇä¡ăéĂŽåÿÿä
èĂŇåĘĚå■ŸèğĘåŻ¿åĹĞçĽĞäÿ■æŸŕè£ŹæăůçŽĎïijŇåőČäżĚäżĚæŸŕåIJÍåůšå■ŸåIJÍçŽĎåĘĚå■ŸäÿŁéİćåŘäå

è£ŸæIJĽå¿Ĺåď ŽçŻÿåĚşçŽĎçńäèŁĆåŘŕäżěåÿőåŁľæĹŚäżňæĽľåśŢè£ŹéĞŇèőÍèőžçŽĎæŰźæąĹãĂĆ
åŘĆèĂČ8.13åŕŘèŁĆä¡£çŤÍæŘŘè£řåŹÍæđĎåżžäÿĂäÿłçśżåđŃçşżçż§ãĂĆ
8.10åŕŘèŁĆæIJĽæŻť åď ŽåĚşäžŐåżűè£§èőąçőŮåśđæĂğåĂijçŽĎèőÍèőžïijŇåźűäÿŤèů§NestedStructæŘŘè£řå
9.19åŕŘèŁĆæIJĽäÿĂäÿłä¡£çŤÍåĚČçśżæİèåĹİåğŃåŇŰçśżæĹŘåŚŸçŽĎä¿Ňå■ŘïijŇåŠŇ

StructureMeta çśżéİđåÿÿçŻÿäijijãĂĆ PythonçŽĎ ctypes
æžŘçăĄåŘŇæăůäź§å¿ĹæIJĽèűčïijŇåőČæŘŘä¿ŻäžĘåŕźåőŽäźĽæŢřæ■őçżŞæđĎãĂĄæŢřæ■őçżŞæđĎåţŇåěŮ

8.13 6.13 æŢřæ■őçŽĎçťŕåŁääÿŐçż§èőąæŞ■ä¡IJ

éŮőéćŸ

ä¡äéIJĂèęĄåď ĎçŘĘäÿĂäÿłå¿Ĺåď ğçŽĎæŢřæ■őéŻĘåźűéIJĂèęĄèőąçőŮæŢřæ■őæĂżåŠŇæĹŰåĚűäżŰçż

èğčåĘşæŰźæąĹ

åŕźäžŐäżżä¡ŢæűĽåŘŁåĹřçż§èőąãĂĄæŮűéŮť åžŘåĹŮäżěåŘŁåĚűäżŰçŻÿåĚşæŁĂæIJŕçŽĎæŢřæ■őåĹĘ
PandasåžŞ ãĂĆ

äÿžäžĘèőľä¡äåĚĹä¡ŞéłŇäÿŃïijŇäÿŃéİćæŸŕäÿĂäÿłä¡£çŤÍPandasæİěåĹĘæđŘèŁİåŁäåŞěå§ŐåÿĆçŽĎ
èĂĄéijäåŠŇåŢőé¡£çśżåŁĺçĽľæŢřæ■őåžŞ çŽĎä¿Ňå■ŘãĂĆ åIJĺæĹŚåĘŹè£ŹçŕĞæŰĞçńăçŽĎæŮűåĂŹïijŇè£Ź

```
>>> import pandas

>>> # Read a CSV file, skipping last line
>>> rats = pandas.read_csv('rats.csv', skip_footer=1)
>>> rats
<class 'pandas.core.frame.DataFrame'>
Int64Index: 74055 entries, 0 to 74054
Data columns:
Creation Date 74055 non-null values
Status 74055 non-null values
Completion Date 72154 non-null values
Service Request Number 74055 non-null values
Type of Service Request 74055 non-null values
Number of Premises Baited 65804 non-null values
Number of Premises with Garbage 65600 non-null values
Number of Premises with Rats 65752 non-null values
Current Activity 66041 non-null values
Most Recent Action 66023 non-null values
Street Address 74055 non-null values
ZIP Code 73584 non-null values
X Coordinate 74043 non-null values
Y Coordinate 74043 non-null values
Ward 74044 non-null values
Police District 74044 non-null values
Community Area 74044 non-null values
Latitude 74043 non-null values
Longitude 74043 non-null values
Location 74043 non-null values
dtypes: float64(11), object(9)

>>> # Investigate range of values for a certain field
```

(continues on next page)

(continued from previous page)

```
>>> rats['Current Activity'].unique()
array([nan, Dispatch Crew, Request Sanitation Inspector],␣
↪dtype=object)
>>> # Filter the data
>>> crew_dispatched = rats[rats['Current Activity'] == 'Dispatch␣
↪Crew']
>>> len(crew_dispatched)
65676
>>>

>>> # Find 10 most rat-infested ZIP codes in Chicago
>>> crew_dispatched['ZIP Code'].value_counts()[:10]
60647 3837
60618 3530
60614 3284
60629 3251
60636 2801
60657 2465
60641 2238
60609 2206
60651 2152
60632 2071
>>>

>>> # Group by completion date
>>> dates = crew_dispatched.groupby('Completion Date')
<pandas.core.groupby.DataFrameGroupBy object at 0x10d0a2a10>
>>> len(dates)
472
>>>

>>> # Determine counts on each day
>>> date_counts = dates.size()
>>> date_counts[0:10]
Completion Date
01/03/2011 4
01/03/2012 125
01/04/2011 54
01/04/2012 38
01/05/2011 78
01/05/2012 100
01/06/2011 100
01/06/2012 58
01/07/2011 1
01/09/2012 12
>>>

>>> # Sort the counts
>>> date_counts.sort()
```

(continues on next page)

(continued from previous page)

```
>>> date_counts[-10:]
Completion Date
10/12/2012 313
10/21/2011 314
09/20/2011 316
10/26/2011 319
02/22/2011 325
10/26/2012 333
03/17/2011 336
10/13/2011 378
10/14/2011 391
10/07/2011 457
>>>
```

åŮŕïijŇçIJŃæăůå■Ř2011åźť10æIJĹ7æŮěåŕźèĂĄéijääžňæİěèŕť æŸŕäÿłå¿Ĺå£ŹçćŇçŽĎæŮěå■ŘåŢŁïijĄ

èőÍèőž

PandasæŸŕäÿĂäÿłæŃěæIJL'å¿Ĺåď'ŽçL'źæĂğçŽĎåď'ğåđŃåĞ¡æŢřåžŞïijŇæĹŚåIJÍè£ŹéĞŃäÿ■åŘŕèČ¡äżŃ
ä¡ĘæŸŕåŘłèęĄä¡ăéIJĂèęĄåŐżåĹĘæđŘåď'ğåđŃæŢřæ■őéŽĘåŘĹãĂĄåŕźæŢřæ■őåĹĘçżĎãĂĄèőąçőŮåŘĎçğ■

9 çňňäÿČçńäïijŽåĞ¡æŢř

ä¡£çŤÍ `def` èŕ■åŘěåőŽäźL'åĞ¡æŢřæŸŕæL'ĂæIJL'çÍŃåžŘçŽĎå§žçąĂãĂĆ
æIJňçńäçŽĎçŻőæăĞæŸŕèőšèğčäÿĂäžŻæŻť'åŁäénŸçžğåŠŃäÿ■åÿÿèğĄçŽĎåĞ¡æŢřåőŽäźL'äÿŐä¡£çŤÍæĺąåijŘ
æűL'åŘŁåĹřçŽĎåĘĚåőźåŃĚæŃňéżŸèőď'åŘĆæŢřãĂĄäżżæĎŘæŢřéĞŘåŘĆæŢřãĂĄåijžåĹűåĚşéŤőå■ŮåŘĆ
åŘęåď'ŰïijŇäÿĂäžŻénŸçžğçŽĎæŐğåĹűæţĄåŠŃåĹľçŤÍåŻđèřČåĞ¡æŢřäijäéĂŠæŢřæ■őçŽĎæŁĂæIJŕåIJÍè£Ź

Contents:

9.1 7.1 åŘŕæŐěåŘŮäżżæĎŘæŢřéĞŘåŘĆæŢřçŽĎåĞ¡æŢř

éŮőécŸ

ä¡ăæČşæđĎéĂääÿĂäÿłåŘŕæŐěåŘŮäżżæĎŘæŢřéĞŘåŘĆæŢřçŽĎåĞ¡æŢřãĂĆ

èğčåĘşæŰźæąĹ

äÿžäžĘèČ¡èőľäÿĂäÿłåĞ¡æŢřæŐěåŘŮäżżæĎŘæŢřéĞŘçŽĎä¡■ç¡őåŘĆæŢřïijŇåŘŕäżěä¡£çŤÍäÿĂäÿł*åŘĆ

```
def avg(first, *rest):
    return (first + sum(rest)) / (1 + len(rest))
```

(continues on next page)

(continued from previous page)

```
# Sample use
avg(1, 2) # 1.5
avg(1, 2, 3, 4) # 2.5
```

åIJÍè£ŹäÿłäĄŃåŃŘäÿŃïijŇrestæŸŕçŤśæĽĂæIJĽåĚűäżŰäĄçŕőåŘĆæŢřçżĎæĹŘçŽĎåĚČçżĎãĂĆçĎűåŘ

äÿžäžĘæŐěåŘŮäżżæĎŘæŢřéĞŘçŽĎåĚşéŤőåŃŮåŘĆæŢřïijŇäĄçŤÍäÿĂäÿłäžě**åijĂåď'ťçŽĎåŘĆæŢřã

```
import html

def make_element(name, value, **attrs):
    keyvals = [' %s="%s"' % item for item in attrs.items()]
    attr_str = ''.join(keyvals)
    element = '<{name}{attrs}>{value}</{name}>'.format(
                name=name,
                attrs=attr_str,
                value=html.escape(value))
    return element

# Example
# Creates '<item size="large" quantity="6">Albatross</item>'
make_element('item', 'Albatross', size='large', quantity=6)

# Creates '<p>&lt;spam&gt;</p>'
make_element('p', '<spam>')
```

åIJÍè£ŹéĞŇïijŇattrsæŸŕäÿĂäÿłåŇĚåŘńæĽĂæIJĽèćńäijžåĚěè£ŻæĮěçŽĎåĚşéŤőåŃŮåŘĆæŢřçŽĎåŃŮåŘ

åęĆæđIJäĄàè£ŸåÿŇæIJŻæ§ŘäÿłåĞĄæŢřèČĄåŘŇæŮűæŐěåŘŮäżżæĎŘæŢřéĞŘçŽĎäĄçĄőåŘĆæŢřåŠŇåŘ

```
def anyargs(*args, **kwargs):
    print(args) # A tuple
    print(kwargs) # A dict
```

äĄçŤÍè£ŹäÿłåĞĄæŢřæŮűïijŇæĽĂæIJĽäĄçĄőåŘĆæŢřäijŽèćńæŤ¿åĹřargsåĚČçżĎäÿŃïijŇæĽĂæIJĽåĚş

èőÍèőž

äÿĂäÿł*åŘĆæŢřåŘłèČĄåĞžçŐřåIJÍåĞĄæŢřåőŽäźĽäÿŃæIJĂåŘŐäÿĂäÿłäĄçĄőåŘĆæŢřåŘŐéĮćïijŇèĂŇ
**åŘĆæŢřåŘłèČĄåĞžçŐřåIJÍæIJĂåŘŐäÿĂäÿłåŘĆæŢřãĂĆ æIJĽäÿĂçĆźèęĄæşĺæĎŘçŽĎæŸŕïijŇåIJÍ*åŘĆæ

```
def a(x, *args, y):
    pass

def b(x, *args, y, **kwargs):
    pass
```

è£ŹçğŃåŘĆæŢřåřśæŸŕæĹŚäżňæĽĂèŕť'çŽĎåijžåĹűåĚşéŤőåŃŮåŘĆæŢřïijŇåIJÍåŘŐéĮć7.2åřŘèŁĆè£ŸäIj

9.2 7.2 åŘłæŐěåŘŮåĚşéŤőå■ŮåŘĆæŢřçŽĎåĞ¡æŢř

éŮőéćŸ

ä¡ăåÿŇæIJŻåĞ¡æŢřçŽĎæ§ŘäžŻåŘĆæŢřåijžåĹűä¡£çŤĺåĚşéŤőå■ŮåŘĆæŢřäijăéĂŠ

èğčåĚşæŰźæąĹ

ářĘåijžåĹűåĚşéŤőå■ŮåŘĆæŢřæŤ¿åĹřæ§Řäÿł*åŘĆæŢřæĹŰèĂĚå■ŢäÿŁ*åŘŐéİćåřśèČ¡è¿¿åĹřè£ŹçğæŢ

```
def recv(maxsize, *, block):
    'Receives a message'
    pass

recv(1024, True) # TypeError
recv(1024, block=True) # Ok
```

åĹľçŤĺè£ŹçğæŁĂæIJŕïijŇæĹŚäžňè£ŸèČ¡åIJĺæŐěåŘŮäżżæĎŘåďŽäÿłä¡■ç¡őåŘĆæŢřçŽĎåĞ¡æŢřäÿ■æ

```
def minimum(*values, clip=None):
    m = min(values)
    if clip is not None:
        m = clip if clip > m else m
    return m

minimum(1, 5, 2, -5, 10) # Returns -5
minimum(1, 5, 2, -5, 10, clip=0) # Returns 0
```

èőĺèőž

å¿Ĺåď ŽæČĚåĘţäÿŃïijŇä¡£çŤĺåijžåĹűåĚşéŤőå■ŮåŘĆæŢřäijŽæŕŤä¡£çŤĺä¡■ç¡őåŘĆæŢřèąĺæĎŘæŻťåŁä
ä¿ŃåęĆïijŇèĂČèŹŚäÿŃåęĆäÿŃäÿĂäÿłåĞ¡æŢřèřČçŤĺïijŽ

```
msg = recv(1024, False)
```

åęĆæđIJèřČçŤĺèĂĚåŕźrecvåĞ¡æŢřåźűäÿ■æŸŕå¿ĹçĘ§æĆĽïijŇéĆčäżŰèĆŕåőŽäÿ■æŸŐçŹ¡éĆčäÿłFalseå
ä¡ĘæŸŕïijŇåęĆæđIJäžčçăĄåŘŸæĹŘäÿŃéİćè£ŹæăůůåŘçŽĎèŕĺåřśæÿĚæěŽåďŽäžĘïijŽ

```
msg = recv(1024, block=False)
```

åŘęåď ŰïijŇä¡£çŤĺåijžåĹűåĚşéŤőå■ŮåŘĆæŢřäź§äijŽæŕŤä¡£çŤĺ**kwargsåŘĆæŢřæŻťåě¡ïijŇåŻääÿžåIJ

```
>>> help(recv)
Help on function recv in module __main__:
recv(maxsize, *, block)
    Receives a message
```

åijžåĹűåĚşéŤőå■ŮåŘĆæŢřåIJÍäÿĂäžŻæŻť énŸçžğåIJžåŘĹåŘŇæăůäź§å¿ĹæIJĽçŤÍãĂĆ
ä¿ŃåęĆïijŇåőČäžňåŘŕäżěèćńçŤÍæĺěåIJÍä¡£çŤÍ*argsåŠŇ**kwargsåŘĆæŢřä¡IJäÿžè¿ŞåĚěçŽĎåĞ¡æŢřäÿ■æŘŠ

9.3 7.3 çżŹåĞ¡æŢřåŘĆæŢřåćđåŁååĚČä£ąæĄŕ

éŮőéćŸ

ä¡ääĘŹåě¡äžĘäÿĂäÿłåĞ¡æŢřïijŇçĎűåŘŐæČşäÿžè£ŹäÿłåĞ¡æŢřçŽĎåŘĆæŢřåćđåŁääÿĂäžŻéćĺåď ŰçŽĎ

èğčåĘşæŰźæąĹ

ä¡£çŤÍåĞ¡æŢřåŘĆæŢřæşÍèğčæŸŕäÿĂäÿłå¿Ĺåĕ¡çŽĎåŁđæşŢïijŇåőČèČ¡æŘŘçď žçÍŃåžŘåŚŸåžŤèŕěæĂŐ
ä¿ŃåęĆïijŇäÿŃéĺćæIJĽäÿĂäÿłèćńæşÍèğčäžĘçŽĎåĞ¡æŢřïijŽ

```
def add(x:int, y:int) -> int:
    return x + y
```

pythonèğčéĞŁåŹÍäÿ■äijŽåŕźè£ŹäžŻæşÍèğčæůżåŁääżżä¡ŢçŽĎèŕ■äźĽãĂĆåőČäžňäÿ■äijŽèćńçśżåđŃæčĂ
çĎűèĂŃïijŇåŕźäžŐéĆčäžŻéŸĚèŕżæžŘçăĄçŽĎäžżæĺěèőšåřšå¿ĹæIJĽåÿőåŁľåŢęãĂĆçňňäÿĽæŰźåůěåĚůåŠŇ

```
>>> help(add)
Help on function add in module __main__:
add(x: int, y: int) -> int
>>>
```

ář¡çőąä¡ääŘŕäżěä¡£çŤÍäżżæĎŘçśżåđŃçŽĎåŕźèśąçżŹåĞ¡æŢřæůżåŁääşÍèğč(ä¿ŃåęĆæŢřå■ŮïijŇå■Ůçňęä

èőÍèőž

åĞ¡æŢřæşÍèğčåŘłå■ŸåĆÍåIJÍåĞ¡æŢřçŽĎ `__annotations__`
åśđæĂğäÿ■ãĂĆä¿ŃåęĆïijŽ

```
>>> add.__annotations__
{'y': <class 'int'>, 'return': <class 'int'>, 'x': <class 'int'>}
```

ář¡çőąæşÍèğčçŽĎä¡£çŤÍæŰźæşŢåŘŕèČ¡æIJĽå¿Ĺåď Žçğ■ïijŇä¡ĘæŸŕåőČäžňçŽĎäÿżèęĄçŤÍéĂŤè£ŸæŸŕ
åŻääÿžpythonåźűæšąæIJĽçśżåđŃåčřæŸŐïijŇéĂŽåÿÿæĺěèőšäżĚäżĚéĂŽè£ĞéŸĚèŕżæžŘçăĄå¿ĹéŽ¿ç§ěéĄŞå
è£ŹæŮűåĂŹä¡£çŤÍæşÍèğčåřśèČ¡çżŹçÍŃåžŘåŚŸæŻť åď ŽçŽĎæŘŘçď žïijŇèőľäžŰäžňåŘŕäżěæ■čçąőçŽĎä¡£çŤ

åŘĆèĂČ9.20åřŘèŁĆçŽĎäÿĂäÿłæŻť åŁäéńŸçžğçŽĎä¿Ňå■ŘïijŇæijŤçď žäžĘåęĆä¡ŢåĹľçŤÍæşÍèğčæĺěåč

9.4 7.4 è£ŤåŻđåď ŽäÿłåĂijçŽĎåĞ¡æŢř

éŮőéćŸ

ä¡ääÿŇæIJŻæđĎéĂääÿĂäÿłåŘŕäżěè£ŤåŻđåď ŽäÿłåĂijçŽĎåĞ¡æŢř

èğčåĘşæŰźæąĹ

äÿžäžĘèČ¡è£ŤåŻđåď'ŽäÿłåĂijïijŇåĞ¡æŢřçŻť'æŐěreturnäÿĂäÿłåĚČçżĎářśèąŇäžĘãĂĆä¿ŃåęĆïijŽ

```
>>> def myfun():
... return 1, 2, 3
...
>>> a, b, c = myfun()
>>> a
1
>>> b
2
>>> c
3
```

èőĺèőž

ář¡çőąmyfun()çIJŃäÿŁåŐżè£ŤåŻđäžĘåď'ŽäÿłåĂijïijŇåőđéŹĚäÿŁæŸřåĚĹåĹŻåżžäžĘäÿĂäÿłåĚČçżĎçĎ
è£Źäÿłèŕ■æşŢçIJŃäÿŁåŐżæŕŤè¿ČåěĞæĂłïijŇåőđéŹĚäÿŁæĹŚäžňä¡£çŤĺçŽĎæŸřéĂŮåŘůæİěçŤ§æĹŘäÿĂäÿ

```
>>> a = (1, 2) # With parentheses
>>> a
(1, 2)
>>> b = 1, 2 # Without parentheses
>>> b
(1, 2)
>>>
```

å¡ŞæĹŚäžňèřČçŤĺè£ŤåŻđäÿĂäÿłåĚČçżĎçŽĎåĞ¡æŢřçŽĎæŮűåĂŹ
ïijŇéĂŽåÿÿæĹŚäžňäijŽåřĘçżŞæđIJèţŃåĂijçżŹåď'ŽäÿłåŘŸéĞŘïijŇåřśåČŘäÿŁéİćçŽĎéĆčæăůãĂĆ
åĚűåőđè£ŹåřśæŸŕ1.1åřŘèŁĆäÿ■æĹŚäžňæĽĂèŕť'çŽĎåĚČçżĎèğčåŇĚãĂĆè£ŤåŻđçżŞæđIJäźŞåŘŕäžěèţŃåĂij
è£ŹæŮűåĂŹè£ŹäÿłåŘŸéĞŘåĂijåřśæŸŕåĞ¡æŢřè£ŤåŻđçŽĎéĆčäÿłåĚČçżĎæIJňèžňäžĘïijŽ

```
>>> x = myfun()
>>> x
(1, 2, 3)
>>>
```

9.5 7.5 åőŽäźĽæIJĽéżŸèőď'åŘĆæŢřçŽĎåĞ¡æŢř

éŮőécŸ

ä¡äæČşåőŽäźĽäÿĂäÿłåĞ¡æŢřæĹŰèĂĚæŰźæşŢïijŇåőČçŽĎäÿĂäÿłæĹŰåď'ŽäÿłåŘĆæŢřæŸřåŘŕéĂĽçŽ

èğčåĘşæŰźæąĹ

åőŽäźĽäÿĂäÿłæIJĽåŘŕéĂĽåŘĆæŢřçŽĎåĞ¡æŢřæŸŕéİđåÿÿçőĂå■ŢçŽĎïijŇçŻť'æŐěåIJÍåĞ¡æŢřåőŽäźĽ

```
def spam(a, b=42):
    print(a, b)

spam(1) # Ok. a=1, b=42
spam(1, 2) # Ok. a=1, b=2
```

åęĆæđIJéżŸèőď'åŘĆæŢřæŸŕäÿĂäÿłåŘŕä£őæŤźçŽĎåőźåŹĺæŕŤåęĆäÿĂäÿłåĹŮèąĺãĂĄéŻĘåŘĹæĹŰèĂĚ

```
# Using a list as a default value
def spam(a, b=None):
    if b is None:
        b = []
    ...
```

åęĆæđIJä¡ăåźűäÿ■æČşæŘŘä¿ŻäÿĂäÿłéżŸèőď'åĂijïijŇèĂŇæŸŕæČşäżĚäżĚæţŃèŕŢäÿŃæ§ŘäÿłéżŸèőď'å

```
_no_value = object()

def spam(a, b=_no_value):
    if b is _no_value:
        print('No b value supplied')
    ...
```

æĹŚäżňæţŃèŕŢäÿŃè£ŹäÿłåĞ¡æŢřijŽ

```
>>> spam(1)
No b value supplied
>>> spam(1, 2) # b = 2
>>> spam(1, None) # b = None
>>>
```

äżŤçżĘèğĆåŕ§åŘŕäżěåŘŚçŐřåĹřäijäéĂŠäÿĂäÿłNoneåĂijåŠŇäÿ■äijäåĂijäÿď'çğ■æČĚåĘţæŸŕæIJĽ'åůőåĺ

èőÍèőž

åőŽäźĽåÿęéżŸèőď'åĂijåŘĆæŢřçŽĎåĞ¡æŢřæŸŕå¿ĹçőĂå■ŢçŽĎïijŇä¡EçżÍäÿ■äżĚäżĚåŘłæŸŕè£ŹäÿłïijŇ

éęŰåĚĹïijŇéżŸèőď'åŘĆæŢřçŽĎåĂijäżĚäżĚåIJÍåĞ¡æŢřåőŽäźĽçŽĎæŮűåĂŹèţŃåĂijäÿĂæňąãĂĆèŕŢçĺĹ

```
>>> x = 42
>>> def spam(a, b=x):
...     print(a, b)
...
>>> spam(1)
1 42
>>> x = 23 # Has no effect
```

(continues on next page)

(continued from previous page)

```
>>> spam(1)
1 42
>>>
```

æşĺæĎŕåĹřå¡ŞæĹŚäžňæŤźåŔŸxçŽĎåĂijçŽĎæŮűåĂŹåŕźéżŸèőď'åŔĆæŢřåĂijåźűæšąæIJL'å¡śåŞ■ïijŇè

åĚűæňąïijŇéżŸèőď'åŔĆæŢřçŽĎåĂijåžŤèŕěæŸŕäÿ■åŔŕåŔŸçŽĎåŕźèśąïijŇæŕŤåęĆNoneãĂĄTrueãĂĄFal
çĽźåĹńçŽĎïijŇå■ČäÿĞäÿ■èęĄåČŔäÿŃéİćè£ŹæăůåĘŹäžčçăĄïijŽ

```
def spam(a, b=[]): # NO!
    ...
```

åęĆæđIJä¡ăè£ŹäźĹåĄŽäžĘïijŇå¡ŞéżŸèőď'åĂijåIJĺåĚűäžŮåIJřæŰźèćńä£őæŤźåŘŐä¡ăářĘäijŽéĄĞåĹřåŘ

```
>>> def spam(a, b=[]):
...     print(b)
...     return b
...
>>> x = spam(1)
>>> x
[]
>>> x.append(99)
>>> x.append('Yow!')
>>> x
[99, 'Yow!']
>>> spam(1) # Modified list gets returned!
[99, 'Yow!']
>>>
```

è£Źçğ■çżŞæđIJåžŤèŕěäÿ■æŸŕä¡ăæČşèęĄçŽĎãĂĆäÿžäžĘéĄ£åĚ■è£Źçğ■æČĚåĘţçŽĎåŘŚçŤ§ïijŇæIJĂå
çĎűåŘŐåIJĺåĞ¡æŢřéĞŃéİćæčĂæ§ěåőČïijŇåL'■éİćçŽĎä¿Ňå■ŘåřśæŸŕè£ŹæăůåĄŽçŽĎãĂĆ

åIJĺæţŃèŕŢNoneåĂijæŮűä¡£çŤĺ is æŞ■ä¡IJçňęæŸŕå¿ĹéĞ■èęĄçŽĎïijŇäź§æŸŕè£Źçğ■æŰźæąĹçŽĎåĚş
æIJL'æŮűåĂŹåď'ğåőűäijŽçŁŕäÿŃäÿŃéİćè£ŹæăůçŽĎéŤŹèŕŕïijŽ

```
def spam(a, b=None):
    if not b: # NO! Use 'b is None' instead
        b = []
    ...
```

è£ŹäźĹåĘŹçŽĎéŮőéćŸåIJĺäžŐåř¡çőąNoneåĂijçąőåőđæŸŕèćńå¡ŞæĹŘFalseïijŇ
ä¡ĘæŸŕè£ŸæIJL'åĚűäžŰçŽĎåŕźèśą(æŕŤåęĆéŢ£åžęäÿž0çŽĎå■ŮçňęäÿšăĂĄåĹŮèąĺãĂĄåĚČçżĎãĂĄå■ŮåĚÿ
åŻăæ■d'ïijŇäÿŁéİćçŽĎäžčçăĄäijŽèŕŕåřĘäÿĂäžŻåĚűäžŰè¿ŞåĚěäź§å¡ŞæĹŘæŸŕæšąæIJL'è¿ŞåĚěãĂĆæŕŤåęĆ

```
>>> spam(1) # OK
>>> x = []
>>> spam(1, x) # Silent error. x value overwritten by default
>>> spam(1, 0) # Silent error. 0 ignored
>>> spam(1, '') # Silent error. '' ignored
>>>
```

æIJĂåŘŐäÿĂäÿłéŮőéćŸæŕŤè¿Čå¿őåęŹïijŇéĆčåřśæŸŕäÿĂäÿłåĞ¡æŢřéIJĂèęĄæţŃèřŢæ§ŘäÿłåŔŕéĂL'åŘ
è£ŹæŮűåĂŹéIJĂèęĄåřŘå£ČçŽĎæŸŕä¡ääÿ■èČ¡çŤĺæ§ŘäÿłéżŸèőď'åĂijæŕŤåęĆNoneãĂĄ
0æĹŰèĂĚFalseåĂijæĭěæţŃèřŢçŤĺæĹůæŘŘä¿ŻçŽĎåĂij(åŻääÿžè£ŹäžŻåĂijéČ¡æŸŕåŘĹæşŢçŽĎåĂijïijŇæŸ
åŻäæ■d'ïijŇä¡äéIJĂèęĄåĚűäżŰçŽĎèğčåEşæŰźæąĹäžEãĂĆ

äÿžäžEèğčåEşè£ŹäÿłéŮőéćŸïijŇä¡ääŔŕäżěåĹŻåżžäÿĂäÿłçŃňäÿĂæŮääžŇçŽĎçğĄæIJL'åŕźèśąåőđä¿Ńïij
åIJĺåĞ¡æŢřéĞŇéĺćïijŇä¡ääŔŕäżěéĂŽè£ĞæčĂæ§ěèćńäijäéĂŠåŔĆæŢřåĂijèů§è£Źäÿłåőđä¿ŃæŸŕåŘęäÿĂæäůä
è£ŹéĞŇçŽĎæĂÌèůŕæŸŕçŤĺæĹůäÿ■åŔŕèČ¡åŐżäijäéĂŠè£ŹäÿŁ_no_valueåőđä¿Ńä¡IJäÿžè¿ŞåĚěãĂĆ
åŻäæ■d'ïijŇè£ŹéĞŇéĂŽè£ĞæčĂæ§ěè£ŹäÿłåĂijåřśèČ¡çąőåőŽæ§ŘäÿłåŔĆæŢřæŸŕåŘęèćńäijäéĂŠè£ŽæÌěäžŘ

è£ŹéĞŇåŕź `object()` çŽĎä¡£çŤĺçIJŃäÿŁåŐżæIJL'çĆźäÿ■åď'łåÿÿèğĄãĂĆ`object`
æŸŕpythonäÿ■æL'ĂæIJL'çśżçŽĎå§žçśżãĂĆ ä¡ääŔŕäżěåĹŻåżž `object`
çśżçŽĎåőđä¿ŃïijŇä¡EæŸŕè£ŹäżŽåőđä¿ŃæšąäżĂäźĹåőđéŹĚçŤĺåď'ĎïijŇåŻääÿžåőČåźűæšąæIJL'äżżä¡ŢæIJL'
äź§æšąæIJL'äżżä¡Ţåőđä¿ŃæŢřæ■ő(åŻääÿžåőČæšąæIJL'äżżä¡ŢçŽĎåőđä¿Ńå■ŮåĚÿïijŇä¡äçŤŽèĞşéČ¡äÿ■èČ¡
ä¡ääŤŕäÿĂèČ¡åĄŽçŽĎåřśæŸŕæţŃèřŢåŘŇäÿĂæĂğãĂĆè£ŹäÿłåĹŽåě¡çňęåŘĹæĹŚçŽĎèęĄæśĆïijŇåŻääÿžæĹ

9.6 7.6 åőŽäźL'åŇ£åŘ■æĹŰåĘĚèĄŤåĞ¡æŢř

éŮőéćŸ

ä¡äæČşäÿž `sort()` æŞ■ä¡IJåĹŻåżžäÿĂäÿłå¿Ĺç§■çŽĎåŻđèřČåĞ¡æŢřïijŇä¡EåŔĹäÿ■æČşçŤĺ
`def` åŐżåĘŹäÿĂäÿłå■ŢèąŇåĞ¡æŢřïijŇ èĂŇæŸŕåÿŇæIJŻéĂŽè£Ğæ§Řäÿłå£ńæ■ůæŰźåijŔäżěåĘĚèĄŤæŰźåi

èğčåEşæŰźæąĹ

å¡ŞäÿĂäžŻåĞ¡æŢřå¿ĹçőĂå■ŢïijŇäżĚäżĚåŔłæŸŕèőąçőŮäÿĂäÿłèąĺè¿¿åijŔçŽĎåĂijçŽĎæŮűåĂŹïijŇåřś

```
>>> add = lambda x, y: x + y
>>> add(2,3)
5
>>> add('hello', 'world')
'helloworld'
>>>
```

è£ŹéĞŇä¡£çŤĺçŽĎlambdaèąĺè¿¿åijŔèů§äÿŇéĺćçŽĎæŢĹæđIJæŸŕäÿĂæäůçŽĎïijŽ

```
>>> def add(x, y):
...     return x + y
...
>>> add(2,3)
5
>>>
```

lambdaèąĺè¿¿åijŔåĚÿåđŇçŽĎä¡£çŤĺåIJżæŹŕæŸŕæŐŠåżŔæĹŰæŢřæ■őreduceç■L'ïijŽ

```
>>> names = ['David Beazley', 'Brian Jones',
...          'Raymond Hettinger', 'Ned Batchelder']
>>> sorted(names, key=lambda name: name.split()[-1].lower())
```

(continues on next page)

(continued from previous page)

```
['Ned Batchelder', 'David Beazley', 'Raymond Hettinger', 'Brian␣
↪Jones']
>>>
```

èőÍèőž

åř¡çőąlambdaèąÍè¿¿åijŘåĚĄèőÿä¡ääőŽäźL'çőĂå■ŢåĞ¡æŢřïijŇä¡ĘæŸŕäőČçŽĎä¡£çŤÍæŸŕæIJL'éŹŘåĹűç
ä¡ääŘłèČ¡æŇĞäőŽå■Ţäÿłèąlè¿¿åijŘïijŇäőČçŽĎåĂijåřśæŸŕæIJĂåŘŐçŽĎè£ŤåŻđåĂijãĂĆäź§åřśæŸŕèŕť'äÿ■è
åŇĚæŃňåď'Žäÿłèŕ■åŘĕãĂĄæİąäżűèąÍè¿¿åijŘãĂĄè£■äžčäžĕåŘŁåijĆåÿÿåď'ĎçŘĘç■L'ç■L'ãĂĆ

ä¡ääŘŕäžĕäÿ■ä¡£çŤÍlambdaèąÍè¿¿åijŘåřśèČ¡çijŰåĘŹåď'ğéČÍåĹĘpythonäžčçăĄãĂĆ
ä¡ĘæŸŕïijŇå¡ŞæIJL'äžžçijŰåĘŹåď'ğéĞŘèőąçőŮèąÍè¿¿åijŘåĂijçŽĎç§■åřŘåĞ¡æŢřæĹŰèĂĚéIJĂèęĄçŤÍæĹůa
ä¡ääřśäijŽçIJŃåĹřlambdaèąÍè¿¿åijŘçŽĎèžńå¡śäžĘãĂĆ

9.7 7.7 åŇ£åŘ■åĞ¡æŢřæ■ŢèŐůåŘŸéĞŘåĂij

éŮőécŸ

ä¡äçŤÍlambdaäőŽäźL'äžĘäÿĂäÿłåŇ£åŘ■åĞ¡æŢřïijŇäźűæČşåIJÍäőŽäźL'æŮűæ■ŢèŐůåĹřæ§ŘäžŻåŘŸéĞ

èğčåĘşæŰźæąĹ

åĚĹçIJŃäÿŃäÿŃéİćäžčçăĄçŽĎæŢĹæđIJïijŽ

```
>>> x = 10
>>> a = lambda y: x + y
>>> x = 20
>>> b = lambda y: x + y
>>>
```

çŐřåIJÍæĹŚéŮőä¡äïijŇa(10)åŠŇb(10)è£ŤåŻđçŽĎçżŞæđIJæŸŕäżĂäźĹïij§åęĆæđIJä¡äèőď'äÿžçżŞæđIJæŸ

```
>>> a(10)
30
>>> b(10)
30
>>>
```

è£ŹåĚűäÿ■çŽĎåěěåęŹåIJÍäžŐlambdaèąÍè¿¿åijŘäÿ■çŽĎxæŸŕäÿĂäÿłèĞłçŤśåŘŸéĞŘïijŇ
åIJÍè£ŘèąŇæŮűçżŚåőŽåĂijïijŇèĂŇäÿ■æŸŕäőŽäźL'æŮűåřśçżŚåőŽïijŇè£Źèů§åĞ¡æŢřçŽĎéżŸèőď'åĂijåŘĆa
åŽäæ■d'ïijŇåIJÍèřČçŤÍè£ŹäÿłlambdaèąÍè¿¿åijŘçŽĎæŮűåĂŹïijŇxçŽĎåĂijæŸŕæL'ğèąŇæŮűçŽĎåĂijãĂĆä¿Ň

```
>>> x = 15
>>> a(10)
25
```

(continues on next page)

(continued from previous page)

```
>>> x = 3
>>> a(10)
13
>>>
```

åęĆæđIJä¡ăæČşèőľæ§ŘäÿłåŇ£åŘ■åĞ¡æŢřåIJÍåőŽäźĽæŮűåřśæ■ŢèŐůåĹřåĂijïijŇåŕŕäżěåřĘéĆčäÿłåŘĆ

```
>>> x = 10
>>> a = lambda y, x=x: x + y
>>> x = 20
>>> b = lambda y, x=x: x + y
>>> a(10)
20
>>> b(10)
30
>>>
```

èőÍèőž

åIJÍè£ŹéĞŇåĹŮåĞžæİěçŽĎéŮőéćŸæŸŕæŰŕæĽŃå¿ĹåőźæŸŞçŁŕçŽĎéŤŹèŕŕïijŇæIJĽäžŻæŰŕæĽŃåŘŕè
æŕŤåęĆïijŇéĂŽè£ĞåIJÍäÿĂäÿłå¿łçŐŕæĹŰåĹŮèąÍæŐÍåŕijäÿ■åĹŻåżžäÿĂäÿłlambdaèąÍè¿¿åijŘåĹŮèąÍïijŇåźűa

```
>>> funcs = [lambda x: x+n for n in range(5)]
>>> for f in funcs:
... print(f(0))
...
4
4
4
4
4
>>>
```

ä¡ĘæŸŕåőđéŹĚæŢĹæđIJæŸŕè£ŘèąŇæŸŕnçŽĎåĂijäÿžè£■äżčçŽĎæIJĂåŘŐäÿĂäÿłåĂijãĂĆçŐŕåIJÍæĹŚä

```
>>> funcs = [lambda x, n=n: x+n for n in range(5)]
>>> for f in funcs:
... print(f(0))
...
0
1
2
3
4
>>>
```

éĂŽè£Ğä¡£çŤÍåĞ¡æŢřéżŸèőď åĂijåŘĆæŢřå¡ćåijŘïijŇlambdaåĞ¡æŢřåIJÍåőŽäźĽæŮűåřśèČ¡çżŚåőŽåĹřå

9.8 7.8 åĞŔåřŚåŔŕèřČçŤĺåŕźèśąçŽĎåŔĆæŢřäÿłæŢř

éŮőéćŸ

ä¡ăæIJL'äÿĂäÿłèćńåĚűäżŰpythonäżčçăĄä¡£çŤĺçŽĎcallableåŕźèśąïijŇåŔŕèČ¡æŸŕäÿĂäÿłåŻđèřČåĞ¡æŢřä
ä¡ĘæŸŕåőČçŽĎåŔĆæŢřåď łåď ŽäžĘïijŇåŕijèĞť èřČçŤĺæŮűåĞžéŤŹãĂĆ

èğčåĘşæŰźæąĹ

åęĆæđIJéIJĂèęĄåĞŔåřŚæ§ŘäÿłåĞ¡æŢřçŽĎåŔĆæŢřäÿłæŢřïijŇä¡ăåŔŕäżěä¡£çŤĺ
`functools.partial()` ãĂĆ `partial()` åĞ¡æŢřåĚĄèőÿä¡ăçżŹäÿĂäÿłæĹŰåď ŽäÿłåŔĆæŢřèő¿ç¡őåŻžä
äÿžäžĘæijŤçď žæÿĚæěŽïijŇåĄĞèő¿ä¡ăæIJL'äÿŃéİćè£ŹæăůçŽĎåĞ¡æŢřïijŽ

```
def spam(a, b, c, d):
    print(a, b, c, d)
```

çŐřåIJĺæĹŚäżňä¡£çŤĺ `partial()` åĞ¡æŢřæİěåŻžăőŽæ§ŘäžŽåŔĆæŢřåĂijïijŽ

```
>>> from functools import partial
>>> s1 = partial(spam, 1) # a = 1
>>> s1(2, 3, 4)
1 2 3 4
>>> s1(4, 5, 6)
1 4 5 6
>>> s2 = partial(spam, d=42) # d = 42
>>> s2(1, 2, 3)
1 2 3 42
>>> s2(4, 5, 5)
4 5 5 42
>>> s3 = partial(spam, 1, 2, d=42) # a = 1, b = 2, d = 42
>>> s3(3)
1 2 3 42
>>> s3(4)
1 2 4 42
>>> s3(5)
1 2 5 42
>>>
```

åŔŕäżěçIJŃåĞž `partial()` åŻžăőŽæ§ŘäžŽåŔĆæŢřåźűè£ŤåŻđäÿĂäÿłæŰřçŽĎcallableåŕźèśąãĂĆè£Źã
çĎűåŘŐèů§äźŃåĹ'■åůšçżŔèţŃåĂijè£ĞçŽĎåŔĆæŢřåŘĹåźűèţůæİěïijŇæIJĂåŘŐåřĘæĹ'ĂæIJL'åŔĆæŢřäijăéĂ

èőĺèőž

æIJňèŁĆèęĄèğčåĘşçŽĎéŮőéćŸæŸŕèőĺ'åŐ§æIJňäÿ■åĚijåőźçŽĎäżčçăĄåŔŕäżěäÿĂèţůåůěä¡IJãĂĆäÿŃéİ

çňňäÿĂäÿłä¿Ňå■ŘæŸŕïijŇåĄĞèő¿ä¡ăæIJL'äÿĂäÿłçĆźçŽĎåĹŮèąĺæİěèąĺçď ž(x,y)åİŘæăĞåĚĆçżĎãĂĆ
ä¡ăåŔŕäżěä¡£çŤĺäÿŃéİćçŽĎåĞ¡æŢřæİěèőąçőŮäÿď çĆźäźŃéŮť çŽĎèůİçężïijŽ

```
points = [ (1, 2), (3, 4), (5, 6), (7, 8) ]

import math
def distance(p1, p2):
    x1, y1 = p1
    x2, y2 = p2
    return math.hypot(x2 - x1, y2 - y1)
```

çŐřåIJÍåĄĞèő¿ä¡ăæČşäżěæ§ŘäÿłçĆźäÿžå§žçĆźïijŇæăźæ■őçĆźåŠŇå§žçĆźäźŃéŮť çŽĎèůİçężæİěæŐŠå
åĹŮèąÍçŽĎ `sort()` æŰźæşŢæŐěåŘŮäÿĂäÿłåĚşéŤőå■ŮåŘĆæŢřæİěèĞłåőŽäźĽæŐŠåžŘéĂżè¿ŚïijŇ
ä¡ĘæŸŕåőČåŘłèČ¡æŐěåŘŮäÿĂäÿłå■ŢäÿłåŘĆæŢřçŽĎåĞ¡æŢř(distance()å¿ĹæŸŐæŸ¿æŸŕäÿ■çňęåŘĹæİąäżŭ
çŐřåIJÍæĹŚäżňåŘŕäżěéĂŽè£Ğä¡£çŤÍ `partial()` æİěèğčåĘşè£ŹäÿłéŮőéćŸïijŽ

```
>>> pt = (4, 3)
>>> points.sort(key=partial(distance,pt))
>>> points
[(3, 4), (1, 2), (5, 6), (7, 8)]
>>>
```

æŻť'è£ŻäÿĂæ■ěïijŇ`partial()` éĂŽåÿÿèćńçŤÍæİěå¿őèřČåĚűäżŮåžŞåĞ¡æŢřæĽĂä¡£çŤÍçŽĎåŻđèřČå
ä¿ŃåęĆïijŇäÿŃéİćæŸŕäÿĂæőţäżčçăĄïijŇä¡£çŤÍ `multiprocessing`
æİěåijĆæ■ěèőąçőŮäÿĂäÿłçżŞæđIJåĂijïijŇ çĎűåŘŐè£ŹäÿłåĂijèćńäijäéĂŞçżŹäÿĂäÿłæŐěåŘŮäÿĂäÿłresultåĂ

```
def output_result(result, log=None):
    if log is not None:
        log.debug('Got: %r', result)

# A sample function
def add(x, y):
    return x + y

if __name__ == '__main__':
    import logging
    from multiprocessing import Pool
    from functools import partial

    logging.basicConfig(level=logging.DEBUG)
    log = logging.getLogger('test')

    p = Pool()
    p.apply_async(add, (3, 4), callback=partial(output_result,␣
↪log=log))
    p.close()
    p.join()
```

å¡ŞçżŹ `apply_async()` æŘŘä¿ŻåŻđèřČåĞ¡æŢřæŮűïijŇéĂŽè£Ğä¡£çŤÍ
`partial()` äijäéĂŠéćİåď'ŰçŽĎ `logging` åŘĆæŢřãĂĆ èĂŇ `multiprocessing`
åŕźè£ŹäźŻäÿĂæŮäæĽĂç§ěâĂŤâĂŤåőČäżĚäźĚåŘłæŸŕä¡£çŤÍå■ŢäÿłåĂijæİěèřČçŤÍåŻđèřČåĞ¡æŢřãĂĆ

ä¡IJäÿžäÿĂäÿłçśżäijijçŽĎä¿Ńå■ŘïijŇèĂČèŹŚäÿŃçijŰåĘŹç¡ŚçżIJæIJ■åŁąåŹÍçŽĎéŮőéćŸïijŇ`sockets`
æÍąåİŮèőľåőČåŘŸå¿Ůå¿Ĺåőźæ ŸŞãĂĆ äÿŃéİćæŸŕäÿłçőĂå■ŢçŽĎechoæIJ■åŁąåŹÍïijŽ

```
from socketserver import StreamRequestHandler, TCPServer

class EchoHandler(StreamRequestHandler):
    def handle(self):
        for line in self.rfile:
            self.wfile.write(b'GOT:' + line)

serv = TCPServer(('', 15000), EchoHandler)
serv.serve_forever()
```

äÿ■è£ĞïijŇåĄĞèő¿ä¡äæČşçżŹEchoHandleråćđåŁääÿĂäÿłåŔŕäżěæŐěåŔŮåĚűäżŰéĚ■ç¡őéĂĽéąźçŽĎ `__init__` æŰźæşŢãĂĆæŕŤåęĆïijŽ

```
class EchoHandler(StreamRequestHandler):
    # ack is added keyword-only argument. *args, **kwargs are
    # any normal parameters supplied (which are passed on)
    def __init__(self, *args, ack, **kwargs):
        self.ack = ack
        super().__init__(*args, **kwargs)

    def handle(self):
        for line in self.rfile:
            self.wfile.write(self.ack + line)
```

è£ŹäźĹä£őæŤźåŘŐïijŇæĹŚäżňåŕśäÿ■éIJĂèęĄæŸ¿åijŔåIJřåIJÍTCPServerçśżäÿ■æůżåŁäåĽ■çijĂäżĘãĂĆ
ä¡ĘæŸŕä¡ääĘ■æňąè£ŘèąŇçÍŃäžŔåŘŐäijŽæŁěçśżäijijäÿŃéÍćçŽĎéŤŹèŕŕïijŽ

```
Exception happened during processing of request from ('127.0.0.1',␣
↪59834)
Traceback (most recent call last):
...
TypeError: __init__() missing 1 required keyword-only argument: 'ack
↪'
```

åĹÍçIJŃèţůæÍěåě¡åČŔå¿ĹéŽ¿ä£őæ■čè£ŹäÿłéŤŹèŕŕïijŇéŹď äžĘä£őæŤź
`socketserver` æĺąåÍŮæžŘäżčçăĄæĹŰèĂĚä¡£çŤÍæ§ŘäžŽåěĞæĂłçŽĎæŰźæşŢäźŃåď ŰãĂĆ
ä¡ĘæŸŕïijŇåęĆæđIJä¡£çŤÍ `partial()` åŕśèČ¡å¿Ĺè¡żæÍ¿çŽĎèğčåĘşâĂŤâĂŤçżŹåőČäijäéĂŠ
`ack` åŔĆæŢřçŽĎåĂijæÍěåĹÍåğŃåŇŰå■şåŔŕïijŇåęĆäÿŃïijŽ

```
from functools import partial
serv = TCPServer(('', 15000), partial(EchoHandler, ack=b'RECEIVED:
↪'))
serv.serve_forever()
```

åIJÍè£Źäÿłä¿Ňå■Řäÿ■ïijŇ`__init__()` æŰźæşŢäÿ■çŽĎack-
åŔĆæŢřåčřæŸŐæŰźåijŔçIJŃäÿŁåŐżå¿ĹæIJĽèűčïijŇåĚűåőđåŕśæŸŕåčřæŸŐackäÿžäÿĂäÿłåijžåĹűåĚşéŤőå■
åĚşäžŐåijžåĹűåĚşéŤőå■ŮåŔĆæŢřéŮőéćŸæĹŚäżňåIJÍ7.2åŕŔèŁĆæĹŚäżňåůšçżŔèőÍèőžè£ĞäžĘïijŇèŕżèĂĚåŔ

å¿Ĺåď ŽæŮűåĂŹ `partial()` èČ¡åőđçŐřçŽĎæŢĹæđIJïijŇlambdaèąÍè¿¿åijŔäź§èČ¡åőđçŐřãĂĆæŕŤåęĆ

```
points.sort(key=lambda p: distance(pt, p))
p.apply_async(add, (3, 4), callback=lambda result: output_
↪result(result,log))
serv = TCPServer(('', 15000),
        lambda *args, **kwargs: EchoHandler(*args, ack=b'RECEIVED:',
↪ **kwargs))
```

è£ŹæăůåĘŹäź§èČ¡åőđçŐřåŘŇæăůçŽĎæŢĹæđIJïijŇäÿ■è£ĞçŻÿæŕŤèĂŇåůšäijŽæŸ¿å¿ŮæŕŤè¿ČèĞČèĆ
è£ŹæŮűůåĂŹä¡£çŤÍ `partial()` åŘŕäżěæŻť åŁăçŻť èğĆçŽĎèąĺè¿¿ä¡ăçŽĎæĎŘåŻ¿(çżŹæ§ŘäžŽåŘĆæŢřéćĎ

9.9 7.9 åřĘå■ŢæŰźæşŢçŽĎçśżè¡ňæ■ćäÿžåĞ¡æŢř

éŮőéćŸ

ä¡ăæIJL'äÿĂäÿłéŹď `__init__()` æŰźæşŢåď ŰåŘłåőŽäźL'äžĘäÿĂäÿłæŰźæşŢçŽĎçśżãĂĆäÿžäžĘçőĂ

èğčåĘşæŰźæąĹ

åď ğåď ŽæŢřæČĚåĘţäÿŃïijŇåŘŕäżěä¡£çŤÍéŮ■åŇĚæİěåřĘå■ŢäÿłæŰźæşŢçŽĎçśżè¡ňæ■ćæĹŘåĞ¡æŢřãĂ
äÿ¿äÿłä¿Ňå■ŘïijŇäÿŃéİćçď žä¿Ňäÿ■çŽĎçśżåĚĄèőÿä¡£çŤÍèĂĚæăźæ■őæ§ŘäÿłæÍąæİ£æŰźæąĹæİěèŐůåŘŰå

```
from urllib.request import urlopen

class UrlTemplate:
    def __init__(self, template):
        self.template = template

    def open(self, **kwargs):
        return urlopen(self.template.format_map(kwargs))

# Example use. Download stock data from yahoo
yahoo = UrlTemplate('http://finance.yahoo.com/d/quotes.csv?s={names}
↪&f={fields}')
for line in yahoo.open(names='IBM,AAPL,FB', fields='sl1c1v'):
    print(line.decode('utf-8'))
```

è£ŹäÿłçśżåŕäżěèćňäÿĂäÿłæŻť çőĂå■ŢçŽĎåĞ¡æŢřæİěäžčæŻ£ïijŽ

```
def urltemplate(template):
    def opener(**kwargs):
        return urlopen(template.format_map(kwargs))
    return opener

# Example use
yahoo = urltemplate('http://finance.yahoo.com/d/quotes.csv?s={names}
↪&f={fields}')
```

(continues on next page)

(continued from previous page)

```
for line in yahoo(names='IBM,AAPL,FB', fields='sl1c1v'):
    print(line.decode('utf-8'))
```

èőÍèőž

åď ğéČÍåĹĘæČĚåĘţäÿŃïijÑä¡ăæŃěæIJĽäÿĂäÿłå■ŢæŰźæşŢçśżçŽĎåŐ§åŻăæŸŕéIJĂèęĄå■ŸåĆÍæ§ŘäžŽ
æŕŤåęĆïijÑåőŽäźĽUrlTemplateçśżçŽĎåŤŕäÿĂçŻőçŽĎåřśæŸŕåĚĹåIJÍæ§ŘäÿłåIJřæŰźå■ŸåĆÍæÍąæİ£åĂijïijÑ

ä¡£çŤÍäÿĂäÿłåĘĚéČÍåĞ¡æŢřæĹŰèĂĚéŮ■åŃĚçŽĎæŰźæąĹéĂŽåÿÿäijŽæŻť äijŸéŻĚäÿĂäžŻãĂĆçőĂå■
åŘłäÿ■è£ĞåIJÍåĞ¡æŢřåĘĚéČÍåÿęäÿŁäžĘäÿĂäÿłéćİåď ŰçŽĎåŘŸéĞŘçŐŕåćČãĂĆéŮ■åŃĚåĚşéŤőçĽźçĆźåřśa
åŻăæ■ď ïijÑåIJÍæĹŚäžňçŽĎèğčåĘşæŰźæąĹäÿ■ïijÑopener() åĞ¡æŢřèőřä¡ŘäžĘ
template åŘĆæŢřçŽĎåĂijïijÑåźűåIJÍæŐěäÿŃæİěçŽĎèřČçŤÍäÿ■ä¡£çŤÍåőČãĂĆ

äżżä¡ŢæŮűåĂŹåŘłèęĄä¡äçćřåĹřéIJĂèęĄçżŹæ§ŘäÿłåĞ¡æŢřåćđåŁäéćİåď ŰçŽĎçŁűæĂĄä£ąæĄŕçŽĎéŮ
çŻÿæŕŤåřĘä¡äçŽĎåĞ¡æŢřè¡ňæ■ćæĹŘäÿĂäÿłçśżèĂŃèÍĂïijÑéŮ■åŃĚéĂŽåÿÿæŸŕäÿĂçğ■æŻť åŁäçőĂæť ĄåŠ

9.10 7.10 åÿęéćİåď ŰçŁűæĂĄä£ąæĄŕçŽĎåŻđèřČåĞ¡æŢř

éŮőéćŸ

ä¡äçŽĎäżčçăĄäÿ■éIJĂèęĄä¿ÌèţŰåĹřåŻđèřČåĞ¡æŢřçŽĎä¡£çŤÍ(æŕŤåęĆäžŃäżűåď ĎçŘĘåŹÍãĂĄç■L'å¿Ě
åźűäÿŤä¡äè£ŸéIJĂèęĄèől'åŻđèřČåĞ¡æŢřæŃěæIJĽéćİåď ŰçŽĎçŁűæĂĄåĂijïijÑäżěä¿£åIJÍåőČçŽĎåĘĚéČÍä

èğčåĘşæŰźæąĹ

è£ŹäÿĂåřŘèŁĆäÿżèęĄèőÍèőžçŽĎæŸŕéĆčäžŻåĞżçŐřåIJÍå¿Ĺåď ŽåĞ¡æŢřåžŞåŠŃæąĘæđűäÿ■çŽĎåŻđèřČ
äÿžäžĘæijŤçď žäÿŐæţŃèŕŢïijÑæĹŚäžňåĚĹåőŽäźĽ'åęĆäÿŃäÿĂäÿłéIJĂèęĄèřČçŤÍåŻđèřČåĞ¡æŢřçŽĎåĞ¡æŢ

```
def apply_async(func, args, *, callback):
    # Compute the result
    result = func(*args)

    # Invoke the callback with the result
    callback(result)
```

åőđéŹĚäÿŁïijÑè£Źæőţäżčçă ĄåŘŕäżěåĄŽäżżä¡ŢæŻť éńŸçžğçŽĎåď ĎçŘĘïijÑåŃĚæŃňçž£çÍŃãĂĄè£Żç
æĹŚäžňäżĚäżĚåŘłéIJĂèęĄåĚşæşĺåŻđèřČåĞ¡æŢřçŽĎèřČçŤÍãĂĆäÿŃéÍćæŸŕäÿĂäÿłæijŤçď žæĂŐæăůä¡£çŤÍã

```
>>> def print_result(result):
...     print('Got:', result)
...
>>> def add(x, y):
...     return x + y
...
>>> apply_async(add, (2, 3), callback=print_result)
```

(continues on next page)

(continued from previous page)

```
Got: 5
>>> apply_async(add, ('hello', 'world'), callback=print_result)
Got: helloworld
>>>
```

æşĺæĎŕåĹř `print_result()` åĞ¡æŢřäżĚäżĚåŕłæŐěåŕŮäÿĂäÿłåŕĆæŢř `result` ãĂĆäÿ■èČ¡åĘ■äijäåĚěåĚűäżŰä£ąæĄŕãĂĆ èĂŇå¡Şä¡äæČşèőľ åŻđèřČåĞ¡æŢřèő£éŮőåĚűäżŰåŕŸéĞŕæĹŰè

äÿžäžĘèőľ åŻđèřČåĞ¡æŢřèő£éŮőåd'ŰéČĺä£ąæĄŕïijŇäÿĂçğ■æŰźæşŢæŸŕä¡£çŤĺäÿĂäÿłçżŚåőŽæŰźæşŢ æŕŤåęĆïijŇäÿŃéİćè£ŹäÿłçśżäijŽä£İå■ŸäÿĂäÿłåĘĚéČĺåžŕåĹŮåŕůïijŇæŕŕäňąæŐěæŤűåĹřäÿĂäÿł `result` çŽĎæŮűåĂŹåžŕåĹŮåŕůåŁä1ïijŽ

```
class ResultHandler:

    def __init__(self):
        self.sequence = 0

    def handler(self, result):
        self.sequence += 1
        print('[{}] Got: {}'.format(self.sequence, result))
```

ä¡£çŤĺè£ŹäÿłçśżçŽĎæŮűåĂŹïijŇä¡åĚĹåĹŻåżžäÿĂäÿłçśżçŽĎåőđä¿ŃïijŇçĎűåŘŐçŤĺåőČçŽĎ `handler()` çżŚåőŽæŰźæşŢæİěåĄŽäÿžåŻđèřČåĞ¡æŢřïijŽ

```
>>> r = ResultHandler()
>>> apply_async(add, (2, 3), callback=r.handler)
[1] Got: 5
>>> apply_async(add, ('hello', 'world'), callback=r.handler)
[2] Got: helloworld
>>>
```

çňňäžŇçğ■æŰźåijŕïijŇä¡IJäÿžçśżçŽĎæŻ£äżčïijŇåŕŕäżěä¡£çŤĺäÿĂäÿłéŮ■åŇĚæ■ŢèŐůçŁűæĂĄåĂijïijŇ

```
def make_handler():
    sequence = 0
    def handler(result):
        nonlocal sequence
        sequence += 1
        print('[{}] Got: {}'.format(sequence, result))
    return handler
```

äÿŃéİćæŸŕä¡£çŤĺéŮ■åŇĚæŰźåijŕçŽĎäÿĂäÿłä¿Ńå■ŘïijŽ

```
>>> handler = make_handler()
>>> apply_async(add, (2, 3), callback=handler)
[1] Got: 5
>>> apply_async(add, ('hello', 'world'), callback=handler)
[2] Got: helloworld
>>>
```

è£ŸæIJL'åŘęåď'ŰäÿĂäÿłæŻť'éńŸçžğçŽĎæŰźæşŢïijŇåŘŕäžěä¡£çŤĺå■ŘçÍŃæİěåőŇæĹŘåŘŇæăůçŽĎäžŇ

```
def make_handler():
    sequence = 0
    while True:
        result = yield
        sequence += 1
        print('[{}] Got: {}'.format(sequence, result))
```

åŕźäžŐå■ŘçÍŃïijŇä¡äéIJĂèęĄä¡£çŤĺåőČçŽĎ `send()` æŰźæşŢä¡IJäÿžåŻđèřČåĞ¡æŢřïijŇåęĆäÿŃæĽĂç

```
>>> handler = make_handler()
>>> next(handler) # Advance to the yield
>>> apply_async(add, (2, 3), callback=handler.send)
[1] Got: 5
>>> apply_async(add, ('hello', 'world'), callback=handler.send)
[2] Got: helloworld
>>>
```

èőÍèőž

å§žäžŐåŻđèřČåĞ¡æŢřçŽĎè¡ŕäžűéĂŽåÿÿéČ¡æIJL'åŘŕèČ¡åŘŸå¿Ůéİđåÿÿåď'■æİĆãĂĆäÿĂéČÍåĹĘåŐ§åŻ
åŻäæ■ď'ïijŇèŕůæśĆæĽ'ğèąŇåŠŇåď'ĎçŘĘçżŞæđIJäźŃéŮť'çŽĎæĽ'ğèąŇçŐŕåćČåőđéŹĚäÿŁåůšçżŘäÿćåď'śäžĘ
éĆčä¡äåŕśå£ĚéążåŐżèğčåĘşåęĆä¡Ţä£Ïå■ŸåŠŇæĄćåď'■çŻÿåĚşçŽĎçŁűæĂĄä£ąæĄŕäžĘãĂĆ

èĞşåřŚæIJL'äÿď'çğ■äÿżèęĄæŰźåijŘæİěæ■ŢèŐůåŠŇä£İå■ŸçŁűæĂĄä£ąæĄŕïijŇä¡äåŘŕäžěåIJÍäÿĂäÿłåŕź
äÿď'çğ■æŰźåijŘçŻÿæŕŤïijŇéŮ■åŇĚæĹŰèőÿæŸŕæŻť'åŁäè¡żéĞŘçžğåŠŇèĞłçĎűäÿĂçĆźïijŇåŻääÿžåőČäżňåĺ
åőČäžňè£ŸèČ¡èĞłåŁíæ■ŢèŐůæĽ'ĂæIJL'èćńä¡£çŤĺåĹřçŽĎåŘŸéĞŘãĂĆåŻäæ■ď'ïijŇä¡äæŮäéIJĂåŐżæŃĚå£Ć

åęĆæđIJä¡£çŤĺéŮ■åŇĚïijŇä¡äéIJĂèęĄæşĺæĎŘåŕźéĆčäžŻåŘŕä£őæŤźåŘŸéĞŘçŽĎæŞ■ä¡IJãĂĆåIJÍäÿŁé
`nonlocal` åčŕæŸŐèŕ■åŘěçŤĺæİěæŇĞçď'żæŐěäÿŃæİěçŽĎåŘŸéĞŘäijŽåIJÍåŻđèřČåĞ¡æŢřäÿ■èćńä£őæŤźã

èĂŇä¡£çŤĺäÿĂäÿłå■ŘçÍŃæİěä¡IJäÿžäÿĂäÿłåŻđèřČåĞ¡æŢřåŕśæŻť'æIJL'èűčäžĘïijŇåőČèů§éŮ■åŇĚæŰźä
æ§Řçğ■æĎŘäźL'äÿŁæİěèőšïijŇåőČæŸ¿å¿ŮæŻť'åŁäçőĂæť'ĄïijŇåŻääÿžæĂżåĚśåŕśäÿĂäÿłåĞ¡æŢřèĂŇåůšãĂ
åźűäÿŤïijŇä¡äåŘŕäžěå¿ĹèĞłçŤśçŽĎä£őæŤźåŘŸéĞŘèĂŇæŮäéIJĂåŐżä¡£çŤĺ `nonlocal`
åčŕæŸŐãĂĆ è£ŹçğğæŰźåijŘåŤŕäÿĂçijžçĆźåŕśæŸŕçŻÿåŕźäžŐåĚűäżŰPythonæŁĂæIJŕèĂŇèĺĂæĹŰèőÿæŕŤè
åŘęåď'Ůè£ŸæIJL'äÿĂäžŻæŕŤè¿ČéŽ¿æĞĆçŽĎéČĺåĹĘïijŇæŕŤåęĆä¡£çŤĺäźŃåL'■éIJĂèęĄèŕČçŤĺ
`next()` ïijŇåőđéŹĚä¡£çŤĺæŮůè£Źäÿłæ■ěéłď'å¿Ĺåőźæ ŸŞèćńå£ŸèőŕãĂĆ
åř¡çőąåęĆæ■ď'ïijŇå■ŘçÍŃè£ŸæIJL'åĚűäżŰçŤĺåď'ĎïijŇæŕŤåęĆä¡IJäÿžäÿĂäÿłåĘĚèĄŤåŻđèřČåĞ¡æŢřçŽĎåőŽ

åęĆæđIJä¡ääżĚäżĚåŘłéIJĂèęĄçżŹåŻđèřČåĞ¡æŢřäijäéĂŠéćÍåď'ŰçŽĎåĂijçŽĎèŕÍïijŇè£ŸæIJL'äÿĂçğ■ä
`partial()` çŽĎæŰźåijŘäź§å¿ĹæIJL'çŤĺãĂĆ åIJÍæšąæIJL'ä¡£çŤĺ `partial()`
çŽĎæŮůåĂŹïijŇä¡äåŘŕèČ¡çżŘåÿÿçIJŃåĹřäÿŃéİćè£Źçğ■ä¡£çŤĺlambdaèąĺè¿¿åijŘçŽĎåď'■æİĆäżčçăĄïijŽ

```
>>> apply_async(add, (2, 3), callback=lambda r: handler(r, seq))
[1] Got: 5
>>>
```

åŘŕäžěåŘĆèĂČ7.8åřŘèŁĆçŽĎåĞääÿłçď'žä¿ŃïijŇæŢŹä¡äåęĆä¡Ţä¡£çŤĺ `partial()`
æİěæŻť'æŤźåŘĆæŢřç■¿åŘ■æİěçőĂåŇŰäÿŁè£řäžčçăĄãĂĆ

9.11 7.11 åĘĚèĄŤåŻđèřČåĞ¡æŢř

éŮőéćŸ

å¡Şä¡ăçijŰåĘŹä¡£çŤĺåŻđèřČåĞ¡æŢřçŽĎäžčçăĄçŽĎæŮűåĂŹïijŇæŃĚå£Čå¿Ĺåď ŽåřŘåĞ¡æŢřçŽĎæĽľ å
ä¡ăåÿŇæIJŻæĽ¿åĹřæ§ŘäÿłæŰźæşŢæĺěőľäžčçăĄçIJŃäÿŁåŐżæŻť åČŘæŸŕäÿĂäÿłæŹőéĂŽçŽĎæĽğèąŇåžĘ

èğčåĘşæŰźæąĹ

éĂŽè£Ğä¡£çŤĺçŤ§æĹŘåŹĺåŠŇå■ŘçĺŃåŘŕäžěä¡£å¿ŮåŻđèřČåĞ¡æŢřåĘĚèĄŤåIJĺæ§ŘäÿłåĞ¡æŢřäÿ■ãĂĆ
äÿžäžĘæijŤçď žèŕť æŸŐïijŇåĄĞèő¿ä¡ăæIJĽåęĆäÿŃæĽĂçď žçŽĎäÿĂäÿłæĽğèąŇæ§Řçĝ■èőąçőŮäżżåŁąçĎŭ

```
def apply_async(func, args, *, callback):
    # Compute the result
    result = func(*args)

    # Invoke the callback with the result
    callback(result)
```

æŐěäÿŃæĺěőľæĹŚäžňçIJŃäÿĂäÿŃäÿŃéĺćçŽĎäžčçăĄïijŇåőČåŇĚåŘńäžĘäÿĂäÿł
`Async` çśżåŠŇäÿĂäÿł `inlined_async` èčĚéěřåŹĺïijŽ

```
from queue import Queue
from functools import wraps

class Async:
    def __init__(self, func, args):
        self.func = func
        self.args = args

def inlined_async(func):
    @wraps(func)
    def wrapper(*args):
        f = func(*args)
        result_queue = Queue()
        result_queue.put(None)
        while True:
            result = result_queue.get()
            try:
                a = f.send(result)
                apply_async(a.func, a.args, callback=result_queue.
↪put)
            except StopIteration:
                break
    return wrapper
```

è£Źäÿď äÿłäžčçăĄçĽĞæőţåĚĄèőÿä¡ăä¡£çŤĺ `yield` èŕ■åŘěåĘĚèĄŤåŻđèřČæ■ěéłď ãĂĆæŕŤåęĆïijŽ

```
def add(x, y):
    return x + y

@inlined_async
def test():
    r = yield Async(add, (2, 3))
    print(r)
    r = yield Async(add, ('hello', 'world'))
    print(r)
    for n in range(10):
        r = yield Async(add, (n, n))
        print(r)
    print('Goodbye')
```

åęĆæđIJä¡ăèřČçŤÍ `test()` ïijŇä¡ăäijŽå¿ŮåĹřçśżäijijåęĆäÿŃçŽĎè¿ŞåĞžïijŽ

```
5
helloworld
0
2
4
6
8
10
12
14
16
18
Goodbye
```

ä¡ăäijŽåŔŚçŐřïijŇéŹď äžĘéĆčäÿłçL'źåĹńçŽĎèčĚéěřåŹĺåŠŇ `yield` èŕ■åŘěåď Ŭïij ŇåĚűäżŰåIJřæŰźåźűæšąæIJL'åĞžçŐřäżżä¡ŢçŽĎåŻđèřČåĞ¡æŢř(åĚűåőđæŸŕåIJĺåŘŐåŘřåőŽäź

èőĺèőž

æIJňåřŘèŁĆäijŽåőđåőđåIJĺåIJĺçŽĎæţŃèŕŢä¡ăåĚşäžŐåŻđèřČåĞ¡æŢřăĂçŤ§æĹŘåŹĺåŠŇæŐğåĹűæţÃçŽ

éęŰåĚĹïijŇåIJĺéIJĂèęĄä¡£çŤĺåĹřåŻđèřČçŽĎäżčçăĄäÿ■ïijŇåĚşéŤőçĆźåIJĺäžŐå¡ŞåL'■èőąçŐŮåůëä¡IJäij
å¡ŞèőąçŐŮéĞ■åŘřæŮűïijŇåŻđèřČåĞ¡æŢřèćńèřČçŤĺæĺěçżğçż■åď ĎçŘĘçżŞæđIJãĂĆapply_async()
åĞ¡æŢřæijŤçď žäžĘæL'ğèąŇåŻđèřČçŽĎåőđéŹĚéĂżè¿ŚïijŇ åř¡çőąåőđéŹĚæČĚåĘţäÿ■åőČåŘřèČ¡äijŽæŻť'åŁ

èőąçŐŮçŽĎæŽĆåĄIJäÿŐéĞ■åŘřæĂĺèůřèů§çŤ§æĹŘåŹĺåĞ¡æŢřçŽĎæL'ğèąŇæĺąåđŃäÿ■èřŃèĂŇåŘĹãĂ
åĚůä¡ŞæĺěèőšïijŇyield æŞ■ä¡IJäijŽä¡£äÿĂäÿłçŤ§æĹŘåŹĺåĞ¡æŢřäžğçŤ§äÿĂäÿłåĂijåźűæŽĆåĄIJãĂĆ
æŐěäÿŃæĺěèřČçŤĺçŤ§æĹŘåŹĺçŽĎ `__next__()` æĹŰ `send()`
æŰźæşŢåŘĹäijŽèőľ'åőČäżŐæŽĆåĄIJåď'Ďçżğçż■æL'ğèąŇãĂĆ

æăźæ■őè£ŹäÿłæĂĺèůřïijŇè£ŹäÿĂåřŘèŁĆçŽĎæăÿå£ČåřśåIJĺ `inline_async()`
èčĚéěřåŹĺåĞ¡æŢřäÿ■äžĘãĂĆ åĚşéŤőçĆźåřśæŸřïijŇèčĚéěřåŹĺäijŽéĂŘæ■ěéĄ■åŐEçŤ§æĹŘåŹĺåĞ¡æŢřçŽĎæ
`yield` èŕ■åŘěïijŇæŕŘäÿĂæňąäÿĂäÿłãĂĆ äÿžäžĘè£ŹæăůůåĄŽïijŇåĹŽåijĂåğŃçŽĎæŮűůåĂŹåĹŽåżžäžĘäÿĂä
`result` éŸ§åĹŮůźůåŘŚéĞŃéĺćæŤ¿åĚěäÿĂäÿł `None` åĂijãĂĆ

çĎűåŘŐåijĂåğŃäÿĂäÿłå¿łçŐŕæŞ■ä¡IJïijŇäżŐéŸ§åĹŮäÿ■åŘŰåĞžçżŞæđIJåĂijåźűåŘŚéĂĄçżŹçŤ§æĹŘåŹĺï
`yield` èŕ■åŘěïijŇ åIJĺè£ŹéĞŇäÿĂäÿł `Async` çŽĎåőđä¿ŃèćńæŐěåŘŮåĹřãĂĆçĎűåŘŐå¿łçŐŕåijĂåğŃæčĂæ
`apply_async()` ãĂĆ çĎűèĂŇïijŇè£ŹäÿłèőąçőŮæIJL'äÿłæIJĂèŕąåijĆéČĺåĹĘæŸŕåőČåźűæšąæIJL'ä¡£çŤĺäÿ
`put()` æŰźæşŢæİěåŻđèřČãĂĆ

è£ŹæŮűåĂŹïijŇæŸŕæŮűåĂŹèŕęçżĘèğčéĞŁäÿŃåĹřåžŢåŘŚçŤ§äžĘäżĂäźĹäžĘãĂĆäÿżå¿łçŐŕçńŃå■şè£'
`get()` æŞ■ä¡IJãĂĆ åęĆæđIJæŢřæ■őå■ŸåIJĺïijŇåőČäÿĂåőŽæŸŕ `put()`
åŻđèřČå■ŸæŤ¿çŽĎçżŞæđIJãĂĆåęĆæđIJæšąæIJL'æŢřæ■őïijŇéĆčäźĹåĚĹæŽĆåĄIJæŞ■ä¡IJåźűç■L'å¿ĚçżŞæ
è£ŹäÿłåĚůä¡ŞæĂŐæăůåőđçŐřæŸŕçŤś `apply_async()` åĞ¡æŢřæİěåĘşåőŽçŽĎãĂĆ
åęĆæđIJä¡ääÿ■çżÿä£ąäijŽæIJL'è£ŹäźĹçěđåěĞçŽĎäžŃæČĚïijŇä¡ääŘŕäžěä¡£çŤĺ
`multiprocessing` åžŞæİěèŕŢäÿĂäÿŃïijŇ åIJĺå■ŢçŃňçŽĎè£ŻçĺŃäÿ■æL'ğèąŇåijĆæ■ěèőąçőŮæŞ■ä¡IJïijŇ

```
if __name__ == '__main__':
    import multiprocessing
    pool = multiprocessing.Pool()
    apply_async = pool.apply_async

    # Run the test function
    test()
```

åőđéŹĚäÿŁä¡ääijŽåŘŚçŐřè£ŹäÿłçIJ§çŽĎåřśæŸŕè£ŹæăůçŽĎïijŇä¡ĘæŸŕèęĄèğčéĞŁæÿĚæěŽåĚůä¡ŞçŽĎ

åřĘåď■æİĆçŽĎæŐğåĹűæţĄéŽŘèŮŘåĹřçŤ§æĹŘåŹĺåĞ¡æŢřèČŇåŘŐçŽĎä¿Ńå■ŘåIJĺæăĞåĞĘåžŞåŠŇç
æŕŤåęĆïijŇåIJĺ `contextlib` äÿ■çŽĎ `@contextmanager`
èčĚéěřåŹĺä¡£çŤĺäžĘäÿĂäÿłäżď'äžžèť'źèğčçŽĎæŁĂåůğïijŇ éĂŽè£ĞäÿĂäÿł `yield`
èŕ■åŘěåřĘè£ŻåĚěåŠŇçężåijĂäÿŁäÿŃæŰĞçőąçŘĘåŹĺçšŸåŘĹåIJĺäÿĂèţůãĂĆ
åŘęåď'ŰéİđåÿÿæţĄèąŇçŽĎ `Twisted` åŇĚäÿ■äź§åŇĚåŘńäžĘéİđåÿÿçśżäijijçŽĎåĘĚèĄŤåŻđèřČãĂĆ

9.12 7.12 èő£éŮőéŮ■åŇĚäÿ■åőŽäźĹçŽĎåŘŸéĞŘ

éŮőéćŸ

ä¡äæČşèęĄæL'l'åśŢåĞ¡æŢřäÿ■çŽĎæ§ŘäÿłéŮ■åŇĚïijŇåĚĄèöÿåőČèČ¡èő£éŮőåŠŇä£őæŤźåĞ¡æŢřçŽĎå

èğčåĘşæŰźæąĹ

éĂŽåÿÿæİěèőšïijŇéŮ■åŇĚçŽĎåĘĚéČĺåŘŸéĞŘåŕźäžŐåď'ŰçŢŇæİěèőšæŸŕåőŇåĚĺéŽŘèŮŘçŽĎãĂĆ
ä¡ĘæŸŕïijŇä¡ääŘŕäžěéĂŽè£ĞçijŰåĘŹèő£éŮőåĞ¡æŢřåźűåřĘåĚűä¡IJäÿžåĞ¡æŢřåśđæĂğçżŚåőŽåĹřéŮ■åŇĚäÿ

```
def sample():
    n = 0
    # Closure function
    def func():
        print('n=', n)

    # Accessor methods for n
    def get_n():
        return n
```

(continues on next page)

(continued from previous page)

```
    def set_n(value):
        nonlocal n
        n = value

    # Attach as function attributes
    func.get_n = get_n
    func.set_n = set_n
    return func
```

äÿŃéİćæŸŕä¡£çŤĺçŽĎä¿Ńå■Ř:

```
>>> f = sample()
>>> f()
n= 0
>>> f.set_n(10)
>>> f()
n= 10
>>> f.get_n()
10
>>>
```

èőĺèőž

äÿžäžĘèŕť æŸŐæÿĚæěŽåőČåęĆä¡Ţåůëä¡IJçŽĎïijŇæIJL'äÿď çĆźéIJĂèęĄèğčéĞŁäÿĂäÿŃãĂĆéęŰåĚĹïij
åčřæŸŐåŘŕäżěèőľæĹŚäżňçijŰåĘźåĞ¡æŢřæİěä£őæŤźåĘĚéČĺåŘŸéĞŘçŽĎåĂijãĂĆ
åĚűæňąïijŇåĞ¡æŢřåśđæĂğåĚĄèőÿæĹŚäżňçŤĺäÿĂçğ■å¿ĹçőĂå■ŢçŽĎæŰźåijŘåřĘèő£éŮőæŰźæşŢçżŚåőŽåĹ

è£ŸåŘŕäżěè£ŻäÿĂæ■ěçŽĎæL'ľåśŢïijŇèőľéŮ■åŇĚæĺąæŃ§çśżçŽĎåőđä¿ŃãĂĆä¡ăèęĄåĄźçŽĎäżĚäżĚa

```
import sys
class ClosureInstance:
    def __init__(self, locals=None):
        if locals is None:
            locals = sys._getframe(1).f_locals

        # Update instance dictionary with callables
        self.__dict__.update((key,value) for key, value in locals.
↪items()
                                if callable(value) )
    # Redirect special methods
    def __len__(self):
        return self.__dict__['__len__']()

# Example use
def Stack():
    items = []
    def push(item):
```

(continues on next page)

(continued from previous page)

```
        items.append(item)

    def pop():
        return items.pop()

    def __len__():
        return len(items)

    return ClosureInstance()
```

äÿŃéİćæŸŕäÿĂäÿłäžď'äžŠåijŔäijŽèŕİæİěæijŤçď'žåőČæŸŕåęĆä¡Ţåůěä¡IJçŽĎïijŽ

```
>>> s = Stack()
>>> s
<__main__.ClosureInstance object at 0x10069ed10>
>>> s.push(10)
>>> s.push(20)
>>> s.push('Hello')
>>> len(s)
3
>>> s.pop()
'Hello'
>>> s.pop()
20
>>> s.pop()
10
>>>
```

æIJL'èűčçŽĎæŸŕïijŇè£ŹäÿłäžčçăĄè£ŘèąŇèţůæİěäijŽæŕŤäÿĂäÿłæŹőéĂŽçŽĎçśżåőŽäźL'èęĄå£ńå¿Ĺåď

```
class Stack2:
    def __init__(self):
        self.items = []

    def push(self, item):
        self.items.append(item)

    def pop(self):
        return self.items.pop()

    def __len__(self):
        return len(self.items)
```

åęĆæđIJè£Źæăůå ĄŽïijŇä¡ääijŽå¿ŮåĹřçśżäijijåęĆäÿŃçŽĎçżŞæđIJïijŽ

```
>>> from timeit import timeit
>>> # Test involving closures
>>> s = Stack()
>>> timeit('s.push(1);s.pop()', 'from __main__ import s')
```

(continues on next page)

(continued from previous page)

```
0.9874754269840196
>>> # Test involving a class
>>> s = Stack2()
>>> timeit('s.push(1);s.pop()', 'from __main__ import s')
1.0707052160287276
>>>
```

çżŞæđIJæŸ¿çď žïijŇéŮ■åŇĚçŽĎæŰźæąĹè£ŘèąŇèţůæİěèęĄå£ńåď ğæęĆ8%ïijŇåď ğéČÍåĹĘåŐ§åŻäæŸ
éŮ■åŇĚæŻť å£ńæŸŕåŻääÿžäÿ■äijŽæűL'åŘŁåĹřéćİåď ŰçŽĎselfåŘŸéĞŘãĂĆ

Raymond HettingerårźäžŐè£ŹäÿłéŮőécŸèő¿èőąåĞžäžĘæŻť åŁäéŽ¿äżěçŘĘèğčçŽĎæŤźè£ŻæŰźæąĹãĂ
èĂŇäÿŤåőČåŘłæŸŕçIJ§åőđçśżçŽĎäÿĂäÿłåěĞæĂłçŽĎæŻ£æ■ćèĂŇåůšïijŇä¿ŃåęĆïijŇçśżçŽĎäÿżèęĄçĹźæ
åźűäÿŤä¡äèęĄåĄŽäÿĂäžŻåĚűäżŰçŽĎåůěä¡IJæĹ■èČ¡èőľäÿĂäžŻçĹźæőŁæŰźæşŢçŤ§æŢĹ(æŕŤåęĆäÿŁéİć
`ClosureInstance` äÿ■éĞ■åĘŹè£ĞçŽĎ `__len__()` åőđçŐřãĂĆ)

æIJĂåŘŐïijŇä¡äåŘŕèČ¡è£ŸäijŽèőľåĚűäżŰéŸĚèŕżä¡ääżčçăĄçŽĎäžžæĎ§åĹřçŰŚæČŚïijŇäÿżäżĂäźĹåőĆ
(å¡ŞçĎűïijŇäżŰäżňäź§æČşç§ěéĄŞäÿżäżĂäźĹåőČè£ŘèąŇèţůæİěäijŽæŻť å£ń)ãĂĆåř¡çőąåęĆæ■ď ïijŇè£Źåŕźä

æĂżä¡ŞäÿŁèőšïijŇåIJĺéĚ■ç¡őçŽĎæŮűåĂŹçżŹéŮ■åŇĚæůżåŁäæŰźæşŢäijŽæIJL'æŻť åď ŽçŽĎåőđçŤĺåŁ
æŕŤåęĆä¡äéIJĂèęĄéĞ■ç¡őåĘĚéČĺçŁűæĂĄãĂĄåĹůæŰřçijŞåĘšåŇžãĂĄæÿĚéŹď çijŞå■ŸæĹŰåĚűäżŰçŽĎåŘ

10 çňňåĚńçńäïijŽçśżäÿŐåŕźèśą

æIJňçńääÿżèęĄåĚşæşĺçĆźçŽĎæŸŕåŠŇçśżåőŽäźL'æIJL'åĚşçŽĎåÿÿèğĄçijŰçĺŃæĺąåđŃãĂĆåŇĚæŃňèőľ
çśżåřĄèčĚæŁĂæIJŕãĂĄçżğæL'£ãĂĄåĘĚå■ŸçőąçŘĘäżèåŘŁæIJL'çŤĺçŽĎèő¿èőąæĺąåijŘãĂĆ

Contents:

10.1 8.1 æŤźåŘŸåŕźèśąçŽĎå■ŮçňęäÿšæŸ¿çď ž

éŮőécŸ

ä¡äæČşæŤźåŘŸåŕźèśąåőđä¿ŃçŽĎæL'Şå■řæĹŰæŸ¿çď žè¿ŞåĞžïijŇèőľåőČäżňæŻť åĚůåŘŕèŕżæĂgãĂĆ

èğčåĘşæŰźæąĹ

èęĄæŤźåŘŸäÿĂäÿłåőđä¿ŃçŽĎå■Ůçňęäÿšèąĺçď žïijŇåŘŕéĞ■æŰřåőŽäźL'åőČçŽĎ
`__str__()` åŠŇ `__repr__()` æŰźæşŢãĂĆä¿ŃåęĆïijŽ

```
class Pair:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def __repr__(self):
```

(continues on next page)

(continued from previous page)

```
        return 'Pair({0.x!r}, {0.y!r})'.format(self)

    def __str__(self):
        return '({0.x!s}, {0.y!s})'.format(self)
```

__repr__() æŰźæşŢè£ŤåŻđäÿĂäÿłåőđä¿ŃçŽĎäžčçăĄèąĺçď žå¡ćåijŘïijŇéĂŽåÿÿçŤĺæĺěéĞ■æŰřæđĎ
åĘĚç¡őçŽĎ repr() åĞ¡æŢřè£ŤåŻđè£ŹäÿłÅ■ŮçňęäÿšïijŇèů§æĹŚäžňä¡£çŤĺäžď äžŠåijŘèğčéĞŁåŹĺæŸ¿çď ž
__str__() æŰźæşŢåřĘåőđä¿Ńè¡ňæ■ćäÿžäÿĂäÿłå■ŮçňęäÿšïijŇä¡£çŤĺ str() æĹŰ
print() åĞ¡æŢřäijŽè¿ŞåĞžè£ŹäÿłÅ■ŮçňęäÿšãĂĆæřŤåęĆïijŽ

```
>>> p = Pair(3, 4)
>>> p
Pair(3, 4) # __repr__() output
>>> print(p)
(3, 4) # __str__() output
>>>
```

æĹŚäžňåIJĺè£ŹéĞŇè£ŸæijŤçď žäžĘåIJĺæăijåijŘåŇŰçŽĎæŮűåĂŹæĂŐæăůä¡£çŤĺäÿ■åŘŇçŽĎå■Ůçňęäÿ
çĽźåĹńæĺěèőšïijŇ!r æăijåijŘåŇŰäžčçăĄæŇĞæŸŐè¿ŞåĞžä¡£çŤĺ
__repr__() æĺěäžčæŻ£éżŸèőď çŽĎ __str__() ãĂĆ
ä¡ššŘŕäžěçŤĺåĽ■éĺćçŽĎçśżæĺěèřŢçĺĂæţŃèřŢäÿŃïijŽ

```
>>> p = Pair(3, 4)
>>> print('p is {0!r}'.format(p))
p is Pair(3, 4)
>>> print('p is {0}'.format(p))
p is (3, 4)
>>>
```

èőĺèőž

èĞłåőŽäźĽ __repr__() åŠŇ __str__() éĂŽåÿÿæŸŕå¿Ĺåě¡çŽĎäźžæČŕïijŇåŻääÿžåőČèČ¡çőĂåŇŰ
ä¿ŃåęĆïijŇåęĆæđIJäżĚäżĚåŘłæŸŕæĽŞå■řè¿ŞåĞžæĹŰæŮěå£Ůè¿ŞåĞžæ§Řäÿłåőđä¿ŃïijŇéĆčäźĹçĺŃåžŘåŚ

__repr__() çŤ§æĹŘçŽĎæŰĞæIJňå■ŮçňęäÿšæăĞåĞĘåĄŽæşŢæŸŕéIJĂèęĄèőĺ
eval(repr(x)) == x äÿžçIJ§ãĂĆ åęĆæđIJåőđåIJĺäÿ■èČ¡è£ŹæăůÅ■ŘåĄŽïijŇåžŤèřěåĹŽåžžäÿĂäÿłæIJL
< åŠŇ > æŃňèţůæĺěãĂĆæřŤåęĆïijŽ

```
>>> f = open('file.dat')
>>> f
<_io.TextIOWrapper name='file.dat' mode='r' encoding='UTF-8'>
>>>
```

åęĆæđIJ __str__() æšąæIJL'èćňåőŽäźL'ïijŇéĆčäźĹåřšäijŽä¡£çŤĺ __repr__()
æĺěäžčæŻ£è¿ŞåĞžãĂĆ

äÿŁéĺćçŽĎ format() æŰźæşŢçŽĎä¡£çŤĺçIJŃäÿŁåŐžå¿ĹæIJL'èűčïijŇæăijåijŘåŇŰäžčçăĄ
{0.x} åŕźåžŤçŽĎæŸŕçňň1äÿłåŘĆæŢřçŽĎxåśđæĂğãĂĆ åŻăæ■ď'ïijŇåIJĺäÿŃéĺćçŽĎåĞ¡æŢřäÿ■ïijŇ0åőđéŹ
self æIJňèžňïijŽ

```
def __repr__(self):
    return 'Pair({0.x!r}, {0.y!r})'.format(self)
```

ä¡IJäÿžè£Źçğ■åőđçŐřçŽĎäÿĂäÿłæŻ£äżčïijŇä¡ääź§åŘŕäżěä¡£çŤÍ %æŞ■ä¡IJçňęïijŇåřśåČŘäÿŃéİćè£ŹæăůïijŽ

```
def __repr__(self):
    return 'Pair(%r, %r)' % (self.x, self.y)
```

10.2 8.2 èĞłåőŽäźĽå■ŮçňęäÿšçŽĎæăijåijŘåŇŰ

éŮőécŸ

ä¡äæČşéĂŽè£Ğ `format()` åĞ¡æŢřåŠŇå■ŮçňęäÿšæŰźæşŢä¡£å¿ŮäÿĂäÿłåŕźèśąèČ¡æŤŕæŇĄèĞłåőŽäźI

èğčåĘşæŰźæąĹ

äÿžäžĘèĞłåőŽäźĽå■ŮçňęäÿšçŽĎæăijåijŘåŇŰïijŇæĹŚäżňéIJĂèęĄåIJÍçśżäÿŁéİćåőŽäźĽ `__format__()` æŰźæşŢãĂĆä¿ŃåęĆïijŽ

```
_formats = {
    'ymd' : '{d.year}-{d.month}-{d.day}',
    'mdy' : '{d.month}/{d.day}/{d.year}',
    'dmy' : '{d.day}/{d.month}/{d.year}'
    }

class Date:
    def __init__(self, year, month, day):
        self.year = year
        self.month = month
        self.day = day

    def __format__(self, code):
        if code == '':
            code = 'ymd'
        fmt = _formats[code]
        return fmt.format(d=self)
```

çŐřåIJÍ `Date` çśżçŽĎåőđä¿ŃåŘŕäżěæŤŕæŇĄæăijåijŘåŇŰæŞ■ä¡IJäžĘïijŇåęĆåŘŇäÿŃéİćè£ŹæăůïijŽ

```
>>> d = Date(2012, 12, 21)
>>> format(d)
'2012-12-21'
>>> format(d, 'mdy')
'12/21/2012'
>>> 'The date is {:ymd}'.format(d)
```

(continues on next page)

(continued from previous page)

```
'The date is 2012-12-21'
>>> 'The date is {:mdy}'.format(d)
'The date is 12/21/2012'
>>>
```

èőĺèőž

`__format__()` æŰźæşŢçżŹPythonçŽĎå■ŮçňęäÿšæäijåijŘåŇŰåŁ§èČ¡æŘŘä¿ŻäžĘäÿĂäÿłéŠľå■ŘãĂĆ
è£ŹéĞŇéIJĂèęĄçIĂéĞ■åijžèřČçŽĎæŸŕæäijåijŘåŇŰäžčçăĄçŽĎèğčæđŘåůëä¡IJåőŇåĚĺçŤśçśżèĞłåůśåĘşåőŽ
ä¿ŃåęĆïijŇåŘĆèĂČäÿŃéĺćæĺěèĞł `datetime` æĺąåĺŮäÿ■çŽĎäžčçăĄïijŽ

```
>>> from datetime import date
>>> d = date(2012, 12, 21)
>>> format(d)
'2012-12-21'
>>> format(d,'%A, %B %d, %Y')
'Friday, December 21, 2012'
>>> 'The end is {:%d %b %Y}. Goodbye'.format(d)
'The end is 21 Dec 2012. Goodbye'
>>>
```

åŕźäžŐåĘĚç¡őçśżåđŃçŽĎæäijåijŘåŇŰæIJL'äÿĂäžŻæăĞåĞĘçŽĎçžęåőŽãĂĆ
åŘŕäžěåŘĆèĂČ stringæĺąåĺŮæŰĞæąč èŕť'æŸŐãĂĆ

10.3 8.3 èőľåŕźèśąæŤŕæŇĄäÿŁäÿŃæŰĞçőąçŘĘå■Ŕèőő

éŮőécŸ

ä¡äæČşèőľä¡äçŽĎåŕźèśąæŤŕæŇĄäÿŁäÿŃæŰĞçőąçŘĘå■Ŕèőő(withèŕ■åŘě)ãĂĆ

èğčåĘşæŰźæąĹ

äÿžäžĘèőľäÿĂäÿłåŕźèśąåĚijåőź `with` èŕ■åŘěïijŇä¡äéIJĂèęĄåőđçŐř `__enter__()`
åŠŇ `__exit__()` æŰźæşŢãĂĆ ä¿ŃåęĆïijŇèĂČèŹŚåęĆäÿŃçŽĎäÿĂäÿłçśżïijŇåőČèČ¡äÿžæĹŚäžňåĹŻåżžäÿ

```
from socket import socket, AF_INET, SOCK_STREAM

class LazyConnection:
    def __init__(self, address, family=AF_INET, type=SOCK_STREAM):
        self.address = address
        self.family = family
        self.type = type
        self.sock = None
```

(continues on next page)

(continued from previous page)

```
    def __enter__(self):
        if self.sock is not None:
            raise RuntimeError('Already connected')
        self.sock = socket(self.family, self.type)
        self.sock.connect(self.address)
        return self.sock

    def __exit__(self, exc_ty, exc_val, tb):
        self.sock.close()
        self.sock = None
```

è£ŹäÿłçśżçŽĎåĚşéŤőçĽźçĆźåIJÍäžŐåőČèąÍçď žäžĘäÿĂäÿłç¡ŚçżIJè£đæŐěïijŇä¡ĘæŸŕåĹİåğŃåŇŰçŽĎa
è£đæŐěçŽĎåżžçńŃåŠŇåĚşéŮ■æŸŕä¡£çŤÍ `with` èŕ■åŘěèĞłåŁÍåőŇæĹŘçŽĎïijŇä¿ŃåęĆïijŽ

```
from functools import partial

conn = LazyConnection(('www.python.org', 80))
# Connection closed
with conn as s:
    # conn.__enter__() executes: connection open
    s.send(b'GET /index.html HTTP/1.0\r\n')
    s.send(b'Host: www.python.org\r\n')
    s.send(b'\r\n')
    resp = b''.join(iter(partial(s.recv, 8192), b''))
    # conn.__exit__() executes: connection closed
```

èőÍèőž

çijŰåĘŹäÿŁäÿŃæŰĞçőąçŘĘåŹÍçŽĎäÿżèęĄåŐ§çŘĘæŸŕä¡ăçŽĎäżčçăĄäijŽæŤ¿åĹř
`with` èŕ■åŘěåÍŮäÿ■æĽğèąŇãĂĆ å¡ŞåĞžçŐř `with` èŕ■åŘěçŽĎæŮűåĂŹïijŇåŕźèśąçŽĎ
`__enter__()` æŰźæşŢèćńèğęåŘŚïijŇ åőČè£ŤåŻđçŽĎåĂij(åęĆæđIJæIJĽçŽĎèŕÍ)äijŽèćńèţŃåĂijçżŹ
`as` åčřæŸŐçŽĎåŘŸéĞŘãĂĆçĎűåŘŐïijŇwith èŕ■åŘěåÍŮéĞŇéÍćçŽĎäżčçăĄäijĂäğŃæĽğèąŇãĂĆ
æIJĂåŘŐïijŇ`__exit__()` æŰźæşŢèćńèğęåŘŚè£ŻèąŃæÿĚçŘĘåůěä¡IJãĂĆ

äÿ■çőą `with` äżčçăĄåÍŮäÿ■åŘŚçŤ§äżĂäźĹïijŇäÿŁéÍćçŽĎæŐğåĹűæţĄéČ¡äijŽæĽğèąŇåőŇïijŇåřśçőŮ
äžŃåőđäÿŁïijŇ`__exit__()` æŰźæşŢçŽĎçňňäÿĽäÿłåŘĆæŢřåŇĚåŘńäžĘåijĆåÿÿçśżåđŃãĂĄäijĆåÿÿåĂijåŠ
`__exit__()` æŰźæşŢèČ¡èĞłåůśåĘşåőŽæĂŐæăůåĹ'çŤÍè£ŹäÿłåijĆåÿÿä£ąæĄŕïijŇæĹŰèĂĚå£¡çŢěåőČåźűě
åęĆæđIJ `__exit__()` è£ŤåŻđ `True` ïijŇéĆčäźĹåijĆåÿÿäijŽèćńæÿĚçl'žïijŇåřśåě¡åČŘäžĂäźĹéČ¡æšąåŘŚçŤ
`with` èŕ■åŘěåŘŐéÍćçŽĎçÍŃåžŘçżğçż■åIJÍæ■čåÿÿæĽğèąŇãĂĆ

è£ŸæIJĽäÿĂäÿłçżĘèŁĆéŮőéćŸåřśæŸŕ `LazyConnection`
çśżæŸŕåŘęåĚĄèöÿåđ'Žäÿł `with` èŕ■åŘěæÍěåţŇåěŮä¡£çŤÍè£đæŐěãĂĆ
å¿ĹæŸ¿çĎűïijŇäÿŁéÍćçŽĎåőŽäźĽäÿ■äÿĂæňąåŘłèČ¡åĚĄèöÿäÿĂäÿłsocketè£đæŐěïijŇåęĆæđIJæ■čåIJÍä¡£çŤ
`with` èŕ■åŘěïijŇ åŕśäijŽäžğçŤ§äÿĂäÿłåijĆåÿÿäžĘãĂĆäÿ■è£Ğä¡ššŘŕäžěåČŘäÿŃéÍćè£ŹæăůäŁőæŤźäÿŃäÿŁé

```
from socket import socket, AF_INET, SOCK_STREAM

class LazyConnection:
```

(continues on next page)

(continued from previous page)

```
    def __init__(self, address, family=AF_INET, type=SOCK_STREAM):
        self.address = address
        self.family = family
        self.type = type
        self.connections = []

    def __enter__(self):
        sock = socket(self.family, self.type)
        sock.connect(self.address)
        self.connections.append(sock)
        return sock

    def __exit__(self, exc_ty, exc_val, tb):
        self.connections.pop().close()

# Example use
from functools import partial

conn = LazyConnection(('www.python.org', 80))
with conn as s1:
    pass
    with conn as s2:
        pass
        # s1 and s2 are independent sockets
```

åIJÍçňňäžŇäÿłçL'ĹæIJňäÿ■ïijŇLazyConnectionçśżåŘŕäżěèćńçIJŃńåĄŽæŸŕæ§Řäÿłè£đæŐěåůěåŐĆã
æŕŘæňą__enter__() æŰźæşŢæL'ğèąŇçŽĎæŮűåĂŹïijŇåőČåď'■åĹűåĹŻåżžäÿĂäÿłæŰřçŽĎè£đæŐěåźűåì
__exit__() æŰźæşŢçőĂå■ŢçŽĎäżŐæäĹäÿ■åijźåĞžæIJĂåŘŐäÿĂäÿłè£đæŐěåźűåĚşéŮ■åőČãĂĆ
è£ŹéĞŇçÍ■å¿őæIJL'çĆźéŽ¿çŘĘèğčïijŇäÿ■è£ĞåőČèČ¡åĚĄèőÿåţŇåěŮä¡£çŤÍ with
èŕ■åŘěåĹŻåżžåď'Žäÿłè£đæŐěïijŇåřśåęĆäÿŁéİćæijŤçď'žçŽĎéĆčæäůãĂĆ

åIJÍéIJĂèęĄçőąçŘĘäÿĂäžŻèţĎæžŘæŕŤåęĆæŰĞäżűãĂĄç¡ŚçżIJè£đæŐěåŠŇéŤĄçŽĎçijŰçÍŃçŐŕåćČäÿ■
è£ŹäžŻèţĎæžŘçŽĎäÿĂäÿłäÿżèęĄçL'źå¿ĄæŸŕåőČäżňå£Ěéążèćńæ L'ŃåŁÍçŽĎåĚşéŮ■æĹŰéĞŁæŤ¿æİěçąőä£
ä¿ŃåęĆïijŇåęĆæđIJä¡äèŕůæśĆäžĘäÿĂäÿłéŤĄïijŇéĆčäźĹä¡äå£Ěéążçąőä£İäźŃåŘŐéĞŁæŤ¿äžĘąőČïijŇåŘęąĹ
éĂŽè£ĞåőđçŐř __enter__() åŠŇ __exit__() æŰźæşŢåźűä¡£çŤÍ with
èŕ■åŘěåŘŕäżěå¿Ĺåőźæ ŸŞçŽĎéĄ£åĚ■è£ŹäžŻéŮőécŸïijŇ åŻääÿž __exit__()
æŰźæşŢåŘŕäżěèőĺ'ä¡äæŮäéIJĂæŃĚå£Čè£ŹäžŻäžĘãĂĆ

åIJÍ contextmanager æĺąåİŮäÿ■æIJL'äÿĂäÿłæăĞåĞĘçŽĎäÿŁäÿŃæŰĞçőąçŘĘæŰźæąĹæĺąæİ£ïijŇå
åŘŇæŮűåIJÍ12.6åřŘèŁĆäÿ■è£ŸæIJL'äÿĂäÿłåŕźæIJňèŁĆçď'žä¿ŃçÍŃåžŘçŽĎçž£çÍŃåőL'åĚĺçŽĎä£őæŤźçL'Ĺ

10.4 8.4 åĹŻåżžåď'ğéĞŘåŕźèśąæŮűèŁĆçIJĄåĘĚå■ŸæŰźæşŢ

éŮőécŸ

ä¡äçŽĎçÍŃåžŘèęĄåĹŻåżžåď'ğéĞŘ(åŘŕèČ¡äÿŁçŹ¿äÿĞ)çŽĎåŕźèśąïijŇåŕijèĞť'å■äçŤÍå¿Ĺåď'ğçŽĎåĘĚå■Ÿ

èğčåĘşæŰźæąĹ

åŕźäžŐäÿżèęĄæŸŕçŤĺæİěå¡ŞæĹŘçőĂå■ŢçŽĎæŢřæ■őçżŞæđĎçŽĎçśżèĂŇèĺĂïijŇä¡ššŘŕäžěéĂŽè£ĞçżŹ
__slots__ åśđæĂğæİěæđĄåď'ğçŽĎåĞŘåřŚåőđä¿ŃæĽ'Ăå■ăçŽĎåĘĚå■ŸãĂĆæŕŤåęĆïijŽ

```
class Date:
    __slots__ = ['year', 'month', 'day']
    def __init__(self, year, month, day):
        self.year = year
        self.month = month
        self.day = day
```

å¡Şä¡ššŽäźĽ'__slots__åŘŐïijŇPythonåřśäijŽäÿžåőđä¿Ńä¡£çŤĺäÿĂçğ■æŻť'åŁăçť'ğåĞŚçŽĎåĘĚéČĺ
åőđä¿ŃéĂŽè£ĞäÿĂäÿłå¿ĹåřŘçŽĎåŻžåőŽåď'ğåřŘçŽĎæŢřçżĎæİěæđĎåżžïijŇèĂŇäÿ■æŸŕäÿžæŕŘäÿłåőđä¿Ń
åIJĺ__slots__äÿ■åĹŮåĞžçŽĎåśđæĂğåŘ■åIJĺåĘĚéČĺèćńæŸŤåřĎåĹřè£ŹäÿłæŢřçżĎçŽĎæŇĞåőŽåřŘæăĞ
ä¡£çŤĺslotsäÿĂäÿłäÿ■åě¡çŽĎåIJřæŰźåřśæŸŕæĹŚäžňäÿ■èČ¡åĘ■çżŹåőđä¿ŃæůżåŁăæŰřçŽĎåśđæĂğäžĘïijŇåĺ
__slots__äÿ■åőŽäźĽçŽĎéĆčäžŽåśđæĂğåŘ■ãĂĆ

èőĺèőž

ä¡£çŤĺslotsåŘŐèŁĆçIJAçŽĎåĘĚå■ŸäijŽèů§å■ŸåĆĺåśđæĂğçŽĎæŢřéĞŘåŠŇçśżåđŃæIJĽåĚşãĂĆ
äÿ■è£ĞïijŇäÿĂèĹňæİěèőšïijŇä¡£çŤĺåĹřçŽĎåĘĚå■ŸæĂżéĞŘåŠŇåřĘæŢřæ■őå■ŸåĆĺåIJĺäÿĂäÿłåĚĆçżĎäÿ■å
äÿžäžĘçżŹä¡ääÿĂäÿłçŻť'èğĆèőď'èŕĘïijŇåĄĞèő¿ä¡ääÿ■ä¡£çŤĺslotsçŻť'æŐěå■ŸåĆĺäÿĂäÿłDateåőđä¿ŃïijŇ
åIJĺ64ä¡■çŽĎPythonäÿŁéİćèęĄå■ăçŤĺ428å■ŮèŁĆïijŇèĂŇåęĆæđIJä¡£çŤĺäžĘslotsïijŇåĘĚå■Ŷå■ăçŤĺäÿŇéŹ
åęĆæđIJçĺŃåžŘäÿ■éIJĂèęĄåŘŇæŮűåĹŻåżžåď'ğéĞŘçŽĎæŮěæIJ§åőđä¿ŃïijŇéĆčäźĹè£ŹäÿłåřśèČ¡æđĄåď'ğ

åř¡çőąslotsçIJŃäÿŁåŐżæŸŕäÿĂäÿłå¿ĹæIJĽçŤĺçŽĎçĽźæĂğïijŇå¿Ĺåď'ŽæŮűåĂŹä¡ăè£ŸæŸŕå¿ŮåĞŘåřS
PythonçŽĎå¿Ĺåď'ŽçĽźæĂğéČ¡ä¿ÌèţŰäžŐæŹőéĂŽçŽĎå§žäžŐå■ŮåĚÿçŽĎåőđçŐřãĂĆ
åŘęåď'ŰïijŇåőŽäźĽäžĘslotsåŘŐçŽĎçśżäÿ■åĘ■æŤŕæŇĄäÿĂäžŽæŹőéĂŽçśżçĽźæĂğäžĘïijŇæŕŤåęĆåď'Žçżğ
åď'ğåď'ŽæŢřæČĚåĘţäÿŃïijŇä¡ššžŤèŕěåŘłåIJĺéĆčäžŽçżŘåÿÿèćńä¡£çŤĺåĹřçŽĎçŤĺä¡IJæŢřæ■őçżŞæđĎçŽĎçś
(æŕŤåęĆåIJĺçĺŃåžŘäÿ■éIJĂèęĄåĹŻåżžæ§ŘäÿłçśżçŽĎåĞăçŹ¿äÿĞäÿłåőđä¿Ńåŕźèśą)ãĂĆ

åĚşäžŐ__slots__çŽĎäÿĂäÿłåÿÿèğĄèŕŕåŇžæŸŕåőČåŘŕäžěä¡IJäÿžäÿĂäÿłåřĄèčĚåůěåĚůæİěéŸšæ■ć
åř¡çőąä¡£çŤĺslotsåŘŕäžěè¿¿åĹřè£ŹæăůçŽĎçŽőçŽĎïijŇä¡ĘæŸŕè£Źäÿłåźűäÿ■æŸŕåőČçŽĎåĹİèąůãĂĆ
__slots__æŻť'åď'ŽçŽĎæŸŕçŤĺæİěä¡IJäÿžäÿĂäÿłåĘĚå■ŸäijŸåŇŰåůěåĚůãĂĆ

10.5 8.5 åIJĺçśżäÿ■åřĄèčĚåśđæĂğåŘ■

éŮőécŸ

ä¡ăæČşåřĄèčĚçśżçŽĎåőđä¿ŃäÿŁéİćçŽĎâĂIJçğĄæIJĽâĂİæŢřæ■őïijŇä¡ĘæŸŕPythonèŕ■èĺĂåźűæšąæIJI

èğčåĘşæŰźæąĹ

PythonçĺŃåžŘåŚŸäÿ■åŐżä¿ÌèţŰèŕ■èĺĂçĽźæĂğåŐżåřĄèčĚæŢřæ■őïijŇèĂŇæŸŕéĂŽè£ĞéĄţå¿łäÿĂåőŽ
çňňäÿĂäÿłçžęåőŽæŸŕäžżä¡Ţäžěå■ŢäÿŃåĹŚçž£_åijĂåď'ťçŽĎåŘ■å■ŮéČ¡åžŤèŕěæŸŕåĘĚéČĺåőđçŐřãĂĆæŕŤå

```
class A:
    def __init__(self):
        self._internal = 0 # An internal attribute
        self.public = 1 # A public attribute

    def public_method(self):
        '''
        A public method
        '''
        pass

    def _internal_method(self):
        pass
```

Pythonáźűäÿ■äijŽçIJ§çŽĎéŸżæ■ćåĹńäžžèő£éŮőåĘĚéČĺåŘ■çğřãĂĆä¡ĘæŸŕåęĆæđIJä¡ăè£ŹäźĹåĄŽèĆŕ
åŘŇæŮűè£ŸèęĄæşĺæĎŔåĹřïijŇä¡£çŤĺäÿŃåĹŠçž£åijĂåď'ťçŽĎçžęåőŽåŘŇæăůéĂĆçŤĺäžŐæĺąåİŮåŘ■åŠŇæ
ä¿ŃåęĆïijŇåęĆæđIJä¡ăçIJŃåĹřæ§ŘäÿłæĺąåİŮåŘ■äžěå■ŢäÿŃåĹŠçž£åijĂåď'ť(æŕŤåęĆ_socket)ïijŇéĆčåőČåřś
çśżäijijçŽĎïijŇæĺąåİŮçžğåĹńåĞ¡æŢřæŕŤåęĆ sys._getframe()
åIJĺä¡£çŤĺçŽĎæŮűåĂŹåřśå¿ŮåŁäåĂ■åřŔå£ČäžĘãĂĆ

ä¡ăè£ŸåŔŕèČ¡äijŽéĄĞåĹřåIJĺçśżåőŽäźL'äÿ■ä¡£çŤĺäÿď'äÿłäÿŃåĹŠçž£(__)åijĂåď'ťçŽĎåŚ¡åŘ■ãĂĆæŕŤå

```
class B:
    def __init__(self):
        self.__private = 0

    def __private_method(self):
        pass

    def public_method(self):
        pass
        self.__private_method()
```

ä¡£çŤĺåŘŇäÿŃåĹŠçž£åijĂåğŃäijŽåŕijèĞť'èő£éŮőåŘ■çğřåŘŸæĹŘåĚűäżŰå¡ćåijŔãĂĆ
æŕŤåęĆïijŇåIJĺåL'■éĺćçŽĎçśżBäÿ■ïijŇçğĄæIJL'åśđæĂğäijŽèćńåĹĘåĹńéĞ■åŚ¡åŘ■äÿž
_B__private åŠŇ _B__private_method ãĂĆ è£ŹæŮűåĂŹä¡ăåŔŕèČ¡äijŽéŮőè£ŹæăůéĞ■åŚ¡åŘ■çŽĎ

```
class C(B):
    def __init__(self):
        super().__init__()
        self.__private = 1 # Does not override B.__private

    # Does not override B.__private_method()
    def __private_method(self):
        pass
```

è£ŹéĞŇïijŇçğĄæIJL'åŘ■çğř __private åŠŇ __private_method
èćńéĞ■åŚ¡åŘ■äÿž _C__private åŠŇ _C__private_method
ïijŇè£Źäÿłèů§çĹűçśżBäÿ■çŽĎåŘ■çğřæŸŕåőŃåĚĺäÿ■åŘŇçŽĎãĂĆ

èőÍèőž

äÿŁéÍćæŘŘåĹřæIJL'äÿď'çğ■äÿ■åŘŇçŽĎçijŰçăAçžęåőŽ(å■ŢäÿŃåĹŠçž£åŠŇåŘŇäÿŃåĹŠçž£)æÍěåŚ¡åŘ
åď'ğåď'ŽæŢřèĂŇèÍĂïijŇä¡äåžŤèŕěèőľä¡äçŽĎéÍđåĚňåĚśåŘ■çğřäžěå■ŢäÿŃåĹŠçž£åijĂåď't'ãĂĆä¡ĘæŸŕïijŇåę
åźűäÿŤæIJL'äžŽåĘĚéČÍåśđæĂğåžŤèŕěåIJÍå■Řçśżäÿ■éŽŘèŮŘèţůæÍěïijŇéĆčäźĹæL'■èĂČèŹŚä¡£çŤÍåŘŇäÿŃ

è£ŸæIJL'äÿĂçĆźèęĄæşÍæĎŘçŽĎæŸŕïijŇæIJL'æŮűåĂŹä¡äåőŽäźL'çŽĎäÿĂäÿłåŘŸéĞŘåŠŇæ§Řäÿłä£Íç

```
lambda_ = 2.0 # Trailing _ to avoid clash with lambda keyword
```

è£ŹéĞŇæĹŚäžňåźűäÿ■ä¡£çŤÍå■ŢäÿŃåĹŠçž£åL'■çijĂçŽĎåŐ§åŻăæŸŕåőČéĄ£åĚ■èŕŕèğčåőČçŽĎä¡£çŤÍ
(åęĆä¡£çŤÍå■ŢäÿŃåĹŠçž£åL'■çijĂçŽĎçŻőçŽĎæŸŕäÿžäžĘéŸšæ■ćåŚ¡åŘ■åĘšçłĄèĂŇäÿ■æŸŕæŇĞæŸŐè£Ź
éĂŽè£Ğä¡£çŤÍå■ŢäÿŃåĹŠçž£åŘŐçijĂåŘŕäžěèğčåĘşè£ŹäÿłéŮőécŸãĂĆ

10.6 8.6 åĹŻåżžåŘŕçőąçŘĘçŽĎåśđæĂğ

éŮőécŸ

ä¡äæČşçżŹæ§Řäÿłåőđä¿ŃattributeåćđåŁäéŹď'èő£éŮőäÿŐä£őæŤźäźŃåď'ŰçŽĎåĚűäżŰåď'ĎçŘĘéĂżè¿Śi

èğčåĘşæŰźæąĹ

èĞłåőŽäźL'æ§ŘäÿłåśđæĂğçŽĎäÿĂçğ■çőĂå■ŢæŰźæşŢæŸŕåřĘåőČåőŽäźL'äÿžäÿĂäÿłpropertyãĂĆ
ä¿ŃåęĆïijŇäÿŃéÍćçŽĎäżčçăĄåőŽäźL'äžĘäÿĂäÿłpropertyïijŇåćđåŁäåŕźäÿĂäÿłåśđæĂğçőĂå■ŢçŽĎçśżåđŃæ

```
class Person:
    def __init__(self, first_name):
        self.first_name = first_name

    # Getter function
    @property
    def first_name(self):
        return self._first_name

    # Setter function
    @first_name.setter
    def first_name(self, value):
        if not isinstance(value, str):
            raise TypeError('Expected a string')
        self._first_name = value

    # Deleter function (optional)
    @first_name.deleter
    def first_name(self):
        raise AttributeError("Can't delete attribute")
```

äÿŁè£řäżčçăĄäÿ■æIJL'äÿL'äÿłçŻÿåĚşèĄŤçŽĎæŰźæşŢïijŇè£ŹäÿL'äÿłæŰźæşŢçŽĎåŘ■å■ŮéČ¡å£Ěéążäÿ
çňňäÿĂäÿłæŰźæşŢæŸŕäÿĂäÿł `getter` åĞ¡æŢřïijŇåőČä¡£å¿Ů `first_name`

æĹŘäÿžäÿĂäÿłåśđæĂğãĂĆ åĚűäżŰäÿď äÿłæŰźæşŢçżŹ `first_name` åśđæĂğæůżåŁääžĘ `setter` åŠŇ `deleter` åĞ¡æŢřãĂĆ éIJĂèęĄåijžèřČçŽĎæŸřåŘłæIJĽåIJÍ `first_name` åśđæĂğèćńåĹŻåżžåŘŐïijŇ åŘŐéİćçŽĎäÿď äÿłèčĚéěřåŹÍ `@first_name.setter` åŠŇ `@first_name.deleter` æĽ■èČ¡èćńåőŽäźĽãĂĆ

propertyçŽĎäÿĂäÿłåĚşéŤőçĽźå¿ĄæŸřåőČçIJŃäÿŁåŐżèů§æŹőéĂŽçŽĎattributeæšąäżĂäźĹäÿď æăůïijŇ ä¡ĘæŸřèő£éŮőåőČçŽĎæŮűůåĂŹäijŽèĞłåŁÍèğęåŘŚ `getter` ãĂĄ`setter` åŠŇ `deleter` æŰźæşŢãĂĆä¿ŃåęĆïijŽ

```
>>> a = Person('Guido')
>>> a.first_name # Calls the getter
'Guido'
>>> a.first_name = 42 # Calls the setter
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "prop.py", line 14, in first_name
        raise TypeError('Expected a string')
TypeError: Expected a string
>>> del a.first_name
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
AttributeError: can`t delete attribute
>>>
```

åIJÍåőđçŐřäÿĂäÿłpropertyçŽĎæŮűůåĂŹïijŇåžŢåśĆæŢřæ■ő(åęĆæđIJæIJĽçŽĎèřİ)äż■çĎűéIJĂèęĄå■ŸåĆ åŻäæ■ď ïijŇåIJÍgetåŠŇsetæŰźæşŢäÿ■ïijŇä¡ääijŽçIJŃåĹřåŕź `_first_name` åśđæĂğçŽĎæŞ■ä¡IJïijŇè£Źäź§æŸřåőđéŹĚæŢřæ■őä£İå■ŸçŽĎåIJřæŰźãĂĆ åŘęåď ŰïijŇä¡ăåŘŕèČ¡è£ŸäijŽéŮőäÿžäżĂäźĹ `__init__()` æŰźæşŢäÿ■èő¿ç¡őäžĘ `self.first_name` èĂŇäÿ■æŸŕ `self._first_name` ãĂĆ åIJÍè£Źäÿłä¿Ńå■Řäÿ■ïijŇæĹŚäżňåĹŻåżžäÿĂäÿłpropertyçŽĎçŽőçŽĎåŕśæŸřåIJÍèő¿ç¡őattributeçŽĎæŮűůåĂŹ åŻäæ■ď ïijŇä¡ăåŘŕèČ¡æČşåIJÍåĹİåğŃåŇŰçŽĎæŮűůåĂŹäź§è£ŻèąŇè£Źçğ■çśżåđŃæčĂæ§ěãĂĆéĂŽè£ĞèőźçŽ `self.first_name` ïijŇèĞłåŁÍèřČçŤÍ `setter` æŰźæşŢïijŇ è£ŹäÿłæŰźæşŢéĞŇéİćäijŽè£ŻèąŇåŘĆæŢřçŽĎæčĂæ§ěïijŇåŘęåĹŹåŕśæŸřçŻť æŐěèő£éŮő `self._first_name` äžĘãĂĆ

è£ŸèČ¡åIJÍåůšå■ŸåIJÍçŽĎgetåŠŇsetæŰźæşŢå§žçąĂäÿłåőŽäźĽpropertyãĂĆä¿ŃåęĆïijŽ

```
class Person:
    def __init__(self, first_name):
        self.set_first_name(first_name)

    # Getter function
    def get_first_name(self):
        return self._first_name

    # Setter function
    def set_first_name(self, value):
        if not isinstance(value, str):
            raise TypeError('Expected a string')
        self._first_name = value
```

(continues on next page)

(continued from previous page)

```
    # Deleter function (optional)
    def del_first_name(self):
        raise AttributeError("Can't delete attribute")

    # Make a property from existing get/set methods
    name = property(get_first_name, set_first_name, del_first_name)
```

èőÍèőž

äÿĂäÿłpropertyåśđæĂğåĚűåőđåřśæŸŕäÿĂçşżåĹŮçŻÿåĚşçżŚåőŽæŰźæşŢçŽĎéŻĘåŘĹãĂĆåęĆæđIJä¡åå
åřśäijŽåŘŚçŐřpropertyæIJňèžńçŽĎfgetãĂĄfsetåŠŇfdelåśđæĂğåřśæŸŕçśżéĞŇéİćçŽĎæŹőéĂŽæŰźæşŢãĂĆ

```
>>> Person.first_name.fget
<function Person.first_name at 0x1006a60e0>
>>> Person.first_name.fset
<function Person.first_name at 0x1006a6170>
>>> Person.first_name.fdel
<function Person.first_name at 0x1006a62e0>
>>>
```

éĂŽåÿÿæİěèőšïijŇä¡ääÿ■äijŽçŻť æŐěåŘŰèřČçŤÍfgetæĹŰèĂĚfsetïijŇåőČäžňäijŽåIJÍèő£éŮőpropertyçŽ

åŘłæIJL'å¡Şä¡äçąőåőđéIJĂèęĄåŕźattributeæL'ğèąŇåĚűäżŰéćİåď'ŰçŽĎæŞ■ä¡IJçŽĎæŮűåĂŹæL'■åžŤèřě
æIJL'æŮűåĂŹäÿĂäžŽäżŐåĚűäžŰçijŰçÍŃèŕ■èÍĂ(æŕŤåęĆJava)è£ĞæİěçŽĎçÍŃåžŘåŚŸæĂżèőď'äÿžæL'ĂæIJL
æL'Ääżěäż Űäžňèőď'äÿžäżčçăĄåžŤèřěåČŘäÿŃéİćè£ŹæăůåĘŹïijŽ

```
class Person:
    def __init__(self, first_name):
        self.first_name = first_name

    @property
    def first_name(self):
        return self._first_name

    @first_name.setter
    def first_name(self, value):
        self._first_name = value
```

äÿ■èęĄåĘŹè£Źçğ■æšąæIJL'åĄŽäżżä¡ŢåĚűäżŰéćİåď'ŰæŞ■ä¡IJçŽĎpropertyãĂĆ
éęŰåĚĹïijŇåőČäijŽèőľ'ä¡äçŽĎäżčçăĄåŘŸå¿Ůå¿ĹèĞČèĆ£ïijŇåźűäÿŤè£ŸäijŽè£ůæČŚéŸĚèŕżèĂĚãĂĆ
åĚűæňąïijŇåőČè£ŸäijŽèőľ'ä¡äçŽĎçÍŃåžŘè£ŘèąŇèţůæİěåŘŸæĚćå¿Ĺåď'ŽãĂĆ
æIJĂåŘŐïijŇè£ŹæăůçŽĎèő¿èőąåźűæšąæIJL'åÿęæİěäżżä¡ŢçŽĎåě¡åď'ĎãĂĆ
çL'źåĹńæŸŕå¡Şä¡ääżěåŘŐæČşçżŹæŹőéĂŽattributeèè£éŮőæůżåŁäéćİåď'ŰçŽĎåď'ĎçŘĘéĂżè¿ŚçŽĎæŮűåĂŹ
ä¡äåŘŕäżěåřĘåőČåŘŸæĹŘäÿĂäÿłpropertyèĂŇæŮäéIJĂæŤźåŘŸåŐ§æİěçŽĎäżčçăĄãĂĆ
åŽääÿżèő£éŮőattributeçŽĎäżčçăĄè£ŸæŸŕä£ÍæŇĄåŐ§æäůãĂĆ

Propertiesè£ŸæŸŕäÿĂçğ■åőŽäźL'åŁÍæĂĄèőąçőŮattributeçŽĎæŰźæşŢãĂĆ
è£Źçğ■çśżåđŇçŽĎattributesåźűäÿ■äijŽèćňåőđéŹĚçŽĎå■ŸåĆÍïijŇèĂŇæŸŕåIJÍéIJĂèęĄçŽĎæŮűåĂŹèőąçőŮ

```
import math
class Circle:
    def __init__(self, radius):
        self.radius = radius

    @property
    def area(self):
        return math.pi * self.radius ** 2

    @property
    def diameter(self):
        return self.radius * 2

    @property
    def perimeter(self):
        return 2 * math.pi * self.radius
```

åIJÍè£ŹéĞŇïijŇæĹŚäżňéĂŽè£Ğä¡£çŤÍpropertiesïijŇåřĘæĽĂæIJĽçŽĎèő£éŮőæŐěåŘčå¡ćåijŘçż§äÿĂèţ
åŕźå■Łå¿ĎãĂĄçŻť å¿ĎãĂĄåŚÍéŢ£åŠŇéİćçğŕçŽĎèő£éŮőéČ¡æŸŕéĂŽè£ĞåśđæĂğèő£éŮőïijŇåŕśèů§èő£éŮőç
åęĆæđIJäÿ■è£ŹæăůåĄŽçŽĎèŕİïijŇéĆčäźĹåŕśèęĄåIJÍäżčçăĄäÿ■æůůåŘĹä¡£çŤÍçőĂå■ŢåśđæĂğèő£éŮőåŠŇæ
äÿŇéİćæŸŕä¡£çŤÍçŽĎåőđä¿ŃïijŽ

```
>>> c = Circle(4.0)
>>> c.radius
4.0
>>> c.area  # Notice lack of ()
50.26548245743669
>>> c.perimeter  # Notice lack of ()
25.132741228718345
>>>
```

åř¡çőąpropertiesåŘŕäżěåőđçŐřäijŸéŻĚçŽĎçijŰçÍŃæŐěåŘčïijŇä¡ĘæIJĽäžŻæŮůåĂŹä¡ăè£ŸæŸŕäijŽæČ

```
>>> p = Person('Guido')
>>> p.get_first_name()
'Guido'
>>> p.set_first_name('Larry')
>>>
```

è£ŹçğæČĚåĘţçŽĎåĞžçŐŕéĂŽåÿÿæŸŕåŻääÿžPythonäżčçăĄèćńéŻĘæĹŘåĹřäÿĂäÿłåď ğåđŃå§žçąÅåźş
ä¿ŃåęĆïijŇæIJĽåŘŕèČ¡æŸŕäÿĂäÿłPythonçśżåĞĘåď ĞåŁäåĚěåĹřäÿĂäÿłå§žäžŐè£IJçÍŃè£ĞçÍŃèřČçŤÍçŽĎåć
è£ŹçğæČĚåĘţäÿŃïijŇçŻť æŐěä¡£çŤÍget/setæŰźæşŢ(æŹőéĂŽæŰźæşŢèřČçŤÍ)èĂŇäÿ■æŸŕpropertyæĹŰèőÿ

æIJĂåŘŐäÿĂçĆźïijŇäÿ■èęĄåČŘäÿŃéİćè£ŹæăůåĘŹæIJĽåď ğéĞŘéĞ■åď ■äżčçăĄçŽĎpropertyåőŽäźĽïi

```
class Person:
    def __init__(self, first_name, last_name):
        self.first_name = first_name
        self.last_name = last_name
```

(continues on next page)

(continued from previous page)

```
    @property
    def first_name(self):
        return self._first_name

    @first_name.setter
    def first_name(self, value):
        if not isinstance(value, str):
            raise TypeError('Expected a string')
        self._first_name = value

    # Repeated property code, but for a different name (bad!)
    @property
    def last_name(self):
        return self._last_name

    @last_name.setter
    def last_name(self, value):
        if not isinstance(value, str):
            raise TypeError('Expected a string')
        self._last_name = value
```

éĞ■åď■äžčçăĄäijŽåŕijèĞť'èĞČèĆ£ãĂĄæŸŞåĞžéŤŹåŠŇäÿŚéŹŃçŽĎçÍŃåžŔãĂĆåě¡æűĹæĄŕæŸŕïijŇéÀ
åŘŕäžěåŘĆèĂČ8.9åŠŇ9.21åřŘèŁĆçŽĎåĘĚåőźãĂĆ

10.7 8.7 èřČçŤÍçĹűçśżæŰźæşŢ

éŮőéćŸ

ä¡ăæČşåIJÍå■Řçśżäÿ■èřČçŤÍçĹűçśżçŽĎæ§ŘäÿłåůšçżŘèćńèęĘçŻŰçŽĎæŰźæşŢãĂĆ

èğčåĘşæŰźæąĹ

äÿžäžĘèřČçŤÍçĹűçśż(èűĚçśż)çŽĎäÿĂäÿłæŰźæşŢïijŇåŘŕäžěä¡£çŤÍ `super()` åĞ¡æŢřïijŇæŕŤåęĆïijŽ

```
class A:
    def spam(self):
        print('A.spam')

class B(A):
    def spam(self):
        print('B.spam')
        super().spam()  # Call parent spam()
```

`super()` åĞ¡æŢřçŽĎäÿĂäÿłåÿÿèğĄçŤÍæşŢæŸŕåIJÍ `__init__()` æŰźæşŢäÿ■çąőä£ÍçĹűçśżèćńæ■čçąőçŽĎåĹİåğŃåŇŰäžĘïijŽ

```
class A:
    def __init__(self):
        self.x = 0

class B(A):
    def __init__(self):
        super().__init__()
        self.y = 1
```

super() çŽĎåŕęåď'ŰäÿĂäÿłåÿÿèğĄçŤÍæşŢåĞžçŐřåIJÍèęEçŻŰPythonçL'źæőŁæŰźæşŢçŽĎäžčçăĄäÿ

```
class Proxy:
    def __init__(self, obj):
        self._obj = obj

    # Delegate attribute lookup to internal obj
    def __getattr__(self, name):
        return getattr(self._obj, name)

    # Delegate attribute assignment
    def __setattr__(self, name, value):
        if name.startswith('_'):
            super().__setattr__(name, value) # Call original __
↪setattr__
        else:
            setattr(self._obj, name, value)
```

åIJÍäÿŁéİćäžčçăĄäÿ■ïijŇ__setattr__() çŽĎåőđçŐřåŇĚåŘńäÿĂäÿłåŘ■å■ŮæčĂæ§ěãĂĆ åęĆæđIJæ§ŘäÿłåśđæĂğåŘ■äžěäÿŇåĹŠçž£(_)åijĂåď't'ïijŇåřśéĂŽè£Ğ super() èřČçŤÍåŐ§åğŇçŽĎ __setattr__() ïijŇ åŘęåĹŹçŽĎèŕİåřśåğŤæť¿çżŹåĘĚéČÍçŽĎäžčçŘĘåŕźèśą self._obj åŐżåď'ĎçŘĘãĂĆ è£ŹçIJŃäÿŁåŐżæIJL'çĆźæĎŘæĂİïijŇåŻääÿžåřśçőŮæšąæIJL'æŸ¿åijŘçŽĎæ super() äż■çĎűåŘŕäžěæIJL'æŢĹçŽĎåůěä¡IJãĂĆ

èőÍèőž

åőđéŹĚäÿŁïijŇåď'ğåőűåŕźäžŐåIJÍPythonäÿ■åęĆä¡Ţæ■čçąőä¡£çŤÍ super() åĞ¡æŢřæŹőéĄ■ç§ěäźŃçŤŽåřŚãĂĆ ä¡ăæIJL'æŮűåĂŹäijŽçIJŃåĹřåČŘäÿŃéİćè£Źæăůç Żť'æŐěěřČçŤÍçĹűçśżç

```
class Base:
    def __init__(self):
        print('Base.__init__')

class A(Base):
    def __init__(self):
        Base.__init__(self)
        print('A.__init__')
```

åř¡çőąåŕźäžŐåď'ğéČÍåĹĘäžčçăĄèĂŇèÍĂè£ŹäźĹåĄŽæšąäžĂäźĹéŮőéćŸïijŇä¡ĘæŸŕåIJÍæŻť'åď'■æİĆçŽĎ æŕŤåęĆïijŇèĂČèŹŚåęĆäÿŃçŽĎæČĚåĘţïijŽ

```
class Base:
    def __init__(self):
        print('Base.__init__')

class A(Base):
    def __init__(self):
        Base.__init__(self)
        print('A.__init__')

class B(Base):
    def __init__(self):
        Base.__init__(self)
        print('B.__init__')

class C(A,B):
    def __init__(self):
        A.__init__(self)
        B.__init__(self)
        print('C.__init__')
```

åęĆæđIJä¡ăè£ŘèąŇè£ŹæőţäżčçăĄåřśäijŽåŘŚçŐř `Base.__init__()` èćńèřČçŤÍäÿď æňąïijŇåęĆäÿŃæĽĂçď žïijŽ

```
>>> c = C()
Base.__init__
A.__init__
Base.__init__
B.__init__
C.__init__
>>>
```

åŘŕèČ¡äÿď æňąèřČçŤÍ `Base.__init__()` æšąäżĂäźĹåİŘåď ĎïijŇä¡ĘæIJĽæŮűåĂŹå■ť äÿ■æŸŕãĂĆ åŘęäÿĂæŰźéİćïijŇåĄĞèő¿ä¡ăåIJÍäżčçăĄäÿ■æ■ćæĹŘä¡£çŤÍ `super()` ïijŇçżŞæđIJåřśå¿ĹåőŇç¿ŐäžĘïijŽ

```
class Base:
    def __init__(self):
        print('Base.__init__')

class A(Base):
    def __init__(self):
        super().__init__()
        print('A.__init__')

class B(Base):
    def __init__(self):
        super().__init__()
        print('B.__init__')

class C(A,B):
```

(continues on next page)

(continued from previous page)

```
    def __init__(self):
        super().__init__()  # Only one call to super() here
        print('C.__init__')
```

è£ŘèąŇè£ŹäÿłæŰřçĽŹłæIJňåŘŐïijŇä¡ääijŽåŘŚçŐřæŕŘäÿł `__init__()` æŰźæşŢåŘłäijŽèćńèřČçŤÍäÿĂæňąäžĘïijŽ

```
>>> c = C()
Base.__init__
B.__init__
A.__init__
C.__init__
>>>
```

äÿžäžĘåijĎæÿĚåőČçŽĎåŐ§çŘĘïijŇæŹŚäžňéIJĂèęĄèŁśçĆźæŮűéŮť'èğčéĞŁäÿŃPythonæŸŕåęĆä¡Ţåőđ
åŕźäžŐä¡äåőŽäźĽçŽĎæŕŘäÿĂäÿłçśżïijŇPythonäijŽèőąçőŮåĞžäÿĂäÿłæĽĂèřŞçŽĎæŰźæşŢèğčæđŘéąžåžŘ(
è£ŹäÿłMROåĹŮèąĺåřśæŸŕäÿĂäÿłçőĂå■ŢçŽĎæĽĂæIJĽå§žçśżçŽĎçž£æĂğéąžåžŘèąĺãĂĆä¿ŃåęĆïijŽ

```
>>> C.__mro__
(<class '__main__.C'>, <class '__main__.A'>, <class '__main__.B'>,
<class '__main__.Base'>, <class 'object'>)
>>>
```

äÿžäžĘåőđçŐřçżğæĽ'£ïijŇPythonäijŽåIJÍMROåĹŮèąĺäÿŁäžŐåůęåĹřåŘşåijĂåğŃæ§ěæĽ'¿å§žçśżïijŇçŻť'

èĂŇè£ŹäÿłMROåĹŮèąĺçŽĎæđĎéĂäæŸŕéĂŽè£ĞäÿĂäÿłC3çž£æĂğåŇŰçőŮæşŢæİěåőđçŐřçŽĎãĂĆ
æĹŚäžňäÿ■åŐżæůśçľ'űè£ŹäÿłçőŮæşŢçŽĎæŢřå■ęåŐ§çŘĘïijŇåőČåőđéŹĚäÿŁåřśæŸŕåŘĹåźűæĽ'ĂæIJĽçĹűçş

- å■ŘçśżäijŽåĚĹäžŐçĹűçśżèćńæčĂæ§ě
- åď'ŽäÿłçĹűçśżäijŽæăźæ■őåőČäžňåIJĺåĹŮèąĺäÿ■çŽĎéąžåžŘèćńæčĂæ§ě
- åęĆæđIJåŕźäÿŃäÿĂäÿłçśżå■ŸåIJĺäÿď'äÿłåŘĹæşŢçŽĎéĂĽ'æŃľ'ïijŇéĂĽ'æŃľ'çňňäÿĂäÿłçĹűçśż

èĂĄåőđèŕť'ïijŇä¡äæĽ'ĂèęĄç§ěéĄŞçŽĎåřśæŸŕMROåĹŮèąĺäÿ■çŽĎçśżéąžåžŘäijŽèőľ'ä¡äåőŽäźĽçŽĎäżz

å¡Şä¡ää¡£çŤÍ `super()` åĞ¡æŢřæŮűïijŇPythonäijŽåIJÍMROåĹŮèąĺäÿŁçżğçż■æŘIJçť'ćäÿŃäÿĂäÿłçśżãĂ
åŘłèęĄæŕŘäÿłéĞ■åőŽäźĽçŽĎæŰźæşŢçż§äÿĂä¡£çŤÍ `super()`
åźűåŘłèřČçŤÍåőČäÿĂæňąïijŇ éĆčäźĹæŐğåĹűæţĄæIJĂçżĹäijŽéĄ■åŐĘåőŃæŢť'äÿłM-
ROåĹŮèąĺïijŇæŕŘäÿłæŰźæşŢäź§åŘłäijŽèćńèřČçŤÍäÿĂæňąãĂĆ
è£Źäź§æŸŕäÿžäžĂäźĹåIJÍçňňäžŃäÿłä¿Ńå■Řäÿ■ä¡ääÿ■äijŽèřČçŤÍäÿď'æňą `Base.__init__()` çŽĎåŐ§åŽäăĂĆ

`super()` æIJĽäÿłäżď'äžžåŘČæČŁçŽĎåIJřæŰźæŸŕåőČåźűäÿ■äÿĂäőŽåŐżæ§ěæĽ'¿æ§Řäÿłçśżåĺ
ä¡äçŤŽèĞşåŘŕäżěåIJÍäÿĂäÿłæšąæIJĽçŻť'æŐěçĹűçśżçŽĎçśżäÿ■ä¡£çŤÍåőČäĂĆä¿ŃåęĆïijŇèĂČèŹŚåęĆäÿŃè

```
class A:
    def spam(self):
        print('A.spam')
        super().spam()
```

åęĆæđIJä¡äèŕŢçĺĂçŻť'æŐěä¡£çŤÍè£ŹäÿłçśżåřśäijŽåĞžéŤŹïijŽ

```
>>> a = A()
>>> a.spam()
A.spam
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "<stdin>", line 4, in spam
AttributeError: 'super' object has no attribute 'spam'
>>>
```

ä¡ĘæŸŕïijŇåęĆæđIJä¡ăä¡£çŤĺåď'ŽçżğæĽ'£çŽĎèŕİçIJŃçIJŃäijŽåŔŚçŤ§äżĂäźĹïijŽ

```
>>> class B:
...     def spam(self):
...         print('B.spam')
...
>>> class C(A,B):
...     pass
...
>>> c = C()
>>> c.spam()
A.spam
B.spam
>>>
```

ä¡ăåŔŕäżěçIJŃåĹřåIJĺçśżAäÿ■ä¡£çŤĺ `super().spam()` åőđéŹĚäÿŁèřČçŤĺçŽĎæŸŕèů§çśżAæŕńæŮăåĚşçşżçŽĎçśżBäÿ■çŽĎ `spam()` æŰźæşŢãĂĆ è£ŹäÿłçŤĺçśżCçŽĎMROåĹŮèąĺåřśåŔŕäżěåőŇåĚĺèğčéĞŁæÿĚæěŽäżĘïijŽ

```
>>> C.__mro__
(<class '__main__.C'>, <class '__main__.A'>, <class '__main__.B'>,
<class 'object'>)
>>>
```

åIJĺåőŽäźĽ'æůůåĚěçśżçŽĎæŮűůåĂŹè£Źæăůä¡£çŤĺ `super()` æŸŕå¿ĹæŹőéĄ■çŽĎãĂĆåŔŕäżěåŔĆèĂČ8.13åŠŇ8.18åřŔèŁĆãĂĆ

çĎűèĂŇïijŇçŤśäżŐ `super()` åŔŕèČ¡äijŽèřČçŤĺäÿ■æŸŕä¡ăæČşèęĄçŽĎæŰźæşŢïijŇä¡ăåžŤèŕěéĄţå¿łäÿ éęŰåĚĹïijŇçąőä£ÌåIJĺçżğæĽ'£ä¡Ş çşżäÿ■æĽ'ĂæIJĽ'çŻÿåŘŇåŘ■å■ŮçŽĎæŰźæşŢæŃěæIJĽ'åŔŕåĚijåőźçŽĎåŔ è£ŹæăůåŔŕäżěçąőä£Ì `super()` èřČçŤĺäÿĂäÿłéÌđçŻť'æŐěçĹűçśżæŰźæşŢæŮűäÿ■äijŽåĞžéŤŹãĂĆ åĚűæňąïijŇæIJĂåě¡çąőä£ÌæIJĂéąűåśĆçŽĎçśżæŔŘä¿ŻäžĘè£ŹäÿłæŰźæşŢçŽĎåőđçŐřïijŇè£ŹæăůçŽĎèŕÌåIJĺ

åIJĺPythonçď¿åŇžäÿ■åŕźäżŐ `super()` çŽĎä¡£çŤĺæIJĽ'æŮűåĂŹäijŽåijŢæÌěäÿĂäžŻäžĽ'èőőãĂĆ åř¡çőąåęĆæ■ď'ïijŇåęĆæđIJäÿĂåĹĞéąžåĹľçŽĎèŕÌïijŇä¡ăåžŤèŕěåIJĺä¡ăæIJĂæŰřäżčçăĄäÿ■ä¡£çŤĺåőĆãĂĆ Raymond Hettingeräÿžæ■ď'åĘŹäžĘäÿĂçŕĞéÌđåÿÿåě¡çŽĎæŰĞçńä âĂIJPythonâĂŹs super() Considered Super!âĂÌ ïijŇ éĂŽè£Ğåď'ğéĞŔçŽĎä¿Ńå■ŘåŘŚæĹŚäżňèğčéĞŁäžĘäÿžäżĂäźĹ `super()` æŸŕæđĄåě¡çŽĎãĂĆ

10.8 8.8 å■Řçśżäÿ■æĽľåśŢproperty

éŮőécŸ

åIJÍå■Řçśżäÿ■ïijŇä¡ăæČşèęĄæĽľåśŢåőŽäźĽåIJÍçĹűçśżäÿ■çŽĎpropertyçŽĎåŁ§èČ¡ãĂĆ

èğčåĘşæŰźæąĹ

èĂČèŹŚåęĆäÿŃçŽĎäżčçăĄïijŇåőČåőŽäźĽäžĘäÿĂäÿłpropertyïijŽ

```
class Person:
    def __init__(self, name):
        self.name = name

    # Getter function
    @property
    def name(self):
        return self._name

    # Setter function
    @name.setter
    def name(self, value):
        if not isinstance(value, str):
            raise TypeError('Expected a string')
        self._name = value

    # Deleter function
    @name.deleter
    def name(self):
        raise AttributeError("Can't delete attribute")
```

äÿŃéİćæŸŕäÿĂäÿłçď žä¿ŃçśżïijŇåőČçżğæĽ£èĞłPersonåźűæĽľåśŢäžĘ name åśđæĂğçŽĎåŁ§èČ¡ïijŽ

```
class SubPerson(Person):
    @property
    def name(self):
        print('Getting name')
        return super().name

    @name.setter
    def name(self, value):
        print('Setting name to', value)
        super(SubPerson, SubPerson).name.__set__(self, value)

    @name.deleter
    def name(self):
        print('Deleting name')
        super(SubPerson, SubPerson).name.__delete__(self)
```

æŐěäÿŃæİěä¡£çŤÍè£ŹäÿłæŰřçśżïijŽ

```
>>> s = SubPerson('Guido')
Setting name to Guido
>>> s.name
Getting name
'Guido'
>>> s.name = 'Larry'
Setting name to Larry
>>> s.name = 42
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "example.py", line 16, in name
        raise TypeError('Expected a string')
TypeError: Expected a string
>>>
```

åęĆæđIJä¡ääżĚäżĚåŘłæČşæL'l'åśŢpropertyçŽĎæ§ŘäÿĂäÿłæŰźæşŢïijŃéĆčäźĹåŘŕäżěåČŘäÿŃéİćè£Źæ

```
class SubPerson(Person):
    @Person.name.getter
    def name(self):
        print('Getting name')
        return super().name
```

æĹŰèĂĚïijŃä¡äåŘłæČşä£őæŤźsetteræŰźæşŢïijŃåŕśè£ŹäźĹåĘŹïijŽ

```
class SubPerson(Person):
    @Person.name.setter
    def name(self, value):
        print('Setting name to', value)
        super(SubPerson, SubPerson).name.__set__(self, value)
```

èőÍèőž

åIJÍå■Řçśżäÿ■æL'l'åśŢäÿĂäÿłpropertyåŘŕèČ¡äijŽåijŢèţůå¿Ĺåď'Žäÿ■æŸŞåŕ§èğL'çŽĎéŮőécŸïijŃ åŻääÿžäÿĂäÿłpropertyåĚűåőđæŸŕ getterãĂĄsetter åŠŃ deleter æŰźæşŢçŽĎéŻĘåŘĹïijŃèĂŃäÿ■æŸŕå■ŢäÿłæŰźæşŢãĂĆ åŻäæ■d'ïijŃå¡Şä¡äæL'l'åśŢäÿĂäÿłpropertyçŽĎæŮűåĂŹïijŃä¡äéIJĂèęĄåĚĹçąőåőŽä¡äæŸŕåŘęèęĄéĞ■æŰřåő

åIJÍçňňäÿĂäÿłä¿Ńå■Řäÿ■ïijŃæL'ĂæIJL'çŽĎpropertyæŰźæşŢéČ¡èćńéĞ■æŰřåőŽäźL'ãĂĆ åIJÍæŕŘäÿĂäÿłæŰźæşŢäÿ■ïijŃä¡£çŤÍäžĘ super() æİěèřČçŤÍçĹűçśżçŽĎåőđçŐřãĂĆ åIJÍ setter åĞ¡æŢřäÿ■ä¡£çŤÍ super(SubPerson, SubPerson).name.__set__(self, value) çŽĎèŕ■åŘěæŸŕæšąæIJL'éŤŹçŽĎãĂĆ äÿžäžĘåğŤæL'ŸçżŹäźŃåL'■åőŽäźL'çŽĎsetteræŰźæşŢïijŃéIJĂèęĄåřĘæŐğåĹűæİČäijäéĂŠçżŹäźŃåL'■åőŽäź __set__() æŰźæşŢãĂĆ äÿ■è£ĞïijŃèŐůåŘŰè£ŹäÿłæŰźæşŢçŽĎåŤŕäÿĂéĂŤå¿ĎæŸŕä¡£çŤÍçśżåŘŸéĞŘèĂ è£Źäź§æŸŕäÿžäżĂäźĹæĹŚäżňèęĄä¡£çŤÍ super(SubPerson, SubPerson) çŽĎåŐ§åŻäãĂĆ

åęĆæđIJä¡ăåŘłæČşéĞ■åőŽäźĽåĚűäÿ■äÿĂäÿłæŰźæşŢïijŇéĆčåŘłä¡£çŤÍ @property æIJňèžńæŸŕäÿ■åď§çŽĎãĂĆæŕŤåęĆïijŇäÿŃéİćçŽĎäžčçăĄåřśæŮăæşŢåůëä¡IJïijŽ

```
class SubPerson(Person):
    @property  # Doesn't work
    def name(self):
        print('Getting name')
        return super().name
```

åęĆæđIJä¡ăèŕŢçİĂè£ŘèąŇäijŽåŘŚçŐřsetteråĞ¡æŢřæŢť äÿłæűĹåď śäžĘïijŽ

```
>>> s = SubPerson('Guido')
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "example.py", line 5, in __init__
        self.name = name
AttributeError: can't set attribute
>>>
```

ä¡ăäžŤèŕěåČŘäźŃåĽ■èŕť è£ĞçŽĎéĆčæăůä£őæŤźäžčçăĄïijŽ

```
class SubPerson(Person):
    @Person.name.getter
    def name(self):
        print('Getting name')
        return super().name
```

è£ŹäźĹåĘŹåŘŐïijŇpropertyäźŃåĽ■åůšçżŘåőŽäźĽè£ĞçŽĎæŰźæşŢäijŽèćnåď■åĹűè£ĞæİěïijŇèĂŇget

```
>>> s = SubPerson('Guido')
>>> s.name
Getting name
'Guido'
>>> s.name = 'Larry'
>>> s.name
Getting name
'Larry'
>>> s.name = 42
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "example.py", line 16, in name
        raise TypeError('Expected a string')
TypeError: Expected a string
>>>
```

åIJÍè£ŹäÿłçľźåĹńçŽĎèğčåEşæŰźæąĹäÿ■ïijŇæĹŚäžňæšąåŁđæşŢä¡£çŤÍæŻť åŁäéĂŽçŤÍçŽĎæŰźåijŘåĆ `Person` çśżåŘ■ãĂĆ åęĆæđIJä¡ääÿ■ç§ěéĄŞåĹřåžŢæŸŕåŞłäÿłå§žçśżåőŽäźĽäžEpropertyïijŇéĆčä¡ăåŘłèČ¡éĂŽè£ĞéĞ■æŰřåőŽäźĽæĽĂæIJĽpropertyåźűä¡£çŤÍ `super()` æİěåřĘæŐğåĹűæİČäijäéĂŠçżŹåĽ■éİćçŽĎåőđçŐřãĂĆ

åĂijå¿ŮæşÍæĎŘçŽĎæŸŕäÿŁéİćæijŤçď žçŽĎçňňäÿĂçğ■æŁĂæIJŕè£ŸåŘŕäžěèćńçŤÍæİěæĽľåśŢäÿĂäÿłæ

```
# A descriptor
class String:
    def __init__(self, name):
        self.name = name

    def __get__(self, instance, cls):
        if instance is None:
            return self
        return instance.__dict__[self.name]

    def __set__(self, instance, value):
        if not isinstance(value, str):
            raise TypeError('Expected a string')
        instance.__dict__[self.name] = value

# A class with a descriptor
class Person:
    name = String('name')

    def __init__(self, name):
        self.name = name

# Extending a descriptor with a property
class SubPerson(Person):
    @property
    def name(self):
        print('Getting name')
        return super().name

    @name.setter
    def name(self, value):
        print('Setting name to', value)
        super(SubPerson, SubPerson).name.__set__(self, value)

    @name.deleter
    def name(self):
        print('Deleting name')
        super(SubPerson, SubPerson).name.__delete__(self)
```

æIJĂåŘŐåĂijå¿ŮæşÍæĎŘçŽĎæŸŕïijŇèŕżåĹřè£ŹéĞŇæŮűïijŇä¡ăåžŤèŕĕäijŽåŘŚçŐřå■ŘçśżåŇŰ setter åŠŇ deleter æŰźæşŢåĚűåőđæŸŕå¿ĹçőĂå■ŢçŽĎãĂĆ è£ŹéĞŇæijŤçď žçŽĎèğčåĘşæŰźæąĹåŘŇæăůéĂĆçŤĺïijŇä¡ĘæŸŕåIJÍ PythonçŽĎissueéąţéĺć æŁĕåŚŁçŽĎäÿĂäÿłbugïijŇæĹŰèőÿäijŽä¡£å¿ŮåřĘæĺĕçŽĎPythonçL'ĹæIJňäÿ■åĞžçŐřäÿĂäÿłæŻť'åŁăçőĂæť

10.9 8.9 åĹŻåżžæŰřçŽĎçśżæĹŰåőđä¿ŃåśđæĂğ

éŮőéćŸ

ä¡ăæČşåĹŻåżžäÿĂäÿłæŰřçŽĎæŃĕæIJL'äÿĂäžŻéćĺåď'ŰåŁ§èČ¡çŽĎåőđä¿ŃåśđæĂğçśżåđŃïijŇæŕŤåęĆç

èğčåĘşæŰźæąĹ

åęĆæđIJä¡ăæČşåĹŻåżżäÿĂäÿłåĚĺæŰřçŽĎåőđä¿ŃåśđæĂğïijŇåŔŕäżěéĂŽè£ĞäÿĂäÿłæŔŔè£řåŹĺçśżçŽĎ

```
# Descriptor attribute for an integer type-checked attribute
class Integer:
    def __init__(self, name):
        self.name = name

    def __get__(self, instance, cls):
        if instance is None:
            return self
        else:
            return instance.__dict__[self.name]

    def __set__(self, instance, value):
        if not isinstance(value, int):
            raise TypeError('Expected an int')
        instance.__dict__[self.name] = value

    def __delete__(self, instance):
        del instance.__dict__[self.name]
```

äÿĂäÿłæŔŔè£řåŹĺåřśæŸŕäÿĂäÿłåőđçŐřäžĘäÿL'äÿłæäÿå£ČçŽĎåśđæĂğèő£éŮőæŞ■ä¡IJ(get, set, delete)çŽĎçśżïijŇ åĹĘåĹńäÿž __get__() ãĂĄ__set__() åŠŇ __delete__() è£ŹäÿL'äÿłçL'źæőŁçŽĎæŰźæşŢãĂĆ è£ŹäžŻæŰźæşŢæŐěåŘŮäÿĂäÿłåőđä¿Ńä¡IJäÿžè¿ŞåĚěïijŇäźŃåŘŐçŻÿäžŤçŽĎæŞ■ä¡IJåőđä¿ŃåžŢåśĆçŽĎå■

äÿžäžĘä¡£çŤĺäÿĂäÿłæŔŔè£řåŹĺïijŇéIJĂåřĘè£ŹäÿłæŔŔè£řåŹĺçŽĎåőđä¿Ńä¡IJäÿžçśżåśđæĂğæŤ¿åĹřäÿĂ

```
class Point:
    x = Integer('x')
    y = Integer('y')

    def __init__(self, x, y):
        self.x = x
        self.y = y
```

å¡Şä¡ăè£ŹæăůåĄŽåŘŐïijŇæL'ĂæIJL'åŕźæŔŔè£řåŹĺåśđæĂğ(æŕŤåęĆxæĹŰy)çŽĎèő£éŮőäijŽèćń __get__() ãĂĄ__set__() åŠŇ __delete__() æŰźæşŢæ■ŢèŐůåĹřãĂĆä¿ŃåęĆïijŽ

```
>>> p = Point(2, 3)
>>> p.x # Calls Point.x.__get__(p,Point)
2
>>> p.y = 5 # Calls Point.y.__set__(p, 5)
>>> p.x = 2.3 # Calls Point.x.__set__(p, 2.3)
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "descrip.py", line 12, in __set__
        raise TypeError('Expected an int')
```

(continues on next page)

(continued from previous page)

```
TypeError: Expected an int
>>>
```

ä¡IJäÿžè¿ŞåĚěïijŇæŘŘè£řåŹÍçŽĎæŕŘäÿĂäÿłæŰźæşŢäijŽæŐěåŘŮäÿĂäÿłæŞ■ä¡IJåőđä¿ŃãĂĆ äÿžäžĘåőđçŐřèŕůæśĆæŞ■ä¡IJïijŇäijŽçŻÿåžŤçŽĎæŞ■ä¡IJåőđä¿ŃåžŢåśĆçŽĎå■ŮåĚÿ(__dict__åśđæĂğ)ãĂĆ æŘŘè£řåŹÍçŽĎ `self.name` åśđæĂğå■ŸåĆÍäžĘåIJÍåőđä¿Ňå■ŮåĚÿäÿ■èćňåőđéŹĚä¡£çŤÍåŁřçŽĎkeyãĂĆ

èőÍèőž

æŘŘè£řåŹÍåŘŕåőđçŐřåď'ğéČÍåŁĘPythonçśżçL'źæĂğäÿ■çŽĎåžŢåśĆé■ŤæşŢïijŇ åŇĚæŃň `@classmethod` ãĂĄ`@staticmethod` ãĂĄ`@property` ïijŇçŤŽèĞşæŸŕ `__slots__` çL'źæĂğãĂĆ

éĂŽè£ĞåőŽäźL'äÿĂäÿłæŘŘè£řåŹÍïijŇä¡ăåŘŕäżěåIJÍåžŢåśĆæ■ŢèŐůæăÿå£ČçŽĎåőđä¿ŃæŞ■ä¡IJ(get, set, delete)ïijŇåźűäÿŤåŘŕåőŇåĚÍèĞłåőŽäźL'åőČäżňçŽĎèąŇäÿžãĂĆ è£ŹæŸŕäÿĂäÿłåijžåď'ğçŽĎåůěåĚůïijŇæIJL'äžĘåőČä¡ăåŘŕäżěåőđçŐřå¿Ĺåď'ŽéńŸçžğåŁ§èČ¡ïijŇåźűäÿŤåőČä

æŘŘè£řåŹÍçŽĎäÿĂäÿłæŕŤè¿ČåŻřæČŚçŽĎåIJřæŰźæŸŕåőČåŘłèČ¡åIJÍçśżçžğåĹńèćňåőŽäźL'ïijŇèĂŇäÿ

```
# Does NOT work
class Point:
    def __init__(self, x, y):
        self.x = Integer('x') # No! Must be a class variable
        self.y = Integer('y')
        self.x = x
        self.y = y
```

åŘŇæŮűïijŇ`__get__()` æŰźæşŢåőđçŐřèţůæİěæŕŤçIJŃäÿŁåŐżèęĄåď'■æĺĆå¿Ůåď'ŽïijŽ

```
# Descriptor attribute for an integer type-checked attribute
class Integer:

    def __get__(self, instance, cls):
        if instance is None:
            return self
        else:
            return instance.__dict__[self.name]
```

`__get__()` çIJŃäÿŁåŐżæIJL'çĆźåď'■æĺĆçŽĎåŐ§åŽäå¡ŠçżŞäžŐåőđä¿ŃåŘŸéĞŘåŠŇçśżåŘŸéĞŘçŽĎ åęĆæđIJäÿĂäÿłæŘŘè£řåŹÍèćňå¡ŞåĄŽäÿĂäÿłçśżåŘŸéĞŘæĺěèő£éŮőïijŇéĆčäźĹ `instance` åŘĆæŢřèćňèő¿ç¡őæĹŘ `None` ãĂĆ è£ŹçğæČĚåĘţäÿŃïijŇæăĞåĞĘåĄŽæşŢåŕśæŸŕçőĂå■ŢçŽĎè£ŤåŽđè£Źä

```
>>> p = Point(2,3)
>>> p.x # Calls Point.x.__get__(p, Point)
2
>>> Point.x # Calls Point.x.__get__(None, Point)
<__main__.Integer object at 0x100671890>
>>>
```

æŘŘè£řåŹĺéĂŽåÿÿæŸŕéĆčäžŻä¡£çŤĺåĹřèčĚéěřåŹĺæĹŰåĚČçśżçŽĎåď ğåđŃæąĘæđűäÿ■çŽĎäÿĂäÿłçżĎ
äÿ¿äÿłä¿Ńå■ŘïijŇäÿŃéĺćæŸŕäÿĂäžŻæŻť éńŸçžğçŽĎå§žäžŐæŘŘè£řåŹĺçŽĎäżčçăĄïijŇåźűæűĽ'åŘŁåĹřäÿĂ

```
# Descriptor for a type-checked attribute
class Typed:
    def __init__(self, name, expected_type):
        self.name = name
        self.expected_type = expected_type
    def __get__(self, instance, cls):
        if instance is None:
            return self
        else:
            return instance.__dict__[self.name]

    def __set__(self, instance, value):
        if not isinstance(value, self.expected_type):
            raise TypeError('Expected ' + str(self.expected_type))
        instance.__dict__[self.name] = value
    def __delete__(self, instance):
        del instance.__dict__[self.name]

# Class decorator that applies it to selected attributes
def typeassert(**kwargs):
    def decorate(cls):
        for name, expected_type in kwargs.items():
            # Attach a Typed descriptor to the class
            setattr(cls, name, Typed(name, expected_type))
        return cls
    return decorate

# Example use
@typeassert(name=str, shares=int, price=float)
class Stock:
    def __init__(self, name, shares, price):
        self.name = name
        self.shares = shares
        self.price = price
```

æIJĂåŘŐèęĄæŃĞåĞžçŽĎäÿĂçĆźæŸŕïijŇåęĆæđIJä¡ăåŘłæŸŕæČşçőĂå■ŢçŽĎèĞłåőŽäźĽæ§Řäÿłçśżçž
è£Źçğ■æČĚåĘţäÿŃä¡£çŤĺ8.6åřŘèŁĆäżŃçż■çŽĎpropertyæŁĂæIJŕäijŽæŻť åŁäåőźæŸŞãĂĆ
å¡ŞçĺŃåžŘäÿ■æIJĽå¿Ĺåď ŽéĞ■åď ■äżčçăĄçŽĎæŮűåĂŹæŘŘè£řåŹĺåřśå¿ĹæIJĽçŤĺäžĘ
(æŕŤåęĆä¡äæČşåIJĺä¡ääżčçăĄçŽĎå¿Ĺåď ŽåIJřæŰźä¡£çŤĺæŘŘè£řåŹĺæŘŘä¿ŻçŽĎåŁ§èČ¡æĹŰèĂĚåřĘåőČä¡I

10.10 8.10 ä¡£çŤĺäżűè£§èőąçőŮåśđæĂğ

éŮőéćŸ

ä¡äæČşåřĘäÿĂäÿłåŘłèŕżåśđæĂğåőŽäźĽæĹŘäÿĂäÿłpropertyïijŇåźűäÿŤåŘłåIJĺèő£éŮőçŽĎæŮűåĂŹæĽ'
ä¡ĘæŸŕäÿĂæŮęèćńèő£éŮőåŘŐïijŇä¡äåÿŃæIJŻçżŞæđIJåĂijèćńçijŞå■ŸèţůæİěïijŇäÿ■çŤĺæŕŘæňąéČ¡åŐżèőą

èğčåĘşæŰźæąĹ

åőŽäźĽäÿĂäÿłåżűè£§åśđæĂğçŽĎäÿĂçğ■éńŸæŢĹæŰźæşŢæŸŕéĂŽè£Ğä¡£çŤĺäÿĂäÿłæŘŔè£řåŹĺçśżïijŃ

```
class lazyproperty:
    def __init__(self, func):
        self.func = func

    def __get__(self, instance, cls):
        if instance is None:
            return self
        else:
            value = self.func(instance)
            setattr(instance, self.func.__name__, value)
            return value
```

ä¡ăéIJĂèęĄåČŔäÿŃéİćè£ŹæăůåIJĺäÿĂäÿłçśżäÿ■ä¡£çŤĺåőČïijŽ

```
import math

class Circle:
    def __init__(self, radius):
        self.radius = radius

    @lazyproperty
    def area(self):
        print('Computing area')
        return math.pi * self.radius ** 2

    @lazyproperty
    def perimeter(self):
        print('Computing perimeter')
        return 2 * math.pi * self.radius
```

äÿŃéİćåIJĺäÿĂäÿłäžď'äžŠçŐŕåćČäÿ■æijŤçď'žåőČçŽĎä¡£çŤĺïijŽ

```
>>> c = Circle(4.0)
>>> c.radius
4.0
>>> c.area
Computing area
50.26548245743669
>>> c.area
50.26548245743669
>>> c.perimeter
Computing perimeter
25.132741228718345
>>> c.perimeter
25.132741228718345
>>>
```

äżŤçżĘèğĆåŕ§ä¡ääijŽåŘŚçŐřæűĹæĄŕ `Computing area` åŠŇ `Computing perimeter` äżĚäżĚåĞžçŐřäÿĂæňąãĂĆ

èőĺèőž

å¿Ĺåď ŽæŮűåĂŹïijŇæđĎéĂääÿĂäÿłåżűè£§èőąçőŮåśđæĂğçŽĎäÿżèęĄçŽőçŽĎæŸŕäÿžäžĘæŘŘå■ĞæĂ
ä¿ŃåęĆïijŇä¡ääŘŕäżěéĄ£åĚ■èőąçőŮè£ŹäžŽåśđæĂğåĂijïijŇéŹď éİđä¡äçIJ§çŽĎéIJĂèęĄåőČäżňãĂĆ
è£ŹéĞŇæijŤçď žçŽĎæŮźæąĹåŕśæŸŕçŤĺæİěåőđçŐřè£ŹæäůçŽĎæŢĹæđIJçŽĎïijŇ
åŘłäÿ■è£ĞåőČæŸŕéĂŽè£ĞäżěéİđåÿÿéńŸæŢĹçŽĎæŮźåijŘä¡£çŤĺæŘŘè£řåŹĺçŽĎäÿĂäÿłçš¿åęŹçĽźæĂğæİě

æ■čåęĆåIJĺåĚűäżŮåřŘèŁĆ(åęĆ8.9åřŘèŁĆ)æĽĂèőšçŽĎéĆčæäůïijŇå¡ŞäÿĂäÿłæŘŘè£řåŹĺèćńæŤ¿åĚěäÿ
æŕŘæňąèő£éŮőåśđæĂğæŮűåőČçŽĎ `__get__()` ãĂĄ `__set__()` åŠŇ `__delete__()` æŮźæşŢåŕśäijŽèćńèğęåŘŚãĂĆ äÿ■è£ĞïijŇåęĆæđIJäÿĂäÿłæŘŘè£řåŹĺäżĚäżĚåŘłäőŽäźĽäžĘäÿĂäÿł `__get__()` æŮźæşŢçŽĎèŕİïijŇäőČæŕŤéĂŽåÿÿçŽĎåĚůæIJĽæŻť åijśçŽĎçżŚäőŽãĂĆ çĽźäĹńåIJřïijŇåŘłæIJĽå¡Şèćńèő£éŮőåśđæĂğäÿ■åIJĺäőđä¿ŃäžŢåśĆçŽĎå■ŮåĚÿäÿ■æŮű `__get__()` æŮźæşŢæĽ■äijŽèćńèğęåŘŚãĂĆ

`lazyproperty` çśżåĹľçŤĺè£ŹäÿĂçĆźïijŇä¡£çŤĺ `__get__()` æŮźæşŢåIJĺäőđä¿Ńäÿ■å■ŸåĆĺèőąçőŮåĞżæİěçŽĎåĂijïijŇ è£Źäÿłäőđä¿Ńä¡£çŤĺçŻÿåŘŇçŽĎåŘ■å■Ůä¡IJäÿžä
è£ŹæäůäÿĂæİěïijŇçżŞæđIJåĂijèćńå■ŸåĆĺåIJĺäőđä¿Ńå■ŮåĚÿäÿ■äźűäÿŤäżěåŘŐäŕśäÿ■éIJĂèęĄåĘ■åŐżèőąçõ
ä¡ääŘŕäżěåřĺèŕŢæŻť æůśåĚěçŽĎä¿Ńå■ŘæİěèğĆåŕ§çżŞæđIJïijŽ

```
>>> c = Circle(4.0)
>>> # Get instance variables
>>> vars(c)
{'radius': 4.0}

>>> # Compute area and observe variables afterward
>>> c.area
Computing area
50.26548245743669
>>> vars(c)
{'area': 50.26548245743669, 'radius': 4.0}

>>> # Notice access doesn't invoke property anymore
>>> c.area
50.26548245743669

>>> # Delete the variable and see property trigger again
>>> del c.area
>>> vars(c)
{'radius': 4.0}
>>> c.area
Computing area
50.26548245743669
>>>
```

è£Źçğ■æŮźæąĹæIJĽäÿĂäÿłåřŘçijżéŹůåŕśæŸŕèőąçőŮåĞżçŽĎåĂijèćńåĹŻåżżåŘŐæŸŕåŘŕäżěèćńä£őæŤ

```
>>> c.area
Computing area
50.26548245743669
>>> c.area = 25
>>> c.area
25
>>>
```

åęĆæđIJä¡ăæŃĚå£Čè£ŹäÿłéŮőéćŸïijŇéĆčäźĹåŔŕäżěä¡£çŤĺäÿĂçğ■çĺ■å¿őæšąéĆčäźĹéńŸæŢĹçŽĎåőđç

```
def lazyproperty(func):
    name = '_lazy_' + func.__name__
    @property
    def lazy(self):
        if hasattr(self, name):
            return getattr(self, name)
        else:
            value = func(self)
            setattr(self, name, value)
            return value
    return lazy
```

åęĆæđIJä¡ăä¡£çŤĺè£ŹäÿłçL'ĹæIJňïijŇåŕśäijŽåŔŚçŐřçŐřåIJĺä£őæŤźæŞ■ä¡IJåůšçżŔäÿ■èćńåĚĄèőÿäžĘïij

```
>>> c = Circle(4.0)
>>> c.area
Computing area
50.26548245743669
>>> c.area
50.26548245743669
>>> c.area = 25
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
AttributeError: can't set attribute
>>>
```

çĎűèĂŃïijŇè£Źçğ■æŰźæąĹæIJL'äÿĂäÿłçijžçĆźåŕśæŸŕæL'ĂæIJL'getæŞ■ä¡IJéČ¡å£ĚéążèćńåőŽåŘŚåĹřä
getter åĞ¡æŢřäÿŁåŐżãĂĆ è£Źäÿłèů§äźŃåL'■çőĂå■ŢçŽĎåIJĺåőđä¿Ńå■ŮåĚÿäÿ■æ§ěæL'¿åĂijçŽĎæŰźæą
åęĆæđIJæČşèŐůåŘŰæŻt'åď'ŽåĚşäžŐpropertyåŠŇåŘŕçőąçŘĘåśđæĂğçŽĎä£ąæĄŕïijŇåŘŕäżěåŘĆèĂČ8.6åřŘ

10.11 8.11 çőĂåŇŰæŢřæ■őçżŞæđĎçŽĎåĹłåğŃåŇŰ

éŮőéćŸ

ä¡ăåĘŹäžĘå¿Ĺåď'ŽäżĚäżĚçŤĺä¡IJæŢřæ■őçżŞæđĎçŽĎçśżïijŇäÿ■æČşåĘŹåď'łåď'ŽçČęäžžçŽĎ
__init__() åĞ¡æŢř

èğčåĘşæŰźæąĹ

åŔŕäżěåIJĺäÿĂäÿłå§žçśżäÿ■åĘŹäÿĂäÿłåĚňçŤĺçŽĎ `__init__()` åĞ¡æŢřïijŽ

```
import math

class Structure1:
    # Class variable that specifies expected fields
    _fields = []

    def __init__(self, *args):
        if len(args) != len(self._fields):
            raise TypeError('Expected {} arguments'.format(len(self.
↪_fields)))
        # Set the arguments
        for name, value in zip(self._fields, args):
            setattr(self, name, value)
```

çĎűåŘŐä¡£ä¡ăçŽĎçśżçżğæL'£èĞłè£ŹäÿłåŞžçśż:

```
# Example class definitions
class Stock(Structure1):
    _fields = ['name', 'shares', 'price']

class Point(Structure1):
    _fields = ['x', 'y']

class Circle(Structure1):
    _fields = ['radius']

    def area(self):
        return math.pi * self.radius ** 2
```

ä¡£çŤĺè£ŹäžŻçśżçŽĎçď žä¿ŃïijŽ

```
>>> s = Stock('ACME', 50, 91.1)
>>> p = Point(2, 3)
>>> c = Circle(4.5)
>>> s2 = Stock('ACME', 50)
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "structure.py", line 6, in __init__
        raise TypeError('Expected {} arguments'.format(len(self._
↪fields)))
TypeError: Expected 3 arguments
```

åęĆæđIJè£ŸæČşæŤŕæŇĄåĚşéŤőå■ŮåŔĆæŢřïijŇåŔŕäżěåřĘåĚşéŤőå■ŮåŔĆæŢřèő¿ç¡őäÿžåőđä¿ŃåśđæĂ

```
class Structure2:
    _fields = []
```

(continues on next page)

(continued from previous page)

```
    def __init__(self, *args, **kwargs):
        if len(args) > len(self._fields):
            raise TypeError('Expected {} arguments'.format(len(self.
↪_fields)))

        # Set all of the positional arguments
        for name, value in zip(self._fields, args):
            setattr(self, name, value)

        # Set the remaining keyword arguments
        for name in self._fields[len(args):]:
            setattr(self, name, kwargs.pop(name))

        # Check for any remaining unknown arguments
        if kwargs:
            raise TypeError('Invalid argument(s): {}'.format(','.
↪join(kwargs)))
# Example use
if __name__ == '__main__':
    class Stock(Structure2):
        _fields = ['name', 'shares', 'price']

    s1 = Stock('ACME', 50, 91.1)
    s2 = Stock('ACME', 50, price=91.1)
    s3 = Stock('ACME', shares=50, price=91.1)
    # s3 = Stock('ACME', shares=50, price=91.1, aa=1)
```

ä¡ăè£ŸèČ¡ářĘäÿ■åIJÍ `_fields` äÿ■çŽĎåŘ■çğřåŁăåĚěåĹřåśđæĂğäÿ■åŐżïijŽ

```
class Structure3:
    # Class variable that specifies expected fields
    _fields = []

    def __init__(self, *args, **kwargs):
        if len(args) != len(self._fields):
            raise TypeError('Expected {} arguments'.format(len(self.
↪_fields)))

        # Set the arguments
        for name, value in zip(self._fields, args):
            setattr(self, name, value)

        # Set the additional arguments (if any)
        extra_args = kwargs.keys() - self._fields
        for name in extra_args:
            setattr(self, name, kwargs.pop(name))

        if kwargs:
```

(continues on next page)

(continued from previous page)

```
            raise TypeError('Duplicate values for {}'.format(','.
↪join(kwargs)))

# Example use
if __name__ == '__main__':
    class Stock(Structure3):
        _fields = ['name', 'shares', 'price']

    s1 = Stock('ACME', 50, 91.1)
    s2 = Stock('ACME', 50, 91.1, date='8/2/2012')
```

èőÍèőž

å¡Şä¡äéIJĂèęĄä¡£çŤÍåď ğéĞŘå¿ĹåřŘçŽĎæŢřæ■őçżŞæđĎçśżçŽĎæŮűåĂŹïijŇ
çŻÿæŕŤæĽŃåůěäÿĂäÿłäÿłäőŽäźĽ `__init__()` æŰźæşŢèĂŇåůšïijŇä¡£çŤÍè£ŹçğŒæŰźåijŘåŘŕäżěåď ğåď ğ

åIJÍäÿŁéİćçŽĎåőđçŐřäÿ■æĹŚäżňä¡£çŤÍäžĘ `setattr()`
åĞ¡æŢřçśżèő¿ç¡őåśđæĂğåĂijïijŇ ä¡ääŘŕèČ¡äÿ■æČşçŤÍè£ŹçğŒæŰźåijŘïijŇèĂŇæŸŕæČşçŻť æŐěæŻť æŰřåőć

```
class Structure:
    # Class variable that specifies expected fields
    _fields= []
    def __init__(self, *args):
        if len(args) != len(self._fields):
            raise TypeError('Expected {} arguments'.format(len(self.
↪_fields)))

        # Set the arguments (alternate)
        self.__dict__.update(zip(self._fields,args))
```

ař¡çőąè£ŹäźŞåŘŕäżěæ■čåÿÿäůěä¡IJïijŇä¡ĘæŸŕå¡ŞäőŽäźĽ'å■ŘçśżçŽĎæŮűåĂŹéŮőéćŸäřśæİěäžĘãĂĆ
å¡ŞäÿĂäÿłä■ŘçśżäőŽäźĽ'äžĘ `__slots__` æĹŰèĂĚéĂŽè£Ğproperty(æĹŰæŘŘè£řäŹÍ)æİěäŇĚèčĚæ§Řäÿłäs
éĆčäźĹçŻť æŐěèő£éŮőäőđä¿Ńå■ŮäĚÿäřśäÿ■èţůä¡IJçŤÍäžĘãĂĆæĹŚäżňäÿŁéİćä¡£çŤÍ
`setattr()` äijŽæŸ¿å¿ŮæŻť éĂŽçŤÍäžŻïijŇäŽääÿžäőČäź§éĂĆçŤÍäžŐä■ŘçśżæČĚäĘţãĂĆ

è£ŹçğŒæŰźæşŢäŤŕäÿĂäÿ■äě¡çŽĎäIJŕæŰźäřśæŸŕäŕźæ§ŘäžŻIDEèĂŇèÍĂïijŇäIJÍæŸ¿çď žäÿőäŁľäĞ¡æŢ

```
>>> help(Stock)
Help on class Stock in module __main__:
class Stock(Structure)
...
| Methods inherited from Structure:
|
| __init__(self, *args, **kwargs)
|
...
>>>
```

åŘŕäżěåŘĆèĂČ9.16åřŘèŁĆæİěåijžåĹűåIJÍ `__init__()` æŰźæşŢäÿ■æŇĞåőŽåŘĆæŢřçŽĎçśżåđŃç■¿åŘ■ãĂĆ

10.12 8.12 åőŽäźĽæŐěåŘčæĹŰèĂĚæŁ¡èśąå§žçśż

éŮőécŸ

ä¡äæČşåőŽäźĽäÿĂäÿłæŐěåŘčæĹŰæŁ¡èśąçśżïijŇåźűäÿŤéĂŽè£ĞæĽğèąŇçśżåđŃæčĂæ§ěæİěçąőä£İå■Ř

èğčåĘşæŰźæąĹ

ä¡£çŤÍ `abc` æÍąåİŮåŘŕäżěå¿Ĺè¡żæİ¿çŽĎåőŽäźĽæŁ¡èśąå§žçśżïijŽ

```
from abc import ABCMeta, abstractmethod

class IStream(metaclass=ABCMeta):
    @abstractmethod
    def read(self, maxbytes=-1):
        pass

    @abstractmethod
    def write(self, data):
        pass
```

æŁ¡èśąçśżçŽĎäÿĂäÿłçĽźçĆźæŸŕåőČäÿ■èČ¡çŻťæŐěèćńåőđä¿ŃåŇŰïijŇæŕŤåęĆä¡äæČşåČŘäÿŃéİćè£Ź

```
a = IStream() # TypeError: Can't instantiate abstract class
              # IStream with abstract methods read, write
```

æŁ¡èśąçśżçŽĎçŻőçŽĎåřśæŸŕèőľåĹńçŽĎçśżçżğæĽ£åőČåźűåőđçŐřçĽźåőŽçŽĎæŁ¡èśąæŰźæşŢïijŽ

```
class SocketStream(IStream):
    def read(self, maxbytes=-1):
        pass

    def write(self, data):
        pass
```

æŁ¡èśąå§žçśżçŽĎäÿĂäÿłäÿżèęĄçŤÍéĂŤæŸŕåIJÍäżčçăĄäÿ■æčĂæ§ěæ§ŘäžŻçśżæŸŕåŘęäÿżçĽźåőŽçśżåđ

```
def serialize(obj, stream):
    if not isinstance(stream, IStream):
        raise TypeError('Expected an IStream')
    pass
```

éŹďäžĘçżğæĽ£è£Źçğ■æŰźåijŔåďŰïijŇè£ŸåŘŕäżěéĂŽè£ĞæşĺåĘŇæŰźåijŔæİěèőľæ§ŘäÿłçśżåőđçŐřæ

```
import io

# Register the built-in I/O classes as supporting our interface
IStream.register(io.IOBase)

# Open a normal file and type check
f = open('foo.txt')
isinstance(f, IStream) # Returns True
```

@abstractmethod è£ŸèČ¡æşÍèğčéİŹæĂĄæŰźæşŢãĂĄçśżæŰźæşŢåŠŇ properties ãĂĆ ä¡ššŘłéIJĂä£İèŕĄè£ŹäÿłæşÍèğčçť ğéİššIJÍåĞ¡æŢřőŽäźĽ'åĽ'■å■şåŘŕïijŽ

```
class A(metaclass=ABCMeta):
    @property
    @abstractmethod
    def name(self):
        pass

    @name.setter
    @abstractmethod
    def name(self, value):
        pass

    @classmethod
    @abstractmethod
    def method1(cls):
        pass

    @staticmethod
    @abstractmethod
    def method2():
        pass
```

èőÍèőž

æăĞåĞĘåžŞäÿ■æIJĽ'å¿Ĺåď'ŽçŤÍåĹřæŁ¡èśąå§žçśżçŽĎåIJřæŰźãĂĆcollections æÍąåİŮåőŽäźĽ'äžĘå¿Ĺåď'Žèů§åőźåŹÍåŠŇè£■äżčåŹÍ(åžŘåĹŮãĂĄæŸššřĎãĂĄéŻĘåŘĹç■Ľ')æIJĽ'åĚşçŽĎæŁ numbers åžŞåőŽäźĽ'äžĘèů§æŢřå■Ůåŕźèšą(æŢť'æŢřãĂĄæţőçĆźæŢřãĂĄæIJĽ'çŘĘæŢřç■Ľ')æIJĽ'åĚşçŽĎå§ åžŞåőŽäźĽ'äžĘå¿Ĺåď'Žèů§I/OæŞ■ä¡IJçŻÿåĚşçŽĎå§žçśżãĂĆ

ä¡ššŘŕäżěä¡£çŤÍéćĎåőŽäźĽ'çŽĎæŁ¡èśąçśżæİěæĽ'ğèąŇæŻť'éĂŽçŤÍçŽĎçśżåđŃæčĂæ§ěïijŇä¿ŃåęĆïijŽ

```
import collections

# Check if x is a sequence
if isinstance(x, collections.Sequence):
...

# Check if x is iterable
```

(continues on next page)

(continued from previous page)

```
if isinstance(x, collections.Iterable):
...

# Check if x has a size
if isinstance(x, collections.Sized):
...

# Check if x is a mapping
if isinstance(x, collections.Mapping):
```

ář¡çőąABCsåŘŕäżěèőľæĹŚäžňå¿ĹæŰźä¿£çŽĎåĄŽçśżåđŃæčĂæ§ěïijŇä¡ĘæŸŕæĹŚäžňåIJĺäżčçăĄäÿ■æĹ
åŻăäÿžPythonçŽĎæIJňèť'ĺæŸŕäÿĂéŮĺåŁĺæĂĄçijŰçĺŃèŕ■èĺĂïijŇåĚűçŻőçŽĎåřśæŸŕçżŹä¡äæŻť'åď'ŽçĄţæť'żä
åijžåĹűçśżåđŃæčĂæ§ěæĹŰèőľä¡ääżčçăĄåŘŸå¿ŮæŻť'åď'■æĺĆïijŇè£ŹæăůåĄŽæŮăåijĆäžŐèĹ■æIJňæśĆæIJ

10.13 8.13 åőđçŐřæŢřæ■őæĺąåđŃçŽĎçśżåđŃçžęæĺ§

éŮőécŸ

ä¡äæČşåőŽäźL'æ§ŘäžŻåIJĺåśđæĂğèţŃåĂijäÿŁéĺćæIJL'éŹŘåĹűçŽĎæŢřæ■őçżŞæđĎãĂĆ

èğčåĘşæŰźæąĹ

åIJĺè£ŹäÿłéŮőécŸäÿ■ïijŇä¡äéIJĂèęĄåIJĺåŕźæ§ŘäžŻåőđä¿ŃåśđæĂğèţŃåĂijæŮűè£ŻèąŇæčĂæ§ěãĂĆ
æL'Ääżěä¡äèęĄèĞłåőŽäźL'åśđæĂğèţŃåĂijåĞ¡æŢřïijŇè£ŹçğæČĚåĘţäÿŃæIJĂåě¡ä¡£çŤĺæŘŕè£řåŹĺãĂĆ

äÿŃéĺćçŽĎäžčçăĄä¡£çŤĺæŘŕè£řåŹĺåőđçŐřäžĘäÿĂäÿłçśżçż§çśżåđŃåŠŇèţŃåĂijéłŇèŕĄæąĘæđűïijŽ

```
# Base class. Uses a descriptor to set a value
class Descriptor:
    def __init__(self, name=None, **opts):
        self.name = name
        for key, value in opts.items():
            setattr(self, key, value)

    def __set__(self, instance, value):
        instance.__dict__[self.name] = value


# Descriptor for enforcing types
class Typed(Descriptor):
    expected_type = type(None)

    def __set__(self, instance, value):
        if not isinstance(value, self.expected_type):
            raise TypeError('expected ' + str(self.expected_type))
        super().__set__(instance, value)
```

(continues on next page)

(continued from previous page)

```
# Descriptor for enforcing values
class Unsigned(Descriptor):
    def __set__(self, instance, value):
        if value < 0:
            raise ValueError('Expected >= 0')
        super().__set__(instance, value)


class MaxSized(Descriptor):
    def __init__(self, name=None, **opts):
        if 'size' not in opts:
            raise TypeError('missing size option')
        super().__init__(name, **opts)

    def __set__(self, instance, value):
        if len(value) >= self.size:
            raise ValueError('size must be < ' + str(self.size))
        super().__set__(instance, value)
```

è£ŹäžŻçśżåřśæŸŕäˡäèęĄåĹŻåżžçŽĎæŢřæ■őæÍąåđŃæĹŰçśżåđŃçşżçż§çŽĎå§žçąĂæđĎåżžæÍąåİŮãĂĆäÿŃéİćåřśæŸŕæĹŚäžňåőđéŹĚåőŽäźĽçŽĎåŘĎçğ■äÿ■åŘŇçŽĎæŢřæ■őçśżåđŃïijŽ

```
class Integer(Typed):
    expected_type = int

class UnsignedInteger(Integer, Unsigned):
    pass

class Float(Typed):
    expected_type = float

class UnsignedFloat(Float, Unsigned):
    pass

class String(Typed):
    expected_type = str

class SizedString(String, MaxSized):
    pass
```

çĎűåŘŐäˡ£çŤÍè£ŹäžŻèĞłåőŽäźĽæŢřæ■őçśżåđŃïijŇæĹŚäžňåőŽäźĽäÿĂäÿłçśżïijŽ

```
class Stock:
    # Specify constraints
    name = SizedString('name', size=8)
    shares = UnsignedInteger('shares')
    price = UnsignedFloat('price')
```

(continues on next page)

(continued from previous page)

```
    def __init__(self, name, shares, price):
        self.name = name
        self.shares = shares
        self.price = price
```

çĎűåŘŐæţŃèŕŢè£ŹäÿłçśżçŽĎåśđæĂğèţŃåĂijçžęæİ§ïijŇåŔŕåŔŚçŐřåŕźæ§ŘäžŻåśđæĂğçŽĎèţŃåĂijè£İ

```
>>> s.name
'ACME'
>>> s.shares = 75
>>> s.shares = -10
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "example.py", line 17, in __set__
        super().__set__(instance, value)
    File "example.py", line 23, in __set__
        raise ValueError('Expected >= 0')
ValueError: Expected >= 0
>>> s.price = 'a lot'
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "example.py", line 16, in __set__
        raise TypeError('expected ' + str(self.expected_type))
TypeError: expected <class 'float'>
>>> s.name = 'ABRACADABRA'
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "example.py", line 17, in __set__
        super().__set__(instance, value)
    File "example.py", line 35, in __set__
        raise ValueError('size must be < ' + str(self.size))
ValueError: size must be < 8
>>>
```

è£ŸæIJL'äÿĂäžŻæŁĂæIJŕåŔŕäžěçőĂåŇŰäÿŁéİćçŽĎäżčçăĄïijŇåĚűäÿ■äÿĂçğ■æŸŕä¡£çŤÍçśżèčĚéěřåŹÍ

```
# Class decorator to apply constraints
def check_attributes(**kwargs):
    def decorate(cls):
        for key, value in kwargs.items():
            if isinstance(value, Descriptor):
                value.name = key
                setattr(cls, key, value)
            else:
                setattr(cls, key, value(key))
        return cls

    return decorate
```

(continues on next page)

(continued from previous page)

```
# Example
@check_attributes(name=SizedString(size=8),
                  shares=UnsignedInteger,
                  price=UnsignedFloat)
class Stock:
    def __init__(self, name, shares, price):
        self.name = name
        self.shares = shares
        self.price = price
```

åŘęåď'ŰäÿĂçğ■æŰźåijŘæŸŕä¡£çŤĺåĚČçśżïijŽ

```
# A metaclass that applies checking
class checkedmeta(type):
    def __new__(cls, clsname, bases, methods):
        # Attach attribute names to the descriptors
        for key, value in methods.items():
            if isinstance(value, Descriptor):
                value.name = key
        return type.__new__(cls, clsname, bases, methods)

# Example
class Stock2(metaclass=checkedmeta):
    name = SizedString(size=8)
    shares = UnsignedInteger()
    price = UnsignedFloat()

    def __init__(self, name, shares, price):
        self.name = name
        self.shares = shares
        self.price = price
```

èőĺèőž

æIJňèŁĆä¡£çŤĺäžĘå¿Ĺåď'ŽéńŸçžğæŁĂæIJŕïijŇåŇĚæŃňæŘŘè£řåŹĺãĂĄæůůåĚěçśżãĂĄ`super()` çŽĎä¡£çŤĺãĂĄçśżèčĚéěřåŹĺåŠŇåĚČçśżãĂĆ äÿ■åŘŕèČ¡åIJĺè£ŹéĞŇäÿĂäÿĂèŕęçżĘåśŢåijĂæĺěèőšïijŇä¡ĘæŸä¡ĘæŸŕïijŇæĹŚåIJĺè£ŹéĞŇè£ŸæŸŕèęĄæŘŘäÿĂäÿŃåĞääÿłéIJĂèęĄæşĺæĎŘçŽĎçĆźãĂĆ

éęŰåĚĹïijŇåIJĺ `Descriptor` å§žçśżäÿ■ä¡ääijŽçIJŃåĹřæIJL'äÿł `__set__()` æŰźæşŢïijŇå■ť'æšąæIJL'çŻÿåžŤçŽĎ `__get__()` æŰźæşŢãĂĆ åęĆæđIJäÿĂäÿłæŘŘè£řäżĚäżĚæŸŕäżŐåžŢåśĆåőđä¿Ňå■ŮåĚÿäÿ■èŐůåŘŰæ§ŘäÿłåśđæĂğåĂijçŽĎèŕĺïijŇéĆ `__get__()` æŰźæşŢãĂĆ

æL'ĂæIJL'æŘŘè£řåŹĺçśżéČ¡æŸŕå§žäžŐæůůåĚěçśżæĺěåőđçŐřçŽĎãĂĆæŕŤåęĆ `Unsigned` åŠŇ `MaxSized` èęĄèů§åĚűäżŰçżğæL'£èĞł `Typed` çśżæůůåĚěãĂĆ è£ŹéĞŇåĹl'çŤĺåď'Žçżğæ L'£æĺěåőđçŐřçŻÿåžŤçŽĎåŁ§èČ¡ãĂĆ

æůůåĚěçśżçŽĎäÿĂäÿłæŕŤè¿ČéŽ¿çŘęèğčçŽĎåIJřæŰźæŸŕïijŇèřČçŤĺ

super() åĞ¡æŢřæŮűïijŇä¡ăăźűäÿ■ç§ěéĄŞçľűçń§èęĄèřČçŤÍåŞłäÿłåĚůä¡ŞçśźãĂĆ
ä¡ăéIJĂèęĄèů§åĚűäżŰçśźçżŞåŘĹåŘŐæĽ■èČ¡æ■čçąőçŽĎä¡£çŤÍïijŇäź§åřśæŸŕå£ĚéążåŘĹä¡IJæĽ■èČ¡äžğçŤ

ä¡£çŤÍçśżèčĚéěřåŹÍåŠŇåĚČçśżéĂŽåÿÿåŔŕäżěçőĂåŇŰäżčçăĄãĂĆäÿŁéİćäÿď äÿłä¿Ňå■Řäÿ■ä¡ääijŽåŘŚ

```
# Normal
class Point:
    x = Integer('x')
    y = Integer('y')

# Metaclass
class Point(metaclass=checkedmeta):
    x = Integer()
    y = Integer()
```

æĽĂæIJĽæŰźæşŢäÿ■ïijŇçśżèčĚéěřåŹÍæŰźæąĹåžŤèřěæŸŕæIJĂçĄţæť żåŠŇæIJĂéńŸæŸŐçŽĎãĂĆ
éęŰåĚĹïijŇåőČåźűäÿ■ä¿İèţŰäżżä¡ŢåĚűäżŰæŰřçŽĎæŁĂæIJŕïijŇæŕŤåęĆåĚČçśżãĂĆåĚűæňąïijŇèčĚéěřåŹÍå

æIJĂåŘŐïijŇèčĚéěřåŹÍè£ŸèČ¡ä¡IJäÿžæůůåĚěçśżçŽĎæŻ£äżčæŁĂæIJŕæİěåőđçŐřåŘŇæăůçŽĎæŢĹæđIJ

```
# Decorator for applying type checking
def Typed(expected_type, cls=None):
    if cls is None:
        return lambda cls: Typed(expected_type, cls)
    super_set = cls.__set__

    def __set__(self, instance, value):
        if not isinstance(value, expected_type):
            raise TypeError('expected ' + str(expected_type))
        super_set(self, instance, value)

    cls.__set__ = __set__
    return cls


# Decorator for unsigned values
def Unsigned(cls):
    super_set = cls.__set__

    def __set__(self, instance, value):
        if value < 0:
            raise ValueError('Expected >= 0')
        super_set(self, instance, value)

    cls.__set__ = __set__
    return cls


# Decorator for allowing sized values
def MaxSized(cls):
    super_init = cls.__init__
```

(continues on next page)

(continued from previous page)

```
    def __init__(self, name=None, **opts):
        if 'size' not in opts:
            raise TypeError('missing size option')
        super_init(self, name, **opts)

    cls.__init__ = __init__

    super_set = cls.__set__

    def __set__(self, instance, value):
        if len(value) >= self.size:
            raise ValueError('size must be < ' + str(self.size))
        super_set(self, instance, value)

    cls.__set__ = __set__
    return cls


# Specialized descriptors
@Typed(int)
class Integer(Descriptor):
    pass


@Unsigned
class UnsignedInteger(Integer):
    pass


@Typed(float)
class Float(Descriptor):
    pass


@Unsigned
class UnsignedFloat(Float):
    pass


@Typed(str)
class String(Descriptor):
    pass


@MaxSized
class SizedString(String):
    pass
```

è£Źçğ■æŰźåijŔåőŽäźĽçŽĎçśżèů§äźŃåĽ■çŽĎæŢĹæđIJäÿĂæăůïijŇèĂŇäÿŤæĽğèąŇéĂ§åžęäijŽæŻť’å

èő¿ç¡őäÿĂäÿłçőĂå■ŢçŽĎçśżåđŃåśđæĂğçŽĎåĂijïijŇèčĚéěřåŹĺæŰźåijŘèęĄæŕŤäźŃåĽ■çŽĎæůůåĚěçśżçŽĎ
çŐřåIJĺä¡ăåžŤèŕěåžĘåźÿèĞłåůśèŕżåőŇäžĘæIJňèŁĆåĚĺéĆĺåĘĚåőźäžĘåŘğïij§^_^

10.14 8.14 åőđçŐřèĞłåőŽäźĽåőźåŹĺ

éŮőéćŸ

ä¡äæČşåőđçŐřäÿĂäÿłèĞłåőŽäźĽçŽĎçśżæİěæĺąæŃ§åĘĚç¡őçŽĎåőźåŹĺçśżåŁ§èČ¡ïijŇæŕŤåęĆåĹŮèąĺåŠŃ

èğčåĘşæŰźæąĹ

`collections` åőŽäźĽäžĘå¿Ĺåď'ŽæŁ¡èśąå§žçśżïijŇå¡Şä¡äæČşèĞłåőŽäźĽåőźåŹĺçśżçŽĎæŮűåĂŹåőČ
æŕŤåęĆä¡äæČşèőľä¡äçŽĎçśżæŤŕæŇĄè£■äžčïijŇéĆčåřśèőľä¡äçŽĎçśżçżğæĽ'£
`collections.Iterable` å■şåŘŕïijŽ

```
import collections
class A(collections.Iterable):
    pass
```

äÿ■è£Ğä¡äéIJĂèęĄåőđçŐř `collections.Iterable` æĽĂæIJĽçŽĎæŁ¡èśąæŰźæşŢïijŇåŘęåĹŹäijŽæŁěéŤŹ:

```
>>> a = A()
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
TypeError: Can't instantiate abstract class A with abstract methods␣
↪__iter__
>>>
```

ä¡ååŘłèęĄåőđçŐř `__iter__()` æŰźæşŢåřśäÿ■äijŽæŁěéŤŹäžĘ(åŘĆèĂČ4.2åŠŇ4.7åřŘèŁĆ)ãĂĆ

ä¡ååŘŕäžěåĚĹèŕŢçĺĂåŐżåőđä¿ŃåŇŰäÿĂäÿłåŕźèśąïijŇåIJĺéŤŹèŕŕæŘŘçď'žäÿ■åŘŕäžěæĽ'¿åĹřéIJĂèęĄåő

```
>>> import collections
>>> collections.Sequence()
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
TypeError: Can't instantiate abstract class Sequence with abstract␣
↪methods \
__getitem__, __len__
>>>
```

äÿŃéĺćæŸŕäÿĂäÿłçőĂå■ŢçŽĎçď'žä¿ŃïijŇçżğæĽ'£èĞłäÿŁéĺćSequenceæŁ¡èśąçśżïijŇåźűäÿŤåőđçŐřåĚČ

```
class SortedItems(collections.Sequence):
    def __init__(self, initial=None):
        self._items = sorted(initial) if initial is not None else []
```

(continues on next page)

(continued from previous page)

```
    # Required sequence methods
    def __getitem__(self, index):
        return self._items[index]

    def __len__(self):
        return len(self._items)

    # Method for adding an item in the right location
    def add(self, item):
        bisect.insort(self._items, item)


items = SortedItems([5, 1, 3])
print(list(items))
print(items[0], items[-1])
items.add(2)
print(list(items))
```

åŕŕäžěçIJŃåĹřïijŇSortedItemsèů§æŹőéĂŽçŽĎåžŕåĹŮæšąäżĂäźĹäÿď æăůïijŇæŤŕæŇĄæĽĂæIJĽåÿÿç

è£ŹéĞŇéİćä¡£çŤĺåĹřäžĘ `bisect` æĺąåİŮïijŇåőČæŸŕäÿĂäÿłåIJĺæŐŠåžŕåĹŮèąĺäÿ■æŕŠåĚěåĚČçť ăçŽ

èőĺèőž

ä¡£çŤĺ `collections` äÿ■çŽĎæŁ¡èśąå§žçśżåŕŕäžěçąőä£İä¡äèĞłåőŽäźĽçŽĎåőźåŹĺåőđçŐřäžĘæĽĂæI
ä¡äçŽĎèĞłåőŽäźĽåőźåŹĺäijŽæżąèűşåď ğéČĺåĹĘçśżåđŃæčĂæ§ěéIJĂèęĄïijŇåęĆäÿŃæĽĂçď žïijŽ

```
>>> items = SortedItems()
>>> import collections
>>> isinstance(items, collections.Iterable)
True
>>> isinstance(items, collections.Sequence)
True
>>> isinstance(items, collections.Container)
True
>>> isinstance(items, collections.Sized)
True
>>> isinstance(items, collections.Mapping)
False
>>>
```

`collections` äÿ■å¿Ĺåď ŽæŁ¡èśąçśżäijŽäÿžäÿĂäžŻåÿÿèğĄåőźåŹĺæŞ■ä¡IJæŕŘä¿Żéż Ÿèőď çŽĎåőđçĆ
è£ŹæăůäÿĂæİěä¡ăåŕłéIJĂèęĄåőđçŐřéĆčäžŻä¡ăæIJĂæĎ§åĚť èűčçŽĎæŰźæşŢå■şåŕŕãĂĆåĄĞèő¿ä¡äçŽĎçśż
`collections.MutableSequence` ïijŇåęĆäÿŃïijŽ

```
class Items(collections.MutableSequence):
    def __init__(self, initial=None):
        self._items = list(initial) if initial is not None else []
```

(continues on next page)

(continued from previous page)

```
    # Required sequence methods
    def __getitem__(self, index):
        print('Getting:', index)
        return self._items[index]

    def __setitem__(self, index, value):
        print('Setting:', index, value)
        self._items[index] = value

    def __delitem__(self, index):
        print('Deleting:', index)
        del self._items[index]

    def insert(self, index, value):
        print('Inserting:', index, value)
        self._items.insert(index, value)

    def __len__(self):
        print('Len')
        return len(self._items)
```

åęĆæđIJä¡ăåĹŻåżž `Items` çŽĎåőđä¿ŃïijŇä¡ăäijŽåŔŚçŐřåőČæŤŕæŇĄåĞäźŐæĽ'ĂæIJĽçŽĎæăÿå£Čå
äÿŃéĺćæŸŕä¡£çŤĺæijŤçď'žïijŽ

```
>>> a = Items([1, 2, 3])
>>> len(a)
Len
3
>>> a.append(4)
Len
Inserting: 3 4
>>> a.append(2)
Len
Inserting: 4 2
>>> a.count(2)
Getting: 0
Getting: 1
Getting: 2
Getting: 3
Getting: 4
Getting: 5
2
>>> a.remove(3)
Getting: 0
Getting: 1
Getting: 2
Deleting: 2
>>>
```

æIJňåřŘèŁĆåŘłæŸŕåŕźPythonæŁ¡èśąçśżåŁ§èČ¡çŽĎæŁŻçăŰåijŢçŐL'ãĂĆnumbers æĺąåİŮæŘŘä¿ŻäžĘäÿĂäÿłçśżäijijçŽĎèů§æŢť'æŢřçśżåđŃçŻÿåĚşçŽĎæŁ¡èśąçśżåđŃéŻĘåŘĹãĂĆ åŘŕäżěåŘĆèĂČ8.12åřŘèŁĆæİěæđĎéĂäæŻť'åď'ŽèĞłåőŽäźĽ'æŁ¡èśąå§žçśżãĂĆ

10.15 8.15 åśđæĂğçŽĎäżčçŘĘèő£éŮő

éŮőéćŸ

ä¡äæČşåřĘæ§Řäÿłåőđä¿ŃçŽĎåśđæĂğèő£éŮőäżčçŘĘåĹřåĘĚéČĺåŘęäÿĂäÿłåőđä¿Ńäÿ■åŐżïijŃçŻőçŽĎå

èğčåĘşæŰźæąĹ

çőĂå■Ţæİěèŕť'ïijŃäżčçŘĘæŸŕäÿĂçğ■çijŰçĺŃæĺąåijŘïijŃåőČåřĘæ§ŘäÿłæŞ■ä¡IJè¡ňçğżçżŹåŘęåď'ŰäÿĂ æIJĂçőĂå■ŢçŽĎå¡ćåijŘåŘŕèČ¡æŸŕåČŘäÿŃéİćè£ŹæăůïijŽ

```
class A:
    def spam(self, x):
        pass

    def foo(self):
        pass


class B1:
    """çőĂå■ŢçŽĎäżčçŘĘ"""

    def __init__(self):
        self._a = A()

    def spam(self, x):
        # Delegate to the internal self._a instance
        return self._a.spam(x)

    def foo(self):
        # Delegate to the internal self._a instance
        return self._a.foo()

    def bar(self):
        pass
```

åęĆæđIJäżĚäżĚåřśäÿď'äÿłæŰźæşŢéIJĂèęĄäżčçŘĘïijŃéĆčäźĹåČŘè£ŹæăůåĘŹåřśèűşåď'§äžĘăĂĆä¡ĘæŸ éĆčäźĹä¡£çŤĺ `__getattr__()` æŰźæşŢæĹŰèőÿæĹŰæŻť'åě¡äžŻïijŽ

```
class B2:
    """ä¡£çŤĺ__getattr__çŽĎäżčçŘĘïijŃäżčçŘĘæŰźæşŢæŕŤè¿Čåď'ŽæŮűåĂŹ"""

    def __init__(self):
```

(continues on next page)

(continued from previous page)

```
        self._a = A()

    def bar(self):
        pass

    # Expose all of the methods defined on class A
    def __getattr__(self, name):
        ""
↪"è£ŹäÿłæŰźæşŢåIJĺèő£éŮőçŽĎattributeäÿ■å■ŸåIJĺçŽĎæŮűåĂŹèćńèřČçŤĺ
        the __getattr__() method is actually a fallback method
        that only gets called when an attribute is not found"""
        return getattr(self._a, name)
```

__getattr__ æŰźæşŢæŸŕåIJĺèő£éŮőattributeäÿ■å■ŸåIJĺçŽĎæŮűåĂŹèćńèřČçŤĺïijŇä¡£çŤĺæijŢçď ži

```
b = B()
b.bar() # Calls B.bar() (exists on B)
b.spam(42) # Calls B.__getattr__('spam') and delegates to A.spam
```

åŘęåď ŰäÿĂäÿłäżčçŘĘä¿Ňå■ŘæŸŕåőđçŐřäżčçŘĘæĺąåijŘïijŇä¿ŇåęĆïijŽ

```
# A proxy class that wraps around another object, but
# exposes its public attributes
class Proxy:
    def __init__(self, obj):
        self._obj = obj

    # Delegate attribute lookup to internal obj
    def __getattr__(self, name):
        print('getattr:', name)
        return getattr(self._obj, name)

    # Delegate attribute assignment
    def __setattr__(self, name, value):
        if name.startswith('_'):
            super().__setattr__(name, value)
        else:
            print('setattr:', name, value)
            setattr(self._obj, name, value)

    # Delegate attribute deletion
    def __delattr__(self, name):
        if name.startswith('_'):
            super().__delattr__(name)
        else:
            print('delattr:', name)
            delattr(self._obj, name)
```

ä¡£çŤĺè£ŹäÿłäżčçŘĘçśżæŮűïijŇä¡ăåŘłéIJĂèęĄçŤĺåőČæĺěåŇĚèčĚäÿŇåĚűäżŰçśżå■şåŘŕïijŽ

```
class Spam:
    def __init__(self, x):
        self.x = x

    def bar(self, y):
        print('Spam.bar:', self.x, y)

# Create an instance
s = Spam(2)
# Create a proxy around it
p = Proxy(s)
# Access the proxy
print(p.x)   # Outputs 2
p.bar(3)   # Outputs "Spam.bar: 2 3"
p.x = 37   # Changes s.x to 37
```

éĂŽè£ĞèĞłåŐŽäźĹ'åśđæĂğèő£éŮőæŰźæşŢïijŇä¡ăåŔŕäżěçŤĺäÿ■åŘŇæŰźåijŔèĞłåŐŽäźĹ'äżčçŘĘçśżèąŇ

èőĺèőž

äżčçŘĘçśżæIJĹ'æŮűåĂŹåŔŕäżěä¡IJäÿžçżğæĹ'£çŽĎæŻ£äżčæŰźæąĹãĂĆä¿ŃåęĆïijŇäÿĂäÿłçőĂå■ŢçŽĎ

```
class A:
    def spam(self, x):
        print('A.spam', x)
    def foo(self):
        print('A.foo')

class B(A):
    def spam(self, x):
        print('B.spam')
        super().spam(x)
    def bar(self):
        print('B.bar')
```

ä¡£çŤĺäżčçŘĘçŽĎèŕĺïijŇåřśæŸŕäÿŃéĺćè£ŹæăůïijŽ

```
class A:
    def spam(self, x):
        print('A.spam', x)
    def foo(self):
        print('A.foo')

class B:
    def __init__(self):
        self._a = A()
    def spam(self, x):
        print('B.spam', x)
        self._a.spam(x)
```

(continues on next page)

(continued from previous page)

```
    def bar(self):
        print('B.bar')
    def __getattr__(self, name):
        return getattr(self._a, name)
```

å¡Ş åőđçŐřäžčçŘĘæĺąåijŕæŮűïijŇè£ŸæIJĽäžŻçżĘèŁĆéIJĂèęĄæşĺæĎŕãĂĆ
éęŰåĚĹïijŇ__getattr__() åőđéŹĚæŸŕäÿĂäÿłåŘŐåď ĞæŰźæşŢïijŇåŕłæIJĽ åIJĺåśđæĂğäÿ■å■ŸåIJĺæŮ
åŻäæ■ď ïijŇåęĆæđIJäžčçŘĘçśżåőđä¿ŃæIJňèžńæIJĽè£ŹäÿłåśđæĂğçŽĎèŕĬïijŇéĆčäźĹäÿ■äijŽèğęåŘŚè£ŹäÿłĂ
åŘęåď ŰïijŇ__setattr__() åŠŇ __delattr__() éIJĂèęĄéćĺåď ŰçŽĎé■ŤæşŢæĺěåŇžåĹĘäžčçŘĘåőđ
_obj çŽĎåśđæĂğãĂĆ äÿĂäÿłéĂŽåÿÿçŽĎçžęåőŽæŸŕåŘłäžčçŘĘéĆčäžŻäÿ■äžěäÿŃåĹŠçž£
_ åijĂåď ť çŽĎåśđæĂğ(äžčçŘĘçśżåŘłæŽť éIJšèćńäžčçŘĘçśżçŽĎåĚňåĚśåśđæĂğ)ãĂĆ

è£ŸæIJĽäÿĂçĆźéIJĂèęĄæşĺæĎŘçŽĎæŸŕïijŇ__getattr__()
åŕźäžŐåď ğéČĺåĹĘäžěåŘŇäÿŃåĹŠçž£(__)åijĂåğŃåŠŇçżŞåř¿çŽĎåśđæĂğåźűäÿ■éĂĆçŤĺãĂĆ
æŕŤåęĆïijŇèĂČèŹŚåęĆäÿŃçŽĎçśżïijŽ

```
class ListLike:
    """__getattr__
↪åŕźäžŐåŘŇäÿŃåĹŠçž£åijĂåğŃåŠŇçżŞåř¿çŽĎæŰźæşŢæŸŕäÿ■èČ¡çŤĺçŽĎïijŇéIJĂèęĄäÿĂäÿłäÿł
↪"""

    def __init__(self):
        self._items = []

    def __getattr__(self, name):
        return getattr(self._items, name)
```

åęĆæđIJæŸŕåĹŻåżžäÿĂäÿłListLikeåŕźèśąïijŇäijŽåŘŚçŐřåőČæŤŕæŇĄæŹőéĂŽçŽĎåĹŮèąĺæŰźæşŢïijŇ
ä¡ĘæŸŕå■ť äÿ■æŤŕæŇĄlen()ãĂĄåĚČçť äæ§ěæĽ¿ç■Ľ ãĂĆä¿ŃåęĆïijŽ

```
>>> a = ListLike()
>>> a.append(2)
>>> a.insert(0, 1)
>>> a.sort()
>>> len(a)
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
TypeError: object of type 'ListLike' has no len()
>>> a[0]
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
TypeError: 'ListLike' object does not support indexing
>>>
```

äÿžäžĘèőľåőČæŤŕæŇĄè£ŹäžŻæŰźæşŢïijŇä¡åå£Ěéążæ Ľ ŃåŁĺçŽĎåőđçŐřè£ŹäžŻæŰźæşŢäžčçŘĘïijŽ

```
class ListLike:
    """__getattr__
↪åŕźäžŐåŘŇäÿŃåĹŠçž£åijĂåğŃåŠŇçżŞåř¿çŽĎæŰźæşŢæŸŕäÿ■èČ¡çŤĺçŽĎïijŇéIJĂèęĄäÿĂäÿłäÿł
↪"""
```

(continues on next page)

(continued from previous page)

```
    def __init__(self):
        self._items = []

    def __getattr__(self, name):
        return getattr(self._items, name)

    # Added special methods to support certain list operations
    def __len__(self):
        return len(self._items)

    def __getitem__(self, index):
        return self._items[index]

    def __setitem__(self, index, value):
        self._items[index] = value

    def __delitem__(self, index):
        del self._items[index]
```

11.8åřŘèŁĆè£ŸæIJL'äÿĂäÿłåIJÍè£IJçÍŃæŰźæşŢèřČçŤÍçŐŕåćČäÿ■ä¡£çŤÍäżčçŘĘçŽĎä¿Ńå■ŘãĂĆ

10.16 8.16 åIJÍçśżäÿ■åőŽäźĽåďŽäÿłæđĎéĂäåŹÍ

éŮőéćŸ

ä¡äæČşåőđçŐřäÿĂäÿłçśżïijŇéŹďäžĘä¡£çŤÍ `__init__()` æŰźæşŢåď'ŰïijŇè£ŸæIJL'åĚűäżŰæŰźåijŘåŘŕäżěåĹİåğŃåŇŰåőČãĂĆ

èğčåĘşæŰźæąĹ

äÿžäžĘåőđçŐřåď'ŽäÿłæđĎéĂäåŹÍïijŇä¡äéIJĂèęĄä¡£çŤÍåĹřçśżæŰźæşŢãĂĆä¿ŃåęĆïijŽ

```
import time

class Date:
    """æŰźæşŢäÿĂïijŽä¡£çŤÍçśżæŰźæşŢ"""
    # Primary constructor
    def __init__(self, year, month, day):
        self.year = year
        self.month = month
        self.day = day

    # Alternate constructor
    @classmethod
    def today(cls):
```

(continues on next page)

(continued from previous page)

```
        t = time.localtime()
        return cls(t.tm_year, t.tm_mon, t.tm_mday)
```

çŻťæŐěèřČçŤÍçśżæŰźæşŢå■şåŔŕïijŇäÿŃéÍćæŸŕä¡£çŤÍçď žä¿ŃïijŽ

```
a = Date(2012, 12, 21) # Primary
b = Date.today() # Alternate
```

èőÍèőž

çśżæŰźæşŢçŽĎäÿĂäÿłäÿżèęAçŤÍéĂŤåřśæŸŕåőŽäźL'åď Žäÿłæđ ĎéĂäåŹÍãĂĆåőČæŐěåŔŮäÿĂäÿł class ä¡IJäÿžçňňäÿĂäÿłåŔĆæŢř(cls)ãĂĆ ä¡ăåžŤèŕěæşÍæĎŔåĹřäžĘè£ŹäÿłçśżèćńçŤÍæÍěåĹŻåżžåźűè£ŤåŻđa

```
class NewDate(Date):
    pass

c = Date.today() # Creates an instance of Date (cls=Date)
d = NewDate.today() # Creates an instance of NewDate (cls=NewDate)
```

10.17 8.17 åĹŻåżžäÿ■èřČçŤÍinitæŰźæşŢçŽĎåőđä¿Ń

éŮőéćŸ

ä¡ăæČşåĹŻåżžäÿĂäÿłåőđä¿ŃïijŇä¡ĘæŸŕåÿŇæIJŻçżŢè£ĞæL'ğèąŇ __init__() æŰźæşŢãĂĆ

èğčåĘşæŰźæąĹ

åŔŕäżěéĂŽè£Ğ __new__() æŰźæşŢåĹŻåżžäÿĂäÿłæIJłåĹÍåğŃåŇŰçŽĎåőđä¿ŃãĂĆä¿ŃåęĆèĂČèŹŚåę

```
class Date:
    def __init__(self, year, month, day):
        self.year = year
        self.month = month
        self.day = day
```

äÿŃéÍćæijŤçď žåęĆä¡Ţäÿ■èřČçŤÍ __init__() æŰźæşŢæÍěåĹŻåżžè£ŹäÿłDateåőđä¿ŃïijŽ

```
>>> d = Date.__new__(Date)
>>> d
<__main__.Date object at 0x1006716d0>
>>> d.year
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
```

(continues on next page)

(continued from previous page)

```
AttributeError: 'Date' object has no attribute 'year'
>>>
```

çżŞæđIJåŘŕäżěçIJŃåĹřïijŇè£ŹäÿłDateåőđä¿ŃçŽĎåśđæĂğyearè£ŸäÿŒåŒŸåIJĺïijŇæĽĂäżěä¡ăéIJĂèęĄæ

```
>>> data = {'year':2012, 'month':8, 'day':29}
>>> for key, value in data.items():
...     setattr(d, key, value)
...
>>> d.year
2012
>>> d.month
8
>>>
```

èőĺèőž

å¡ŞæĹŚäżňåIJĺåŘŒåžŘåĹŮåŕźèśąæĹŰèĂĚåőđçŐřæ§ŘäÿłçśżæŰźæşŢæđĎéĂåĂĞ¡æŢřæŮűéIJĂèęĄçżŢè
`__init__()` æŰźæşŢæİěåĹŻåżżåŕźèśąãĂĆ ä¿ŃåęĆïijŇåŕźäžŐäÿŁéİćçŽĎDateæİěèőšïijŇæIJĽæŮűåĂŹä¡ă
`today()` ïijŽ

```
from time import localtime

class Date:
    def __init__(self, year, month, day):
        self.year = year
        self.month = month
        self.day = day

    @classmethod
    def today(cls):
        d = cls.__new__(cls)
        t = localtime()
        d.year = t.tm_year
        d.month = t.tm_mon
        d.day = t.tm_mday
        return d
```

åŘŇæăůïijŇåIJĺä¡ăåŒŒåžŘåĹŮåŇŰJSONæŢřæŒőæŮűäžğçŢ§äÿĂäÿłåęĆäÿŃçŽĎåŒŮåĚÿåŕźèśąïijŽ

```
data = { 'year': 2012, 'month': 8, 'day': 29 }
```

åęĆæđIJä¡ăæČşåřĘåőČè¡ňæŒćæĹŘäÿĂäÿłDateçśżåđŃåőđä¿ŃïijŇåŘŕäżěä¡£çŢĺäÿłéİćçŽĎæŁĂæIJŕãĂĆ

å¡Şä¡ăéĂŽè£Ğè£Źçğ■éİđåÿÿèğĎæŰźåijŘæİěåĹŻåżżåőđä¿ŃçŽĎæŮűåĂŹïijŇæIJĂåě¡äÿ■èęĄçŻť æŐěåĆ
åŘęåĹŹçŽĎèŕİïijŇåęĆæđIJè£Źäÿłçśżä¡£çŢĺäžĘ `__slots__` ãĂĄproperties ãĂĄdescriptors æĹŰåĚüäżŰéńŸçžğæŁĂæIJŕçŽĎæŮűåĂŹäżčçăĄåřśäijŽåď śæŢĹãĂĆ
èĂŇè£ŹæŮűåĂŹä¡£çŢĺ `setattr()` æŰźæşŢäijŽèőľä¡äçŽĎäżčçăĄåŘŸå¿ŮæŻť åŁăéĂŽçŢĺãĂĆ

10.18 8.18 åĹľçŤÍMixinsæL'ľåśŢçśżåŁ§èČ¡

éŮőéćŸ

ä¡ăæIJL'å¿Ĺåď'ŽæIJL'çŤÍçŽĎæŰźæşŢïijŇæČşä¡£çŤÍåőČäżňæİěæL'ľåśŢåĚűäżŰçśżçŽĎåŁ§èČ¡ãĂĆä¡Ęa
åŻăæ■ď'ä¡ääÿ■èČ¡çőĂå■ŢçŽĎåřĘè£ŹäžŻæŰźæşŢæŤ¿åĚěäÿĂäÿłå§žçśżïijŇçĎűåŘŐèćńåĚűäżŰçśżçżğæL'£ã

èğčåĘşæŰźæąĹ

éĂŽåÿÿå¡Şä¡äæČşèĞłåőŽäźL'çśżçŽĎæŮűåĂŹäijŽçćřäÿŁè£ŹäžŻéŮőéćŸãĂĆåŘŕèČ¡æŸŕæ§ŘäÿłåžŞæŘ
ä¡äåŘŕäżěåĹľçŤÍåőČäżňæİěæđĎéĂää¡äèĞłåůśçŽĎçśżãĂĆ

åĄĞèő¿ä¡äæČşæL'ľåśŢæŸäåřĎåŕźèśąïijŇçżŹåőČäżňæůżåŁäæŮěå£ŮãĂĄåŤŕäÿĂæĂğèő¿ç¡őãĂĄçśżåđĎ

```
class LoggedMappingMixin:
    """
    Add logging to get/set/delete operations for debugging.
    """
    __slots__ = ()   #␣
↪æůůåĚěçśżéČ¡æšąæIJL'åőđä¿ŃåŘŸéĞŘïijŇåŻääÿžçŻt'æŐěåőđä¿ŃåŇŰæůůåĚěçśżæšąæIJL'äżż

    def __getitem__(self, key):
        print('Getting ' + str(key))
        return super().__getitem__(key)

    def __setitem__(self, key, value):
        print('Setting {} = {!r}'.format(key, value))
        return super().__setitem__(key, value)

    def __delitem__(self, key):
        print('Deleting ' + str(key))
        return super().__delitem__(key)


class SetOnceMappingMixin:
    '''
    Only allow a key to be set once.
    '''
    __slots__ = ()

    def __setitem__(self, key, value):
        if key in self:
            raise KeyError(str(key) + ' already set')
        return super().__setitem__(key, value)


class StringKeysMappingMixin:
    '''
    Restrict keys to strings only
```

(continues on next page)

(continued from previous page)

```
    '''
    __slots__ = ()

    def __setitem__(self, key, value):
        if not isinstance(key, str):
            raise TypeError('keys must be strings')
        return super().__setitem__(key, value)
```

è£ŹäžŻçśżå■ŢçŃňä¡£çŤÍèţůæİěæšąæIJL'äżżä¡ŢæĎŘäźL'ïijŇäžŃåőđäÿŁåęĆæđIJä¡ăåŐżåőđä¿ŃåŇŰäżżä
åőČäžňæŸŕçŤÍæİěéĂŽè£ĞåđŽçżğæL'£æİěåŠŇåĚűäżŰæŸăåřĎåŕźèśąæůůåĚěä¡£çŤÍçŽĎãĂĆä¿ŃåęĆïijŽ

```
class LoggedDict(LoggedMappingMixin, dict):
    pass

d = LoggedDict()
d['x'] = 23
print(d['x'])
del d['x']

from collections import defaultdict

class SetOnceDefaultDict(SetOnceMappingMixin, defaultdict):
    pass


d = SetOnceDefaultDict(list)
d['x'].append(2)
d['x'].append(3)
# d['x'] = 23  # KeyError: 'x already set'
```

è£ŹäÿŁä¿Ňå■Řäÿ■ïijŇåŔŕäżěçIJŃåĹřæůůåĚěçśżèů§åĚűäżŰåůšå■ŸåIJÍçŽĎçśż(æŕŤåęĆdictãĂĄdefaultd
çżŞåŘĹåŘŐåřśèČ¡åŘŚæŇěæ■čåÿÿåŁ§æŢĹäžĘãĂĆ

èőÍèőž

æůůåĚěçśżåIJÍæăĞåĞĘåžŞäÿ■å¿Ĺåď'ŽåIJřæŰźéČ¡åĞžçŐřè£ĞïijŇéĂŽåÿÿéČ¡æŸŕçŤÍæİěåČŔäÿŁéİćéĆ
åőČäžňäź§æŸŕåď'Žçżğæ L'£çŽĎäÿĂäÿłäÿżèęĄçŤÍéĂŤãĂĆæŕŤåęĆïijŇå¡Şä¡ăçijŰåĘŹç¡ŚçżIJäżčçăĄæŮůåĂŹ
ä¡ăäijŽçżŘåÿÿä¡£çŤÍ socketserver æĺąåİŮäÿ■çŽĎ ThreadingMixIn
æİěçżŹåĚűäżŰç¡ŚçżIJçŻÿåĚşçśżåćđåŁåđ'Žçž£çÍŃæŤŕæŇĄãĂĆ
ä¿ŃåęĆïijŇäÿŃéİćæŸŕäÿĂäÿłđ'Žçž£çÍŃçŽĎXML-RPCæIJ■åŁąïijŽ

```
from xmlrpc.server import SimpleXMLRPCServer
from socketserver import ThreadingMixIn
class ThreadedXMLRPCServer(ThreadingMixIn, SimpleXMLRPCServer):
    pass
```

åŘŇæŮűåIJÍäÿĂäžŻåď'ğđŃåžŞåŠŇæąĘæđűäÿ■äź§äijŽåŔŚçŐřæůůåĚěçśżçŽĎä¡£çŤÍïijŇçŤÍéĂŤåŘŇæ

åŕźäžŐæůůåĚěçśżïijŇæIJL'åĞăçĆźéIJĂèęĄèőŕä¡ŘãĂĆéęŰåĚĹæŸŕïijŇæůůåĚěçśżäÿ■èČ¡çŻť'æŐěèćńåőđ

åĚűæňąïijŇæůůåĚěçśżæšąæIJL'èĞłåůšçŽĎçŁűæĂĄä£ąæĄŕïijŇäź§åřśæŸŕèŕť'åőČäżňåźűæšąæIJL'åőŽäźL'
__init__() æŰźæşŢïijŇåźűäÿŤæšąæIJL'åőđä¿ŃåśđæĂğãĂĆ
è£Źäź§æŸŕäÿžäżĂäźĹæĹŚäżňåIJĺäÿŁéİćæŸŐçąőåőŽäźL'äžĘ __slots__ = () ãĂĆ

è£ŸæIJL'äÿĂçğ■åőđçŐřæůůåĚěçśżçŽĎæŰźåijŘåřśæŸŕä¡£çŤĺçśżèčĚéěřåŹĺïijŇąęĆäÿŃæL'Ăçd'žïijŽ

```
def LoggedMapping(cls):
    """çňňäžŇçğ■æŰźåijŘïijžä¡£çŤĺçśżèčĚéěřåŹĺ"""
    cls_getitem = cls.__getitem__
    cls_setitem = cls.__setitem__
    cls_delitem = cls.__delitem__

    def __getitem__(self, key):
        print('Getting ' + str(key))
        return cls_getitem(self, key)

    def __setitem__(self, key, value):
        print('Setting {} = {!r}'.format(key, value))
        return cls_setitem(self, key, value)

    def __delitem__(self, key):
        print('Deleting ' + str(key))
        return cls_delitem(self, key)

    cls.__getitem__ = __getitem__
    cls.__setitem__ = __setitem__
    cls.__delitem__ = __delitem__
    return cls


@LoggedMapping
class LoggedDict(dict):
    pass
```

è£ŹäÿłæŢĹæđIJèů§äźŃåL'■çŽĎæŸŕäÿĂæăůçŽĎïijŇèĂŇäÿŤäÿ■åĘ■éIJĂèęĄä¡£çŤĺåď'Žçżğæ L'£äžĘãĂ
åŘĆèĂČ8.13åřŘèŁĆæ§ěçIJŃæŻť'åď'ŽæůůåĚěçśżåŠŇçśżèčĚéěřåŹĺçŽĎä¿Ńå■ŘãĂĆ

10.19 8.19 åőđçŐřçŁűæĂĄåŕźèśąæĹŰèĂĚçŁűæĂĄæIJž

éŮőéćŸ

ä¡äæČşåőđçŐřäÿĂäÿłçŁűæĂĄæIJžæĹŰèĂĚæŸŕåIJĺäÿ■åŘŇçŁűæĂĄäÿŃæL'ğèąŇæŞ■ä¡IJçŽĎåŕźèśąïijl

èğčåĘşæŰźæąĹ

åIJĺå¿Ĺåď'ŽçĺŃåžŘäÿ■ïijŇæIJL'äžŻåŕźèśąäijŽæăźæ■őçŁűæĂĄçŽĎäÿ■åŘŇæĺěæL'ğèąŇäÿ■åŘŇçŽĎæŞ

```
class Connection:
    """æŹőéĂŽæŰźæąĹïijŇåě¡åď ŽäÿłåĹď æŰ■èŕ■åŘěïijŇæŢĹçŐĞä¡ŐäÿŃ~~"""

    def __init__(self):
        self.state = 'CLOSED'

    def read(self):
        if self.state != 'OPEN':
            raise RuntimeError('Not open')
        print('reading')

    def write(self, data):
        if self.state != 'OPEN':
            raise RuntimeError('Not open')
        print('writing')

    def open(self):
        if self.state == 'OPEN':
            raise RuntimeError('Already open')
        self.state = 'OPEN'

    def close(self):
        if self.state == 'CLOSED':
            raise RuntimeError('Already closed')
        self.state = 'CLOSED'
```

è£ŹæăůåĘŹæIJL'å¿Ĺåď ŽçijžçĆźïijŇéęŰåĚĹæŸŕäżčçăĄåď łåď■æİĆäžĘïijŇåě¡åď ŽçŽĎæİąäżűåĹď æŰ■
åŻääÿžäÿĂäžŽåÿÿèğĄçŽĎæŞ■ä¡IJæŕŤåęĆread()ãĂĄwrite()æŕŘæňąæL'ğèąŇåL'■éČ¡éIJĂèęĄæL'ğèąŇæčĂæ§

äÿĂäÿłæŻť'åě¡çŽĎåŁđæşŢæŸŕäÿžæŕŘäÿłçŁűæĂĄåőŽäźL'äÿĂäÿłåŕźèśąïijŽ

```
class Connection1:
    """æŰřæŰźæąĹâĂŤâĂŤåŕźæŕŘäÿłçŁűæĂĄåőŽäźL'äÿĂäÿłçśż"""

    def __init__(self):
        self.new_state(ClosedConnectionState)

    def new_state(self, newstate):
        self._state = newstate
        # Delegate to the state class

    def read(self):
        return self._state.read(self)

    def write(self, data):
        return self._state.write(self, data)

    def open(self):
        return self._state.open(self)
```

(continues on next page)

(continued from previous page)

```
    def close(self):
        return self._state.close(self)


# Connection state base class
class ConnectionState:
    @staticmethod
    def read(conn):
        raise NotImplementedError()

    @staticmethod
    def write(conn, data):
        raise NotImplementedError()

    @staticmethod
    def open(conn):
        raise NotImplementedError()

    @staticmethod
    def close(conn):
        raise NotImplementedError()


# Implementation of different states
class ClosedConnectionState(ConnectionState):
    @staticmethod
    def read(conn):
        raise RuntimeError('Not open')

    @staticmethod
    def write(conn, data):
        raise RuntimeError('Not open')

    @staticmethod
    def open(conn):
        conn.new_state(OpenConnectionState)

    @staticmethod
    def close(conn):
        raise RuntimeError('Already closed')


class OpenConnectionState(ConnectionState):
    @staticmethod
    def read(conn):
        print('reading')

    @staticmethod
    def write(conn, data):
```

(continues on next page)

(continued from previous page)

```
        print('writing')

    @staticmethod
    def open(conn):
        raise RuntimeError('Already open')

    @staticmethod
    def close(conn):
        conn.new_state(ClosedConnectionState)
```

äÿŃéİćæŸŕä¡£çŤÍæijŤçď žïijŽ

```
>>> c = Connection()
>>> c._state
<class '__main__.ClosedConnectionState'>
>>> c.read()
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "example.py", line 10, in read
        return self._state.read(self)
    File "example.py", line 43, in read
        raise RuntimeError('Not open')
RuntimeError: Not open
>>> c.open()
>>> c._state
<class '__main__.OpenConnectionState'>
>>> c.read()
reading
>>> c.write('hello')
writing
>>> c.close()
>>> c._state
<class '__main__.ClosedConnectionState'>
>>>
```

èőÍèőž

åęĆæđIJäžčçăĄäÿ■åĞžçŐŕåď łåď ŽçŽĎæİąäżűåĹď æŰ■èŕ■åŘěçŽĎèŕİïijŇäžčçăĄåřśäijŽåŘŸå¿ŮéŽ¿äžě
è£ŹéĞŃçŽĎèğčåĘşæŰźæąĹæŸŕåřĘæŕŘäÿłçŁűæĂĄæŁ¡åŘŰåĞžæİěåőŽäźL'æĹŘäÿĂäÿłçśžãĂĆ

è£ŹéĞŃçIJŃäÿŁåŐżæIJL'çĆźåěĞæĂłïijŇæŕŘäÿłçŁűæĂĄåŕźèśąéČ¡åŘłæIJL'éİŹæĂĄæŰźæşŢïijŇåźűæš
åőđéŹĚäÿŁïijŇæL'ĂæIJL'çŁűæĂĄä£ąæĄŕéČ¡åŘłå■ŸåĆÍåIJÍ `Connection`
åőđä¿ŃäÿŁãĂĆ åIJÍå§žçśżäÿ■åőŽäźL'çŽĎ `NotImplementedError`
æŸŕäÿžäžĘçąőä£İå■ŘçśżåőđçŐŕäžĘçŻÿåžŤçŽĎæŰźæşŢãĂĆ è£ŹéĞŃä¡äæĹŰèőÿè£ŸæČşä¡£çŤÍ8.12åřŘèŁĆ

èő¿èőąæÍąåijŘäÿ■æIJL'äÿĂçğ■æÍąåijŘåŘńçŁűæĂĄæÍąåijŘïijŇè£ŹäÿĂåřŘèŁĆçőŮæŸŕäÿĂäÿłåĹİæ■ěåĚ

10.20 8.20 éĂŽè£Ğå■ŮçňęäÿšèřČçŤÍåŕźèśąæŰźæşŢ

éŮőéćŸ

ä¡ăæIJL'äÿĂäÿłå■Ůçňęäÿšå¡ćåijŔçŽĎæŰźæşŢåŘ■çğřijŇæČşéĂŽè£ĞåőČèřČçŤÍæ§ŘäÿłåŕźèśąçŽĎåŕźå

èğčåĘşæŰźæąĹ

æIJĂçőĂå■ŢçŽĎæČĚåĘţïijŇåŔŕäżěä¡£çŤÍ `getattr()` ïijŽ

```
import math

class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def __repr__(self):
        return 'Point({!r:},{!r:})'.format(self.x, self.y)

    def distance(self, x, y):
        return math.hypot(self.x - x, self.y - y)


p = Point(2, 3)
d = getattr(p, 'distance')(0, 0)  # Calls p.distance(0, 0)
```

åŔęåď'ŰäÿĂçğ■æŰźæşŢæŸŕä¡£çŤÍ `operator.methodcaller()` ïijŇä¿ŃåęĆïijŽ

```
import operator
operator.methodcaller('distance', 0, 0)(p)
```

å¡Şä¡ăéIJĂèęĄéĂŽè£ĞçŻÿåŘŇçŽĎåŔĆæŢřåď'ŽæňąèřČçŤÍæ§ŘäÿłæŰźæşŢæŮűïijŇä¡£çŤÍ `operator.methodcaller` åŕśå¿ĹæŰźä¿£äžĘãĂĆ æŕŤåęĆä¡ăéIJĂèęĄæŐŠåžŘäÿĂçşżåĹŮçŽĎçĆźïijŇåŕ

```
points = [
    Point(1, 2),
    Point(3, 0),
    Point(10, -3),
    Point(-5, -7),
    Point(-1, 8),
    Point(3, 2)
]
# Sort by distance from origin (0, 0)
points.sort(key=operator.methodcaller('distance', 0, 0))
```

èőÍèőž

èřČçŤÍäÿĂäÿłæŰźæşŢåőđéŹĚäÿŁæŸŕäÿď'éČÍçŃňçńŃæŞ■ä¡IJïijŇçňňäÿĂæ■ěæŸŕæ§ěæĽ¿åśđæĂğïijŇç
åŻăæ■ď'ïijŇäÿžäžĘèřČçŤÍæ§ŘäÿłæŰźæşŢïijŇä¡ăåŘŕäżěéęŰåĚĹéĂŽè£Ğ `getattr()`
æİěæ§ěæĽ¿åĹřè£ŹäÿłåśđæĂğïijŇçĎűåŘŐåĘ■åŐżäžěåĞ¡æŢřæŰźåijŘèřČçŤÍłőČå■şåŘŕãĂĆ

`operator.methodcaller()` åĹŻåżžäÿĂäÿłåŘŕèřČçŤÍłŕźèśąïijŇåźűåŘŇæŮűæŘŘä¿ŻæĽĂæIJĽ'å£
çĎűåŘŐèřČçŤÍçŽĎæŮűåĂŹåŘłéIJĂèęĄåřĘåőđä¿ŃåŕźèśąäijăéĂŠçżŹåőČå■şåŘŕïijŇæŕŤåęĆïijŽ

```
>>> p = Point(3, 4)
>>> d = operator.methodcaller('distance', 0, 0)
>>> d(p)
5.0
>>>
```

éĂŽè£ĞæŰźæşŢåŘ■çğřå■ŮçňęäÿšæİěèřČçŤÍæŰźæşŢéĂŽåÿÿåĞžçŐřåIJÍéIJĂèęĄæÍąæŃ§
`case` èŕ■åŘěæĹŰåőđçŐřèő£éŮőèĂĚæÍąåijŘçŽĎæŮűåĂŹãĂĆ
åŘĆèĂČäÿŃäÿĂåřŘèŁĆèŐůåŘŰæŻť'åď'ŽéńŸçžğä¿Ńå■ŘãĂĆ

10.21 8.21 åőđçŐřèő£éŮőèĂĚæÍąåijŘ

éŮőécŸ

ä¡ăèęĄåď'ĎçŘĘçŤśåď'ğéĞŘäÿ■åŘŃçśżåđŃçŽĎåŕźèśąçżĎæĹŘçŽĎåď'■æİĆæŢřæ■őçżŞæđĎïijŇæŕŘäÿĂ
æŕŤåęĆïijŇéĄ■åŐĘäÿĂäÿłæăŚå¡ćçżŞæđĎïijŇçĎűåŘŐæăźæ■őæŕŘäÿłèŁĆçĆźçŽĎçŻÿåžŤçŁűæĂĄæĽ'ğèąŃ

èğčåĘşæŰźæąĹ

è£ŹéĞŃéĄĞåĹřçŽĎéŮőécŸåIJÍçijŰçÍŃécĘå§§äÿ■æŸŕå¿ĹæŹőéĄ■çŽĎïijŇæIJĽ'æŮűåĂŹäijŽæđĎåżžä
åĄĞèő¿ä¡ăèęĄåĘŹäÿĂäÿłèąÍçď'žæŢřå■ęèąÍè¿¿åijŘçŽĎçÍŃåžŘïijŇéĆčäźĹä¡ăåŘŕèČ¡éIJĂèęĄåőŽäźĽ'åęĆäÿŃ

```
class Node:
    pass

class UnaryOperator(Node):
    def __init__(self, operand):
        self.operand = operand

class BinaryOperator(Node):
    def __init__(self, left, right):
        self.left = left
        self.right = right

class Add(BinaryOperator):
    pass

class Sub(BinaryOperator):
```

(continues on next page)

(continued from previous page)

```
    pass

class Mul(BinaryOperator):
    pass

class Div(BinaryOperator):
    pass

class Negate(UnaryOperator):
    pass

class Number(Node):
    def __init__(self, value):
        self.value = value
```

çĎűåŘŐåĹľçŤĺè£ŹäžŻçśżæđĎåżžåţŇåěŮæŢřæ■őçżŞæđĎïijŇåęĆäÿŃæĽĂçď žïijŽ

```
# Representation of 1 + 2 * (3 - 4) / 5
t1 = Sub(Number(3), Number(4))
t2 = Mul(Number(2), t1)
t3 = Div(t2, Number(5))
t4 = Add(Number(1), t3)
```

è£ŹæăůåĄŽçŽĎéŮőéćŸæŸŕåŕźäžŐæŕŘäÿłèąĺè¿¿åijŘïijŇæŕŘæňąéČ¡èęĄéĞ■æŰřåőŽäźĽäÿĂéĄ■ïijŇæl
è£ŹéĞŇæĹŚäžňä¡£çŤĺèő£éŮőèĂĚæĺąåijŘåŘŕäžěè¿¿åĹřè£ŹæăůçŽĎçŻőçŽĎïijŽ

```
class NodeVisitor:
    def visit(self, node):
        methname = 'visit_' + type(node).__name__
        meth = getattr(self, methname, None)
        if meth is None:
            meth = self.generic_visit
        return meth(node)

    def generic_visit(self, node):
        raise RuntimeError('No {} method'.format('visit_' +␣
↪type(node).__name__))
```

äÿžäžĘä¡£çŤĺè£ŹäÿłçśżïijŇåŘŕäžěåőŽäźĽäÿĂäÿłçśżçżğæĽ£åőČåźűäÿŤåőđçŐřåŘĎçğ■
`visit_Name()` æŰźæşŢïijŇåĚűäÿ■NameæŸŕnodeçśżåđŃãĂĆ
ä¿ŃåęĆïijŇåęĆæđIJä¡äæČşæśĆèąĺè¿¿åijŘçŽĎåĂijïijŇåŘŕäžěè£ŹæăůåĘŹïijŽ

```
class Evaluator(NodeVisitor):
    def visit_Number(self, node):
        return node.value

    def visit_Add(self, node):
        return self.visit(node.left) + self.visit(node.right)
```

(continues on next page)

(continued from previous page)

```
    def visit_Sub(self, node):
        return self.visit(node.left) - self.visit(node.right)

    def visit_Mul(self, node):
        return self.visit(node.left) * self.visit(node.right)

    def visit_Div(self, node):
        return self.visit(node.left) / self.visit(node.right)

    def visit_Negate(self, node):
        return -node.operand
```

ä¡£çŤÍçď'žä¿ŃïijŽ

```
>>> e = Evaluator()
>>> e.visit(t4)
0.6
>>>
```

ä¡IJäÿžäÿĂäÿłäÿ■åŘŇçŽĎä¿Ńå■ŘïijŇäÿŃéĺćåőŽäźĽäÿĂäÿłçśżåIJÍäÿĂäÿłæăĹäÿŁéĺćåřĘäÿĂäÿłèąÍè¿¿å

```
class StackCode(NodeVisitor):
    def generate_code(self, node):
        self.instructions = []
        self.visit(node)
        return self.instructions

    def visit_Number(self, node):
        self.instructions.append(('PUSH', node.value))

    def binop(self, node, instruction):
        self.visit(node.left)
        self.visit(node.right)
        self.instructions.append((instruction,))

    def visit_Add(self, node):
        self.binop(node, 'ADD')

    def visit_Sub(self, node):
        self.binop(node, 'SUB')

    def visit_Mul(self, node):
        self.binop(node, 'MUL')

    def visit_Div(self, node):
        self.binop(node, 'DIV')

    def unaryop(self, node, instruction):
        self.visit(node.operand)
```

(continues on next page)

(continued from previous page)

```
        self.instructions.append((instruction,))

    def visit_Negate(self, node):
        self.unaryop(node, 'NEG')
```

ä¡£çŤÍçď žä¿ŃïijŽ

```
>>> s = StackCode()
>>> s.generate_code(t4)
[('PUSH', 1), ('PUSH', 2), ('PUSH', 3), ('PUSH', 4), ('SUB',),
('MUL',), ('PUSH', 5), ('DIV',), ('ADD',)]
>>>
```

èőÍèőž

åĹŽåijĂåğŃçŽĎæŮűåĂŹä¡ăåŘŕèČ¡äijŽåĘŹåď ğéĞŘçŽĎif/elseèŕ■åŘěæİěåőđçŐřïijŃ è£ŹéĞŃèő£éŮőèĂĚæĺąåijŘçŽĎåě¡åď ĎåřśæŸŕéĂŽè£Ğ getattr() æİěèŐůåŘŰçŻÿåžŤçŽĎæŰźæşŢïijŃåźűåĹľçŤĺéĂŠå¡ŠæİěéĄ■åŐĘæĽĂæIJĽçŽĎèŁĆçĆźïijŽ

```
def binop(self, node, instruction):
    self.visit(node.left)
    self.visit(node.right)
    self.instructions.append((instruction,))
```

è£ŸæIJĽäÿĂçĆźéIJĂèęĄæŃĞåĞžçŽĎæŸŕïijŃè£Źçğ■æŁĂæIJŕäź§æŸŕåőđçŐřåĚűäżŰèŕ■èĺĂäÿ■switch æŕŤåęĆïijŃåęĆæđIJä¡ăæ■čåIJĺåĘŹäÿĂäÿłHTTPæąĘæđűïijŃä¡ăåŘŕèČ¡äijŽåĘŹè£ŹæăůäÿĂäÿłèŕůæśĆåĹĘåŘ

```
class HTTPHandler:
    def handle(self, request):
        methname = 'do_' + request.request_method
        getattr(self, methname)(request)
    def do_GET(self, request):
        pass
    def do_POST(self, request):
        pass
    def do_HEAD(self, request):
        pass
```

èő£éŮőèĂĚæĺąåijŘäÿĂäÿłçijžçĆźåřśæŸŕåőČäÿěéĞ■ä¿ÌèţŰéĂŠå¡ŠïijŃåęĆæđIJæŢřæ■őçżŞæđĎåţŃåěŮ æIJĽæŮűåĂŹäijŽèűĚè£ĞPythonçŽĎéĂŠå¡ŠæůśåžęéŹŘåĹű(åŘĆèĂČ sys.getrecursionlimit())ãĂĆ

åŘŕäżěåŘĆçĚğ8.22åřŘèŁĆïijŃåĹľçŤĺçŤ§æĹŘåŹĺæĹŰè£■äżčåŹĺæİěåőđçŐřéİđéĂŠå¡ŠéĄ■åŐĘçőŮæşŢ

åIJĺèů§èğčæđŘåŠŃçijŰèŕŚçŻÿåĚşçŽĎçijŰçĺŃäÿ■ä¡£çŤĺèő£éŮőèĂĚæĺąåijŘæŸŕéİđåÿÿåÿÿèğĄçŽĎãĂĆ PythonæIJňèžńçŽĎ ast æĺąåİŮåĂijå¿ŮåĚşæşĺäÿŃïijŃåŘŕäżěåŐżçIJŃçIJŃæžŘçăĄãĂĆ 9.24åřŘèŁĆæijŤçď žäžĘäÿĂäÿłåĹľçŤĺ ast æĺąåİŮæİěåď ĎçŘĘPythonæžŘäżčçăĄçŽĎä¿Ńå■ŘãĂĆ

10.22 8.22 äÿ■çŤíéĂŠå¡Šåőđç Ő řèő£éŮőèĂĚæÍąåijŘ

éŮőéćŸ

ä¡ăä¡£çŤÍèő£éŮőèĂĚæÍąåijŘéĄ■åŐĘäÿĂäÿłå¿ĹæůśçŽĎåţŇåěŮæăŚå¡ćæŢřæ■őçżŞæđĎïijŇåźűäÿŤåŻää
ä¡ăæČşæűĹéŹď'éĂŠå¡ŠïijŇåźűåŘŇæŮűä£ÍæŇĄèő£éŮőèĂĚçijŰçÍŃæÍąåijŘãĂĆ

èğčåĘşæŰźæąĹ

éĂŽè£ĞåůğåęŹçŽĎä¡£çŤÍçŤ§æĹŘåŹÍåŘŕäżěåIJÍæăŚéĄ■åŐĘæĹŰæŘIJçť'ćçőŮæşŢäÿ■æűĹéŹď'éĂŠå¡Š
åIJÍ8.21åřŘèŁĆäÿ■ïijŇæĹŚäżňçżŹåĞžäžĘäÿĂäÿłèő£éŮőèĂĚçśżãĂĆ
äÿŃéÍćæĹŚäżňåĹ'çŤÍäÿĂäÿłæăĹåŠŇçŤ§æĹŘåŹÍéĞ■æŰřåőđçŐřè£ŹäÿłçśżïijŽ

```
import types

class Node:
    pass

class NodeVisitor:
    def visit(self, node):
        stack = [node]
        last_result = None
        while stack:
            try:
                last = stack[-1]
                if isinstance(last, types.GeneratorType):
                    stack.append(last.send(last_result))
                    last_result = None
                elif isinstance(last, Node):
                    stack.append(self._visit(stack.pop()))
                else:
                    last_result = stack.pop()
            except StopIteration:
                stack.pop()

        return last_result

    def _visit(self, node):
        methname = 'visit_' + type(node).__name__
        meth = getattr(self, methname, None)
        if meth is None:
            meth = self.generic_visit
        return meth(node)

    def generic_visit(self, node):
        raise RuntimeError('No {} method'.format('visit_' +␣
↪type(node).__name__))
```

åęĆæđIJä¡ăä¡£çŤÍè£ŹäÿłçśżïijŇäź§èČ¡è¿¿åĹřçŻÿåŘŇçŽĎæŢĹæđIJãĂĆäžŃåőđäÿŁä¡ăăőŇåĚÍåŘŕäżěåřŘ

èĂČèŹŚąęĆäÿŃäżčçăAïijŇéĄ■åŐĘäÿĂäÿłèąĺè¿¿åijŘçŽĎæăŚïijŽ

```
class UnaryOperator(Node):
    def __init__(self, operand):
        self.operand = operand

class BinaryOperator(Node):
    def __init__(self, left, right):
        self.left = left
        self.right = right

class Add(BinaryOperator):
    pass

class Sub(BinaryOperator):
    pass

class Mul(BinaryOperator):
    pass

class Div(BinaryOperator):
    pass

class Negate(UnaryOperator):
    pass

class Number(Node):
    def __init__(self, value):
        self.value = value

# A sample visitor class that evaluates expressions
class Evaluator(NodeVisitor):
    def visit_Number(self, node):
        return node.value

    def visit_Add(self, node):
        return self.visit(node.left) + self.visit(node.right)

    def visit_Sub(self, node):
        return self.visit(node.left) - self.visit(node.right)

    def visit_Mul(self, node):
        return self.visit(node.left) * self.visit(node.right)

    def visit_Div(self, node):
        return self.visit(node.left) / self.visit(node.right)

    def visit_Negate(self, node):
        return -self.visit(node.operand)
```

(continues on next page)

(continued from previous page)

```
if __name__ == '__main__':
    # 1 + 2*(3-4) / 5
    t1 = Sub(Number(3), Number(4))
    t2 = Mul(Number(2), t1)
    t3 = Div(t2, Number(5))
    t4 = Add(Number(1), t3)
    # Evaluate it
    e = Evaluator()
    print(e.visit(t4))  # Outputs 0.6
```

åęĆæđIJåţŇåěŮåśĆæňąåď łæůśéĆčäźĹäÿŁè£řçŽĎEvaluatorårśäijŽåď śæŢĹïijŽ

```
>>> a = Number(0)
>>> for n in range(1, 100000):
... a = Add(a, Number(n))
...
>>> e = Evaluator()
>>> e.visit(a)
Traceback (most recent call last):
...
    File "visitor.py", line 29, in _visit
return meth(node)
    File "visitor.py", line 67, in visit_Add
return self.visit(node.left) + self.visit(node.right)
RuntimeError: maximum recursion depth exceeded
>>>
```

çŐřåIJĺæĹŚäżňçĺ■å¿őä£őæŤźäÿŃäÿŁéİćçŽĎEvaluatorïijŽ

```
class Evaluator(NodeVisitor):
    def visit_Number(self, node):
        return node.value

    def visit_Add(self, node):
        yield (yield node.left) + (yield node.right)

    def visit_Sub(self, node):
        yield (yield node.left) - (yield node.right)

    def visit_Mul(self, node):
        yield (yield node.left) * (yield node.right)

    def visit_Div(self, node):
        yield (yield node.left) / (yield node.right)

    def visit_Negate(self, node):
        yield - (yield node.operand)
```

åĘ■æňąè£ŘèąŇïijŇåřśäÿ■äijŽæŁěéŤŹäžĘïijŽ

```
>>> a = Number(0)
>>> for n in range(1,100000):
...     a = Add(a, Number(n))
...
>>> e = Evaluator()
>>> e.visit(a)
4999950000
>>>
```

åęĆæđIJä¡ăè£ŸæČşæůżåŁååĚűäżŰèĞłåőŽäźL'éĂżè¿Śäź§æšąéŮőéćŸïijŽ

```
class Evaluator(NodeVisitor):
    ...
    def visit_Add(self, node):
        print('Add:', node)
        lhs = yield node.left
        print('left=', lhs)
        rhs = yield node.right
        print('right=', rhs)
        yield lhs + rhs
    ...
```

äÿŃéİćæŸŕçőĂå■ŢçŽĎæţŃèŕŢïijŽ

```
>>> e = Evaluator()
>>> e.visit(t4)
Add: <__main__.Add object at 0x1006a8d90>
left= 1
right= -0.4
0.6
>>>
```

èőÍèőž

è£ŹäÿĂåřŔèŁĆæĹŚäżňæijŤçd'žäžEçŤ§æĹŘåŹÍåŠŇå■ŘçÍŃåIJÍçÍŃåžŘæŐğåĹűæţĄæŰźéİćçŽĎåijžåď'ğ
éĄ£åĚ■éĂŠå¡ŠçŽĎäÿĂäÿłéĂŽåÿÿæŰźæşŢæŸŕä¡£çŤÍäÿĂäÿłæăĹæĹŰéŸ§åĹŮçŽĎæŢřæ■őçżŞæđĎãĂĆ
ä¿ŃåęĆïijŇæůśåžęäijŸåĚĹçŽĎéĄ■åŐEçőŮæşŢïijŇçňňäÿĂæňąçćřåĹřäÿĂäÿłèŁĆçĆźæŮűåřĘåĚűåŐŃåĚěæă
æŰźæşŢçŽĎæäÿå£ČæĂÌèůŕåřśæŸŕè£ŹæăůãĂĆ

åŘęåď'ŰäÿĂäÿłéIJĂèęĄçŘEęğčçŽĎåřśæŸŕçŤ§æĹŘåŹÍäÿ■yieldèŕ■åŘěãĂĆå¡ŞçćřåĹřyieldèŕ■åŘěæŮűïij
äÿŁéÍćçŽĎä¿Ńå■Ŕä¡£çŤÍè£ŹäÿłæŁĂæIJŕæÌěäžčæŻ£ážEéĂŠå¡ŠãĂĆä¿ŃåęĆïijŇäźŃåL'■æĹŚäżňæŸŕè£Źæăŭ

```
value = self.visit(node.left)
```

çŐřåIJÍæ■ćæĹŘyieldèŕ■åŘěïijŽ

```
value = yield node.left
```

åőČäijŽåřĘ `node.left` è£ŤåŻđçżŹ `visit()` æŰźæşŢïijŇçĎűåŘŐ `visit()`

æŰźæşŢèřČçŤÍéĆčäÿłèŁĆçĆźçŻÿåžŤçŽĎ `visit_Name()` æŰźæşŢãĂĆ yield-
æŽĆæŮűåřĘçÍŃåžŘæŐğåĹűåŹÍèőľåĞžçżŹèřČçŤÍèĂĚïijŇå¡ŞæĽğèąŇåőŇåŘŐïijŇçżŞæđIJäijŽèţŃåĂijçżŹv

çIJŃåőŇè£ŹäÿĂåřŘèŁĆïijŇä¡ääź§èöÿæČşåŐżåŕżæĽ¿åĚűåőČæšąæIJĽyieldèŕ■åŘěçŽĎæŰźæąĹãĂĆä¡
ä¿ŃåęĆïijŇäÿžäžĘæűĹéŹď´éĂŠå¡ŠïijŇä¡ää£ĚéążèęĄçżť´æŁď´äÿĂäÿłæăĹçżŞæđĎïijŇåęĆæđIJäÿ■ä¡£çŤÍçŤ§
åőđéŹĚäÿŁïijŇä¡£çŤÍyieldèŕ■åŘěåŘŕäžěèőľ´ä¡ääĘŹåĞžéİđåÿÿæijĆäžőçŽĎäžčçăĄïijŇåőČæűĹéŹď´äžĘéĂŠå¡

10.23 8.23 å¿łçŐŕåijŢçŤÍæŢřæ■őçżŞæđĎçŽĎåĘĚå■ŸçőąçŘĘ

éŮőécŸ

ä¡äçŽĎçÍŃåžŘåĹŻåżžäžĘå¿Ĺåď´Žå¿łçŐŕåijŢçŤÍæŢřæ■őçżŞæđĎ(æŕŤåęĆæăŚãĂĄåŻ¿ãĂĄèğĆåŕ§èĂĚæ

èğčåĘşæŰźæąĹ

äÿĂäÿłçőĂå■ŢçŽĎå¿łçŐŕåijŢçŤÍæŢřæ■őçżŞæđĎä¿Ňå■ŘåŕśæŸŕäÿĂäÿłæăŚå¡ćçżŞæđĎïijŇåŘŇäžšèŁĆ
è£Źçğ■æČĚåĘţäÿŃïijŇåŘŕäżěèĂČèŹŚä¡£çŤÍ `weakref` åžŞäÿ■çŽĎåijśåijŢçŤÍãĂĆä¿ŃåęĆïijŽ

```
import weakref

class Node:
    def __init__(self, value):
        self.value = value
        self._parent = None
        self.children = []

    def __repr__(self):
        return 'Node({!r:})'.format(self.value)

    # property that manages the parent as a weak-reference
    @property
    def parent(self):
        return None if self._parent is None else self._parent()

    @parent.setter
    def parent(self, node):
        self._parent = weakref.ref(node)

    def add_child(self, child):
        self.children.append(child)
        child.parent = self
```

è£Źçğ■æŸŕæČşæŰźåijŘåĚĄèőÿparentéİŹéżŸçżĹæ■ćãĂĆä¿ŃåęĆïijŽ

```
>>> root = Node('parent')
>>> c1 = Node('child')
>>> root.add_child(c1)
>>> print(c1.parent)
```

(continues on next page)

(continued from previous page)

```
Node('parent')
>>> del root
>>> print(c1.parent)
None
>>>
```

èőÍèőž

å¿łçŐŕåijŢçŤÍçŽĎæŢřæ■őçżŞæđĎåIJÍPythonäÿ■æŸŕäÿĂäÿłå¿ĹæčŸæĽŃçŽĎéŮőéćŸïijŇåŻääÿžæ■čåÿ
ä¿ŃåęĆèĂČèŹŚåęĆäÿŃäżčçăĄïijŽ

```
# Class just to illustrate when deletion occurs
class Data:
    def __del__(self):
        print('Data.__del__')

# Node class involving a cycle
class Node:
    def __init__(self):
        self.data = Data()
        self.parent = None
        self.children = []

    def add_child(self, child):
        self.children.append(child)
        child.parent = self
```

äÿŃéİćæĹŚäżňä¡£çŤÍè£ŹäÿłäżčçăĄæİěåĄŽäÿĂäžŽåđČåIJ¿åŻđæŤűèŕŢéłŇïijŽ

```
>>> a = Data()
>>> del a # Immediately deleted
Data.__del__
>>> a = Node()
>>> del a # Immediately deleted
Data.__del__
>>> a = Node()
>>> a.add_child(Node())
>>> del a # Not deleted (no message)
>>>
```

åŘŕäżěçIJŃåĹřïijŇæIJĂåŘŐäÿĂäÿłçŽĎåĹăéŹď æŮűæĽŞå■řèŕ■åŘěæšąæIJĽåĞżçŐřãĂĆåŐ§åŻăæŸŕPy
å¡ŞäÿĂäÿłåŕźèśąçŽĎåijŢçŤÍæŢřåŘŸæĹŘ0çŽĎæŮűåĂŹæĽ■äijŽçńŃå■şåĹăéŹď æŐĽãĂĆèĂŇåŕźäžŐå¿łçŐŕ
åŻăæ■d'ïijŇåIJÍäÿŁéİćä¿Ňå■Řäÿ■æIJĂåŘŐéČÍåĹĘïijŇçĹűèŁĆçĆźåŠŇå■l'å■ŘèŁĆçĆźäžŠçŻÿæŃěæIJĽåŕźa

PythonæIJĽåŘęåď ŰçŽĎåđČåIJ¿åŻđæŤűåŹÍæİěäÿŞéŮÍéŠĹåŕźå¿łçŐŕåijŢçŤÍçŽĎïijŇä¡ĘæŸŕä¡ăæřÿè£IJ
åŘęåď Űä¡ăè£ŸåŘŕäżěæĽŃåŁÍçŽĎèğęåŘŚåőČïijŇä¡ĘæŸŕäżčçăĄçIJŃäÿŁåŐżå¿ĹæŇńïijŽ

```
>>> import gc
>>> gc.collect() # Force collection
Data.__del__
Data.__del__
>>>
```

åęĆæđIJå¿łçŐŕåijŢçŤĺçŽĎåŕźèśąèĞłåůśè£ŸåőŽäźĽäžĘèĞłåůśçŽĎ `__del__()` æŰźæşŢïijŇéĆčäźĹäijŽèőľæČĚåĘţåŘŸå¿ŮæŻť'çş§çşŢãĂĆ åĄĞèő¿ä¡ääČŔäÿŃéİćè£ŹæăůçżŹNodeåőŽäźĽèĞłåůśçŽĎ `__del__()` æŰźæşŢïijŽ

```
# Node class involving a cycle
class Node:
    def __init__(self):
        self.data = Data()
        self.parent = None
        self.children = []

    def add_child(self, child):
        self.children.append(child)
        child.parent = self

    # NEVER DEFINE LIKE THIS.
    # Only here to illustrate pathological behavior
    def __del__(self):
        del self.data
        del.parent
        del.children
```

è£Źçğ■æČĚåĘţäÿŃïijŇåđČåIJ¿åŻđæŤűæřÿè£IJéČ¡äÿ■äijŽåŐżåŻđæŤűè£ŹäÿłåŕźèśąçŽĎïijŇè£ŸäijŽåŕij åęĆæđIJä¡äèŕŢçİĂåŐżè£ŘèąŇåőČäijŽåŘŚçŐřïijŇ`Data.__del__` æűĹæĄŕæřÿè£IJäÿ■äijŽåĞžçŐřäžĘ,çŤŽèĞşåIJĺä¡ääijžåĹűåĘĚå■ŸåŻđæŤűæŮűïijŽ

```
>>> a = Node()
>>> a.add_child(Node())
>>> del a # No message (not collected)
>>> import gc
>>> gc.collect() # No message (not collected)
>>>
```

åijśåijŢçŤĺæűĹéŹďäžĘåijŢçŤĺå¿łçŐŕçŽĎè£ŹäÿłéŮőécŸïijŇæIJňèťĺæİěèőšïijŇåijśåijŢçŤĺåřśæŸŕäÿĂäÿ ä¡ääŘŕäžěéĂŽè£Ğ `weakref` æİěåĹŻåżžåijśåijŢçŤĺãĂĆä¿ŃåęĆïijŽ

```
>>> import weakref
>>> a = Node()
>>> a_ref = weakref.ref(a)
>>> a_ref
<weakref at 0x100581f70; to 'Node' at 0x1005c5410>
>>>
```

äÿžäžĘèő£éŮőåijśåijŢçŤĺæĽĂåijŢçŤĺçŽĎåŕźèśąïijŇä¡ääŘŕäžěåČŔåĞ¡æŢřäÿĂæăůåŐżèřČçŤĺåőČå■şåŘ

çŤśäžŐåŐ§åğŃåŕźèśąçŽĎåijŢçŤÍèőąæŢřæšąæIJL’åćđåŁäïijŇéĆčäźĹåřśåŘŕäżěåŐżåĹäéŹď’åőČäžĘãĂĆä¿Ńåę

```
>>> print(a_ref())
<__main__.Node object at 0x1005c5410>
>>> del a
Data.__del__
>>> print(a_ref())
None
>>>
```

éĂŽè£Ğè£ŹéĞŇæijŤçď’žçŽĎåijśåijŢçŤÍæŁĂæIJŕïijŇä¡ääijŽåŘŚçŐřäÿ■åĘ■æIJL’å¿łçŐŕåijŢçŤÍéŮőéćŸ
ä¡äè£ŸèČ¡åŘĆèĂČ8.25åřŘèŁĆåĚşäžŐåijśåijŢçŤÍçŽĎåŘęåď’ŰäÿĂäÿłä¿Ńå■ŘãĂĆ

10.24 8.24 èőľçśżæŤŕæŇĄæŕŤè¿ČæŞ■ä¡IJ

éŮőéćŸ

ä¡äæČşèőľæ§ŘäÿłçśżçŽĎåőđä¿ŃæŤŕæŇĄæäĞåĞĘçŽĎæŕŤè¿Čè£ŘçőŮ(æŕŤåęĆ>=,!=,<=,<ç■L’)ïijŇä¡E

èğčåĘşæŰźæąĹ

PythonçśżåŕźæŕŘäÿłæŕŤè¿ČæŞ■ä¡IJéČ¡éIJĂèęĄåőđçŐřäÿĂäÿłçL’źæőŁæŰźæşŢæİěæŤŕæŇĄãĂĆ
ä¿ŃåęĆäÿžäžĘæŤŕæŇĄ>=æŞ■ä¡IJçňęïijŇä¡äéIJĂèęĄåőŽäźL’äÿĂäÿł `__ge__()`
æŰźæşŢãĂĆ åř¡çőąåőŽäźL’äÿĂäÿłæŰźæşŢæšąäżĂäźĹéŮőéćŸïijŇä¡ĘåęĆæđIJèęĄä¡äåőđçŐřæL’ĂæIJL’åŘŕè

èčĚéěřåŹÍ `functools.total_ordering` åřśæŸŕçŤÍæİěçőĂåŇŰè£Źäÿłåď’ĎçŘĘçŽĎãĂĆ
ä¡£çŤÍåőČæİěèčĚéěřäÿĂäÿłæİěïijŇä¡äåŘłéIJĂåőŽäźL’äÿĂäÿł `__eq__()` æŰźæşŢïijŇ
åď’ŰåŁäåĚűäżŰæŰźæşŢ(`__lt__`, `__le__`, `__gt__`, or `__ge__`)äÿ■çŽĎäÿĂäÿłå■şåŘŕãĂĆ
çĎűåŘŐèčĚéěřåŹÍäijŽèĞłåŁÍäÿžä¡äąńåĚĚåĚűåőČæŕŤè¿ČæŰźæşŢãĂĆ

ä¡IJäÿžä¿Ńå■ŘïijŇæĹŚäžňæđĎåżžäÿĂäżŻæĹ£å■ŘïijŇçĎűåŘŐçżŹåőČäžňåćđåŁääÿĂäžŻæĹ£éŮť’ïijŇæ

```
from functools import total_ordering

class Room:
    def __init__(self, name, length, width):
        self.name = name
        self.length = length
        self.width = width
        self.square_feet = self.length * self.width

@total_ordering
class House:
    def __init__(self, name, style):
        self.name = name
        self.style = style
        self.rooms = list()
```

(continues on next page)

(continued from previous page)

```
    @property
    def living_space_footage(self):
        return sum(r.square_feet for r in self.rooms)

    def add_room(self, room):
        self.rooms.append(room)

    def __str__(self):
        return '{}: {} square foot {}'.format(self.name,
                self.living_space_footage,
                self.style)

    def __eq__(self, other):
        return self.living_space_footage == other.living_space_
↪footage

    def __lt__(self, other):
        return self.living_space_footage < other.living_space_
↪footage
```

è£ŹéĞŇæĹŚäżňåŘłæŸŕçżŹHouseçśżåőŽäźĽäžĘäÿď äÿłæŰźæşŢïijŽ__eq__() åŠŇ __lt__() ïijŇåőČåřśèČ¡æŤŕæŇĄæĽĂæIJĽçŽĎæŕŤè¿ČæŞ■ä¡IJïijŽ

```
# Build a few houses, and add rooms to them
h1 = House('h1', 'Cape')
h1.add_room(Room('Master Bedroom', 14, 21))
h1.add_room(Room('Living Room', 18, 20))
h1.add_room(Room('Kitchen', 12, 16))
h1.add_room(Room('Office', 12, 12))
h2 = House('h2', 'Ranch')
h2.add_room(Room('Master Bedroom', 14, 21))
h2.add_room(Room('Living Room', 18, 20))
h2.add_room(Room('Kitchen', 12, 16))
h3 = House('h3', 'Split')
h3.add_room(Room('Master Bedroom', 14, 21))
h3.add_room(Room('Living Room', 18, 20))
h3.add_room(Room('Office', 12, 16))
h3.add_room(Room('Kitchen', 15, 17))
houses = [h1, h2, h3]
print('Is h1 bigger than h2?', h1 > h2) # prints True
print('Is h2 smaller than h3?', h2 < h3) # prints True
print('Is h2 greater than or equal to h1?', h2 >= h1) # Prints False
print('Which one is biggest?', max(houses)) # Prints 'h3: 1101-
↪square-foot Split'
print('Which is smallest?', min(houses)) # Prints 'h2: 846-square-
↪foot Ranch'
```

èőÍèőž

åĚűåőđ `total_ordering` èčĚéěřåŹÍäź§æšąéĆčäźĹçěđçğŸãĂĆ
åőČåřśæŸřåőŽäźĽäžĘäÿĂäÿłäżŐæřŘäÿłæřŤè¿ČæŤřæŇĄæŰźæşŢåĹřæĽĂæIJĽéIJĂèęĄåőŽäźĽçŽĎåĚűäżŰ
æřŤåęĆä¡äåőŽäźĽäžĘ `__le__()` æŰźæşŢïijŇéĆčäźĹåőČåřśèćńçŤÍæİěæđĎåżžæĽĂæIJĽåĚűäżŰçŽĎéIJĂè
åőđéŹĚäÿŁåřśæŸřåIJÍçśżéĞŇéİćåČŘäÿŃéİćè£ŹæăůåőŽäźĽäžĘäÿĂäžŻçĽźæőŁæŰźæşŢïijŽ

```
class House:
    def __eq__(self, other):
        pass
    def __lt__(self, other):
        pass
    # Methods created by @total_ordering
    __le__ = lambda self, other: self < other or self == other
    __gt__ = lambda self, other: not (self < other or self == other)
    __ge__ = lambda self, other: not (self < other)
    __ne__ = lambda self, other: not self == other
```

å¡ŞçĎűïijŇä¡äèĞłåůśåŐżåĘŹäź§å¿ĹåőźæŸŞïijŇä¡ĘæŸřä¡£çŤÍ `@total_ordering`
åŘřäžěçőĂåŇŰäżčçăĄïijŇä¡ŢäźŘèĂŇäÿ■äÿžåŚćãĂĆ

10.25 8.25 åĹŻåżžçijŞå■Ÿåőđä¿Ń

éŮőéćŸ

åIJÍåĹŻåżžäÿĂäÿłçśżçŽĎåŕźèśąæŮűïijŇåęĆæđIJäźŃåĽ■ä¡£çŤÍåŘŇæăůåŘĆæŢřåĹŻåżžè£Ğè£Źäÿłåŕźèś
ä¡äæČşè£ŤåŻđåőČçŽĎçijŞå■ŸåijŢçŤÍãĂĆ

èğčåĘşæŰźæąĹ

è£ŹçğŒéĂŽåÿÿæŸřåŻääÿžä¡äåÿŇæIJŻçŻÿåŘŇåŘĆæŢřåĹŻåżžçŽĎåŕźèśąæŮűå■Ţä¿ŃçŽĎãĂĆ
åIJÍå¿Ĺåď ŽåžŞäÿ■éČ¡æIJĽåőđéŹĚçŽĎä¿Ńå■ŘïijŇæřŤåęĆ `logging`
æÍąåİŮïijŇä¡£çŤÍçŻÿåŘŇçŽĎåŘ■çğřåĹŻåżžçŽĎ `logger` åőđä¿Ńæřÿè£IJåŘłæIJĽäÿĂäÿłãĂĆä¿ŃåęĆïijŽ

```
>>> import logging
>>> a = logging.getLogger('foo')
>>> b = logging.getLogger('bar')
>>> a is b
False
>>> c = logging.getLogger('foo')
>>> a is c
True
>>>
```

äÿžäžĘè¿¿åĹřè£ŹæăůçŽĎæŢĹæđIJïijŇä¡äéIJĂèęĄä¡£çŤÍäÿĂäÿłåŠŇçśżæIJňèžńåĹĘåijĂçŽĎåůěåŐĆåĞ

```
# The class in question
class Spam:
    def __init__(self, name):
        self.name = name

# Caching support
import weakref
_spam_cache = weakref.WeakValueDictionary()
def get_spam(name):
    if name not in _spam_cache:
        s = Spam(name)
        _spam_cache[name] = s
    else:
        s = _spam_cache[name]
    return s
```

çĎűåŘŐåĄŽäÿĂäÿłæţŃèŕŢïijŇä¡ăäijŽåŔŚçŐřèů§äźŃåĽ’■éĆčäÿłæŮěå£ŮåŕźèśąçŽĎåĹŻåżžèąŇäÿžæŸŕ

```
>>> a = get_spam('foo')
>>> b = get_spam('bar')
>>> a is b
False
>>> c = get_spam('foo')
>>> a is c
True
>>>
```

èőĺèőž

çijŰåĘŹäÿĂäÿłåůěåŐĆåĞ¡æŢřæİěä£őæŤźæŹőéĂŽçŽĎåőđä¿ŃåĹŻåżžèąŇäÿžéĂŽåÿÿæŸŕäÿĂäÿłæŕŤè¿ä¡ĘæŸŕæĹŚäżňè£ŸèČ¡åŘęæĽ’¿åĹřæŻť’äijŸéŻĚçŽĎèğčåĘşæŰźæąĹåŚćïij§

ä¿ŃåęĆïijŇä¡ăåŔŕèČ¡äijŽèĂČèŹŚéĞ■æŰřåőŽäźĽ’çśżçŽĎ `__new__()` æŰźæşŢïijŇářśåČŔäÿŃéİćè£ŹæăůïijŽ

```
# Note: This code doesn't quite work
import weakref

class Spam:
    _spam_cache = weakref.WeakValueDictionary()
    def __new__(cls, name):
        if name in cls._spam_cache:
            return cls._spam_cache[name]
        else:
            self = super().__new__(cls)
            cls._spam_cache[name] = self
            return self
    def __init__(self, name):
```

(continues on next page)

(continued from previous page)

```
        print('Initializing Spam')
        self.name = name
```

åĹİçIJŃèţůæİěåě¡åČŔåŔŕäžěè¿¿åĹřéćĎæIJ§æŢĹæđIJïijŇä¡ĘæŸŕéŮőéćŸæŸŕ
`__init__()` æŕŔæňąéČ¡äijŽèćńèřČçŤĺïijŇäÿ■çőąè£Źäÿłåőđä¿ŃæŸŕåŘęèćńçijŞå■ŸäžĘãĂĆä¿ŃåęĆïijŽ

```
>>> s = Spam('Dave')
Initializing Spam
>>> t = Spam('Dave')
Initializing Spam
>>> s is t
True
>>>
```

è£ŹäÿłæĹŰèőÿäÿ■æŸŕä¡äæČşèęAçŽĎæŢĹæđIJïijŇåŻäæ■ď è£Źçğ■æŰźæşŢåźűäÿ■åŔŕåŔŰãĂĆ

äÿŁéİćæĹŚäžňä¡£çŤĺåĹřäžĘåijśåijŢçŤĺèőąæŢřïijŇåŕźäžŐåđČåIJ¿åŻđæŤűæİěèőšæŸŕå¿ĹæIJL'åÿőåŁľçŽ
å¡ŞæĹŚäžňä£ÍæŇĄåőđä¿ŃçijŞå■ŸæŮűïijŇä¡ääŔŕèČ¡åŔłæČşåIJĺçĺŃåžŘäÿ■ä¡£çŤĺåĹřäőČäžňæŮűæL'■ä£Íå■
äÿĂäÿł `WeakValueDictionary` åőđä¿ŃåŔłäijŽä£Íå■ŸéĆčäžŻåIJĺåĚűäőČåIJřæŰźè£ŸåIJĺèćńä¡£çŤĺçŽĎå
åŘęåĹŹçŽĎèŕÍïijŇåŔłèęĄåőđä¿Ńäÿ■åĘ■èćńä¡£çŤĺäžĘïijŇäőČåřśäžŐå■ŮåĚÿäÿ■èćńçğżéŹď äžĘãĂĆèğĆåŕ§ä

```
>>> a = get_spam('foo')
>>> b = get_spam('bar')
>>> c = get_spam('foo')
>>> list(_spam_cache)
['foo', 'bar']
>>> del a
>>> del c
>>> list(_spam_cache)
['bar']
>>> del b
>>> list(_spam_cache)
[]
>>>
```

åŕźäžŐåď ğéČĺåĹęçĺŃåžŘèĂŇåůšïijŇè£ŹéĞŇäżčçăĄåůšçżŘåď §çŤĺäžĘãĂĆäÿ■è£Ğè£ŸæŸŕæIJL'äÿĂäż

éęŰåĚĹæŸŕè£ŹéĞŇä¡£çŤĺåĹřäžĘäÿĂäÿłåĚĺåśĂåŘŸéĞŘïijŇåźűäÿŤåůěåŐĆåĞ¡æŢřèů§çśżæŤ¿åIJĺäÿĂä

```
import weakref

class CachedSpamManager:
    def __init__(self):
        self._cache = weakref.WeakValueDictionary()

    def get_spam(self, name):
        if name not in self._cache:
            s = Spam(name)
            self._cache[name] = s
        else:
```

(continues on next page)

(continued from previous page)

```
            s = self._cache[name]
        return s

    def clear(self):
            self._cache.clear()

class Spam:
    manager = CachedSpamManager()
    def __init__(self, name):
        self.name = name

    def get_spam(name):
        return Spam.manager.get_spam(name)
```

è£ŹæăůçŽĎèŕÍäžčçăĄæŻť æÿĚæŹřïijŇåźűäÿŤäź§æŻť çĄţæť żïijŇæĹŚäžňåŘŕäžěåćđåŁăæŻť åď ŽçŽĎçij

è£ŸæIJĽäÿĂçĆźářśæŸŕïijŇæĹŚäžňæŽť éIJšäžĘçśżçŽĎåőđä¿ŃåŇŰçżŹçŤĺæĹůïijŇçŤĺæĹůå¿ĹåőźæŸ Ş

```
>>> a = Spam('foo')
>>> b = Spam('foo')
>>> a is b
False
>>>
```

æIJĽåĞăçğ■æŰźåijŔåŘŕäžěéŸšæ■ćçŤĺæĹůè£ŹæăůåĄŽïijŇçňňäÿĂäÿłæŸŕåřĘçśżçŽĎåŘ■å■Ůä£őæŤźäÿ çňňäžŇçğ■åřśæŸŕèőľè£ŹäÿłçśżçŽĎ `__init__()` æŰźæşŢæŁŻåĞžäÿĂäÿłåijĆåÿÿïijŇèőľåőČäÿ■èČ¡èćńåĹ

```
class Spam:
    def __init__(self, *args, **kwargs):
        raise RuntimeError("Can't instantiate directly")

    # Alternate constructor
    @classmethod
    def _new(cls, name):
        self = cls.__new__(cls)
        self.name = name
```

çĎűåŘŐä£őæŤźçijŞå■ŸçőąçŘĘåŹĺäžčçăĄïijŇä¡£çŤĺ `Spam._new()` æĺěåĹŻåżžåőđä¿ŃïijŇèĂŇäÿ■æŸŕçŻť æŐěěřČçŤĺ `Spam()` æđĎéĂäåĞ¡æŢřïijŽ

```
# ------------------------æIJĂåŘŐçŽĎä£őæ■čæŰźæąĹ---------------------
↪---
class CachedSpamManager2:
    def __init__(self):
        self._cache = weakref.WeakValueDictionary()

    def get_spam(self, name):
        if name not in self._cache:
            temp = Spam3._new(name)  # Modified creation
            self._cache[name] = temp
```

(continues on next page)

(continued from previous page)

```
        else:
            temp = self._cache[name]
        return temp

    def clear(self):
            self._cache.clear()

class Spam3:
    def __init__(self, *args, **kwargs):
        raise RuntimeError("Can't instantiate directly")

    # Alternate constructor
    @classmethod
    def _new(cls, name):
        self = cls.__new__(cls)
        self.name = name
        return self
```

æIJĂåŘŐè£ŹæăůçŽĎæŰźæąĹåřśåůšçżŘèűşåď§åě¡äžĘãĂĆ
çijŞå■ŸåŠŇåĚűäžŰæđĎéĂăæĺąåijŘè£ŸåŘŕäžěä¡£çŤÍ9.13åřŘèŁĆäÿ■çŽĎåĚČçśżåőđçŐřçŽĎæŻť äijŸéŻĚäÿ

11 çňňäźĺçńäïijŽåĚČçijŰçÍŃ

è¡ŕäżűåijĂåŘŚéćĘå§§äÿ■æIJĂçżŘåĚÿçŽĎåŘčåď'ť çęĚåřśæŸŕâĂIJdonâĂŹt repeat yourselfâĂĺãĂĆ äź§åřśæŸŕèŕť'ïijŇäżżä¡ŢæŮűåĂŹå¡Şä¡äçŽĎçÍŃåžŘäÿ■å■ŸåIJÍéńŸåžęéĞ■åď■(æŁŰèĂĚæŸŕéĂ
åIJÍPythonå¡Şäÿ■ïijŇéĂŽåÿÿéČ¡åŘŕäżěéĂŽè£ĞåĚČçijŰçÍŃæĺěèğčåĘşè£ŹçśżéŮőéćŸãĂĆ
çőĂèĂŇèÍĂäźŃïijŇåĚČçijŰçÍŃåřśæŸŕåĚşäžŐåŁŻåżżæŞ■ä¡IJæžŘäżčçăĄ(æŕŤåęĆä£őæŤźãĂĄçŤ§æŁŘæŁŰ
äÿżèęĄæŁĂæIJŕæŸŕä¡£çŤÍèčĚéěřåŹÍãĂĄçśżèčĚéěřåŹÍåŠŇåĚČçśżãĂĆäÿ■è£Ğè£ŸæIJL'äÿĂäžŻåĚűäżŰæŁĂ
åŇĚæŇňç■¿åŘ■åŕźèśąãĂĄä¡£çŤÍ exec() æL'ğèąŇäżčçăĄäżěåŘŁåŕźåĘĚéČÍåĞ¡æŢřåŠŇçśżçŽĎåŘ■åřĎæŁ
æIJňçńäçŽĎäÿżèęĄçŻőçŽĎæŸŕåŘŚåď'ğåőűäżŃçż■è£ŹäžŻåĚČçijŰçÍŃæŁĂæIJŕïijŇåźűäÿŤçżŹåĞžåőđä¿Ńæ

Contents:

11.1 9.1 åIJÍåĞ¡æŢřäÿŁæůżåŁäåŇĚèčĚåŹÍ

éŮőéćŸ

ä¡äæČşåIJÍåĞ¡æŢřäÿŁæůżåŁääÿĂäÿłåŇĚèčĚåŹÍïijŇåćđåŁäéćÍåď'ŰçŽĎæŞ■ä¡IJåď'ĎçŘĘ(æŕŤåęĆæŮěå

èğčåĘşæŰźæąĹ

åęĆæđIJä¡äæČşä¡£çŤÍéćÍåď'ŰçŽĎäżčçăĄåŇĚèčĚäÿĂäÿłåĞ¡æŢřïijŇåŕźäżěåőŽäźL'äÿĂäÿłèčĚéěřåŹÍåĞ

```
import time
from functools import wraps

def timethis(func):
    '''
    Decorator that reports the execution time.
    '''
    @wraps(func)
    def wrapper(*args, **kwargs):
        start = time.time()
        result = func(*args, **kwargs)
        end = time.time()
        print(func.__name__, end-start)
        return result
    return wrapper
```

äÿŃéĮćæŸŕä¡£çŤĺèčĚéěřåŹĺçŽĎä¿Ńå■ŘïijŽ

```
>>> @timethis
... def countdown(n):
...     '''
...     Counts down
...     '''
...     while n > 0:
...         n -= 1
...
>>> countdown(100000)
countdown 0.008917808532714844
>>> countdown(10000000)
countdown 0.87188299392912
>>>
```

èőĺèőž

äÿĂäÿłèčĚéěřåŹĺåřśæŸŕäÿĂäÿłåĞ¡æŢřïijŃåőČæŐěåŘŮäÿĂäÿłåĞ¡æŢřä¡IJäÿžåŘĆæŢřåźűè£ŤåŻđäÿĂäÿ
å¡Şä¡ăåČŘäÿŃéĮćè£ŹæăůåĘŹïijŽ

```
@timethis
def countdown(n):
    pass
```

èů§åČŘäÿŃéĮćè£ŹæăůåĘŹåĚűåőđæŢĹæđIJæŸŕäÿĂæăůçŽĎïijŽ

```
def countdown(n):
    pass
countdown = timethis(countdown)
```

éąžä¿£èŕť äÿĂäÿŃïijŃåĘĚç¡őçŽĎèčĚéěřåŹĺæŕŤåęĆ `@staticmethod`, `@classmethod`,`@property` åŐ§çŘĘäź§æŸŕäÿĂæăůçŽĎãĂĆ

ä¿ŃåęĆïijŇäÿŃéİćè£Źäÿď äÿłäžčçăĄçĽĞæőţæŸŕç■Ľäžůç ŽĎïijŽ

```
class A:
    @classmethod
    def method(cls):
        pass

class B:
    # Equivalent definition of a class method
    def method(cls):
        pass
    method = classmethod(method)
```

åIJÍäÿŁéİćçŽĎ `wrapper()` åĞ¡æŢŕäÿ■ïijŇ èčĚéěřåŹÍåĘĚéČÍåőŽäźĽäžĘäÿĂäÿłä¡£çŤÍ `*args` åŠŇ `**kwargs` æİěæŐěåŘŮäżżæĎŘåŘĆæŢřçŽĎåĞ¡æŢřãĂĆ
åIJÍè£ŹäÿłåĞ¡æŢřéĞŇéİćèřČçŤÍäžĘåŐ§åğŃåĞ¡æŢřåźűåřĘåĚűçżŞæđIJè£ŤåŻđïijŇäÿ■è£Ğä¡äè£ŸåŘŕäżěæůż
çĎűåŘŐè£ŹäÿłæŰřçŽĎåĞ¡æŢřåŇĚèčĚåŹÍèćńä¡IJäÿžçżŞæđIJè£ŤåŻđæİěäžčæŻ£åŐ§åğŃåĞ¡æŢřãĂĆ

éIJĂèęĄåijžèřČçŽĎæŸŕèčĚéěřåŹÍåźűäÿ■äijŽä£őæŤźåŐ§åğŃåĞ¡æŢřçŽĎåŘĆæŢřç■¿åŘ■äżěåŘŁè£ŤåŻ
ä¡£çŤÍ `*args` åŠŇ `**kwargs` çŻőçŽĎåřśæŸŕçąőä£İäżżä¡ŢåŘĆæŢřéČ¡èČ¡éĂĆçŤÍãĂĆ
èĂŇè£ŤåŻđçżŞæđIJåĂijå§žæIJňéČ¡æŸŕèřČçŤÍåŐ§åğŃåĞ¡æŢř `func(*args, **kwargs)` çŽĎè£ŤåŻđçżŞæđIJïijŇåĚűäÿ■funcåřśæŸŕåŐ§åğŃåĞ¡æŢřãĂĆ

åĹŽåijĂåğŃå■ęäźžèčĚéěřåŹÍçŽĎæŮűåĂŹïijŇäijŽä¡£çŤÍäÿĂäžŻçőĂå■ŢçŽĎä¿Ńå■Řæİěèŕť æŸŐïijŇæŕ
äÿ■è£ĞåőđéŹĚåIJžæŹŕä¡£çŤÍæŮűïijŇè£ŸæŸŕæIJL'äÿĂäžŻçżĘèŁĆéŮőécŸèęĄæşÍæĎŘçŽĎãĂĆ
æŕŤåęĆäÿŁéİcä¡£çŤÍ `@wraps(func)` æşÍèğčæŸŕå¿ĹéĞ■èęĄçŽĎïijŇ
åőČèČ¡ä£ÍçŢŹåŐ§åğŃåĞ¡æŢřçŽĎåĚČæŢřæ■ő(äÿŃäÿĂåřŘèŁĆäijŽèőšåĹř)ïijŇæŰřæĽŃçżŘåÿÿäijŽå£¡çŢěè
æŐěäÿŃæİěçŽĎåĞääÿłåřŘèŁĆæĹŚäžňäijŽæŻť åŁäæůśåĚěçŽĎèőšèğčèčĚéěřåŹÍåĞ¡æŢřçŽĎçżĘèŁĆéŮőécŸ

11.2 9.2 åĹŻåżżèčĚéěřåŹÍæŮűä£ÍçŢŹåĞ¡æŢřåĚČä£ąæĄŕ

éŮőécŸ

ä¡ăåĘźäžĘäÿĂäÿłèčĚéěřåŹÍä¡IJçŤÍåIJÍæ§ŘäÿłåĞ¡æŢřäÿŁïijŇä¡ĘæŸŕè£ŹäÿłåĞ¡æŢřçŽĎéĞ■èęĄçŽĎåĚ

èğčåĘşæŰźæąĹ

äżżä¡ŢæŮűåĂŹä¡ăőŽäźĽèčĚéěřåŹÍçŽĎæŮűåĂŹïijŇéČ¡ăžŤèŕěä¡£çŤÍ `functools` ăžŞäÿ■çŽĎ `@wraps` èčĚéěřåŹÍæİěæşÍèğčăžŢăśĆåŇĚèčĚåĞ¡æŢřãĂĆä¿ŃåęĆïijŽ

```
import time
from functools import wraps
def timethis(func):
    '''
    Decorator that reports the execution time.
    '''
    @wraps(func)
    def wrapper(*args, **kwargs):
```

(continues on next page)

(continued from previous page)

```
        start = time.time()
        result = func(*args, **kwargs)
        end = time.time()
        print(func.__name__, end-start)
        return result
    return wrapper
```

äÿŃéİćæĹŚäżňä¡£çŤÍè£ŹäÿłèćńåŃĚèčĚåŘŐçŽĎåĞ¡æŢřåźűæčĂæ§ěåőČçŽĎåĚČä£ąæĄŕïijŽ

```
>>> @timethis
... def countdown(n):
...     '''
...     Counts down
...     '''
...     while n > 0:
...         n -= 1
...
>>> countdown(100000)
countdown 0.008917808532714844
>>> countdown.__name__
'countdown'
>>> countdown.__doc__
'\n\tCounts down\n\t'
>>> countdown.__annotations__
{'n': <class 'int'>}
>>>
```

èőÍèőž

åIJÍçijŰåĘŹèčĚéěřåŹÍçŽĎæŮűåĂŹåď■åĹűåĚČä£ąæĄŕæŸŕäÿĂäÿłéİđåÿÿéĞ■èęĄçŽĎéČÍåĹĘãĂĆåęĆæ
`@wraps` ïijŃ éĆčäźĹä¡ääijŽåŘŚçŐřèćńèčĚéěřåĞ¡æŢřäÿćåď śäžĘæĽĂæIJĽæIJĽçŤÍçŽĎä£ąæĄŕãĂĆæŕŤåęĆ
`@wraps` åŘŐçŽĎæŢĹæđIJæŸŕäÿŃéİćè£ŹæäůçŽĎïijŽ

```
>>> countdown.__name__
'wrapper'
>>> countdown.__doc__
>>> countdown.__annotations__
{}
>>>
```

`@wraps` æIJĽäÿĂäÿłéĞ■èęĄçĽźå¿ĄæŸŕåőČèČ¡èőĺď ä¡äéĂŽè£ĞåśđæĂğ `__wrapped__` çŻť æŐěèő£éŮőèćńåŃĚèčĚåĞ¡æŢřãĂĆä¿ŃåęĆ:

```
>>> countdown.__wrapped__(100000)
>>>
```

`__wrapped__` åśđæĂğè£ŸèČ¡èőĺď èćńèčĚéěřåĞ¡æŢřæ■čçąőæŽť éIJšåžŢåśĆçŽĎåŘĆæŢřç■¿åŘ■ä£ąæĄ

```
>>> from inspect import signature
>>> print(signature(countdown))
(n:int)
>>>
```

äÿĂäÿłå¿ĹæŹőéĄ■çŽĎéŮőécŸæŸŕæĂŐæăůèőľèčĚéěřåŹĺåŐżçŻť'æŐěåď■åĹűåŐ§åğŃåĞ¡æŢřçŽĎåŔĆ
åęĆæđIJæČşèĞłåůśæĽŃåŁĺåőđçŐřçŽĎèřĺéIJĂèęĄåĄŽåď'ğéĞŔçŽĎåůěä¡IJïijŇæIJĂåě¡åřśçőĂå■ŢçŽĎä¡£çŤ
`@wraps` èčĚéěřåŹĺãĂĆ éĂŽè£ĞåžŢåśĆçŽĎ `__wrapped__`
åśđæĂğèő£éŮőåĹřåĞ¡æŢřç■¿åŘ■ä£ąæĄŕãĂĆæŻť'åď'ŽåĚşäžŐç■¿åŘ■çŽĎåĘĚåőźåŔŕäżěåŔĆèĂČ9.16åřŔèŁ

11.3 9.3 èğčéŹď'äÿĂäÿłèčĚéěřåŹĺ

éŮőécŸ

äÿĂäÿłèčĚéěřåŹĺåůšçżŔä¡IJçŤĺåIJĺäÿĂäÿłåĞ¡æŢřäÿŁïijŇä¡ăæČşæŠď'éŤĂåőČïijŇçŻť'æŐěèő£éŮőåŐ§åğ

èğčåĘşæŰźæąĹ

åĄĞèő¿èčĚéěřåŹĺæŸŕéĂŽè£Ğ `@wraps` (åŔĆèĂČ9.2åřŔèŁĆ)æĺěåőđçŐřçŽĎïijŇéĆčäźĹä¡ăåŔŕäżěéĂŽè
`__wrapped__` åśđæĂğæĺěèő£éŮőåŐ§åğŃåĞ¡æŢřïijŽ

```
>>> @somedecorator
>>> def add(x, y):
...     return x + y
...
>>> orig_add = add.__wrapped__
>>> orig_add(3, 4)
7
>>>
```

èőĺèőž

çŻť'æŐěèő£éŮőæIJłåŇĚèčĚçŽĎåŐ§åğŃåĞ¡æŢřåIJĺèřČèŕŢãĂĄåĘĚçIJĄåŠŇåĚűäżŰåĞ¡æŢřæŞ■ä¡IJæŮ
ä¡ĘæŸŕæĹŚäżňè£ŹéĞŇçŽĎæŰźæąĹäżĚäżĚéĂĆçŤĺäžŐåIJĺåŇĚèčĚåŹĺäÿ■æ■čçąőä¡£çŤĺäžĘ
`@wraps` æĹŰèĂĚçŻť'æŐěèő¿ç¡őäžĘ `__wrapped__` åśđæĂğçŽĎæČĚåĘţãĂĆ

åęĆæđIJæIJL'åď'ŽäÿłåŇĚèčĚåŹĺïijŇéĆčäźĹèő£éŮő `__wrapped__`
åśđæĂğçŽĎèąŇäÿžæŸŕäÿ■åŔŕéćĎç§ěçŽĎïijŇåžŤèŕěéĄ£åĚ■è£ŹæăůåĄžãĂĆ
åIJĺPython3.3äÿ■ïijŇåőČäijŽçŢěè£ĞæL'ĂæIJL'çŽĎåŇĚèčĚåśĆïijŇæŕŤåęĆïijŇåĄĞåęĆä¡ăæIJL'åęĆäÿŃçŽĎ

```
from functools import wraps

def decorator1(func):
    @wraps(func)
    def wrapper(*args, **kwargs):
```

(continues on next page)

(continued from previous page)

```
        print('Decorator 1')
        return func(*args, **kwargs)
    return wrapper

def decorator2(func):
    @wraps(func)
    def wrapper(*args, **kwargs):
        print('Decorator 2')
        return func(*args, **kwargs)
    return wrapper

@decorator1
@decorator2
def add(x, y):
    return x + y
```

äÿŃéİćæĹŚäżňåIJÍPython3.3äÿŃæţŃèŕŢïijŽ

```
>>> add(2, 3)
Decorator 1
Decorator 2
5
>>> add.__wrapped__(2, 3)
5
>>>
```

äÿŃéİćæĹŚäżňåIJÍPython3.4äÿŃæţŃèŕŢïijŽ

```
>>> add(2, 3)
Decorator 1
Decorator 2
5
>>> add.__wrapped__(2, 3)
Decorator 2
5
>>>
```

æIJĂåŘŐèęĄèŕť'çŽĎæŸŕïijŃåźűäÿ■æŸŕæL'ĂæIJL'çŽĎèčĚéěřåŹÍéČ¡ä¡£çŤÍäžĘ `@wraps` ïijŃåŽäæ■d'è£ŹéĞŃçŽĎæŰźæąĹåźűäÿ■åĚÍéČÍéĂĆçŤÍãĂĆçL'źåĹńçŽĎïijŃåĘĚç¡őçŽĎèčĚéěřåŹÍ `@staticmethod` åŠŃ `@classmethod` åŕśæšąæIJL'éĄţå¿łè£ŹäÿłçžęåőŽ (åőČäżňæŁŁåŐ§åğŃåĞ¡æŢřå■ŸåĆÍåIJÍåśđæĂğ `__func__` äÿ■)ãĂĆ

11.4 9.4 åőŽäźL'äÿĂäÿłåÿęåŔĆæŢřçŽĎèčĚéěřåŹÍ

éŮőécŸ

ä¡äæČşåőŽäźL'äÿĂäÿłåŔŕäżěæŐěåŔŮåŔĆæŢřçŽĎèčĚéěřåŹÍ

èğčåĘşæŰźæąĹ

æĹŚäżňçŤĺäÿĂäÿłä¿Ńå■ŘèŕęçżĘéŸŘè£řäÿŃæŐěåŘŮåŘĆæŢřçŽĎåď ĎçŘĘè£ĞçÍŃãĂĆ
åĄĞèő¿ä¡äæČşåĘŹäÿĂäÿłèčĚéěřåŹÍïijŃçżŹåĞ¡æŢřæůżåŁäæŮěå£ŮåŁ§èČ¡ïijŃåŘŃæŮűåĚĄèőÿçŤÍæĹůæŃĆ
äÿŃéÍćæŸŕè£ŹäÿłèčĚéěřåŹÍçŽĎåőŽäźĽ'åŠŃä¡£çŤÍçď žä¿ŃïijŽ

```
from functools import wraps
import logging

def logged(level, name=None, message=None):
    """
    Add logging to a function. level is the logging
    level, name is the logger name, and message is the
    log message. If name and message aren't specified,
    they default to the function's module and name.
    """
    def decorate(func):
        logname = name if name else func.__module__
        log = logging.getLogger(logname)
        logmsg = message if message else func.__name__

        @wraps(func)
        def wrapper(*args, **kwargs):
            log.log(level, logmsg)
            return func(*args, **kwargs)
        return wrapper
    return decorate

# Example use
@logged(logging.DEBUG)
def add(x, y):
    return x + y

@logged(logging.CRITICAL, 'example')
def spam():
    print('Spam!')
```

åĹÍçIJŃèţůæÍěïijŃè£Źçğ■åőđçŐřçIJŃäÿŁåŐżå¿Ĺåď■æÍĆïijŃä¡ĘæŸŕæăÿå£ČæĂÍæČşå¿ĹçőĂå■ŢãĂĆ
æIJĂåď ŰåśĆçŽĎåĞ¡æŢř `logged()` æŐěåŘŮåŘĆæŢřåźűåřĘåőČäżňä¡IJçŤĺåIJĺåĘĚéČĺçŽĎèčĚéěřåŹĺåĞ¡æŢ
åĘĚåśĆçŽĎåĞ¡æŢř `decorate()` æŐěåŘŮäÿĂäÿłåĞ¡æŢřä¡IJäÿżåŘĆæŢřïijŃçĎűåŘŐåIJĺåĞ¡æŢřäÿŁéÍćæŤ
è£ŹéĞŃçŽĎåĚşéŤőçĆźæŸŕåŃĚèčĚåŹĺæŸŕåŘŕäżěä¡£çŤĺäijäéĂŠçżŹ `logged()`
çŽĎåŘĆæŢřçŽĎãĂĆ

èőĺèőž

åőŽäźĽäÿĂäÿłæŐěåŘŮåŘĆæŢřçŽĎåŃĚèčĚåŹĺçIJŃäÿŁåŐżæŕŤè¿Čåď■æÍĆäÿżèęĄæŸŕåŽääÿžåžŢåśĆ

```
@decorator(x, y, z)
def func(a, b):
```

(continues on next page)

(continued from previous page)

```
    pass
```

èčĚéěřåŹĺåď ĎçŘĘè£ĞçÍŃèů§äÿŃéİćçŽĎèřČçŤÍæŸŕç■Ľ æŢĹçŽĎ;

```
def func(a, b):
    pass
func = decorator(x, y, z)(func)
```

decorator(x, y, z) çŽĎè£ŤåŻđçżŞæđIJå£ĚéążæŸŕäÿĂäÿłåŘŕèřČçŤÍåŕźèśąïijŃåőČæŐěåŘŮäÿĂ åŘŕäżěåŘĆèĂČ9.7åřŘèŁĆäÿ■åŘęåď ŰäÿĂäÿłåŘŕæŐěåŘŮåŘĆæŢřçŽĎåŃĚèčĚåŹĺä¿Ńå■ŘãĂĆ

11.5 9.5 åŘŕèĞłåőŽäźĽåśđæĂğçŽĎèčĚéěřåŹĺ

éŮőéćŸ

ä¡ăæČşåĘŹäÿĂäÿłèčĚéěřåŹĺæİěåŃĚèčĚäÿĂäÿłåĞ¡æŢřïijŃåźűäÿŤåĚĄèőÿçŤÍæĹůæŘŘä¿ŻåŘĆæŢřåIJÍè

èğčåĘşæŰźæąĹ

åijŢåĚěäÿĂäÿłèő£éŮőåĞ¡æŢřïijŃä¡£çŤÍ nonlocal æİěä£őæŤźåĘĚéČĺåŘŸéĞŘãĂĆ çĎűåŘŐè£Źäÿłèő£éŮőåĞ¡æŢřèćńä¡IJäÿžäÿĂäÿłåśđæĂğèţŃåĂijçżŹåŃĚèčĚåĞ¡æŢřãĂĆ

```
from functools import wraps, partial
import logging
# Utility decorator to attach a function as an attribute of obj
def attach_wrapper(obj, func=None):
    if func is None:
        return partial(attach_wrapper, obj)
    setattr(obj, func.__name__, func)
    return func

def logged(level, name=None, message=None):
    '''
    Add logging to a function. level is the logging
    level, name is the logger name, and message is the
    log message. If name and message aren't specified,
    they default to the function's module and name.
    '''
    def decorate(func):
        logname = name if name else func.__module__
        log = logging.getLogger(logname)
        logmsg = message if message else func.__name__

        @wraps(func)
        def wrapper(*args, **kwargs):
            log.log(level, logmsg)
```

(continues on next page)

(continued from previous page)

```
            return func(*args, **kwargs)

        # Attach setter functions
        @attach_wrapper(wrapper)
        def set_level(newlevel):
            nonlocal level
            level = newlevel

        @attach_wrapper(wrapper)
        def set_message(newmsg):
            nonlocal logmsg
            logmsg = newmsg

        return wrapper

    return decorate

# Example use
@logged(logging.DEBUG)
def add(x, y):
    return x + y

@logged(logging.CRITICAL, 'example')
def spam():
    print('Spam!')
```

äÿŃéİćæŸŕäžďäžŠçŐŕåćČäÿŃçŽĎä¡£çŤĺä¿Ńå■ŘïijŽ

```
>>> import logging
>>> logging.basicConfig(level=logging.DEBUG)
>>> add(2, 3)
DEBUG:__main__:add
5
>>> # Change the log message
>>> add.set_message('Add called')
>>> add(2, 3)
DEBUG:__main__:Add called
5
>>> # Change the log level
>>> add.set_level(logging.WARNING)
>>> add(2, 3)
WARNING:__main__:Add called
5
>>>
```

èőÍèőž

è£ŹäÿĂåřŔèŁĆçŽĎåĚşéŤőçĆźåIJÍäžŐèő£éŮőåĞ¡æŢř(åęĆ `set_message()` åŠŇ `set_level()`)ïijŇåőČäżňèćńä¡IJäÿžåśđæĂğèţŃçżŹåŇĚèčĚåŹÍãĂĆ æŕŔäÿłèő£éŮőåĞ¡æŢřåĚĄèőÿä¡£çŤÍ `nonlocal` æİěä£őæŤźåĞ¡æŢřåĘĚéČÍçŽĎåŔŸéĞŔãĂĆ

è£ŸæIJL'äÿĂäÿłäžd'äžžåŘČæČŁçŽĎåIJřæŰźæŸŕèő£éŮőåĞ¡æŢřäijŽåIJÍåd'ŽåśĆèčĚéěřåŹÍéŮt'äijäæŠ■ `@functools.wraps` æşÍèğč)ãĂĆ ä¿ŃåęĆïijŇåĄĞèő¿ä¡ääijŢåĚěåŔęåd'ŰäÿĂäÿłèčĚéěřåŹÍïijŇæŕŤåęĆ9.2å `@timethis` ïijŇåČŔäÿŃéİćè£ŹæăůïijŽ

```
@timethis
@logged(logging.DEBUG)
def countdown(n):
    while n > 0:
        n -= 1
```

ä¡ääijŽåŔŚçŐřèő£éŮőåĞ¡æŢřä¿ÍæŮğæIJL'æŢĹïijŽ

```
>>> countdown(10000000)
DEBUG:__main__:countdown
countdown 0.8198461532592773
>>> countdown.set_level(logging.WARNING)
>>> countdown.set_message("Counting down to zero")
>>> countdown(10000000)
WARNING:__main__:Counting down to zero
countdown 0.8225970268249512
>>>
```

ä¡äè£ŸäijŽåŔŚçŐřå■şä¡£èčĚéěřåŹÍåČŔäÿŃéİćè£ŹæăůäżèçŻÿåŔ■çŽĎæŰźåŘŚæŐŠæŤ¿ïijŇæŢĹæđIJäź

```
@logged(logging.DEBUG)
@timethis
def countdown(n):
    while n > 0:
        n -= 1
```

è£ŸèČ¡éĂŽè£Ğä¡£çŤÍlambdaèąÍè¿¿åijŔäžčçăĄæÍěèőľèő£éŮőåĞ¡æŢřçŽĎè£ŤåŻđäÿ■åŘŇçŽĎèő¿åőŽåź

```
@attach_wrapper(wrapper)
def get_level():
    return level

# Alternative
wrapper.get_level = lambda: level
```

äÿĂäÿłæŕŤè¿ČéŽ¿çŘĚèğčçŽĎåIJřæŰźåřśæŸŕåŕźäžŐèő£éŮőåĞ¡æŢřçŽĎéęŰæňąä¡£çŤÍãĂĆä¿ŃåęĆïijŇ

```
@wraps(func)
def wrapper(*args, **kwargs):
    wrapper.log.log(wrapper.level, wrapper.logmsg)
    return func(*args, **kwargs)
```

(continues on next page)

(continued from previous page)

```
# Attach adjustable attributes
wrapper.level = level
wrapper.logmsg = logmsg
wrapper.log = log
```

è£ŹäÿłæŰźæşŢäź§åŔŕèČ¡æ■čåÿÿåůëä¡IJïijŇä¡ĘåĽ■æŘŕæŸŕåőČå£ĚéążæŸŕæ IJĂåď ŰåśÇçŽĎèčĚéěřåŹ
åęĆæđIJåőČçŽĎäÿŁéİćè£ŸæIJĽåŔęåď ŰçŽĎèčĚéěřåŹÍ(æŕŤåęĆäÿŁéİćæŘŘåĹřçŽĎ
@timethis ä¿Ňå■Ř)ïijŇéĆčäźĹåőČäijŽéŽŘèŮŔåžŢåśĆåśđæĂğïijŇä¡£å¿Ůä£őæŤźåőČäżňæšąæIJĽäżżä¡Ţ
èĂŇéĂŽè£Ğä¡£çŤĺèő£éŮőåĞ¡æŢřåŕśèČ¡éĄ£åĚ■è£ŹæăůçŽĎåśĂéŹŘæĂğãĂĆ

æIJĂåŘŐæŘŘäÿĂçĆźïijŇè£ŹäÿĂåřŔèŁĆçŽĎæŰźæąĹäź§åŔŕäżěä¡IJäÿž9.9åřŔèŁĆäÿ■èčĚéěřåŹĺçśżçŽĎ

11.6 9.6 åÿęåŔŕéĂĽåŔĆæŢřçŽĎèčĚéěřåŹĺ

éŮőécŸ

ä¡ăæČşåĘŹäÿĂäÿłèčĚéěřåŹĺïijŇæŮćåŔŕäżěäÿ■äijăåŔĆæŢřçżŹåőČïijŇæŕŤåęĆ
@decorator ïijŇ äź§åŔŕäżěäijăéĂŠåŔŕéĂĽåŔĆæŢřçżŹåőČïijŇæŕŤåęĆ
@decorator(x,y,z) ãĂĆ

èğčåĘşæŰźæąĹ

äÿŃéİćæŸŕ9.5åřŔèŁĆäÿ■æŮěå£ŮèčĚéěřåŹĺçŽĎäÿĂäÿłä£őæŤźçĽĹæIJňïijŽ

```
from functools import wraps, partial
import logging

def logged(func=None, *, level=logging.DEBUG, name=None,␣
↪message=None):
    if func is None:
        return partial(logged, level=level, name=name,␣
↪message=message)

    logname = name if name else func.__module__
    log = logging.getLogger(logname)
    logmsg = message if message else func.__name__

    @wraps(func)
    def wrapper(*args, **kwargs):
        log.log(level, logmsg)
        return func(*args, **kwargs)

    return wrapper

# Example use
```

(continues on next page)

(continued from previous page)

```
@logged
def add(x, y):
    return x + y

@logged(level=logging.CRITICAL, name='example')
def spam():
    print('Spam!')
```

åŘŕäžěçIJŃåĹřïijŇ`@logged`èčĚéěřåŹĺåŘŕäžěåŘŇæŮűäÿ■åÿęåŘĆæŢřæĹŰåÿęåŘĆæŢřãĂĆ

èőĺèőž

è£ŹéĞŇæŘŘåĹřçŽĎè£ŹäÿłéŮőéćŸåřśæŸŕéĂŽåÿÿæĽĂèŕť çŽĎçijŰçĺŃäÿĂèĞť æĂğéŮőéćŸãĂĆ å¡ŞæĹŚäžňä¡£çŤĺèčĚéěřåŹĺçŽĎæŮűåĂŹïijŇåď ğéČĺåĹĘçĺŃåžŘåŚŸäźăæČŕäžĘèęĄäźĹäÿ■çżŹåőČäžňäijăéĂ åĚűåőđäžŐæŁĂæIJŕäÿŁæİěèőšïijŇæĹŚäžňåŘŕäžěåőŽäźĽäÿĂäÿłæĽĂæIJĽåŘĆæŢřéČ¡æŸŕåŘŕéĂĽçŽĎèčĚ

```
@logged()
def add(x, y):
    return x+y
```

ä¡ĘæŸŕïijŇè£Źçğ■åĘŹæşŢåźűäÿ■çňęåŘĹæĹŚäžňçŽĎäźăæČŕïijŇæIJĽæŮűåĂŹçĺŃåžŘåŚŸå£ŸèőŕåŁää è£ŹéĞŇæĹŚäžňåŘŚä¡ăåśŢçď žäžĘåęĆä¡ŢäžěäÿĂèĞť çŽĎçijŰçĺŃéčŐæäijæİěåŘŇæŮűæżąèűşæšąæIJĽæŃňå

äÿžäžĘçŘĘèğčäžčçăĄæŸŕåęĆä¡Ţåůěä¡IJçŽĎïijŇä¡äéIJĂèęĄéİđåÿÿçĘ§æĆĽèčĚéěřåŹĺæŸŕåęĆä¡Ţä¡IJçŤ åŕźäžŐäÿĂäÿłåČŘäÿŃéİćè£ŹæăůçŽĎçőĂå■ŢèčĚéěřåŹĺïijŽ

```
# Example use
@logged
def add(x, y):
    return x + y
```

è£ŹäÿłèřČçŤĺåžŘåĹŮèů§äÿŃéİćç■ĽäžůïijŽ

```
def add(x, y):
    return x + y

add = logged(add)
```

è£ŹæŮűåĂŹïijŇèćńèčĚéěřåĞ¡æŢřäijŽèćńå¡ŞåĄŽçňňäÿĂäÿłåŘĆæŢřçŻť æŐěäijăéĂŠçżŹ `logged`èčĚéěřåŹĺãĂĆ åŻăæ■ď ïijŇ`logged()` äÿ■çŽĎçňňäÿĂäÿłåŘĆæŢřåŕśæŸŕèćńåŇĚèčĚåĞ¡æŢřæIJňè

èĂŇåŕźäžŐäÿĂäÿłäÿŃéİćè£ŹæăůæIJĽåŘĆæŢřçŽĎèčĚéěřåŹĺïijŽ

```
@logged(level=logging.CRITICAL, name='example')
def spam():
    print('Spam!')
```

èřČçŤĺåžŘåĹŮèů§äÿŃéİćç■ĽäžůïijŽ

```
def spam():
    print('Spam!')
spam = logged(level=logging.CRITICAL, name='example')(spam)
```

åĹİåğŃèřČçŤÍ `logged()` åĞ¡æŢřæŮűïijŃèćńåŃĚèčĚåĞ¡æŢřåźűæšąæIJĽäijăéĂŠè£ŻæİěãĂĆ
åŻăæ■ď åIJÍèčĚéěřåŹÍåĘĚïijŃåőČå£ĚéążæŸřåŘřéĂĽçŽĎãĂĆè£ŹäÿłåŘ■è£ĞæİěäijŽè£ńä¡£åĚűäżŰåŘĆæŢřà
åźűäÿŤïijŃä¡Ęè£ŹäžŻåŘĆæŢřèćńäijăéĂŠè£ŻæİěåŘŐïijŃèčĚéěřåŹÍèęĄè£ŤåŻđäÿĂäÿłæŐěåŘŮäÿĂäÿłåĞ¡æŢ
äÿžäžĘè£ŹæăůåĄŽïijŃæĹŚäżňä¡£çŤÍäžĘäÿĂäÿłæŁĂåůğïijŃářśæŸřåĹľçŤÍ `functools.partial` ãĂĆ åőČäijŽè£ŤåŻđäÿĂäÿłæIJłåőŃåĚÍåĹİåğŃåŃŰçŽĎèĞłèžńïijŃéŹď äžĘèćńåŃĚèčĚåĞ¡æŢřåď
åŘřäżěåŘĆèĂČ7.8åřŘèŁĆèŐůåŘŰæŻť åď Ž `partial()` æŰźæşŢçŽĎç§ěèřĘãĂĆ

11.7 9.7 åĹľçŤÍèčĚéěřåŹÍåijžåĹűåĞ¡æŢřäÿŁçŽĎçśżåđŃæčĂæ§ě

éŮőéćŸ

ä¡IJäÿžæ§Řçğ■çijŰçÍŃèğĎçžęïijŃä¡ăæČşåIJÍåřźåĞ¡æŢřåŘĆæŢřè£ŻèąŃåijžåĹűçśżåđŃæčĂæ§ěãĂĆ

èğčåĘşæŰźæąĹ

åIJÍæijŤçď žåőđéŹĚäżčçăĄåĽ■ïijŃåĚĹèřť æŸŐæĹŚäżňçŽĎçŻőæăĞïijŽèČ¡åřźåĞ¡æŢřåŘĆæŢřçśżåđŃè

```
>>> @typeassert(int, int)
... def add(x, y):
...     return x + y
...
>>>
>>> add(2, 3)
5
>>> add(2, 'hello')
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "contract.py", line 33, in wrapper
TypeError: Argument y must be <class 'int'>
>>>
```

äÿŃéĺćæŸřä¡£çŤÍèčĚéěřåŹÍæŁĂæIJřæİěåőđçŐř `@typeassert` ïijŽ

```
from inspect import signature
from functools import wraps

def typeassert(*ty_args, **ty_kwargs):
    def decorate(func):
        # If in optimized mode, disable type checking
        if not __debug__:
            return func

        # Map function argument names to supplied types
```

(continues on next page)

(continued from previous page)

```
        sig = signature(func)
        bound_types = sig.bind_partial(*ty_args, **ty_kwargs).
↪arguments

        @wraps(func)
        def wrapper(*args, **kwargs):
            bound_values = sig.bind(*args, **kwargs)
            # Enforce type assertions across supplied arguments
            for name, value in bound_values.arguments.items():
                if name in bound_types:
                    if not isinstance(value, bound_types[name]):
                        raise TypeError(
                            'Argument {} must be {}'.format(name,␣
↪bound_types[name])
                            )
            return func(*args, **kwargs)
        return wrapper
    return decorate
```

åŘŕäżěçIJŃåĞžè£ŹäÿłèčĚéěřåŹĺéİđåÿÿçĄţæťżïijŇæŮćåŘŕäżěæŇĞåőŽæĽĂæIJĽåŘĆæŢřçśżåđŃïijŇäź
åźűäÿŤåŘŕäżěéĂŽè£Ğä¡■ç¡őæĹŰåĚşéŤőå■ŮæĺěæŇĞåőŽåŘĆæŢřçśżåđŃãĂĆäÿŃéİćæŸŕä¡£çŤĺçď žä¿ŃïijŽ

```
>>> @typeassert(int, z=int)
... def spam(x, y, z=42):
...     print(x, y, z)
...
>>> spam(1, 2, 3)
1 2 3
>>> spam(1, 'hello', 3)
1 hello 3
>>> spam(1, 'hello', 'world')
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
File "contract.py", line 33, in wrapper
TypeError: Argument z must be <class 'int'>
>>>
```

èőĺèőž

è£ŹèŁĆæŸŕéńŸçžğèčĚéěřåŹĺçď žä¿ŃïijŇåijŢåĚěäžĘå¿Ĺåď ŽéĞ■èęĄçŽĎæęĆå£ţãĂĆ

éęŰåĚĹïijŇèčĚéěřåŹĺåŘłäijŽåIJĺåĞ¡æŢřåőŽäźĽæŮűèćńèřČçŤĺäÿĂæňąãĂĆ
æIJĽæŮűåĂŹä¡ääŐżæŐĽèčĚéěřåŹĺçŽĎåŁ§èČ¡ïijŇéĆčäźĹä¡ääŘłéIJĂèęĄçőĂå■ŢçŽĎè£ŤåŻđèćńèčĚéěřåĞ¡
äÿŃéİćçŽĎäžčçăĄäÿ■ïijŇåęĆæđIJåĚĺåśĂåŘŸéĞŘãĂĂ__debug__
èćńèő¿ç¡őæĹŘäžĘFalse(å¡Şä¡ää¡£çŤĺ-OæĹŰ-OOåŘĆæŢřçŽĎäijŸåŇŰæĺąåijŘæĽğèąŇçĺŃåžŘæŮűï)ïijŇ
éĆčäźĹåřśçŻť æŐèè£ŤåŻđæIJłä£őæŤźè£ĞçŽĎåĞ¡æŢřæIJňèžńïijŽ

```
def decorate(func):
    # If in optimized mode, disable type checking
    if not __debug__:
        return func
```

åĚűæňąïijŇè£ŹéĞŇè£ŸåŕźèćńåŇĚèčĚåĞ¡æŢřçŽĎåŘĆæŢřç■¿åŘ■è£ŻèąŇäžĘæčĂæ§ěïijŇæĹŚäžňä¡£çŤ
inspect.signature() åĞ¡æŢřãĂĆ çőĂå■ŢæĺěèőšïijŇåőČè£ŘèąŇä¡ăæŘŘåŘŰäÿĂäÿłåŘŕèřČçŤĺåŕźèšą

```
>>> from inspect import signature
>>> def spam(x, y, z=42):
...     pass
...
>>> sig = signature(spam)
>>> print(sig)
(x, y, z=42)
>>> sig.parameters
mappingproxy(OrderedDict([('x', <Parameter at 0x10077a050 'x'>),
('y', <Parameter at 0x10077a158 'y'>), ('z', <Parameter at␣
↪0x10077a1b0 'z'>)]))
>>> sig.parameters['z'].name
'z'
>>> sig.parameters['z'].default
42
>>> sig.parameters['z'].kind
<_ParameterKind: 'POSITIONAL_OR_KEYWORD'>
>>>
```

èčĚéěřåŹĺçŽĎåijĂåğŇéČĺåĹĘïijŇæĹŚäžňä¡£çŤĺäžĘ bind_partial() æŰźæşŢæĺěæĽğèąŇäżŐæŇĞåőŽçśżåđŇåĹřåŘ■çğřçŽĎéČĺåĹĘçżŚåőŽãĂĆ äÿŃéĺćæŸŕä¿Ňå■ŘæijŤçď žïijŽ

```
>>> bound_types = sig.bind_partial(int,z=int)
>>> bound_types
<inspect.BoundArguments object at 0x10069bb50>
>>> bound_types.arguments
OrderedDict([('x', <class 'int'>), ('z', <class 'int'>)])
>>>
```

åIJĺè£ŹäÿłéČĺåĹĘçżŚåőŽäÿ■ïijŇä¡ăåŘŕäżěæşĺæĎŘåĹřçijžåď śçŽĎåŘĆæŢřèćńå£¡çŢěäžĘ(æŕŤåęĆåźűæš
äÿ■è£ĞæIJĂéĞ■èęĄçŽĎæŸŕåĹŻåżžäžĘäÿĂäÿłæIJĽåžŘå■ŮåĚÿ bound_types.
arguments ãĂĆ è£ŹäÿłåŮåĚÿäijŽåřĘåŘĆæŢřåŘ■äżěåĞ¡æŢřç■¿åŘ■äÿ■çŻÿåŘŇéąžåžŘæŸăåřĎåĹřæŇĞå
åIJĺæĹŚäžňçŽĎèčĚéěřåŹĺä¿Ňå■Řäÿ■ïijŇè£ŹäÿłæŸăåřĎåŇĚåŘńäžĘæĹŚäžňèęĄåijžåĹűæŇĞåőŽçŽĎçśżåđŇ

åIJĺèčĚéěřåŹĺåĹŻåżžçŽĎåőđéŹĚåŇĚèčĚåĞ¡æŢřäÿ■ä¡£çŤĺåĹřäžĘ
sig.bind() æŰźæşŢãĂĆ bind() èů§ bind_partial()
çśżäijijïijŇä¡ĘæŸŕåőČäÿ■åĚĄèöÿå£¡çŢěäżżä¡ŢåŘĆæŢřãĂĆåŻăæ■ď æIJĽäžĘäÿŃéĺćçŽĎçżŞæđIJïijŽ

```
>>> bound_values = sig.bind(1, 2, 3)
>>> bound_values.arguments
OrderedDict([('x', 1), ('y', 2), ('z', 3)])
```

(continues on next page)

(continued from previous page)

```
>>>
```

ä¡£çŤÍè£ŹäÿłæŸăářĎæĹŚäžňåŘŕäžěå¿Ĺè¡żæİ¿çŽĎåőđçŐřæĹŚäžňçŽĎåijžåĹűçśżåđŃæčĂæ§ěïijŽ

```
>>> for name, value in bound_values.arguments.items():
...     if name in bound_types.arguments:
...         if not isinstance(value, bound_types.arguments[name]):
...             raise TypeError()
...
>>>
```

äÿ■è£Ğè£ŹäÿłæŰźæąĹè£ŸæIJĽçĆźářŘçŚŢçŰţïijŇåőČářźäžŐæIJĽéżŸèőď˙åĂijçŽĎåŘĆæŢřåźűäÿ■éĂĆ
æŕŤåęĆäÿŃéİćçŽĎäžčçăĄåŘŕäžěæ■čåÿÿåůěä¡IJïijŇåř¡çőąitemsçŽĎçśżåđŃæŸŕéŤŹèŕŕçŽĎïijŽ

```
>>> @typeassert(int, list)
... def bar(x, items=None):
...     if items is None:
...         items = []
...     items.append(x)
...     return items
>>> bar(2)
[2]
>>> bar(2,3)
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "contract.py", line 33, in wrapper
TypeError: Argument items must be <class 'list'>
>>> bar(4, [1, 2, 3])
[1, 2, 3, 4]
>>>
```

æIJĂåŘŐäÿĂçĆźæŸŕåĚşäžŐéĂĆçŤÍèčĚéěřåŹÍåŘĆæŢřåŠŇåĞ¡æŢřæşÍèğčäźŃéŮť˙çŽĎäžĽèőžãĂĆ
ä¿ŃåęĆïijŇäÿžäżĂäźĹäÿ■åČŘäÿŃéİćè£ŹæăůåĘŹäÿĂäÿłèčĚéěřåŹÍæİěæ§ěæĽ¿åĞ¡æŢřäÿ■çŽĎæşÍèğčåŚćïij§

```
@typeassert
def spam(x:int, y, z:int = 42):
    print(x,y,z)
```

äÿĂäÿłåŘŕèČ¡çŽĎåŐ§åŻăæŸŕåęĆæđIJä¡£çŤÍäžĘåĞ¡æŢřåŘĆæŢřæşÍèğčïijŇéĆčäźĹåřśèćńéŹŘåĹűäžĘãA
åęĆæđIJæşÍèğčèćńçŤÍæİěåĄŽçśżåđŃæčĂæ§ěåřśäÿ■èČ¡åĄŽåĚűäżŰäžŃæČĚäžĘãĂĆèĂŃäÿŤ
`@typeassert` äÿ■èČ¡åĘ■çŤÍäžŐä¡£çŤÍæşÍèğčåĄŽåĚűäżŰäžŃæČĚçŽĎåĞ¡æŢřäžĘãĂĆ
èĂŃä¡£çŤÍäÿŁéİćçŽĎèčĚéěřåŹÍåŘĆæŢřçĄţæť˙żæĂğåď˙ğåď˙ŽäžĘïijŇäź§æŻť˙åŁäéĂŽçŤÍãĂĆ

åŘŕäžěåIJÍPEP 362äžěåŘŁ `inspect` æĺąåİŮäÿ■æĽ¿åĹřæŻť˙åď˙ŽåĚşäžŐåĞ¡æŢřåŘĆæŢřåźèśąçŽĎä£ąa

11.8 9.8 åřĘèčĚéěřåŹÍåőŽäźĽäÿžçśżçŽĎäÿĂéČÍåĹĘ

éŮőéćŸ

ä¡ăæČşåIJÍçśżäÿ■åőŽäźĽèčĚéěřåŹÍïijŇåźűåřĘåĚűä¡IJçŤÍåIJÍåĚűäżŰåĞ¡æŢřæĹŰæŰźæşŢäÿŁãĂĆ

èğčåĘşæŰźæąĹ

åIJÍçśżéĞŇéİćåőŽäźĽèčĚéěřåŹÍå¿ĹçőĂå■ŢïijŇä¡ĘæŸŕä¡äéęŰåĚĹèęĄçąőèőď åőČçŽĎä¡£çŤÍæŰźåijŔãĂäÿŃéİćæĹŚäżňçŤÍä¿Ńå■ŘæİěéŸŘè£řåőČäżňçŽĎäÿ■åŘŇïijŽ

```
from functools import wraps

class A:
    # Decorator as an instance method
    def decorator1(self, func):
        @wraps(func)
        def wrapper(*args, **kwargs):
            print('Decorator 1')
            return func(*args, **kwargs)
        return wrapper

    # Decorator as a class method
    @classmethod
    def decorator2(cls, func):
        @wraps(func)
        def wrapper(*args, **kwargs):
            print('Decorator 2')
            return func(*args, **kwargs)
        return wrapper
```

äÿŃéİćæŸŕäÿĂä¡£çŤÍä¿Ńå■ŘïijŽ

```
# As an instance method
a = A()
@a.decorator1
def spam():
    pass
# As a class method
@A.decorator2
def grok():
    pass
```

äżŤçżĘèğĆåŕ§åŘŕäżèåŘŚçŐŕäÿĂäÿłæŸŕåőđä¿ŃèřČçŤÍïijŇäÿĂäÿłæŸŕçśżèřČçŤÍãĂĆ

èőÍèőž

åIJÍçśżäÿ■åőŽäźL'èčĚéěřåŹÍåĹiçIJŃäÿŁåŐżåě¡åČŘå¿ĹåěĞæĂłïijŇä¡ĘæŸřåIJÍæăĞåĞĘåžŞäÿ■æIJL'å¿Ĺ
çL'źåĹńçŽĎïijŇ`@property` èčĚéěřåŹÍåőđéŹĚäÿŁæŸřäÿĂäÿłçśżïijŇåőČéĞŇéİćåőŽäźL'äžĘäÿL'äÿłæŰźæş
`getter()`, `setter()`, `deleter()`, æŕŘäÿĂäÿłæŰźæşŢéČ¡æŸřäÿĂäÿłèčĚéěřåŹÍãĂĆä¿ŃåęĆïijŽ

```
class Person:
    # Create a property instance
    first_name = property()

    # Apply decorator methods
    @first_name.getter
    def first_name(self):
        return self._first_name

    @first_name.setter
    def first_name(self, value):
        if not isinstance(value, str):
            raise TypeError('Expected a string')
        self._first_name = value
```

åőČäÿžäżĂäźĹèęĄè£ŹäźĹåőŽäźL'çŽĎäÿżèęĄåŐ§åŽäæŸřåŘĎçğ■äÿ■åŘŇçŽĎèčĚéěřåŹÍæŰźæşŢäijŽåĹ
`property` åőđä¿ŃäÿŁæŞ■ä¡IJåőČçŽĎçŁűæĂĄãĂĆ åŽäæ■ď'ïijŇäżżä¡ŢæŮűåĂŹåŘłèęĄä¡äçćřåĹřéIJĂèęĄå

åIJÍçśżäÿ■åőŽäźL'èčĚéěřåŹÍæIJL'äÿłéŽ¿çŘĘèğčçŽĎåIJřæŰźåřśæŸřåřźäžŐéćÍåď'ŰåŘĆæŢř
`self` æĹŰ `cls` çŽĎæ■čçąőä¡£çŤÍãĂĆ åř¡çőąæIJĂåď'ŰåśĆçŽĎèčĚéěřåŹÍåĞ¡æŢřæŕŤåęĆ
`decorator1()` æĹŰ `decorator2()` éIJĂèęĄæŘŘä¿ŻäÿĂäÿł `self`
æĹŰ `cls` åŘĆæŢřïijŇ ä¡ĘæŸřåIJÍäÿď'äÿłèčĚéěřåŹÍåĘĚéČÍèćńåĹŻåżžçŽĎ
`wrapper()` åĞ¡æŢřåźűäÿ■éIJĂèęĄåŇĚåŘńè£ŹäÿŁ `self` åŘĆæŢřãĂĆ
ä¡ääŤřäÿĂéIJĂèęĄè£ŹäÿłåŘĆæŢřæŸřåIJÍä¡äçąőåőđèęĄèő£éŮőåŇĚèčĚåŹÍäÿ■è£Źäÿłåőđä¿ŃçŽĎæ§ŘäżŻéČ

åŕźäžŐçśżéĞŇéİćåőŽäźL'çŽĎåŇĚèčĚåŹÍè£ŸæIJL'äÿĂçĆźæŕŤè¿ČéŽ¿çŘĘèğčïijŇåřśæŸřåIJÍæűL'åŘŁåÍ
ä¿ŃåęĆïijŇåĄĞèő¿ä¡äæČşèőľ'åIJÍAäÿ■åőŽäźL'çŽĎèčĚéěřåŹÍä¡IJçŤÍåIJÍå■ŘçśżBäÿ■ãĂĆä¡äéIJĂèęĄåČŘäÿŃ

```
class B(A):
    @A.decorator2
    def bar(self):
        pass
```

äź§åřśæŸřèřť'ïijŇèčĚéěřåŹÍèęĄèćńåőŽäźL'æĹŘçśżæŰźæşŢåźűäÿŤä¡äåå£ĚéążæŸ¿åijŘçŽĎä¡£çŤÍçĹűçśż
ä¡ääÿ■èČ¡ä¡£çŤÍ `@B.decorator2` ïijŇåŽääÿžåIJÍæŰźæşŢåőŽäźL'æŮűïijŇè£ŹäÿłçśżBè£ŸæšąæIJL'èćńåĹŽ

11.9 9.9 åřĘèčĚéěřåŹÍåőŽäźL'äÿžçśż

éŮőéćŸ

ä¡äæČşä¡£çŤÍäÿĂäÿłèčĚéěřåŹÍåŐżåŇĚèčĚåĞ¡æŢřïijŇä¡ĘæŸřåÿŇæIJŻè£ŤåŽđäÿĂäÿłåŘŕèřČçŤÍçŽĎåőć
ä¡äéIJĂèęĄőľ'ä¡äçŽĎèčĚéěřåŹÍåŘŕäżèåŘŇæŮűåůěä¡IJåIJÍçśżåőŽäźL'çŽĎåĘĚéČÍåŠŇåď'ŰéČÍãĂĆ

èğčåĘşæŰźæąĹ

äÿžäžĘåřĘèčĚéěřåŹÍåőŽäźĽæĹŘäÿĂäÿłåőđä¿ŃïijŇä¡ăéIJĂèęĄçąőä£ÍåőČåőđçŐřäžĘ
`__call__()` åŠŇ `__get__()` æŰźæşŢãĂĆ ä¿ŃåęĆïijŇäÿŃéÍćçŽĎäžčçăĄåőŽäźĽäžĘäÿĂäÿłçśżïijŇåőČå

```
import types
from functools import wraps

class Profiled:
    def __init__(self, func):
        wraps(func)(self)
        self.ncalls = 0

    def __call__(self, *args, **kwargs):
        self.ncalls += 1
        return self.__wrapped__(*args, **kwargs)

    def __get__(self, instance, cls):
        if instance is None:
            return self
        else:
            return types.MethodType(self, instance)
```

ä¡åăŘŕäžěåřĘåőČå¡ŞåĄŽäÿĂäÿłæŹőéĂŽçŽĎèčĚéěřåŹÍæÍěä¡£çŤÍïijŇåIJÍçśżéĞŇéÍćæĹŰåď ŰéÍćéČ¡åŘŕ

```
@Profiled
def add(x, y):
    return x + y

class Spam:
    @Profiled
    def bar(self, x):
        print(self, x)
```

åIJÍäžď äžŠçŐŕåćČäÿ■çŽĎä¡£çŤÍçď žä¿ŃïijŽ

```
>>> add(2, 3)
5
>>> add(4, 5)
9
>>> add.ncalls
2
>>> s = Spam()
>>> s.bar(1)
<__main__.Spam object at 0x10069e9d0> 1
>>> s.bar(2)
<__main__.Spam object at 0x10069e9d0> 2
>>> s.bar(3)
<__main__.Spam object at 0x10069e9d0> 3
>>> Spam.bar.ncalls
3
```

èőÍèőž

åřĘèčĚéěřåŹÍåőŽäźĽæĹŘçśżéĂŽåÿÿæŸŕå¿ÍçőĂå■ŢçŽĎãĂĆä¡ĘæŸŕè£ŹéĞŇè£ŸæŸŕæIJĽäÿĂäžŻçżĘ

éęŰåĚĹïijŇä¡£çŤÍ `functools.wraps()` åĞ¡æŢřçŽĎä¡IJçŤÍèů§äźŃåĽ■è£ŸæŸŕäÿĂæăůïijŇåřĘèćńå

åĚűæňąïijŇéĂŽåÿÿå¿ĹåőźæŸ$äijŽå£¡èğĘäÿŁéİćçŽĎ `__get__()` æŰźæşŢãĂĆåęĆæđIJä¡åå£¡çŢěåőČïijŇä£ÍæŇĄåĚűäżŰäžčçăĄäÿ■åŘŸåĘ■æňąè£ŘèąŇïijŇä¡ääijŽåŘŚçŐřå¡$ä¡åőŻèřČçŤÍèćńèčĚéěřåőđä¿ŃæŰźæşŢæŮűåĞžçŐřå¿ĹåěĞæĂłçŽĎéŮőécŸãĂĆä¿ŃåęĆï

```
>>> s = Spam()
>>> s.bar(3)
Traceback (most recent call last):
...
TypeError: bar() missing 1 required positional argument: 'x'
```

åĞžéŤŹåŐ§åŽäæŸŕå¡$æŰźæşŢåĞ¡æŢřåIJÍäÿĂäÿłçśżäÿ■èćńæ§ěæĽ¿æŮűïijŇåőČäžňçŽĎ `__get__()` æŰźæşŢä¿Íæ■őæŘŘè£řåŹÍå■ŘèőőèćńěřČçŤÍïijŇ åIJÍ8.9åřŘèŁĆåůšçżŘèőšè£řè£ĞæŘŘè£řåŹÍå■Řèőőäž$ãĂĆåIJÍè£ŹéĞŇïijŇ `__get__()` çŽĎçŻőçŽĎæŸŕåĹŽåżžäÿĂäÿłçżŚåőŽæŰźæşŢåŕźèśą (æIJĂçżĹäijŽçżŹè£ŹäÿłæŰźæşŢäijäéĂŠselfåŘĆæŢř)ã

```
>>> s = Spam()
>>> def grok(self, x):
...     pass
...
>>> grok.__get__(s, Spam)
<bound method Spam.grok of <__main__.Spam object at 0x100671e90>>
>>>
```

`__get__()` æŰźæşŢæŸŕäÿžäžĘçąőä£ĺçżŚåőŽæŰźæşŢåŕźèśąèČ¡èćńæ■čçąőçŽĎåĹŽåżžãĂĆ `type.MethodType()` æĽŃåŁĹåĹŽåżžäÿĂäÿłçżŚåőŽæŰźæşŢæĺěä¡£çŤĹãĂĆåŘłæIJĽå¡Ş$åőđä¿Ńèćńä¡£çŤ åęĆæđIJè£ŹäÿłæŰźæşŢæŸŕåIJĹçśżäÿŁéİćæĺěèő£éŮőïijŇ éĆčäźĹ `__get__()` äÿ■çŽĎinstanceåŘĆæŢřäijŽèćńèő¿ç¡őæĹŘNoneåźűçŽť'æŐěè£ŤåŽđ `Profiled` åőđä¿ŃæIJňèžňãĂĆ è£ŹæăůçŽĎèřĺæĹŚäžňåřśåŘřäžěæŘŘåŘŰåőČçŽĎ `ncalls` åśđæĂğäžĘãĂĆ

åęĆæđIJä¡äæČşéĄ£åĚ■äÿĂäžŻæůůäźśïijŇäź§åŘřäžěèĂČèŹŚåŘęåď'ŰäÿĂäÿłä¡£çŤĹéŮ■åŇĚåŠŇ `nonlocal` åŘŸéĞŘåőđçŐřçŽĎèčĚéěřåŹÍïijŇè£ŹäÿłåIJÍ9.5åřŘèŁĆæIJĽèőšåĹřãĂĆä¿ŃåęĆïijŽ

```
import types
from functools import wraps

def profiled(func):
    ncalls = 0
    @wraps(func)
    def wrapper(*args, **kwargs):
        nonlocal ncalls
        ncalls += 1
        return func(*args, **kwargs)
    wrapper.ncalls = lambda: ncalls
    return wrapper
```

(continues on next page)

(continued from previous page)

```
# Example
@profiled
def add(x, y):
    return x + y
```

è£ŹäÿłæŰźåijŘèů§äźŃåĽ■çŽĎæŢĹæđIJåĞäźŐäÿĂæăůïijŇéŹď äžĘåŕźäžŐ `ncalls` çŽĎèő£éŮőçŐřåIJÍæŸŕéĂŽè£ĞäÿĂäÿłèćńçżŚåőŽäÿžåśđæĂğçŽĎåĞ¡æŢřæİěåőđçŐřïijŇä¿ŃåęĆïijŽ

```
>>> add(2, 3)
5
>>> add(4, 5)
9
>>> add.ncalls()
2
>>>
```

11.10 9.10 äÿžçśżåŠŇéİŹæĂĄæŰźæşŢæŘŘä¿ŻèčĚéěřåŹÍ

éŮőéćŸ

ä¡ăæČşçżŹçśżæĹŰéİŹæĂĄæŰźæşŢæŘŘä¿ŻèčĚéěřåŹÍãĂĆ

èğčåĘşæŰźæąĹ

çżŹçśżæĹŰéİŹæĂĄæŰźæşŢæŘŘä¿ŻèčĚéěřåŹÍæŸŕå¿ĹçőĂå■ŢçŽĎïijŇäÿ■è£ĞèęĄçąőä£ÌèčĚéěřåŹÍåIJ `@classmethod` æĹŰ `@staticmethod` äźŃåĽ■ãĂĆä¿ŃåęĆïijŽ

```
import time
from functools import wraps

# A simple decorator
def timethis(func):
    @wraps(func)
    def wrapper(*args, **kwargs):
        start = time.time()
        r = func(*args, **kwargs)
        end = time.time()
        print(end-start)
        return r
    return wrapper

# Class illustrating application of the decorator to different␣
↪kinds of methods
class Spam:
    @timethis
    def instance_method(self, n):
```

(continues on next page)

(continued from previous page)

```
        print(self, n)
        while n > 0:
            n -= 1

    @classmethod
    @timethis
    def class_method(cls, n):
        print(cls, n)
        while n > 0:
            n -= 1

    @staticmethod
    @timethis
    def static_method(n):
        print(n)
        while n > 0:
            n -= 1
```

èčĚéěřåŘŐçŽĎçśżåŠŇéİŹæĂĄæŰźæşŢåŔŕæ■čåÿÿåůěä¡IJïijŇåŘłäÿ■è£ĞåćđåŁääžĘéćİåď ŰçŽĎèőąæŰ

```
>>> s = Spam()
>>> s.instance_method(1000000)
<__main__.Spam object at 0x1006a6050> 1000000
0.11817407608032227
>>> Spam.class_method(1000000)
<class '__main__.Spam'> 1000000
0.11334395408630371
>>> Spam.static_method(1000000)
1000000
0.11740279197692871
>>>
```

èőÍèőž

åęĆæđIJä¡ăæŁŁèčĚéěřåŹÍçŽĎéąžåžŘåĘŹéŤŹäžĘåřśäijŽåĞžéŤŹãĂĆä¿ŃåęĆïijŇåĄĞèő¿ä¡ăåČŘäÿŃéİć

```class Spam:
    @timethis
    @staticmethod
    def static_method(n):
        print(n)
        while n > 0:
            n -= 1
```

éĆčäźĹä¡ăèřČçŤÍè£ŹäÿłéİŹæĂĄæŰźæşŢæŮűåřśäijŽæŁěéŤŹïijŽ

```
>>> Spam.static_method(1000000)
Traceback (most recent call last):
```

(continues on next page)

(continued from previous page)

```
File "<stdin>", line 1, in <module>
File "timethis.py", line 6, in wrapper
start = time.time()
TypeError: 'staticmethod' object is not callable
>>>
```

éŮőécŸåIJÍäžŐ @classmethod åŠŇ @staticmethod åőđéŹĚäÿŁåźűäÿ■äijŽåĹŻåżžåŘŕçŻť'æŐěèřČçŤÍçŽĎåŕźèšąïijŇ
èĂŇæŸŕåĹŻåżžçL'źæőŁçŽĎæŘŘè£řåŹÍåŕźèšą(åŘĆèĂČ8.9åřŘèŁĆ)ãĂĆåŻäæ■đ'å¡Şä¡ăèŕŢçİĂåIJÍåĚűäżŰèč
çąőä£Ìè£Źçğ■èčĚéěřåŹÍåĞžçŐřåIJÍèčĚéěřåŹÍéŞ¿äÿ■çŽĎçňňäÿĂäÿłä¡■ç¡őåŘŕäżěä£őåđ'■è£ŹäÿłéŮőécŸãĂĆ

å¡ŞæĹŚäżňåIJÍæŁ¡èšąå§žçśżäÿ■åőŽäźL'çśżæŰźæşŢåŠŇéİŹæĂĄæŰźæşŢ(åŘĆèĂČ8.12åřŘèŁĆ)æŮűïijŇ
ä¿ŇåęĆïijŇåęĆæđIJä¡ăæČşåőŽäźL'äÿĂäÿłæŁ¡èšąçśżæŰźæşŢïijŇåŘŕäżěä¡£çŤÍçśżäijijäÿŇéİćçŽĎäżčçăĄïijŽ

```
from abc import ABCMeta, abstractmethod
class A(metaclass=ABCMeta):
    @classmethod
    @abstractmethod
    def method(cls):
        pass
```

åIJÍè£Źæőţäżčçă Ąäÿ■ïijŇ@classmethod èů§ @abstractmethod
äÿd'èĂĚçŽĎéąžåžŘæŸŕæIJL'èőšçl'űçŽĎïijŇåęĆæđIJä¡ăèřČæ■ćåőČäżňçŽĎéąžåžŘåřśäijŽåĞžéŤŹãĂĆ

11.11 9.11 èčĚéěřåŹÍäÿžèćńåŇĚèčĚåĞ¡æŢřåćđåŁäåŘĆæŢř

éŮőécŸ

ä¡ăæČşåIJÍèčĚéěřåŹÍäÿ■çżŹèćńåŇĚèčĚåĞ¡æŢřåćđåŁäéćİåd'ŰçŽĎåŘĆæŢřïijŇä¡ĘæŸŕäÿ■èČ¡å¡śåŞ■è£Ź

èğčåĘşæŰźæąĹ

åŘŕäżěä¡£çŤÍåĚşéŤőå■ŮåŘĆæŢřæİěçżŹèćńåŇĚèčĚåĞ¡æŢřåćđåŁäéćİåd'ŰåŘĆæŢřãĂĆèĂČèŹŚäÿŃéİćç

```
from functools import wraps

def optional_debug(func):
    @wraps(func)
    def wrapper(*args, debug=False, **kwargs):
        if debug:
            print('Calling', func.__name__)
        return func(*args, **kwargs)

    return wrapper
```

```
>>> @optional_debug
... def spam(a,b,c):
...     print(a,b,c)
...
>>> spam(1,2,3)
1 2 3
>>> spam(1,2,3, debug=True)
Calling spam
1 2 3
>>>
```

èőÍèőž

éĂŽè£ĞèčĚéěřåŹÍæİěçżŹèćńåŇĚèčĚåĞ¡æŢřåćđåŁäåŔĆæŢřçŽĎåĄŽæşŢåźűäÿ■åÿÿèğĄãĂĆ
åř¡çőąåęĆæ■ď'ïijŇæIJL'æŮűåĂŹåőČåŔŕäżěéĄ£åĚ■äÿĂäžŽéĞ■åď'■äżčçăĄãĂĆä¿ŃåęĆïijŇåęĆæđIJä¡ăæIJL'

```
def a(x, debug=False):
    if debug:
        print('Calling a')

def b(x, y, z, debug=False):
    if debug:
        print('Calling b')

def c(x, y, debug=False):
    if debug:
        print('Calling c')
```

éĆčäźĹä¡ăåŔŕäżěářĘåĚűéĞ■æđĎæĹŘè£ŹæăůïijŽ

```
from functools import wraps
import inspect

def optional_debug(func):
    if 'debug' in inspect.getargspec(func).args:
        raise TypeError('debug argument already defined')

    @wraps(func)
    def wrapper(*args, debug=False, **kwargs):
        if debug:
            print('Calling', func.__name__)
        return func(*args, **kwargs)
    return wrapper

@optional_debug
def a(x):
    pass
```

(continues on next page)

(continued from previous page)

```
@optional_debug
def b(x, y, z):
    pass

@optional_debug
def c(x, y):
    pass
```

è£ŹçğŃåőđçŐřæŰźæąĹäźŃæĽĂäżěèąŇå¿ŮéĂŽïijŇåIJÍäžŐåijžåĹűåĚşéŤőå■ŮåŘĆæŢřå¿ĹåőźæŸŞèćńa
`*args` åŠŇ `**kwargs` åŘĆæŢřçŽĎåĞ¡æŢřäÿ■ãĂĆ éĂŽè£Ğä¡£çŤĺåijžåĹűåĚşéŤőå■ŮåŘĆæŢřïijŇåőČèćńä
åźűäÿŤæŐěäÿŃæİěäżĚäżĚä¡£çŤĺåĽľä¡ŹçŽĎä¡■ç¡őåŠŇåĚşéŤőå■ŮåŘĆæŢřåŐżèřČçŤĺè£ŹäÿłåĞ¡æŢřæŮűïij
äź§åřśæŸŕèŕťïijŇåőČåźűäÿ■äijŽèćńçžşåĚěåĹř `**kwargs` äÿ■åŐżãĂĆ

è£ŸæIJĽäÿĂäÿłéŽ¿çĆźåřśæŸŕåęĆä¡ŢåŐżåď ĎçŘĘèćńæůżåŁäçŽĎåŘĆæŢřäÿŐèćńåŇĚèčĚåĞ¡æŢřåŘĆ
ä¿ŃåęĆïijŇåęĆæđIJèčĚéěřåŹÍ `@optional_debug` ä¡IJçŤĺåIJÍäÿĂäÿłåůšçżŘæŃěæIJĽäÿĂäÿł
`debug` åŘĆæŢřçŽĎåĞ¡æŢřäÿŁæŮűäijŽæIJĽéŮőéćŸãĂĆ è£ŹéĞŇæĹŚäžňåćđåŁäážĘäÿĂæ■ěåŘ■å■ŮæčĂæ

äÿŁéĺćçŽĎæŰźæąĹè£ŸåŘŕäżěæŻť åőŇç¿ŐäÿĂçĆźïijŇåŻääÿžçš¿æŸŐçŽĎçÍŃåžŘåŚŸåžŤèŕěåŘŚçŐřäž

```
>>> @optional_debug
... def add(x,y):
...     return x+y
...
>>> import inspect
>>> print(inspect.signature(add))
(x, y)
>>>
```

éĂŽè£ĞåęĆäÿŃçŽĎä£őæŤźïijŇåŘŕäżěèğčåĘşè£ŹäÿłéŮőéćŸïijŽ

```
from functools import wraps
import inspect

def optional_debug(func):
    if 'debug' in inspect.getargspec(func).args:
        raise TypeError('debug argument already defined')

    @wraps(func)
    def wrapper(*args, debug=False, **kwargs):
        if debug:
            print('Calling', func.__name__)
        return func(*args, **kwargs)

    sig = inspect.signature(func)
    parms = list(sig.parameters.values())
    parms.append(inspect.Parameter('debug',
                inspect.Parameter.KEYWORD_ONLY,
                default=False))
    wrapper.__signature__ = sig.replace(parameters=parms)
    return wrapper
```

éĂŽè£Ğè£ŹæăůçŽĎä£őæŤźïijŇåŇĚèčĚåŘŐçŽĎåĞ¡æŢřç■¿åŘ■åřśèČ¡æ■čçąőçŽĎæŸ¿çď'ž `debug` åŘĆæŢřçŽĎå■ŸåIJÍäžĘãĂĆä¿ŃåęĆïijŽ

```
>>> @optional_debug
... def add(x,y):
...     return x+y
...
>>> print(inspect.signature(add))
(x, y, *, debug=False)
>>> add(2,3)
5
>>>
```

åŘĆèĂČ9.16åřŘèŁĆèŐůåŘŰæŻť'åď'ŽåĚşäžŐåĞ¡æŢřç■¿åŘ■çŽĎä£ąæĄŕãĂĆ

11.12 9.12 ä¡£çŤÍèčĚéěřåŹÍæĽľåĚĚçśżçŽĎåŁ§èČ¡

éŮőéćŸ

ä¡ăæČşéĂŽè£ĞåŘ■çIJĄæŁŰèĂĚéĞ■åĘŹçśżåőŽäźĽçŽĎæ§ŘéČÍåŁĘæĺěä£őæŤźåőČçŽĎèąŇäÿžïijŇä¡E

èğčåĘşæŰźæąĹ

è£ŹçğŋæČĚåĘţåŘŕèČ¡æŸŕçśżèčĚéěřåŹÍæIJĂåě¡çŽĎä¡£çŤÍåIJžæŹŕäžĘãĂĆä¿ŃåęĆïijŇäÿŃéĺćæŸŕäÿĂä `__getattribute__` çŽĎçśżèčĚéěřåŹÍïijŇ åŘŕäżěæĽŞå■řæŮěå£ŮïijŽ

```
def log_getattribute(cls):
    # Get the original implementation
    orig_getattribute = cls.__getattribute__

    # Make a new definition
    def new_getattribute(self, name):
        print('getting:', name)
        return orig_getattribute(self, name)

    # Attach to the class and return
    cls.__getattribute__ = new_getattribute
    return cls

# Example use
@log_getattribute
class A:
    def __init__(self,x):
        self.x = x
    def spam(self):
        pass
```

äÿŃéĺćæŸŕä¡£çŤÍæŢĹæđIJïijŽ

```
>>> a = A(42)
>>> a.x
getting: x
42
>>> a.spam()
getting: spam
>>>
```

èőÍèőž

çśżèčĚéěřåŹÍéĂŽåÿÿåŘŕäżěä¡IJäÿžåĚűäżŰéńŸçžğæŁĂæIJŕæŕŤåęĆæůůåĚěæĹŰåĚČçśżçŽĎäÿĂçğ■éİđ
æŕŤåęĆïijŇäÿŁéİćçď žä¿Ňäÿ■çŽĎåŘęåď ŰäÿĂçğ■åőđçŐřä¡£çŤÍåĹřçżğæL'£ïijŽ

```
class LoggedGetattribute:
    def __getattribute__(self, name):
        print('getting:', name)
        return super().__getattribute__(name)

# Example:
class A(LoggedGetattribute):
    def __init__(self,x):
        self.x = x
    def spam(self):
        pass
```

è£Źçğ■æŰźæąĹäź§èąŇå¿ŮéĂŽïijŇä¡ĘæŸŕäÿžäžĘåŐżçŘĘèğčåőČïijŇä¡åářśå£Ěéążç§ěéĄŞæŰźæşŢèřČ
äżěåŘŁåĚűåőČ8.7åřŘèŁĆäżŃçż■çŽĎçżğæL'£ç§ěèřĘãĂĆ æ§Řçğ■çÍŃåžęäÿŁæİěèőšïijŇçśżèčĚéěřåŹÍæŰźæą
åŻääÿžäżŰåźűäÿ■ä¿İèţŰ `super()` åĞ¡æŢřãĂĆ

åęĆæđIJä¡äçşżæČşåIJÍäÿĂäÿłçśżäÿŁéİćä¡£çŤÍåď ŽäÿłçśżèčĚéěřåŹÍïijŇéĆčäźĹåřśéIJĂèęĄæşÍæĎŘäÿŃé
ä¿ŃåęĆïijŇäÿĂäÿłèčĚéěřåŹÍAäijŽåřĘåĚűèčĚéěřçŽĎæŰźæşŢåőŇæŢť æŻ£æ■ćæĹŘåŘęäÿĂçğ■åőđçŐřïijŇ
èĂŇåŘęäÿĂäÿłèčĚéěřåŹÍBåŘłæŸŕçőĂå■ŢçŽĎåIJÍåĚűèčĚéěřçŽĎæŰźæşŢäÿ■æůżåŁäçĆźéćİåď ŰéĂżè¿ŚãĂĆ
éĆčäźĹè£ŹæŮűåĂŹèčĚéěřåŹÍAåřśéIJĂèęĄæŤ¿åIJÍèčĚéěřåŹÍBçŽĎåL'■éİćãĂĆ

ä¡ăè£ŸåŘŕäżěåŻđéą¿äÿĂäÿŃ8.13åřŘèŁĆåŘęåď ŰäÿĂäÿłåĚşäžŐçśżèčĚéěřåŹÍçŽĎæIJL'çŤÍçŽĎä¿Ňå■Ř

11.13 9.13 ä¡£çŤÍåĚČçśżæŐğåĹűåőđä¿ŃçŽĎåĹŻåżž

éŮőéćŸ

ä¡ăæČşéĂŽè£ĞæŤźåŘŸåőđä¿ŃåĹŻåżžæŰźåijŘæİěåőđçŐřå■Ţä¿ŃãĂĄçijŞå■ŸæĹŰåĚűäżŰçśżäijijçŽĎ

èğčåĘşæŰźæąĹ

PythonçÍŃåžŘåŚŸéČ¡ç§ěéĄŞïijŇåęĆæđIJä¡ăáőŽäźL'äžĘäÿĂäÿłçśżïijŇåřśèČ¡åČŘåĞ¡æŢřäÿĂæăůçŽĎè

```
class Spam:
    def __init__(self, name):
        self.name = name

a = Spam('Guido')
b = Spam('Diana')
```

åęĆæđIJä¡ăæČşèĞłåőŽäźL'è£Źäÿłæ■ěéłď'ïijŇä¡ăåŘŕäżěåőŽäźL'äÿĂäÿłåĚČçśżåźűèĞłåůśåőđçŐř `__call__()` æŰźæşŢãĂĆ

äÿžäžĘæijŤçď'žïijŇåĄĞèő¿ä¡ăäÿ■æČşäżżä¡ŢäžžåĹŻåżžè£ŹäÿłçśżçŽĎåőđä¿ŃïijŽ

```
class NoInstances(type):
    def __call__(self, *args, **kwargs):
        raise TypeError("Can't instantiate directly")

# Example
class Spam(metaclass=NoInstances):
    @staticmethod
    def grok(x):
        print('Spam.grok')
```

è£ŹæăůçŽĎèŕÍïijŇçŤÍæĹůåŘłèČ¡èřČçŤÍè£ŹäÿłçśżçŽĎéÍŹæĂĄæŰźæşŢïijŇèĂŇäÿ■èČ¡ä¡£çŤÍéĂŽåÿÿç

```
>>> Spam.grok(42)
Spam.grok
>>> s = Spam()
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "example1.py", line 7, in __call__
        raise TypeError("Can't instantiate directly")
TypeError: Can't instantiate directly
>>>
```

çŐřåIJÍïijŇåĄĞåęĆä¡ăæČşåőđçŐřå■Ţä¿ŃæÍąåijŘïijĹåŘłèČ¡åĹŻåżžåŤŕäÿĂåőđä¿ŃçŽĎçśżïijL'ïijŇåőđçŐ

```
class Singleton(type):
    def __init__(self, *args, **kwargs):
        self.__instance = None
        super().__init__(*args, **kwargs)

    def __call__(self, *args, **kwargs):
        if self.__instance is None:
            self.__instance = super().__call__(*args, **kwargs)
            return self.__instance
        else:
            return self.__instance

# Example
class Spam(metaclass=Singleton):
```

(continues on next page)

(continued from previous page)

```
    def __init__(self):
        print('Creating Spam')
```

éĆčäźĹSpamçśżářśåŘłèČ¡åĹŻåżžåŤŕäÿĂçŽĎåőđä¿ŃäžĘïijŇæijŤçď žåęĆäÿŃïijŽ

```
>>> a = Spam()
Creating Spam
>>> b = Spam()
>>> a is b
True
>>> c = Spam()
>>> a is c
True
>>>
```

æIJĂåŘŐïijŇåĄĞèő¿ä¡ăæČşåĹŻåżž8.25ářŘèŁĆäÿ■éĆčæăůçŽĎçijŞå■ŸåőđÄ¿ŃãĂĆäÿŃéİćæĹŚäżňåŘŕä

```
import weakref

class Cached(type):
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.__cache = weakref.WeakValueDictionary()

    def __call__(self, *args):
        if args in self.__cache:
            return self.__cache[args]
        else:
            obj = super().__call__(*args)
            self.__cache[args] = obj
            return obj

# Example
class Spam(metaclass=Cached):
    def __init__(self, name):
        print('Creating Spam({!r})'.format(name))
        self.name = name
```

çĎűåŘŐæĹŚäź§æİĕæţŃèŕŢäÿĂäÿŃïijŽ

```
>>> a = Spam('Guido')
Creating Spam('Guido')
>>> b = Spam('Diana')
Creating Spam('Diana')
>>> c = Spam('Guido') # Cached
>>> a is b
False
>>> a is c # Cached value returned
True
>>>
```

èőÍèőž

åĹľçŤÍåĚČçśżåőđçŐřåď Žçğ■åőđä¿ŃåĹŻåżžæÍąåijŔéĂŽåÿÿèęĄæŕŤäÿ■ä¡£çŤÍåĚČçśżçŽĎæŰźåijŔäijŸ

åĄĞèő¿ä¡ääÿ■ä¡£çŤÍåĚČçśżïijŇä¡ääŔŕèČ¡éIJĂèęĄåřEçśżéŽŘèŮŔåIJÍæ§ŘäžŻåůěåŐĆåĞ¡æŢřåŘŐéİćãĂ
æŕŤåęĆäÿžäžEåőđçŐřäÿĂäÿłå■Ţä¿ŃïijŇä¡ää¡ääŔŕèČ¡äijŽåČŔäÿŃéİćè£ŹæăůåEŹïijŽ

```
class _Spam:
    def __init__(self):
        print('Creating Spam')

_spam_instance = None

def Spam():
    global _spam_instance

    if _spam_instance is not None:
        return _spam_instance
    else:
        _spam_instance = _Spam()
        return _spam_instance
```

ář¡çőąä¡£çŤÍåĚČçśżåŔŕèČ¡äijŽæűL'åŘŁåĹřæŕŤè¿ČéńŸçžğçĆźçŽĎæŁĂæIJŕïijŇä¡ĘæŸŕåőČçŽĎäżčçăĄ

æŻť'åď ŽåĚşäžŐåĹŻåżžçijŞå■Ÿåőđä¿ŃãĂĄåijśåijŢçŤÍç■L'åĘĚåőźïijŇèŕůåŘĆèĂČ8.25åřŘèŁĆãĂĆ

11.14 9.14 æ■ŢèŐůçśżçŽĎåśđæĂğåőŽäźL'éąžåžŘ

éŮőéćŸ

ä¡äæČşèĞłåŁÍèőřå¡ŢäÿĂäÿłçśżäÿ■åśđæĂğåŠŇæŰźæşŢåőŽäźL'çŽĎéąžåžŘïijŇ
çĎűåŘŐåŘŕäżěåĹľçŤÍåőČæİěåĄŽå¿Ĺåď ŽæŞ■ä¡IJïijĹæŕŤåęĆåžŘåĹŮåŇŰãĂĄæŸäářĎåĹřæŢřæ■őåžŞç■L'ç

èğčåĘşæŰźæąĹ

åĹľçŤÍåĚČçśżåŘŕäżěå¿Ĺåőźæ ŸŞçŽĎæ■ŢèŐůçśżçŽĎåőŽäźL'ä£ąæĄŕãĂĆäÿŃéİćæŸŕäÿĂäÿłä¿Ńå■ŘïijŇ

```
from collections import OrderedDict

# A set of descriptors for various types
class Typed:
    _expected_type = type(None)
    def __init__(self, name=None):
        self._name = name

    def __set__(self, instance, value):
        if not isinstance(value, self._expected_type):
            raise TypeError('Expected ' + str(self._expected_type))
```

(continues on next page)

(continued from previous page)

```
        instance.__dict__[self._name] = value

class Integer(Typed):
    _expected_type = int

class Float(Typed):
    _expected_type = float

class String(Typed):
    _expected_type = str

# Metaclass that uses an OrderedDict for class body
class OrderedMeta(type):
    def __new__(cls, clsname, bases, clsdict):
        d = dict(clsdict)
        order = []
        for name, value in clsdict.items():
            if isinstance(value, Typed):
                value._name = name
                order.append(name)
        d['_order'] = order
        return type.__new__(cls, clsname, bases, d)

    @classmethod
    def __prepare__(cls, clsname, bases):
        return OrderedDict()
```

åIJÍè£ŹäÿłåĚČçśżäÿ■ïijŇæL'ğèąŇçśżäÿżä¡ŞæŮűæŘŘè£řåŹÍçŽĎåőŽäźL'éążåžŘäijŽèćńäÿĂäÿł
OrderedDict``æ■ŢèŐůåĹřïijŇ çŤŞæĹŘçŽĎæIJL'åžŘåŘ■çğřäżŐå■ŮåĚÿäÿ■æŘŘåŘŰåĞžæİě
``_order äÿ■ãĂĆè£ŹæăůçŽĎèŕÍçśżäÿ■çŽĎæŰźæşŢåŘŕäżěéĂŽè£ĞåďŽçğ■æŰźåijŘæİěä¡£çŤÍåőČãĂĆ
ä¿ŇåęĆïijŇäÿŇéÍćæŸŕäÿĂäÿłçőĂå■ŢçŽĎçśżïijŇä¡£çŤÍè£ŹäÿłæŐŠåžŘå■ŮåĚÿæİěåőđçŐřåřĘäÿĂäÿłçśżåőđä

```
class Structure(metaclass=OrderedMeta):
    def as_csv(self):
        return ','.join(str(getattr(self,name)) for name in self._
↪order)

# Example use
class Stock(Structure):
    name = String()
    shares = Integer()
    price = Float()

    def __init__(self, name, shares, price):
        self.name = name
        self.shares = shares
        self.price = price
```

æĹŚäżňåIJÍäžď'äžŠåijŘçŐřåćČäÿ■æţŃèŕŢäÿĂäÿŃè£ŹäÿłStockçśżïijŽ

```
>>> s = Stock('GOOG',100,490.1)
>>> s.name
'GOOG'
>>> s.as_csv()
'GOOG,100,490.1'
>>> t = Stock('AAPL','a lot', 610.23)
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "dupmethod.py", line 34, in __init__
TypeError: shares expects <class 'int'>
>>>
```

èőÍèőž

æIJňèŁĆäÿĂäÿłåĚşéŤőçĆźåřśæŸŕOrderedMetaåĚČçśżäÿ■åőŽäźL'çŽĎ ‘‘ __prepare__()‘‘ æŰźæşŢãĂĆ è£ŹäÿłæŰźæşŢäijŽåIJĺåijĂåğŃåőŽäźL'çśżåŠŇåőČçŽĎçĹűçśżçŽĎæŮűåĂŹèćńæL'ğè
æĹŚäżňè£ŹéĞŇéĂŽè£Ğè£ŤåŻđäžĘäÿĂäÿłOrderedDictèĂŇäÿ■æŸŕäÿĂäÿłæŹőéĂŽçŽĎå■ŮåĚÿïijŇåŕŕäżěå

åęĆæđIJä¡ăæČşæđĎéĂăèĞłåůśçŽĎçśżå■ŮåĚÿåŕźèśąïijŇåŕŕäżěå¿Ĺåőźæ ŸŞçŽĎæL'ľåśŢè£ŹäÿłåŁ§èČ¡ã

```
from collections import OrderedDict

class NoDupOrderedDict(OrderedDict):
    def __init__(self, clsname):
        self.clsname = clsname
        super().__init__()
    def __setitem__(self, name, value):
        if name in self:
            raise TypeError('{} already defined in {}'.format(name,␣
↪self.clsname))
        super().__setitem__(name, value)

class OrderedMeta(type):
    def __new__(cls, clsname, bases, clsdict):
        d = dict(clsdict)
        d['_order'] = [name for name in clsdict if name[0] != '_']
        return type.__new__(cls, clsname, bases, d)

    @classmethod
    def __prepare__(cls, clsname, bases):
        return NoDupOrderedDict(clsname)
```

äÿŃéĬćæĹŚäżňæţŃèŕŢéĞ■åď■çŽĎåőŽäźL'äijŽåĞžçŐŕäżĂäźĹæČĚåĘţïijŽ

```
>>> class A(metaclass=OrderedMeta):
... def spam(self):
... pass
... def spam(self):
... pass
```

(continues on next page)

(continued from previous page)

```
...
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "<stdin>", line 4, in A
    File "dupmethod2.py", line 25, in __setitem__
        (name, self.clsname))
TypeError: spam already defined in A
>>>
```

æIJĂåŘŐè£ŸæIJL'äÿĂçĆźå¿ĹéĞ■èęĄïijŇåŕśæŸŕŕåIJÍ `__new__()` æŰźæşŢäÿ■åŕźäžŐåĚČçśżäÿ■èćńä£őæŤźå■ŮåĚÿçŽĎåď'ĎçŘĘãĂĆ åř¡çőąçśżä¡£çŤÍäžĘåŘęåď'ŰäÿĂäÿłå■ŮåĚÿæÍěåőŽäźL'ïijŇåIJÍæđĎéĂăæIJĂçżĹçŽĎ `class` åŕźèśąçŽĎæŮűåĂŹïijŇ æĹŚäžňäž■çĎűéIJĂèęĄåřĘè£Źäÿłå■ŮåĚÿè¡ňæ■ćäÿžäÿĂäÿłæ■čçąőçŽĎ `dict` åőđä¿ŃãĂĆ éĂŽè£ĞèŕåŘě `d = dict(clsdict)` æÍěåőŇæĹŘè£ŹäÿłæŢĹæđIJãĂĆ

åŕźäžŐå¿Ĺåď'ŽäžŤçŤÍçÍŃäžŘèĂŇåůšïijŇèČ¡åď'§æ■ŢèŐůçśżåőŽäźL'çŽĎéąžäžŘæŸŕäÿĂäÿłçIJŃäijijäÿ■ä¿ŃåęĆïijŇåIJÍåŕźèśąåĚşçşżæŸăåřĎäÿ■ïijŇæĹŚäžňéĂŽäÿÿäijŽçIJŃåĹřäÿŃéÍćè£Źçğ■æŰźåijŘåőŽäźL'çŽĎç

```
class Stock(Model):
    name = String()
    shares = Integer()
    price = Float()
```

åIJÍæąĘæđűäžŢäśĆïijŇæĹŚäžňå£Ěéążæ■ŢèŐůåőŽäźL'çŽĎéąžäžŘæÍěåřĘåŕźèśąæŸăåřĎåĹřåĚČçżĎæĹŮ `as_csv()` çŽĎåŁ§èČ¡ïijL'ãĂĆ è£ŹèŁĆæijŤçď'žçŽĎæŁĂæIJŕéÍđåÿÿçőĂå■ŢïijŇäźűäÿŤéĂŽäÿÿäijŽæŕŤåĚŮ

11.15 9.15 åőŽäźL'æIJL'åŘŕéĂL'åŘĆæŢřçŽĎåĚČçśż

éŮőécŸ

ä¡ăæČşåőŽäźL'äÿĂäÿłåĚČçśżïijŇåĚĄèőÿçśżåőŽäźL'æŮűæŘŘä¿ŻåŘŕéĂL'åŘĆæŢřïijŇè£ŹæăůåŘŕäžěæĆ

èğčåĘşæŰźæąĹ

åIJÍåőŽäźL'çśżçŽĎæŮűåĂŹïijŇPythonåĚĄèőÿæĹŚäžňä¡£çŤÍ "metaclass"åĚşéŤőå■ŮåŘĆæŢřæÍěæŇĞåőŽçL'źåőŽçŽĎåĚČçśżãĂĆ ä¿ŃåęĆä¡£çŤÍæŁ¡èśąå§žçśżïijŽ

```
from abc import ABCMeta, abstractmethod
class IStream(metaclass=ABCMeta):
    @abstractmethod
    def read(self, maxsize=None):
        pass

    @abstractmethod
```

(continues on next page)

(continued from previous page)

```
    def write(self, data):
        pass
```

çĎűèĂŇïijŇåIJÍèĞłåőŽäźL'åĚČçśżäÿ■æĹŚäžňè£ŸåŘŕäżěæŘŘä¿ŻåĚűäżŰçŽĎåĚşéŤőå■ŮåŘĆæŢřïijŇå

```
class Spam(metaclass=MyMeta, debug=True, synchronize=True):
    pass
```

äÿžäžĘä¡£åĚČçśżæŤŕæŇĄè£ŹäžŻåĚşéŤőå■ŮåŘĆæŢřïijŇä¡ăå£Ěéążçąőä£ĺåIJÍ
`__prepare__()` , `__new__()` åŠŇ `__init__()` æŰźæşŢäÿ■
éČ¡ä¡£çŤÍåijžåĹűåĚşéŤőå■ŮåŘĆæŢřãĂĆåřśåČŘäÿŃéİćè£ŹæăůïijŽ

```
class MyMeta(type):
    # Optional
    @classmethod
    def __prepare__(cls, name, bases, *, debug=False,␣
↪synchronize=False):
        # Custom processing
        pass
        return super().__prepare__(name, bases)

    # Required
    def __new__(cls, name, bases, ns, *, debug=False,␣
↪synchronize=False):
        # Custom processing
        pass
        return super().__new__(cls, name, bases, ns)

    # Required
    def __init__(self, name, bases, ns, *, debug=False,␣
↪synchronize=False):
        # Custom processing
        pass
        super().__init__(name, bases, ns)
```

èőÍèőž

çżŹäÿĂäÿłåĚČçśżæůżåŁäåŘŕéĂL'åĚşéŤőå■ŮåŘĆæŢřéIJĂèęĄä¡ăäőŇåĚÍåijĎæĞĆçśżåĹŻåżžçŽĎæL'Ăa
åŻääÿžè£ŹäžŻåŘĆæŢřäijŽèćńäijžéĂŠçżŹæŕŘäÿĂäÿłçŻÿåĚşçŽĎæŰźæşŢãĂĆ
`__prepare__()` æŰźæşŢåIJÍæL'ĂæIJL'çśżåőŽäźL'åijĂåğŃæL'ğèąŇåL'■éęŰåĚĹèćńèřĆçŤÍïijŇçŤÍæİèåĹŻ
éĂŽåÿÿæİèèőšïijŇè£ŹäÿłæŰźæşŢåŘłæŸŕçőĂå■ŢçŽĎè£ŤåŻđäÿĂäÿłå■ŮåĚÿæĹŰåĚűäżŰæŸäåřĎåŕźèśąãĂĆ
`__new__()` æŰźæşŢèćńçŤÍæİèåőđä¿ŃåŇŰæIJĂçżĹçŽĎçśżåŕźèśąãĂĆåőČåIJÍçśżçŽĎäÿżä¡Şèćńæ L'ğèąŇåő
`__init__()` æŰźæşŢæIJĂåŘŐèćńèřĆçŤÍïijŇçŤÍæİèæL'ğèąŇåĚűäżŰçŽĎäÿĂäżŽåĹIåğŃåŇŰåůěä¡IJãĂĆ

å¡ŞæĹŚäžňæđĎéĂäåĚČçśżçŽĎæŮűåĂŹïijŇéĂŽåÿÿåŘłéIJĂèęĄäőŽäźL'äÿĂäÿł
`__new__()` æĹŰ `__init__()` æŰźæşŢïijŇä¡Ęäÿ■æŸŕäÿď'äÿłéČ¡äőŽäźL'ãĂĆ
ä¡ĘæŸŕïijŇäęĆæđIJéIJĂèęĄæŐěåŘŮåĚűäżŰçŽĎåĚşéŤőå■ŮåŘĆæŢřçŽĎèŕİïijŇè£Źäÿď'äÿłæŰźæşŢåŕśèęĄå
éżŸèőď'çŽĎ `__prepare__()` æŰźæşŢæŐěåŘŮäżżæĎŘçŽĎåĚşéŤőå■ŮåŘĆæŢřïijŇä¡ĘæŸŕäijŽå£¡çŢěäőĆ

æĽĂäžěåŘłæIJĽå¡Ş è£ŹäžŻéćÍåď ŰçŽĎåŘĆæŢřåŘŕèČ¡äijŽå¡śåŞ■åĽřçśžåŚ¡åŘ■çľžéÚť çŽĎåĽŻåžžæŮűä¡ăa
`__prepare__()` æŰźæşŢãĂĆ

éĂŽè£Ğä¡£çŤĺåijžåĽűåĚşéŤőå■ŮåŘĆæŢřïijŇåIJÍçśžçŽĎåĽŻåžžè£ĞçÍŃäÿ■æĽŚäžňå£ĚéąžéĂŽè£ĞåĚş

ä¡£çŤĺåĚşéŤőå■ŮåŘĆæŢřéĚ■ç¡őäÿĂäÿłåĚČçśžè£ŸåŘŕäžěèğĘä¡IJåŕźçśžåŘŸéĞŘçŽĎäÿĂçğ■æŻ£äžčæU

```
class Spam(metaclass=MyMeta):
    debug = True
    synchronize = True
    pass
```

åřĘè£ŹäžŻåśđæĂğåőŽäźĽäÿžåŘĆæŢřçŽĎåě¡åď ĎåIJÍäžŐåőČäžňäÿ■äijŽæśąæ§Ş çśžçŽĎåŘ■çğřçľžéÚť
è£ŹäžŻåśđæĂğäžĚäžĚåŘłäžŐåśđäžŐçśžçŽĎåĽŻåžžéŸűæőţïijŇèĂŇäÿ■æŸŕçśžäÿ■çŽĎèŕ■åŘěæĽğèąŇéŸűa
åŘęåď ŰïijŇåőČäžňåIJÍ `__prepare__()` æŰźæşŢäÿ■æŸŕåŘŕäžěèćńèő£éŮőçŽĎïijŇåŽääÿžè£ŹäÿłæŰźæşŢ
ä¡ĘæŸŕçśžåŘŸéĞŘåŘłèČ¡åIJÍåĚČçśžçŽĎ `__new__()` åŠŇ `__init__()`
æŰźæşŢäÿ■åŘŕèğĄãĂĆ

11.16 9.16 *argsåŠŇ**kwargsçŽĎåijžåĽűåŘĆæŢřç■¿åŘ■

éŮőécŸ

ä¡ăæIJĽäÿĂäÿłåĞ¡æŢřæĽŰæŰźæşŢïijŇåőČä¡£çŤĺ*argsåŠŇ**kwargsä¡IJäÿžåŘĆæŢřïijŇè£Źæăůÿĺ¡£å¿Ú
ä¡ĘæIJĽæŮűåĂŹä¡ăæČşæčĂæ§ěäijăéĂŠè£ŻæĺěçŽĎåŘĆæŢřæŸŕäÿ■æŸŕæ§Řäÿłä¡ăæČşèęĄçŽĎçśžåđŇãĂĆ

èğčåĘşæŰźæąĹ

åŕźäżżä¡ŢæűĽåŘŁåĹřæŞ■ä¡IJåĞ¡æŢřèřČçŤĺç■¿åŘ■çŽĎéŮőécŸïijŇä¡ăéČ¡åžŤèŕěä¡£çŤĺ
`inspect` æĺąåİŮäÿ■çŽĎç■¿åŘ■çĽźæĂğãĂĆ æĽŚäžňæIJĂäÿżèęĄåĚşæşĺäÿď äÿłçśżïijŽ`Signature`
åŠŇ `Parameter` ãĂĆäÿŇéİćæŸŕäÿĂäÿłåĹŻåžžåĞ¡æŢřåĽ■éİćçŽĎäžď äžŠä¿Ňå■ŘïijŽ

```
>>> from inspect import Signature, Parameter
>>> # Make a signature for a func(x, y=42, *, z=None)
>>> parms = [ Parameter('x', Parameter.POSITIONAL_OR_KEYWORD),
...           Parameter('y', Parameter.POSITIONAL_OR_KEYWORD,␣
↪default=42),
...           Parameter('z', Parameter.KEYWORD_ONLY, default=None) ]
>>> sig = Signature(parms)
>>> print(sig)
(x, y=42, *, z=None)
>>>
```

äÿĂæŮęä¡ăæIJĽäžĘäÿĂäÿłç■¿åŘ■åŕźèśąïijŇä¡ăåřśåŘŕäżěä¡£çŤĺåőČçŽĎ `bind()`
æŰźæşŢå¿Łåőźæ ŸŞçŽĎåřĘåőČçżŚåőŽåĹř `*args` åŠŇ `**kwargs` äÿŁåŐżãĂĆ
äÿŇéİćæŸŕäÿĂäÿłçőĂå■ŢçŽĎæijŤçď žïijŽ

```
>>> def func(*args, **kwargs):
...     bound_values = sig.bind(*args, **kwargs)
```

(continues on next page)

(continued from previous page)

```
...         for name, value in bound_values.arguments.items():
...             print(name,value)
...
>>> # Try various examples
>>> func(1, 2, z=3)
x 1
y 2
z 3
>>> func(1)
x 1
>>> func(1, z=3)
x 1
z 3
>>> func(y=2, x=1)
x 1
y 2
>>> func(1, 2, 3, 4)
Traceback (most recent call last):
...
    File "/usr/local/lib/python3.3/inspect.py", line 1972, in _bind
        raise TypeError('too many positional arguments')
TypeError: too many positional arguments
>>> func(y=2)
Traceback (most recent call last):
...
    File "/usr/local/lib/python3.3/inspect.py", line 1961, in _bind
        raise TypeError(msg) from None
TypeError: 'x' parameter lacking default value
>>> func(1, y=2, x=3)
Traceback (most recent call last):
...
    File "/usr/local/lib/python3.3/inspect.py", line 1985, in _bind
        '{arg!r}'.format(arg=param.name))
TypeError: multiple values for argument 'x'
>>>
```

åŘŕäžěçIJŃåĞžæİěïijŇéĂŽè£ĞåřEç■¿åŘ■åŠŇäijăéĂŠçŽĎåŘĆæŢřçżŚåőŽèţůæİěïijŇåŘŕäžěåijžåĹűåĞ¡a

äÿŃéİćæŸŕäÿĂäÿłåijžåĹűåĞ¡æŢřç■¿åŘ■æŻťåĚůä¡ŞçŽĎä¿Ńå■ŘãĂĆåIJÍäžčçăĄäÿ■ïijŇæĹŚäžňåIJÍå§žç
__init__() æŰźæşŢïijŇ çĎűåŘŐæĹŚäžňåijžåĹűæL'ĂæIJL'çŽĎå■Řçśžå£ĚéążæŘŘä¿ŻäÿĂäÿłçL'źåőŽçŽĎ

```
from inspect import Signature, Parameter

def make_sig(*names):
    parms = [Parameter(name, Parameter.POSITIONAL_OR_KEYWORD)
             for name in names]
    return Signature(parms)

class Structure:
```

(continues on next page)

(continued from previous page)

```
    __signature__ = make_sig()
    def __init__(self, *args, **kwargs):
        bound_values = self.__signature__.bind(*args, **kwargs)
        for name, value in bound_values.arguments.items():
            setattr(self, name, value)

# Example use
class Stock(Structure):
    __signature__ = make_sig('name', 'shares', 'price')

class Point(Structure):
    __signature__ = make_sig('x', 'y')
```

äÿŃéİćæŸŕä¡£çŤĺè£ŹäÿŁ `Stock` çśżçŽĎçď žä¿ŃïijŽ

```
>>> import inspect
>>> print(inspect.signature(Stock))
(name, shares, price)
>>> s1 = Stock('ACME', 100, 490.1)
>>> s2 = Stock('ACME', 100)
Traceback (most recent call last):
...
TypeError: 'price' parameter lacking default value
>>> s3 = Stock('ACME', 100, 490.1, shares=50)
Traceback (most recent call last):
...
TypeError: multiple values for argument 'shares'
>>>
```

èőĺèőž

åIJĺæĹŚäżňéIJĂèęĄæđĎåżžéĂŽçŤĺåĞ¡æŢřåžŞãĂĄçijŰåĘŹèčĚéěřåŹĺæĹŰåőđçŐřäżčçŘĘçŽĎæŮűåĂŹ
`*args` åŠŇ `**kwargs` çŽĎä¡£çŤĺæŸŕå¿ĹæŹőéĄ■çŽĎãĂĆ
ä¡ĘæŸŕïijŇè£ŹæăůçŽĎåĞ¡æŢřæIJL'äÿĂäÿłçijžçĆźåřśæŸŕå¡Şä¡äæČşèęĄåőđçŐřèĞłåůśçŽĎåŘĆæŢřæčĂéłŃ
è£ŹæŮűåĂŹæĹŚäżňåŘŕäżěéĂŽè£ĞäÿĂäÿłç■¿åŘ■åŕźèśąæİěçőĂåŇŰåőČãĂĆ

åIJĺæIJĂåŘŐçŽĎäÿĂäÿłæŰźæąĹåőđä¿Ńäÿ■ïijŇæĹŚäżňè£ŸåŘŕäżěéĂŽè£Ğä¡£çŤĺèĞłåőŽäźL'åĚČçśżæİ

```
from inspect import Signature, Parameter

def make_sig(*names):
    parms = [Parameter(name, Parameter.POSITIONAL_OR_KEYWORD)
             for name in names]
    return Signature(parms)

class StructureMeta(type):
    def __new__(cls, clsname, bases, clsdict):
        clsdict['__signature__'] = make_sig(*clsdict.get('_fields',
↪[]))
```

(continues on next page)

(continued from previous page)

```
        return super().__new__(cls, clsname, bases, clsdict)

class Structure(metaclass=StructureMeta):
    _fields = []
    def __init__(self, *args, **kwargs):
        bound_values = self.__signature__.bind(*args, **kwargs)
        for name, value in bound_values.arguments.items():
            setattr(self, name, value)

# Example
class Stock(Structure):
    _fields = ['name', 'shares', 'price']

class Point(Structure):
    _fields = ['x', 'y']
```

å¡ŞæĹŚäžňèĞłåőŽäźL'ç■¿åŘ■çŽĎæŮűåĂŹïijŇåřĘç■¿åŘ■å■ŸåĆĺåIJĺçL'źåőŽçŽĎåśđæĂğ `__signature__` äÿ■éĂŽåÿÿæŸŕå¿ĹæIJL'çŤĺçŽĎãĂĆ è£ŹæăůçŽĎèŕİïijŇåIJĺä¡£çŤĺ `inspect` æĺąåİŮæL'ğèąŇåĘĚçIJĄçŽĎäžčçăĄåřśèČ¡åŘŚçŐřç■¿åŘ■åźűåřĘąőČä¡IJäÿžèřČçŤĺçžęåőŽãĂĆ

```
>>> import inspect
>>> print(inspect.signature(Stock))
(name, shares, price)
>>> print(inspect.signature(Point))
(x, y)
>>>
```

11.17 9.17 åIJĺçśżäÿŁåijžåĹűä¡£çŤĺçijŰçĺŃèğĎçžę

éŮőéćŸ

ä¡ăçŽĎçĺŃåžŘåŇĚåŘńäÿĂäÿłå¿Ĺåď'ğçŽĎçśżçżğæL'£ä¡ŞçşżïijŇä¡ăäÿŇæIJŻåijžåĹűæL'ğèąŇæ§ŘäžŻçij

èğčåĘşæŰźæąĹ

ąęĆæđIJä¡ăæČşçŻŚæŐğçśżçŽĎăőŽäźL'ïijŇéĂŽåÿÿåŘŕäżěéĂŽè£ĞăőŽäźL'äÿĂäÿłåĚČçśżãĂĆäÿĂäÿłå§ `type` åźűéĞ■åőŽäźL'åőČçŽĎ `__new__()` æŰźæşŢ æĹŰèĂĚæŸŕ `__init__()` æŰźæşŢãĂĆæŕŤåęĆïijŽ

```
class MyMeta(type):
    def __new__(self, clsname, bases, clsdict):
        # clsname is name of class being defined
        # bases is tuple of base classes
        # clsdict is class dictionary
        return super().__new__(cls, clsname, bases, clsdict)
```

åŘęäÿĂçğ■æŸŕïijŇåőŽäźĽ `__init__()` æŰźæşŢïijŽ

```
class MyMeta(type):
    def __init__(self, clsname, bases, clsdict):
        super().__init__(clsname, bases, clsdict)
        # clsname is name of class being defined
        # bases is tuple of base classes
        # clsdict is class dictionary
```

äÿžäžĘä¡£çŤÍè£ŹäÿłåĚČçśżïijŇä¡äéĂŽåÿÿèęĄåřĘåőČæŤ¿åĹřåĹřäÿĂäÿłéąűçžğçĹűçśżåőŽäźĽäÿ■ïijŇçĎ

```
class Root(metaclass=MyMeta):
    pass

class A(Root):
    pass

class B(Root):
    pass
```

åĚČçśżçŽĎäÿĂäÿłåĚşéŤőçĽźçĆźæŸŕåőČåĚĄèőÿä¡ääIJÍåőŽäźĽçŽĎæŮűåĂŹæčĂæ§ěçśżçŽĎåĘĚåőźã
`__init__()` æŰźæşŢäÿ■ïijŇ ä¡ääŘŕäžěå¿Ĺè¡żæĬ¿çŽĎæčĂæ§ěçśżå■ŮåĚÿãĂĄçĹűçśżç■Ľç■Ľ'ãĂĆåźűäÿŤi
åŽäæ■ď'ïijŇäÿĂäÿłæąĘæđűçŽĎæđĎåżžèĂĚåřśèČ¡åIJÍåď'ğåđŃçŽĎçżğæĽ'£ä¡Şçşżäÿ■éĂŽè£ĞçżŹäÿĂäÿłéąű

ä¡IJäÿžäÿĂäÿłåĚůä¡ŞçŽĎåžŤçŤÍä¿Ńå■ŘïijŇäÿŃéĬćåőŽäźĽäžĘäÿĂäÿłåĚČçśżïijŇäőČäijŽæŃŠçżĬäżżä¡

```
class NoMixedCaseMeta(type):
    def __new__(cls, clsname, bases, clsdict):
        for name in clsdict:
            if name.lower() != name:
                raise TypeError('Bad attribute name: ' + name)
        return super().__new__(cls, clsname, bases, clsdict)

class Root(metaclass=NoMixedCaseMeta):
    pass

class A(Root):
    def foo_bar(self): # Ok
        pass

class B(Root):
    def fooBar(self): # TypeError
        pass
```

ä¡IJäÿžæŻť'éńŸçžğåŠŇåőđçŤÍçŽĎä¿Ńå■ŘïijŇäÿŃéĬćæIJĽäÿĂäÿłåĚČçśżïijŇäőČçŤÍæĬěæčĂæţŃéĞ■è¡¡

```
from inspect import signature
import logging

class MatchSignaturesMeta(type):
```

(continues on next page)

(continued from previous page)

```
    def __init__(self, clsname, bases, clsdict):
        super().__init__(clsname, bases, clsdict)
        sup = super(self, self)
        for name, value in clsdict.items():
            if name.startswith('_') or not callable(value):
                continue
            # Get the previous definition (if any) and compare the␣
↪signatures
            prev_dfn = getattr(sup,name,None)
            if prev_dfn:
                prev_sig = signature(prev_dfn)
                val_sig = signature(value)
                if prev_sig != val_sig:
                    logging.warning('Signature mismatch in %s. %s !
↪= %s',
                                    value.__qualname__, prev_sig,␣
↪val_sig)

# Example
class Root(metaclass=MatchSignaturesMeta):
    pass

class A(Root):
    def foo(self, x, y):
        pass

    def spam(self, x, *, z):
        pass

# Class with redefined methods, but slightly different signatures
class B(A):
    def foo(self, a, b):
        pass

    def spam(self,x,z):
        pass
```

åęĆæđIJä¡ăè£ŘèąŇè£Źæőţäżčçă Ąïij Ňåřśäij Žå¿ŮåĹřäÿŃéİćè£ŹæăůçŽĎè¿ŞåĞžçżŞæđIJïij Ž

```
WARNING:root:Signature mismatch in B.spam. (self, x, *, z) != (self,
↪ x, z)
WARNING:root:Signature mismatch in B.foo. (self, x, y) != (self, a,␣
↪b)
```

è£Źçğ■è■ęåŚŁä£ąæĄŕåŕźäžŐæ■ŢèŐůäÿĂäžŻå¿őåęŹçŽĎçĺŃåžŘbugæŸŕå¿ĹæIJL'çŤĺçŽĎãĂĆä¿ŃåęĆïij
éĆčäźĹå¡Şå■ŘçśżæŤźå ŘŸåŘĆæŢřåŘ■å■ŮçŽĎæŮűåĂŹåřśäijŽèřČçŤĺåĞžéŤŹãĂĆ

èőÍèőž

åIJÍåď ğåđŃéİćåŘŚåŕźèśąçŽĎçÍŃåžŘäÿ■ïijŇéĂŽåÿÿåřĘçśżçŽĎåőŽäźĽæŤ¿åIJÍåĚČçśżäÿ■æŐğåĹűæŸŕå
åĚČçśżåŘŕäžěçŻŚæŐğçśżçŽĎåőŽäźĽ'ïijŇè■ęåŚŁçijŰçÍŃäžžåŚŸæ§ŘäžŻæšąæIJĽ'æşÍæĎŘåĹřçŽĎåŘŕèČ¡åĞ

æIJĽäžžåŘŕèČ¡äijŽèŕť'ïijŇåČŘè£ŹæăůçŽĎéŤŹèŕŕåŘŕäžěéĂŽè£ĞçÍŃåžŘåĹĘæđŘåůěåĚůæĹŰIDEåŐżå
ä¡ĘæŸŕïijŇåęĆæđIJä¡ăåIJÍæđĎåżžäÿĂäÿłæąĘæđűæĹŰåĞ¡æŢřåžŞä¿ŻåĚűäžŰäžžä¡£çŤÍïijŇéČčäźĹä¡ăæšąåŁ
åŻăæ■d'ïijŇåŕźäžŐè£Źçğ■çśżåđŃçŽĎçÍŃåžŘïijŇåęĆæđIJåŘŕäżěåIJÍåĚČçśżäÿ■åĄŽæčĂæţŃæĹŰèőÿåŘŕäžěä

åIJÍåĚČçśżäÿ■éĂĽæŃĽéĞ■æŰřåőŽäźĽ `__new__()` æŰźæşŢè£ŸæŸŕ
`__init__()` æŰźæşŢåŘŰåĘşäžŐä¡ăæČşæĂŐæăůä¡£çŤÍçżŞæđIJçśżãĂĆ `__new__()`
æŰźæşŢåIJÍçśżåĹŻåżžäźŃåĽ■èćńèřČçŤÍïijŇéĂŽåÿÿçŤÍäžŐéĂŽè£ĞæŞŘçğ■æŰźåijŘïijĹæŕŤåęĆéĂŽè£ĞæŤ
èĂŃ `__init__()` æŰźæşŢæŸŕåIJÍçśżèćńåĹŻåżžäźŃåŘŐèćńèřČçŤÍïijŇå¡Şä¡ăéIJĂèęĄåőŃæŢť'æđĎåżžçśż
åIJÍæIJĂåŘŐäÿĂäÿłä¿Ńå■Řäÿ■ïijŇè£ŹæŸŕå£ĚèęĄçŽĎïijŇåŻääÿžåőČä¡£çŤÍäžĘ `super()`
åĞ¡æŢřæİěæŘIJçť'ćäźŃåĽ■çŽĎåőŽäźĽ'ãĂĆ åőČåŘłèČ¡åIJÍçśżçŽĎåőđä¿ŃèćńåĹŻåżžäźŃåŘŐïijŇåźűäÿŤçŻÿ

æIJĂåŘŐäÿĂäÿłä¿Ńå■Řè£ŸæijŤçď'žäžĘPythonçŽĎåĞ¡æŢřç■¿åŘ■åŕźèśąçŽĎä¡£çŤÍãĂĆ
åőđéŹĚäÿŁïijŇåĚČçśżåřĘæŕŘäÿłåŘŕèřČçŤÍåőŽäźĽ'æŤ¿åIJÍäÿĂäÿłçśżäÿ■ïijŇæŘIJçť'ćåĽ■äÿĂäÿłåőŽäźĽ'ïijĹ
çĎűåŘŐéĂŽè£Ğä¡£çŤÍ `inspect.signature()` æİěçőĂå■ŢçŽĎæŕŤè¿ČåőČäžňçŽĎèřČçŤÍç■¿åŘ■ãĂĆ

æIJĂåŘŐäÿĂçĆźïijŇäžčçăĄäÿ■æIJĽäÿĂèąŇä¡£çŤÍäžĘ `super(self, self)`
åźűäÿ■æŸŕæŐŚçĽĹéŤŹèŕŕãĂĆ å¡Şä¡£çŤÍåĚČçśżçŽĎæŮűåĂŹïijŇæĹŚäžňèęĄæŮűåĹżèőŕä¡ŘäÿĂçĆźåřśæŸ
`self` åőđéŹĚäÿŁæŸŕäÿĂäÿłçśżåŕźèśąãĂĆ åŻăæ■d'ïijŇè£Źæİąèŕ■åŘěåĚűåőđåřśæŸŕçŤÍæİěåŕżæĽ'¿ä¡■äžŐçż
`self` çĹűçśżçŽĎåőŽäźĽ'ãĂĆ

11.18 9.18 äžěçijŰçÍŃæŰźåijŘåőŽäźĽçśż

éŮőécŸ

ä¡ăåIJÍåĘŹäÿĂæőţäžčçăĄïijŇæIJĂçżĹéIJĂèęĄåĹŻåżžäÿĂäÿłæŰřçŽĎçśżåŕźèśąãĂĆä¡äèĂČèŹŚåřĘçśżçŽ
åźűäÿŤä¡£çŤÍåĞ¡æŢřæŕŤåęĆ `exec()` æİěæĽ'ğèąŇåőČïijŇä¡ĘæŸŕä¡ăæČşåŕżæĽ'¿äÿĂäÿłæŻť'åŁääijŸéŻĚçŽ

èğčåĘşæŰźæąĹ

ä¡ăåŘŕäžěä¡£çŤÍåĞ¡æŢř `types.new_class()` æİěåĹİåğŃåŇŰæŰřçŽĎçśżåŕźèśąãĂĆ
ä¡äéIJĂèęĄåĄŽçŽĎåŘłæŸŕæŘŘä¿ŻçśżçŽĎåŘ■å■ŮãĂĄçĹűçśżåĚČçżĎãĂĄåĚşéŤőå■ŮåŘĆæŢřïijŇäžěåŘŁ

```
# stock.py
# Example of making a class manually from parts

# Methods
def __init__(self, name, shares, price):
    self.name = name
    self.shares = shares
    self.price = price
def cost(self):
    return self.shares * self.price
```

(continues on next page)

(continued from previous page)

```
cls_dict = {
    '__init__' : __init__,
    'cost' : cost,
}

# Make a class
import types

Stock = types.new_class('Stock', (), {}, lambda ns: ns.update(cls_
↪dict))
Stock.__module__ = __name__
```

è£Źçğ■æŰźåijŔäijŽæđĎåżžäÿĂäÿłæŹőéĂŽçŽĎçśżåŕźèśąïijŇåźűäÿŤæŇL'çĚğä¡ăçŽĎæIJ§æIJŻåůěä¡IJïij

```
>>> s = Stock('ACME', 50, 91.1)
>>> s
<stock.Stock object at 0x1006a9b10>
>>> s.cost()
4555.0
>>>
```

è£Źçğ■æŰźæşŢäÿ■ïijŇäÿĂäÿłæŕŤè¿ČéŽ¿çŘĘèğčçŽĎåIJřæŰźæŸŕåIJÍèřČçŤÍåőŇ types.new_class() åŕź Stock.__module__ çŽĎèţŇåĂijãĂĆ æŕŘæňąå¡ŞäÿĂäÿłçśżèćńåőŽäźL'åŘŐïijŇåőČçŽĎ __module__ åśđæĂğåŇĚåŘńåőŽäźL'åőČçŽĎæÍąåİŮåŘ■ãĂĆ è£ŹäÿłåŘ■å■ŮçŤÍäžŐçŤ§æĹŘ __repr__() æŰźæşŢçŽĎè¿ŞåĞžãĂĆåőČåŘŇæăůäź§èćńçŤÍäžŐå¿Ĺåď'ŽåžŞïijŇæŕŤåęĆ pickle ãĂĆ åŻăæ■d'ïijŇäÿžäžĘèől'ä¡ăåĹŻåżžçŽĎçśżæŸŕâĂIJæ■čçąőâĂİçŽĎïijŇä¡ăéIJĂèęAçąőä£Ìè£ŹäÿŁ

åęĆæđIJä¡ăæČşåĹŻåżžçŽĎçśżéIJĂèęĄäÿĂäÿłäÿ■åŘŇçŽĎåĚČçśżïijŇåŕäżěéĂŽè£Ğ types.new_class() çňňäÿL'äÿłåŘĆæŢřäijăéĂŠçżŹåőČãĂĆä¿ŃåęĆïijŽ

```
>>> import abc
>>> Stock = types.new_class('Stock', (), {'metaclass': abc.ABCMeta},
...                         lambda ns: ns.update(cls_dict))
...
>>> Stock.__module__ = __name__
>>> Stock
<class '__main__.Stock'>
>>> type(Stock)
<class 'abc.ABCMeta'>
>>>
```

çňňäÿL'äÿłåŘĆæŢřè£ŸåŘŕäżěåŇĚåŘńåĚűäżŰçŽĎåĚşéŤőå■ŮåŘĆæŢřăĂĆæŕŤåęĆïijŇäÿĂäÿłçśżçŽĎåĉ

```
class Spam(Base, debug=True, typecheck=False):
    pass
```

éĆčäźĹåŘŕäżěåřĘåĚűç£żèŕŚæĹŘåęĆäÿŃçŽĎ new_class() èřČçŤÍå¡ćåijŘïijŽ

```
Spam = types.new_class('Spam', (Base,),
                        {'debug': True, 'typecheck': False},
                        lambda ns: ns.update(cls_dict))
```

new_class() çňňåŻŻäÿłåŔĆæŢřæIJĂçěđçğŸïijŇåőČæŸŕäÿĂäÿłçŤÍæİěæŐěåŔŮçśżåŚ¡åŘ■çľžéŬť çŻ
éĂŽåÿÿè£ŹæŸŕäÿĂäÿłæŹőéĂŽçŽĎå■ŮåĚÿïijŇä¡ĘæŸŕåőČåőđéŹĚäÿŁæŸŕ
__prepare__() æŰźæşŢè£ŤåŻđçŽĎäżżæĎŘåŕźèśąïijŇè£ŹäÿłåIJÍ9.14åřŘèŁĆåůšçżŘäżŃçż■è£ĞäžĘãĂĆ
è£ŹäÿłåĞ¡æŢřéIJĂèęĄä¡£çŤÍäÿŁéİćæijŤçď žçŽĎ update()
æŰźæşŢçżŹåŚ¡åŘ■çľžéŬť åćđåŁäåĘĚåőźãĂĆ

èőÍèőž

å¿ŁåďŽæŮűåĂŹåęĆæđIJèČ¡æđĎéĂäæŰřçŽĎçśżåŕźèśąæŸŕå¿ŁæIJL'çŤÍçŽĎãĂĆ
æIJL'äÿłå¿ŁçĘ§æĆL'çŽĎä¿Ńå■ŘæŸŕèřČçŤÍ collections.namedtuple()
åĞ¡æŢřïijŇä¿ŃåęĆïijŽ

```
>>> Stock = collections.namedtuple('Stock', ['name', 'shares',
↪'price'])
>>> Stock
<class '__main__.Stock'>
>>>
```

namedtuple() ä¡£çŤÍ exec() èĂŃäÿ■æŸŕäÿŁéİćäżŃçż■çŽĎæŁĂæIJŕãĂĆä¡ĘæŸŕïijŇäÿŃéİćéĂŽè
æĹŚäżňçŻť æŐěåĹŻåżžäÿĂäÿłçśżïijŽ

```
import operator
import types
import sys

def named_tuple(classname, fieldnames):
    # Populate a dictionary of field property accessors
    cls_dict = { name: property(operator.itemgetter(n))
                 for n, name in enumerate(fieldnames) }

    # Make a __new__ function and add to the class dict
    def __new__(cls, *args):
        if len(args) != len(fieldnames):
            raise TypeError('Expected {} arguments'.
↪format(len(fieldnames)))
        return tuple.__new__(cls, args)

    cls_dict['__new__'] = __new__

    # Make the class
    cls = types.new_class(classname, (tuple,), {},
                          lambda ns: ns.update(cls_dict))

    # Set the module to that of the caller
```

(continues on next page)

(continued from previous page)

```
    cls.__module__ = sys._getframe(1).f_globals['__name__']
    return cls
```

è£ŹæőţäžčçăĄçŽĎæIJĂåŘŐéČĺåĹĘä¡£çŤĺäžĘäÿĂäÿłæĽĂèřŞçŽĎâĂİæąĘæđűé■ŤæşŢâĂÏïijŇéĂŽè£Ğè
sys._getframe() æİěèŐůåŘŰèřČçŤĺèĂĚçŽĎæĺąåİŮůåŘ■ãĂĆ
åŘęåď ŰäÿĂäÿłæąĘæđűé■ŤæşŢä¿Ńå■ŘåIJĺ2.15åřŘèŁĆäÿ■æIJĽäžŃçż■è£ĞãĂĆ

äÿŃéİćçŽĎä¿Ńå■ŘæijŤçď žäžĘåĽ■éİćçŽĎäžčçăĄæŸŕåęĆä¡Ţåůěä¡IJçŽĎïijŽ

```
>>> Point = named_tuple('Point', ['x', 'y'])
>>> Point
<class '__main__.Point'>
>>> p = Point(4, 5)
>>> len(p)
2
>>> p.x
4
>>> p.y
5
>>> p.x = 2
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
AttributeError: can't set attribute
>>> print('%s %s' % p)
4 5
>>>
```

è£ŹéąźæŁĂæIJŕäÿĂäÿłå¿ĹéĞ■èęĄçŽĎæŰźéİćæŸŕåőČåŕźäžŐåĚČçśżçŽĎæ■čçąőä¡£çŤĺãĂĆ
ä¡ššŘŕèČ¡åČŘéĂŽè£ĞçŻť æŐěåőđä¿ŃåŇŰäÿĂäÿłåĚČçśżæİěçŻť æŐěåĹŻåżžäÿĂäÿłçśżïijŽ

```
Stock = type('Stock', (), cls_dict)
```

è£Źçğ■æŰźæşŢçŽĎéŮőécŸåIJĺäžŐåőČå£¡çŢěäžĘäÿĂäžŻåĚşéŤőæ■ěéłď ïijŇæŕŤåęĆåŕźäžŐåĚČçśżäÿ■
__prepare__() æŰźæşŢçŽĎèřČçŤĺãĂĆ éĂŽè£Ğä¡£çŤĺ types.new_class()
ïijŇä¡ššŘŕäžěä£ÌèŕĄæĽĂæIJĽçŽĎå£ĚèęĄåĹİåğŃåŇŰæ■ěéłď éČ¡èČ¡å¿ŮůåĹřæĽ ğèąŇãĂĆ
æŕŤåęĆïijŇtypes.new_class() çňňåŻŻäÿłåŘĆæŢřçŽĎåŻđèřČåĞ¡æŢřæŐěåŘŮ
__prepare__() æŰźæşŢè£ŤåŻđçŽĎæŸššřĎåŕźèśąãĂĆ

åęĆæđIJä¡ššżĚäżĚåŘłæŸŕæČşæĽ ğèąŇåĞĘåď Ğæ■ěéłď ïijŇåŘŕäžěä¡£çŤĺ types.
prepare_class() ãĂĆä¿ŃåęĆïijŽ

```
import types
metaclass, kwargs, ns = types.prepare_class('Stock', (), {'metaclass
↪': type})
```

åőČäijŽæ§ěæĽ¿åŘĹéĂĆçŽĎåĚČçśżåźůèřČçŤĺåőČçŽĎ __prepare__()
æŰźæşŢãĂĆ çĎűåŘŐè£ŹäÿłåĚČçśżä£Ìå■ŸåőČçŽĎåĚşéŤőå■ŮåŘĆæŢřïijŇåĞĘåď ĞåŚ¡åŘ■çľ žéŮť åŘŐèćńě

æŻť åď Žä£ąæĄŕ, èŕůåŘĆèĂČ PEP 3115 , äžěåŘŁ Python documentation .

11.19 9.19 åIJÍåőŽäźĽçŽĎæŮűåĂŹåĹİåğŃåŇŰçśżçŽĎæĹŘåŚŸ

éŮőéćŸ

ä¡ăæČşåIJÍçśżèćńåőŽäźĽçŽĎæŮűåĂŹåřśåĹİåğŃåŇŰäÿĂéČÍåĹĘçśżçŽĎæĹŘåŚŸïijŇèĂŇäÿ■æŸŕèęAç

èğčåĘşæŰźæąĹ

åIJÍçśżåőŽäźĽæŮűåřśæĽğèąŇåĹİåğŃåŇŰæĹŰèő¿ç¡őæŞ■ä¡IJæŸŕåĚČçśżçŽĎäÿĂäÿłåĚÿåđŃåžŤçŤÍåIJ è£ŹæŮűåĂŹä¡ăåŘŕäżěæĽğèąŇäÿĂäžŻéćİåď ŰçŽĎæŞ■ä¡IJãĂĆ

äÿŃéİćæŸŕäÿĂäÿłä¿Ńå■ŘïijŇåĹľçŤÍè£ŹäÿłæĂİèůŕæİěåĹŻåżžçśżäijijäžŐ `collections` æÍąåİŮäÿ■çŽĎåŚ¡åŘ■åĚČçżĎçŽĎçśżïijŽ

```
import operator

class StructTupleMeta(type):
    def __init__(cls, *args, **kwargs):
        super().__init__(*args, **kwargs)
        for n, name in enumerate(cls._fields):
            setattr(cls, name, property(operator.itemgetter(n)))

class StructTuple(tuple, metaclass=StructTupleMeta):
    _fields = []
    def __new__(cls, *args):
        if len(args) != len(cls._fields):
            raise ValueError('{} arguments required'.format(len(cls.
↪_fields)))
        return super().__new__(cls,args)
```

è£ŹæőţäżčçăĄåŘŕäżěçŤÍæİěåőŽäźĽçőĂå■ŢçŽĎå§žäžŐåĚČçżĎçŽĎæŢřæ■őçżŞæđĎïijŇåęĆäÿŃæĽĂç

```
class Stock(StructTuple):
    _fields = ['name', 'shares', 'price']

class Point(StructTuple):
    _fields = ['x', 'y']
```

äÿŃéİćæijŤçď žåőČåęĆä¡Ţåůěä¡IJïijŽ

```
>>> s = Stock('ACME', 50, 91.1)
>>> s
('ACME', 50, 91.1)
>>> s[0]
'ACME'
>>> s.name
'ACME'
>>> s.shares * s.price
4555.0
```

(continues on next page)

(continued from previous page)

```
>>> s.shares = 23
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
AttributeError: can't set attribute
>>>
```

èőÍèőž

è£ŹäÿÄåřŔèŁĆäÿ■ïijŇçśż `StructTupleMeta` èŐůåŔŰåĹřçśżåśđæĂğ `_fields` äÿ■çŽĎåśđæĂğåŘ■å■ŮåĹŮèąÍïijŇ çĎűåŘŐåřĘåőČäžňè¡ňæ■ćæĹŘçŻÿåžŤçŽĎåŔŕèő£éŮőçĽźåőŽåĚČçżĎæ
`operator.itemgetter()` åĹŻåżžäÿĂäÿłèő£éŮőåŹĺåĞ¡æŢřïijŇ çĎűåŘŐ `property()` åĞ¡æŢřåřĘåĚűè¡ňæ■ćæĹŘäÿĂäÿłåśđæĂğãĂĆ

æIJňèŁĆæIJĂéŽ¿æĞĆçŽĎéČĺåĹĘæŸŕç§ěéĄŞäÿ■åŘŇçŽĎåĹİåğŃåŇŰæ■ěéłď'æŸŕäżĂäźĹæŮűåĂŹåŔŚ
`StructTupleMeta` äÿ■çŽĎ `__init__()` æŰźæşŢåŔłåIJĺæŕŔäÿłçśżèćńåőŽäźĽ'æŮűèćńèřČçŤĺäÿĂæňąã
`cls` åŔĆæŢřåřśæŸŕéĆčäÿłèćńåőŽäźĽçŽĎçśżãĂĆåőđéŹĚäÿŁïijŇäÿŁè£řäżčçăĄä¡£çŤĺäžĘ
`_fields` çśżåŔŸéĞŔæİěä£İå■ŸæŰřçŽĎèćńåőŽäźĽçŽĎçśżïijŇ
çĎűåŘŐçżŹåőČåĘ■æůżåŁääÿĂçĆźæŰřçŽĎäÿIJèě£ãĂĆ

`StructTuple` çśżä¡IJäÿžäÿĂäÿłæŹőéĂŽçŽĎå§žçśżïijŇä¿ŻåĚűäżŰä¡£çŤĺèĂĚæİěçżğæĽ'£ãĂĆ
è£Źäÿłçśżäÿ■çŽĎ `__new__()` æŰźæşŢçŤĺæİěæđĎéĂăæŰřçŽĎåőđä¿ŃãĂĆ è£ŹéĞŇä¡£çŤĺ
`__new__()` åźűäÿ■æŸŕå¿ĹåÿÿèğĄïijŇäÿżèęĄæŸŕåŻääÿżæĹŚäżňèęĄä£őæŤźåĚČçżĎçŽĎèřČçŤĺç■¿åŘ■ïij
ä¡£å¿ŮæĹŚäżňåŔŕäżěåČŔæŹőéĂŽçŽĎåőđä¿ŃěřČçŤĺéĆčæůůåĹŻåżžåőđä¿ŃãĂĆåřśåČŔäÿŃéİćè£ŹæůůïijŽ

```
s = Stock('ACME', 50, 91.1) # OK
s = Stock(('ACME', 50, 91.1)) # Error
```

èů§ `__init__()` äÿ■åŘŇçŽĎæŸŕïijŇ `__new__()` æŰźæşŢåIJĺåőđä¿ŃèćńåĹŻåżžäźŃåĽ■èćńèğęåŔŚ
çŤśäžŐåĚČçżĎæŸŕäÿ■åŔŕä£őæŤźçŽĎïijŇæĽĂäżěäÿĂæŮęåőČäžňèćńåĹŻåżžäžĘåřśäÿ■åŔŕèČ¡åŕźåőČåĄŽä
`__init__()` äijŽåIJĺåőđä¿ŃåĹŻåżžçŽĎæIJĂåŘŐèćńèğęåŔŚïijŇ
è£ŹæůůçŽĎèŕİæĹŚäżňåřśåŔŕäżěåĄŽæĹŚäżňæČşåĄŽçŽĎäžĘãĂĆè£Źäź§æŸŕäÿžäżĂäźĹ
`__new__()` æŰźæşŢåůšçżŔèćńåőŽäźĽ'äžĘãĂĆ

åř¡çőąæIJňèŁĆå¿Ĺç§■ïijŇè£ŸæŸŕéIJĂèęĄä¡äèČ¡äżŤçżĘçăŤèŕżïijŇæůśåĚěæĂIèĂČPythonçśżæŸŕåęĆä
è£ŸæIJĽåřśæŸŕåĚČçśżåŠŇçśżçŽĎåŘĎäÿłäÿ■åŘŇçŽĎæŰźæşŢçľ'ůçń§åIJĺäżĂäźĹæŮűåĂŹèćńèřČçŤĺãĂĆ

PEP 422 æŘŘä¿ŽäżĘäÿĂäÿłèğčåĘşæIJňèŁĆéŮőécŸçŽĎåŔęåď'ŰäÿĂçğ■æŰźæşŢãĂĆ
ä¡ĘæŸŕïijŇæĹłæ■ćåĹřæĹŚåĘŹè£ŹæIJňäźęçŽĎæŮűåĂŹïijŇåőČè£ŸæšąèćńéĞĞçžşåŠŇæŐěåŔŮãĂĆ
åř¡çőąåęĆæ■ď'ïijŇåęĆæđIJä¡ää¡£çŤĺçŽĎæŸŕPython 3.3æĹŮæŻť'éńŸçŽĎçĽ'ĹæIJňïijŇéĆčäźĹè£ŸæŸŕåĂijå

11.20 9.20 åĹľçŤĺåĞ¡æŢřæşĺèğčåőđçŐřæŰźæşŢéĞ■è¡¡

éŮőécŸ

ä¡ăåůšçżŔå■ęè£ĞæĂŐæăůä¡£çŤĺåĞ¡æŢřåŔĆæŢřæşĺèğčïijŇéĆčäźĹä¡ăåŔŕèČ¡äijŽæČşåĹľçŤĺåőČæİěåőć
ä¡ĘæŸŕä¡ääÿ■çąőåőŽåžŤèŕěæĂŐæăůåŐżåőđçŐřïijĹæĹŮèĂĚåĹřåžŢèąŇå¿ŮéĂŽäÿ■ïijĽãĂĆ

èğčåĘşæŰźæąĹ

æIJňåřŕèŁĆçŽĎæŁĂæIJŕæŸŕå§žäžŐäÿĂäÿłçőĂå■ŢçŽĎæŁĂæIJŕïijŇéĆčåřśæŸŕPythonåĚĄèőÿåŔĆæŢ

```
class Spam:
    def bar(self, x:int, y:int):
        print('Bar 1:', x, y)

    def bar(self, s:str, n:int = 0):
        print('Bar 2:', s, n)

s = Spam()
s.bar(2, 3) # Prints Bar 1: 2 3
s.bar('hello') # Prints Bar 2: hello 0
```

äÿŃéĺćæŸŕæĹŚäżňçňňäÿĂæ■ěçŽĎåřĺèŕŢïijŇä¡£çŤĺåĹřäžĘäÿĂäÿłåĚČçśżåŠŇæŔŕèłřåŹĺïijŽ

```
# multiple.py
import inspect
import types

class MultiMethod:
    '''
    Represents a single multimethod.
    '''
    def __init__(self, name):
        self._methods = {}
        self.__name__ = name

    def register(self, meth):
        '''
        Register a new method as a multimethod
        '''
        sig = inspect.signature(meth)

        # Build a type signature from the method's annotations
        types = []
        for name, parm in sig.parameters.items():
            if name == 'self':
                continue
            if parm.annotation is inspect.Parameter.empty:
                raise TypeError(
                    'Argument {} must be annotated with a type'.
↪format(name)
                )
            if not isinstance(parm.annotation, type):
                raise TypeError(
                    'Argument {} annotation must be a type'.
↪format(name)
                )
```

(continues on next page)

(continued from previous page)

```
            if parm.default is not inspect.Parameter.empty:
                self._methods[tuple(types)] = meth
            types.append(parm.annotation)

        self._methods[tuple(types)] = meth

    def __call__(self, *args):
        '''
        Call a method based on type signature of the arguments
        '''
        types = tuple(type(arg) for arg in args[1:])
        meth = self._methods.get(types, None)
        if meth:
            return meth(*args)
        else:
            raise TypeError('No matching method for types {}'.
↪format(types))

    def __get__(self, instance, cls):
        '''
        Descriptor method needed to make calls work in a class
        '''
        if instance is not None:
            return types.MethodType(self, instance)
        else:
            return self

class MultiDict(dict):
    '''
    Special dictionary to build multimethods in a metaclass
    '''
    def __setitem__(self, key, value):
        if key in self:
            # If key already exists, it must be a multimethod or␣
↪callable
            current_value = self[key]
            if isinstance(current_value, MultiMethod):
                current_value.register(value)
            else:
                mvalue = MultiMethod(key)
                mvalue.register(current_value)
                mvalue.register(value)
                super().__setitem__(key, mvalue)
        else:
            super().__setitem__(key, value)

class MultipleMeta(type):
    '''
    Metaclass that allows multiple dispatch of methods
```

(continues on next page)

(continued from previous page)

```
    '''
    def __new__(cls, clsname, bases, clsdict):
        return type.__new__(cls, clsname, bases, dict(clsdict))

    @classmethod
    def __prepare__(cls, clsname, bases):
        return MultiDict()
```

äÿžäžĘä¡£çŤÍè£ŹäÿłçśżïijŇä¡ššŘŕäżěšČŘäÿŇéÍćè£ŹæăůšĘŹïijŽ

```
class Spam(metaclass=MultipleMeta):
    def bar(self, x:int, y:int):
        print('Bar 1:', x, y)

    def bar(self, s:str, n:int = 0):
        print('Bar 2:', s, n)

# Example: overloaded __init__
import time

class Date(metaclass=MultipleMeta):
    def __init__(self, year: int, month:int, day:int):
        self.year = year
        self.month = month
        self.day = day

    def __init__(self):
        t = time.localtime()
        self.__init__(t.tm_year, t.tm_mon, t.tm_mday)
```

äÿŇéÍćæŸŕäÿĂäÿłäžď'äžŠçď'žä¿ŇæÍěéłŇèŕĄšőČèČ¡æ■čçąőçŽĎšůěä¡IJïijŽ

```
>>> s = Spam()
>>> s.bar(2, 3)
Bar 1: 2 3
>>> s.bar('hello')
Bar 2: hello 0
>>> s.bar('hello', 5)
Bar 2: hello 5
>>> s.bar(2, 'hello')
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "multiple.py", line 42, in __call__
        raise TypeError('No matching method for types {}'.
↪format(types))
TypeError: No matching method for types (<class 'int'>, <class 'str
↪'>)
>>> # Overloaded __init__
>>> d = Date(2012, 12, 21)
```

(continues on next page)

(continued from previous page)

```
>>> # Get today's date
>>> e = Date()
>>> e.year
2012
>>> e.month
12
>>> e.day
3
>>>
```

èőĺèőž

åİęçŹ¡æİěèőšïijŇçŻÿåŕźäžŐéĂŽåÿÿçŽĎäžčçăĄèĂŇåůšæIJňèŁĆä¡£çŤĺåĹřäžĘå¿Ĺåď'ŽçŽĎé■ŤæşŢäżčçä
ä¡ĘæŸŕïijŇåőČå■ť'èČ¡èőľ'æĹŚäžňæůśåĚěçŘĘèğčåĚČçśżåŠŇæŘŕè£řåŹĺçŽĎåžŢåśĆåůěä¡IJåŐ§çŘĘïijŇ
åźűèČ¡åŁäæůśåŕźè£ŹäžŻæęĆå£ţçŽĎå■řèśąãĂĆåŻäæ■d'ïijŇåřśçőŮä¡ääźűäÿ■äijŽçńŇå■şåŐżäžŤçŤĺæIJňèŁĆ
åőČçŽĎäÿĂäžŻåžŢåśĆæĂİæČşå■ť'äijŽå¡śåŞ■åĹřåĚűåőČæűL'åŘŁåĹřåĚČçśżãĂĄæŘŕè£řåŹĺåŠŇåĞ¡æŢřæş

æIJňèŁĆçŽĎåőđçŐřäÿ■çŽĎäÿżèęĄæĂİèůŕåĚűåőđæŸŕå¿ĹçőĂå■ŢçŽĎãĂĆMutipleMeta
åĚČçśżä¡£çŤĺåőČçŽĎ __prepare__() æŰźæşŢ æİěæŘŘä¿ŻäÿĂäÿłä¡IJäÿž MultiDict
åőđä¿ŃçŽĎèĞłåőŽäźĹ'å■ŮåĚÿãĂĆè£Źäÿłèů§æŹőéĂŽå■ŮåĚÿäÿ■äÿĂæůçŽĎæŸŕïijŇ
MultiDict äijŽåIJĺåĚČçť'äèćńèő¿ç¡őçŽĎæŮűåĂŹæčĂæ§ěæŸŕåŘęåůšçżŘå■ŸåIJĺïijŇåęĆæđIJå■ŸåIJĺçŽĎ
MultiMethod åőđä¿ŃäÿŕåŘĹåźűãĂĆ

MultiMethod åőđä¿ŃéĂŽè£ĞæđĎåżžäżŐçśżåđŃç■¿åŘ■åĹřåĞ¡æŢřçŽĎæŸäåřĎæİěæŤűéŻĘæŰźæşŢ
åIJĺè£ŹäÿłæđĎåżžè£ĞçĺŃäÿ■ïijŇåĞ¡æŢřæşĺèğčèćńçŤĺæİěæŤűéŻĘè£ŹäžŻç■¿åŘ■çĎűåŘŐæđĎåżžè£ŹäÿłæŸ
è£ŹäÿłèĞçĺŃåIJĺ MultiMethod.register() æŰźæşŢäÿ■åőđçŐřãĂĆ
è£Źçğ■æŸäåřĎçŽĎäÿĂäÿłåĚşéŤőçĹźçĆźæŸŕåŕźäžŐåď'ŽäÿłæŰźæşŢïijŇæĹ'ĂæIJĹ'åŘĆæŢřçśżåđŃéČ¡å£Ĕé

äÿžäžĘèőľ MultiMethod åőđä¿ŃæĺąæŃ§äÿĂäÿłèřČçŤĺïijŇåőČçŽĎ
__call__() æŰźæşŢèćńåőđçŐřäžĘãĂĆ è£ŹäÿłæŰźæşŢäżŐæĹ'ĂæIJĹ'æŐŠéŹď' slef
çŽĎåŘĆæŢřäÿ■æđĎåżžäÿĂäÿłçśżåđŃåĚČçżĎïijŇåIJĺåĘĚéČĺmapäÿ■æ§ěæĹ'¿è£ŹäÿłæŰźæşŢïijŇ
çĎűåŘŐèřČçŤĺçŻÿåžŤçŽĎæŰźæşŢãĂĆäÿžäžĘèČ¡èőľ MultiMethod
åőđä¿ŃåIJĺçśżåőŽäźĹ'æŮűæ■čçąőæŞ■ä¡IJïijŇ__get__() æŸŕå£Ĕéążå¿ŮåőđçŐřçŽĎãĂĆ
åőČèćńçŤĺæİěæđĎåżžæ■čçąőçŽĎçżŚåőŽæŰźæşŢãĂĆæŕŤåęĆïijŽ

```
>>> b = s.bar
>>> b
<bound method Spam.bar of <__main__.Spam object at 0x1006a46d0>>
>>> b.__self__
<__main__.Spam object at 0x1006a46d0>
>>> b.__func__
<__main__.MultiMethod object at 0x1006a4d50>
>>> b(2, 3)
Bar 1: 2 3
>>> b('hello')
Bar 2: hello 0
>>>
```

äÿ■è£ĞæIJňèŁĆçŽĎåőđçŐřè£ŸæIJL'äÿĂäžŻéŹŘåĹűïijŇåĚűäÿ■äÿĂäÿłæŸŕåőČäÿ■èČ¡ä¡£çŤĺåĚşéŤőå■

```
>>> s.bar(x=2, y=3)
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
TypeError: __call__() got an unexpected keyword argument 'y'

>>> s.bar(s='hello')
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
TypeError: __call__() got an unexpected keyword argument 's'
>>>
```

äź§èőÿæIJL’åĚűäżŰçŽĎæŰźæşŢèČ¡æůżåŁäè£Źçğ■æŤŕæŇĄïijŇä¡ĘæŸŕåőČéIJĂèęĄäÿĂäÿłåőŇåĚÍäÿ■
éŮőéćŸåIJÍäžŐåĚşéŤőå■ŮåŘĆæŢřçŽĎåĞžçŐřæŸŕæšąæIJL’éąžåžŘçŽĎãĂĆå¡ŞåőČèů§ä¡■ç¡őåŘĆæŢřæůůåĚ
éĆčä¡äçŽĎåŘĆæŢřåřśäijŽåŘŸå¿ŮæŕŤè¿ČæůůäźśäžĘïijŇè£ŹæŮűåĂŹä¡ääÿ■å¿Ůäÿ■åIJÍ
`__call__()` æŰźæşŢäÿ■åĚĹåŐżåĄŽäÿłæŐŠåžŘãĂĆ

åŘŇæăůåŕźäžŐçżğæL’£äź§æŸŕæIJL’éŹŘåĹűçŽĎïijŇä¿ŃåęĆïijŇçśżäijijäÿŃéİćè£Źçğ■äżčçăĄåřśäÿ■èČ¡

```
class A:
    pass

class B(A):
    pass

class C:
    pass

class Spam(metaclass=MultipleMeta):
    def foo(self, x:A):
        print('Foo 1:', x)

    def foo(self, x:C):
        print('Foo 2:', x)
```

åŐ§åŻäæŸŕåŻääÿž `x:A` æşĺèğčäÿ■èČ¡æĹŘåŁ§åŇźéĚ■å■Řçśżäőđä¿ŃïijĹæŕŤåęĆBçŽĎäőđä¿ŃïijL’ïijŇå

```
>>> s = Spam()
>>> a = A()
>>> s.foo(a)
Foo 1: <__main__.A object at 0x1006a5310>
>>> c = C()
>>> s.foo(c)
Foo 2: <__main__.C object at 0x1007a1910>
>>> b = B()
>>> s.foo(b)
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "multiple.py", line 44, in __call__
        raise TypeError('No matching method for types {}'.
↪format(types))
```

(continues on next page)

(continued from previous page)

```
TypeError: No matching method for types (<class '__main__.B'>,)
>>>
```

ä¡IJäÿžä¡£çŤÍåĚČçśżåŠŇæşÍèğčçŽĎäÿĂçğ■æŻ£äżčæŰźæąĹïijŇåŘŕäżěéĂŽè£ĞæŘŘè£řåŹÍæİěåőđçŐřç

```
import types

class multimethod:
    def __init__(self, func):
        self._methods = {}
        self.__name__ = func.__name__
        self._default = func

    def match(self, *types):
        def register(func):
            ndefaults = len(func.__defaults__) if func.__defaults__␣
↪else 0
            for n in range(ndefaults+1):
                self._methods[types[:len(types) - n]] = func
            return self
        return register

    def __call__(self, *args):
        types = tuple(type(arg) for arg in args[1:])
        meth = self._methods.get(types, None)
        if meth:
            return meth(*args)
        else:
            return self._default(*args)

    def __get__(self, instance, cls):
        if instance is not None:
            return types.MethodType(self, instance)
        else:
            return self
```

äÿžäžĘä¡£çŤÍæŘŘè£řåŹÍçĹ'ĹæIJňïijŇä¡ăéIJĂèęĄåČŘäÿŃéİćè£ŹæăůåĘŹïijŽ

```
class Spam:
    @multimethod
    def bar(self, *args):
        # Default method called if no match
        raise TypeError('No matching method for bar')

    @bar.match(int, int)
    def bar(self, x, y):
        print('Bar 1:', x, y)

    @bar.match(str, int)
```

(continues on next page)

(continued from previous page)

```
    def bar(self, s, n = 0):
        print('Bar 2:', s, n)
```

æŘŘè£řåŹÍæŰźæąĹåŘŇæăůäź§æIJĽåĽ■éİćæŘŘåĹřçŽĎéŹŘåĹűïijĹäÿ■æŤŕæŇĄåĚşéŤőå■ŮåŘĆæŢřåŠ

æĽĂæIJĽäžŃçĽľéČ¡æŸŕåźşç■ĽçŽĎïijŇæIJĽåě¡æIJĽåİŘïijŇäź§èőÿæIJĂåě¡çŽĎåŁđæşŢářśæŸŕåIJÍæŽ
äÿ■è£ĞæIJĽäžŻçĽźæőŁæČĚåĘţäÿŃè£ŸæŸŕæIJĽæĎŘäźĽçŽĎïijŇæŕŤåęĆå§žäžŐæÍąåijŘåŇźéĚ■çŽĎæŰźa
äÿ¿äÿłä¿Ňå■Řïij Ň8.21ářŘèŁĆäÿ■çŽĎèő£éŮőèĂĚæÍąåijŘåŘŕäżëä£őæŤźäÿžäÿĂäÿłä¡£çŤÍæŰźæşŢéĞ■è¡¡çŽĎ
ä¡ĘæŸŕïijŇéŹďäžĘè£ŹäÿłäżëåďŰïijŇéĂŽåÿÿäÿ■ážŤèŕëä¡£çŤÍæŰźæşŢéĞ■è¡¡ïijĹářśçőĂå■ŢçŽĎä¡£çŤÍäÿ■å

åIJÍPythonçď¿åŇžåŕźäžŐåőđçŐřæŰźæşŢéĞ■è¡¡çŽĎèőÍèőžåůšçżŘçŤśæİěåůšäźĚãĂĆ
åŕźäžŐåijŢåŘŚè£ŹäÿłäžĽèőžçŽĎåŐ§åŻäïijŇåŘŕäżëåŘĆèĂČäÿŃGuido van
RossumçŽĎè£ŹçŕĞå■Žåőćïij Ž Five-Minute Multimethods in Python

11.21 9.21 éĄ£åĚ■éĞ■åď■çŽĎåśđæĂğæŰźæşŢ

éŮőécŸ

ä¡ăåIJÍçśżäÿ■éIJĂèęĄéĞ■åď■çŽĎåőŽäźĽäÿĂäžŻæĽğèąŇçŻÿåŘŇéĂżè¿ŚçŽĎåśđæĂğæŰźæşŢïijŇæŕŤ

èğčåĘşæŰźæąĹ

èĂČèŹŚäÿŃäÿĂäÿłçőĂå■ŢçŽĎçśżïijŇåőČçŽĎåśđæĂğçŤśåśđæĂğæŰźæşŢåŇĚèčĚïijŽ

```
class Person:
    def __init__(self, name ,age):
        self.name = name
        self.age = age

    @property
    def name(self):
        return self._name

    @name.setter
    def name(self, value):
        if not isinstance(value, str):
            raise TypeError('name must be a string')
        self._name = value

    @property
    def age(self):
        return self._age

    @age.setter
    def age(self, value):
        if not isinstance(value, int):
```

(continues on next page)

(continued from previous page)

```
            raise TypeError('age must be an int')
        self._age = value
```

åŘŕäžěçIJŃåĹřïijŇäÿžäžĘåŏđçŐřåśđæĂğåĂijçŽĎçśżåđŃæčĂæ§ěæĹŚäżňåĘŹäžĘå¿Ĺåď'ŽçŽĎéĞ■åď■ä
åŘłèęĄä¡ăäżěåŘŐçIJŃåĹřçśżäijijè£ŹæăůçŽĎäżčçăĄïijŇä¡ăéČ¡åžŤèřěæČşåŁđæşŢåŐżçőĂåŇŰåőČãĂĆ
äÿĂäÿłåŘŕèąŇçŽĎæŰźæşŢæŸŕåĹŻåżžäÿĂäÿłåĞ¡æŢřçŤĺæĺěåőŽäźĹ'åśđæĂğåźűè£ŤåŽđåőČãĂĆä¿ŃåęĆïijŽ

```
def typed_property(name, expected_type):
    storage_name = '_' + name

    @property
    def prop(self):
        return getattr(self, storage_name)

    @prop.setter
    def prop(self, value):
        if not isinstance(value, expected_type):
            raise TypeError('{} must be a {}'.format(name, expected_
↪type))
        setattr(self, storage_name, value)

    return prop

# Example use
class Person:
    name = typed_property('name', str)
    age = typed_property('age', int)

    def __init__(self, name, age):
        self.name = name
        self.age = age
```

èőĺèőž

æIJňèŁĆæĹŚäżňæijŤçď'žåĘĚéČĺåĞ¡æŢřæĹŰèĂĚéŮ■åŇĚçŽĎäÿĂäÿłéĞ■èęĄçL'źæĂğïijŇåőČäżňå¿ĹåĆ
`typed_property()` çIJŃäÿŁåŐżæIJL'çĆźéŽ¿çŘĘèğčïijŇåĚűåőđåőČæL'ĂåĄŽçŽĎäżĚäżĚåřśæŸŕäÿžä¡ăç
åŽăæ■ď'ïijŇå¡ŞåIJĺäÿĂäÿłçśżäÿ■ä¡£çŤĺåőČçŽĎæŮűåĂŹïijŇæŢĹæđIJèů§åřĘåőČéĞŇéĺćçŽĎäżčçăĄæŤ¿åĹř
åř¡çőąåśđæĂğçŽĎ `getter` åŠŇ `setter` æŰźæşŢèő£éŮőäžĘæIJňåIJřåŘŸéĞŘåęĆ `name` ,
`expected_type` äżěåŘŁ `storate_name` ïijŇè£ŹäÿłåĺĹæ■čåÿÿïijŇè£ŹäżŽåŘŸéĞŘçŽĎåĂijäijŽä£Ĺå■Ÿ

æĹŚäżňè£ŸåŘŕäżěä¡£çŤĺ `functools.partial()` æĺěçĺ■çĺ■æŤźåŘŸäÿŃè£ŹäÿłäĺĹŃå■ŘïijŇåĺĹæIJI

```
from functools import partial

String = partial(typed_property, expected_type=str)
Integer = partial(typed_property, expected_type=int)

# Example:
```

(continues on next page)

(continued from previous page)

```
class Person:
    name = String('name')
    age = Integer('age')

    def __init__(self, name, age):
        self.name = name
        self.age = age
```

åĚűåőđä¡ăåŘŕäżěåŘŚçŐřïijŇè£ŹéĞŇçŽĎäżčçăĄèů§8.13åřŘèŁĆäÿ■çŽĎçśżåđŃçşżçż§æŘŘè£řåŹÍäżčçă

11.22 9.22 åőŽäźĽäÿŁäÿŃæŰĞçőąçŘĘåŹÍçŽĎçőĂå■ŢæŰźæşŢ

éŮőécŸ

ä¡ăæČşèĞłåůśåŐżåőđçŐřäÿĂäÿłæŰřçŽĎäÿŁäÿŃæŰĞçőąçŘĘåŹÍïijŇäżěä¿£ä¡£çŤÍwithèŕ■åŘěãĂĆ

èğčåĘşæŰźæąĹ

åőđçŐřäÿĂäÿłæŰřçŽĎäÿŁäÿŃæŰĞçőąçŘĘåŹÍçŽĎæIJĂçőĂå■ŢçŽĎæŰźæşŢåřśæŸŕä¡£çŤÍ `contexlib` æÍąåİŮäÿ■çŽĎ `@contextmanager` èčĚéěřåŹÍãĂĆäÿŃéİćæŸŕäÿĂäÿłåőđçŐřäžĘäżčçăĄåİŮèőąæŮűåŁ§èČ¡çŽĎäÿŁäÿŃæŰĞçőąçŘĘåŹÍä¿Ńå■ŘïijŽ

```
import time
from contextlib import contextmanager

@contextmanager
def timethis(label):
    start = time.time()
    try:
        yield
    finally:
        end = time.time()
        print('{}: {}'.format(label, end - start))

# Example use
with timethis('counting'):
    n = 10000000
    while n > 0:
        n -= 1
```

åIJÍåĞ¡æŢř `timethis()` äÿ■ïijŇ`yield` äźŃåĽ■çŽĎäżčçăĄäijŽåIJÍäÿŁäÿŃæŰĞçőąçŘĘåŹÍäÿ■ä¡IJäÿ `__enter__()` æŰźæşŢæĽ'ğèąŇïijŇ æĽĂæIJĽåIJÍ `yield` äźŃåŘŐçŽĎäżčçăĄäijŽä¡IJäÿž `__exit__()` æŰźæşŢæĽ'ğèąŇăĂĆ åęĆæđIJåĞžçŐřäžĘåijĆåÿÿïijŇåijĆåÿÿäijŽåIJÍyield-èŕ■åŘěéĆčéĞŇæŁŽåĞžãĂĆ

äÿŃéİćæŸŕäÿĂäÿłæŻť'åŁäéńŸçžğäÿĂçĆźçŽĎäÿŁäÿŃæŰĞçőąçŘĘåŹÍïijŇåőđçŐřäžĘåĹŮèąĺåŕźèśąäÿŁç

```
@contextmanager
def list_transaction(orig_list):
    working = list(orig_list)
    yield working
    orig_list[:] = working
```

è£Źæőţäžčçă ĄçŽĎä¡IJçŤĺæŸŕäżżä¡Ţåŕźå ĹŮèąĺçŽĎä£őæŤźåŘłæIJL’å¡ŞæL’ĂæIJL’äžčçă Ąè£ŘèąŇåőŇæ
äÿŃéĺćæĹŚäžňæĺěæijŤçď žäÿĂäÿŃïijŽ

```
>>> items = [1, 2, 3]
>>> with list_transaction(items) as working:
...     working.append(4)
...     working.append(5)
...
>>> items
[1, 2, 3, 4, 5]
>>> with list_transaction(items) as working:
...     working.append(6)
...     working.append(7)
...     raise RuntimeError('oops')
...
Traceback (most recent call last):
    File "<stdin>", line 4, in <module>
RuntimeError: oops
>>> items
[1, 2, 3, 4, 5]
>>>
```

èőĺèőž

éĂŽåÿÿæČĚåĘţäÿŃïijŇåęĆæđIJèęĄåĘŹäÿĂäÿłäÿŁäÿŃæŰĞçőąçŘĘåŹĺïijŇä¡ăéIJĂèęĄåőŽäźL’äÿĂäÿłç
`__enter__()` åŠŇäÿĂäÿł `__exit__()` æŰźæşŢïijŇåęĆäÿŃæL’Ăçď žïijŽ

```
import time

class timethis:
    def __init__(self, label):
        self.label = label

    def __enter__(self):
        self.start = time.time()

    def __exit__(self, exc_ty, exc_val, exc_tb):
        end = time.time()
        print('{}: {}'.format(self.label, end - self.start))
```

ář¡çőąè£Źäÿłäź§äÿ■éŽ¿åĘŹïijŇä¡ĘæŸŕçŻÿæŕŤè¿ČåĘŹäÿĂäÿłçőĂå■ŢçŽĎä¡£çŤĺ
`@contextmanager` æşĺèğčçŽĎåĞ¡æŢřèĂŇèĺĂè£ŸæŸŕçĺ■æŸ¿äźŘåŚşãĂĆ

`@contextmanager` åžŤèŕěäžĚäžĚçŤĺæİěåĘŹèĞłåŇĚåŘńçŽĎäÿŁäÿŃæŰĞçőąçŘĘåĞ¡æŢřãĂĆ åęĆæđIJä¡ăæIJĽäÿĂäžŻåŕźèśą(æŕŤåęĆäÿĂäÿłæŰĞäžűãĂĄç¡ŚçżIJè£đæŐěæĹŰéŤĄ)ïijŇéIJĂèęĄæŤŕæŇĄ `with` èŕ■åŘěïijŇéĆčäźĹä¡ăåřśéIJĂèęĄå■ŢçŃňåőđçŐř `__enter__()` æŰźæşŢåŠŇ `__exit__()` æŰźæşŢãĂĆ

11.23 9.23 åIJĺåśĂéČĺåŘŸéĞŘå§§äÿ■æĽğèąŇäžčçăĄ

éŮőécŸ

ä¡ăæČşåIJĺä¡£çŤĺèŇČåŻť åĘĚæĽğèąŇæ§Řäÿłäžčçă ĄçĽĞæőţïijŇåźűäÿŤåÿŇæIJŻåIJĺæĽğèąŇåŘŐæĽĂ

èğčåĘşæŰźæąĹ

äÿžäžĘçŘĘèğčè£ŹäÿłéŮőécŸïijŇåĚĹèŕŢèŕŢäÿĂäÿłçőĂå■ŢåIJžæŹŕãĂĆéęŰåĚĹïijŇåIJĺåĚĺåśĂåŚ¡åŘ■çĺ

```
>>> a = 13
>>> exec('b = a + 1')
>>> print(b)
14
>>>
```

çĎűåŘŐïijŇåĘ■åIJĺäÿĂäÿłåĞ¡æŢřäÿ■æĽğèąŇåŘŇæăůçŽĎäžčçăĄïijŽ

```
>>> def test():
...     a = 13
...     exec('b = a + 1')
...     print(b)
...
>>> test()
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "<stdin>", line 4, in test
NameError: global name 'b' is not defined
>>>
```

åŘŕäžěçIJŃåĞžïijŇæIJĂåŘŐæŁŻåĞžäžĘäÿĂäÿłNameErroråijĆåÿÿïijŇåřśèů§åIJĺ `exec()` èŕ■åŘěäžŐæšąæĽğèąŇè£ĞäÿĂæăůãĂĆ èęĄæŸŕä¡ăæČşåIJĺåŘŐéĺćçŽĎèőąçőŮäÿ■ä¡£çŤĺåĹř `exec()` æĽğèąŇçżŞæđIJçŽĎèŕİåřśäijŽæIJĽéŮőécŸäžĘãĂĆ

äÿžäžĘä£őæ■čè£ŹæăůçŽĎéŤŹèŕŕïijŇä¡ăéIJĂèęĄåIJĺèřČçŤĺ `exec()` äźŃåĽ■ä¡£çŤĺ `locals()` åĞ¡æŢřæİěå¿ŮåĹřäÿĂäÿłåśĂéČĺåŘŸéĞŘå■ŮåĚÿãĂĆ äźŃåŘŐä¡ăåřśèČ¡äžŐåśĂéČĺå■ŮåĚÿäÿ■èŐůåŘŰä£őæŤźè£ĞåŘŐçŽĎåŘŸéĞŘåĂijäžĘãĂĆä¿ŃåęĆïijŽ

```
>>> def test():
...     a = 13
...     loc = locals()
...     exec('b = a + 1')
...     b = loc['b']
```

(continues on next page)

(continued from previous page)

```
...     print(b)
...
>>> test()
14
>>>
```

èőÍèőž

åőđéŹĚäÿŁåŕźäžŐ `exec()` çŽĎæ■čçąőä¡£çŤÍæŸŕæŕŤè¿ČéŽ¿çŽĎãĂĆåď ğåď ŽæŢŕæČĚåĘţäÿŃå¡Şä¡ăè
`exec()` çŽĎæŮűåĂŹïijŃ è£ŸæIJĽåŘęåď ŰæŻť åě¡çŽĎèğčåĘşæŰźæąĹïijĹæŕŤåęĆèčĚéěřåŹÍãĂĄéŮ■åŃĚã

çĎűèĂŃïijŃåęĆđIJä¡ăäż■çĎűěęĄä¡£çŤÍ `exec()` ïijŃæIJňèŁĆåĹŮåĞžäžĘäÿĂäžŻåęĆä¡Ţæ■čçąőä¡£ç
éżŸèőď æČĚåĘţäÿŃïijŃ`exec()` äijŽåIJÍèřČçŤÍèĂĚåśĂéČÍåŠŃåĚÍåśĂèŃČåŻť åĘĚæĽ ğèąŃäżčçăĄãĂĆçĎű
äijăéĂŠçżŹ `exec()` çŽĎåśĂéČÍèŃČåŻť æŸŕæŃůèť Ïåőđé ŹĚåśĂéČÍåŘŸéĞŘçżĎæĹŘçŽĎäÿĂäÿłå■ŮåĚÿãĂ
åŻăæ■ď ïijŃåęĆđIJ `exec()` åęĆđIJæĽ ğèąŃäžĘä£őæŤźæŞ■ä¡IJïijŃè£ŹçğŽ■ä£őæŤźåŘŐçŽĎçżŞđIJåŕźåć
äÿŃéÍćæŸŕåŘęåď ŰäÿĂäÿłæijŤçď žåőČçŽĎä¿Ńå■ŘïijŽ

```
>>> def test1():
...     x = 0
...     exec('x += 1')
...     print(x)
...
>>> test1()
0
>>>
```

äÿŁéÍćäżčçăĄéĞŃïijŃå¡Şä¡ăèřČçŤÍ `locals()` èŐůåŘŰåśĂéČÍåŘŸéĞŘæŮűïijŃä¡ăèŐůå¿ŮçŽĎæŸŕäi
`exec()` çŽĎåśĂéČÍåŘŸéĞŘçŽĎäÿĂäÿłæŃůèť ÍãĂĆ éĂŽè£ĞåIJÍäżčçăĄæĽ ğèąŃåŘŐåőąæ§ěè£ŹäÿŁå■ŮåĚÿ

```
>>> def test2():
...     x = 0
...     loc = locals()
...     print('before:', loc)
...     exec('x += 1')
...     print('after:', loc)
...     print('x =', x)
...
>>> test2()
before: {'x': 0}
after: {'loc': {...}, 'x': 1}
x = 0
>>>
```

äżŤçżĘèğĆåŕ§æIJĂåŘŐäÿĂæ■ěçŽĎè¿ŞåĞžïijŃéŹď éÍđä¡ăåřĘ `loc`
äÿ■èćńä£őæŤźåŘŐçŽĎåĂijæĽŃåŁÍèţŃåĂijçżŹxïijŃåŘęåŹźxåŘŸéĞŘåĂijæŸŕäÿ■äijŽåŘŸçŽĎãĂĆ

åIJÍä¡£çŤÍ `locals()` çŽĎæŮűåĂŹïijŃä¡ăéIJĂèęĄæşÍæĎŘæŞ■ä¡IJéąžåžŘãĂĆæŕŘæňąåőČèćńèřČçŤÍçŽ
`locals()` äijŽèŐůåŘŰåśĂéČÍåŘŸéĞŘåĂijäÿ■çŽĎåĂijåźűèęĘçŻŰå■ŮåĚÿäÿ■çŻÿåžŤçŽĎåŘŸéĞŘãĂĆ
èřůæşÍæĎŘèğĆåŕ§äÿŃäÿŃéÍćè£ŹäÿłèřŢéłŃçŽĎè¿ŞåĞžçżŞæđIJïijŽ

```
>>> def test3():
...     x = 0
...     loc = locals()
...     print(loc)
...     exec('x += 1')
...     print(loc)
...     locals()
...     print(loc)
...
>>> test3()
{'x': 0}
{'loc': {...}, 'x': 1}
{'loc': {...}, 'x': 0}
>>>
```

æşĺæĎŕæIJĂåŘŐäÿĂæňąèřČçŤĺ `locals()` çŽĎæŮűåĂŹxçŽĎåĂijæŸŕåęĆä¡ŢèćńèęęçżŰæŐĽçŽĎã

ä¡IJäÿž `locals()` çŽĎäÿĂäÿłæŻ£äżčæŰźæąĹïijŇä¡ăåŕŕäżěä¡£çŤĺä¡äèĞłåůśçŽĎå■ŮåĚÿïijŇåźűåřĘåő `exec()` ãĂĆä¿ŃåęĆïijŽ

```
>>> def test4():
...     a = 13
...     loc = { 'a' : a }
...     glb = { }
...     exec('b = a + 1', glb, loc)
...     b = loc['b']
...     print(b)
...
>>> test4()
14
>>>
```

åď ğéČĺåĹĘæČĚåĘţäÿŃïijŇè£Źçğ■æŰźåijŕæŸŕä¡£çŤĺ `exec()` çŽĎæIJĂä¡şåőđèůţãĂĆ ä¡ăåŕłéIJĂèęĄä£ÌèřĄåĚĺåśĂåŠŇåśĂéČĺå■ŮåĚÿåIJĺåŘŐéİćäżčçăĄèő£éŮőæŮűåůšçżŕèćńåĹİåğŃåŇŰãĂĆ

è£ŸæIJĽäÿĂçĆźïijŇåIJĺä¡£çŤĺ `exec()` äźŃåĽ■ïijŇä¡ăåŕŕèČ¡éIJĂèęĄéŮőäÿŃèĞłåůśæŸŕåŘęæIJĽåĚ åď ğåď ŽæŢřæČĚåĘţäÿŃå¡Şä¡ăèęĄèĂČèŹŚä¡£çŤĺ `exec()` çŽĎæŮűåĂŹïijŇ è£ŸæIJĽåŕęåď ŰæŽť åě¡çŽĎèğčåĘşæŰźæąĹïijŇæŕŤåęĆèčĚéěřåŹĺãĂĄéŮ■åŇĚãĂĄåĚČçśżïijŇæĹŰåĚűäżŮ

11.24 9.24 èğčæđŘäÿŐåĹĘæđŘPythonæžŘçăĄ

éŮőécŸ

ä¡ăæČşåĘŹèğčæđŘåźűåĹĘæđŘPythonæžŘäżčçăĄçŽĎçĺŃåžŕãĂĆ

èğčåĘşæŰźæąĹ

åď ğéČĺåĹĘçĺŃåžŕåŚŸç§ěéĄŞPythonèČ¡åď §èőąçőŮæĹŰæĽğèąŇå■Ůçňęäÿšå¡ćåijŕçŽĎæžŘäżčçăĄãĂ

```
>>> x = 42
>>> eval('2 + 3*4 + x')
56
>>> exec('for i in range(10): print(i)')
0
1
2
3
4
5
6
7
8
9
>>>
```

ář¡çőąąęĆæ■ď'ïijŇast æÍąąİŮèČ¡èćńçŤÍæİěářĘPythonæžŘçăAçijŰèřŚæĹŘäÿĂäÿłąŘŕèćńąĹĘæđŘçŽĎa

```
>>> import ast
>>> ex = ast.parse('2 + 3*4 + x', mode='eval')
>>> ex
<_ast.Expression object at 0x1007473d0>
>>> ast.dump(ex)
"Expression(body=BinOp(left=BinOp(left=Num(n=2), op=Add(),
right=BinOp(left=Num(n=3), op=Mult(), right=Num(n=4))), op=Add(),
right=Name(id='x', ctx=Load())))"

>>> top = ast.parse('for i in range(10): print(i)', mode='exec')
>>> top
<_ast.Module object at 0x100747390>
>>> ast.dump(top)
"Module(body=[For(target=Name(id='i', ctx=Store()),
iter=Call(func=Name(id='range', ctx=Load()), args=[Num(n=10)],
keywords=[], starargs=None, kwargs=None),
body=[Expr(value=Call(func=Name(id='print', ctx=Load()),
args=[Name(id='i', ctx=Load())], keywords=[], starargs=None,
kwargs=None))], orelse=[])])"
>>>
```

åĹĘæđŘæžŘçăĄæăŚéIJĂèęĄä¡ăèĞłåůśæŻť'åď'ŽçŽĎå■ęäźăïijŇåőČæŸŕçŤśäÿĂçşżåĹŮASTèŁĆçĆźçżĎ
åĹĘæđŘè£ŹäžŻèŁĆçĆźæIJĂçőĂå■ŢçŽĎæŰźæşŢåŕśæŸŕåőŽäźĹ'äÿĂäÿłèő£éŮőèĂĚçśżïijŇåőđçŐřå¿Ĺåď'Ž
visit_NodeName() æŰźæşŢïijŇ NodeName() åŇźéĚ■éĆčäžŻä¡ăæĎ§åĚť'èűčçŽĎèŁĆçĆźãĂĆäÿŃéÍćæ

```
import ast

class CodeAnalyzer(ast.NodeVisitor):
    def __init__(self):
        self.loaded = set()
        self.stored = set()
        self.deleted = set()
```

(continues on next page)

(continued from previous page)

```
    def visit_Name(self, node):
        if isinstance(node.ctx, ast.Load):
            self.loaded.add(node.id)
        elif isinstance(node.ctx, ast.Store):
            self.stored.add(node.id)
        elif isinstance(node.ctx, ast.Del):
            self.deleted.add(node.id)

# Sample usage
if __name__ == '__main__':
    # Some Python code
    code = '''
    for i in range(10):
        print(i)
    del i
    '''

    # Parse into an AST
    top = ast.parse(code, mode='exec')

    # Feed the AST to analyze name usage
    c = CodeAnalyzer()
    c.visit(top)
    print('Loaded:', c.loaded)
    print('Stored:', c.stored)
    print('Deleted:', c.deleted)
```

åęĆæđIJä¡ăè£ŘèąŇè£ŹäÿłçÍŃåžŘïijŇä¡ääijŽå¿ŮåĹřäÿŃéÌćè£ŹæăůçŽĎè¿ŞåĞžïijŽ

```
Loaded: {'i', 'range', 'print'}
Stored: {'i'}
Deleted: {'i'}
```

æIJĂåŘŐïijŇASTåŘŕäżěéĂŽè£Ğ `compile()` åĞ¡æŢřæÌěçijŰèŕŚåźűæL'ğèąŇăĂĆä¿ŃåęĆïijŽ

```
>>> exec(compile(top,'<stdin>', 'exec'))
0
1
2
3
4
5
6
7
8
9
>>>
```

èőÍèőž

å¡Şä¡äèČ¡åď§åĹĘæđŘæžŘäżčçăĄåźűäżŐäÿ■èŐůåŘŰä£ąæĄŕçŽĎæŮűůåĂŹïijÑä¡äářśèČ¡åĘŹå¿Ĺåď Žäżč
ä¿ŃåęĆïijÑçŻÿæŕŤçŽšçŽőçŽĎäijäéĂŠäÿĂäžŻäżčçăĄçĽĞæőţåĹřçśżäijij
exec() åĞ¡æŢřäÿ■ïijÑä¡äåŘŕäżěåĚĹářĘåőČè¡ňæ■ćæĹŘäÿĂäÿłASTïijÑ
çĎűůåŘŐèğĆåŕ§åőČçŽĎçżĘèŁĆçIJŃåőČåĹřåžŢæŸŕæĂŐæäůůåĄŽçŽĎãĂĆ
ä¡äè£ŸåŘŕäżěåĘŹäÿĂäžŻåůěåĚůæİěæ§ěçIJŃæ§ŘäÿłæĺąåİŮçŽĎåĚĺéČĺæžŘçăĄïijÑåźűäÿŤåIJĺæ■ď å§žçąĂäÿ

éIJĂèęĄæşĺæĎŘçŽĎæŸŕïijÑåęĆæđIJä¡äç§ěéĄŞèĞłåůšåIJĺåźšåŢěïijÑä¡äè£ŸèČ¡åď§éĞ■åĘŹASTæİěèą
äÿŃéİćæŸŕäÿĂäÿłèčĚéěřåŹĺä¿Ńå■ŘïijÑåŘŕäżěéĂŽè£ĞéĞ■æŰřèğčæđŘåĞ¡æŢřä¡ŞæžŘçăĄãĂĄ
éĞ■åĘŹASTåźűéĞ■æŰřåĹŻåżžåĞ¡æŢřäżčçăĄåŕźèśąæİěåřĘåĚĺåśĂèő£éŮőåŘŸéĞŘéŹ■äÿžåĞ¡æŢřä¡Şä¡IJçŤ

```
# namelower.py
import ast
import inspect

# Node visitor that lowers globally accessed names into
# the function body as local variables.
class NameLower(ast.NodeVisitor):
    def __init__(self, lowered_names):
        self.lowered_names = lowered_names

    def visit_FunctionDef(self, node):
        # Compile some assignments to lower the constants
        code = '__globals = globals()\n'
        code += '\n'.join("{0} = __globals['{0}']".format(name)
                          for name in self.lowered_names)
        code_ast = ast.parse(code, mode='exec')

        # Inject new statements into the function body
        node.body[:0] = code_ast.body

        # Save the function object
        self.func = node

# Decorator that turns global names into locals
def lower_names(*namelist):
    def lower(func):
        srclines = inspect.getsource(func).splitlines()
        # Skip source lines prior to the @lower_names decorator
        for n, line in enumerate(srclines):
            if '@lower_names' in line:
                break

        src = '\n'.join(srclines[n+1:])
        # Hack to deal with indented code
        if src.startswith((' ','\t')):
            src = 'if 1:\n' + src
        top = ast.parse(src, mode='exec')
```

(continues on next page)

(continued from previous page)

```
        # Transform the AST
        cl = NameLower(namelist)
        cl.visit(top)

        # Execute the modified AST
        temp = {}
        exec(compile(top,'','exec'), temp, temp)

        # Pull out the modified code object
        func.__code__ = temp[func.__name__].__code__
        return func
    return lower
```

äÿžäžĘä¡£çŤÍè£ŹäÿłäżčçăĄïijŇä¡ăåŘŕäżěåČŘäÿŃéİćè£ŹæăůåĘŹïijŽ

```
INCR = 1
@lower_names('INCR')
def countdown(n):
    while n > 0:
        n -= INCR
```

èčĚéěřåŹÍäijŽåřĘ `countdown()` åĞ¡æŢřéĞ■åĘŹäÿžçśżäijijäÿŃéİćè£ŹæăůåŕïijŽ

```
def countdown(n):
    __globals = globals()
    INCR = __globals['INCR']
    while n > 0:
        n -= INCR
```

åIJÍæĂğèČ¡æţŃèŕŢäÿ■ïijŇåőČäijŽèőľåĞ¡æŢřè£ŘèąŇå£ń20%

çŐřåIJÍïijŇä¡ăæŸŕäÿ■æŸŕæČşäÿžä¡ăæL'ĂæIJL'çŽĎåĞ¡æŢřéČ¡åŁääÿŁè£ŹäÿłèčĚéěřåŹÍåŚćïij§æĹŰèőÿä
ä¡ĘæŸŕïijŇè£Źå■ť'æŸŕåŕźäžŐäÿĂäžŻéńŸçžğæŁĂæIJŕæŕŤåęĆASTæŞ■ä¡IJãĂĄæžŘçăĄæŞ■ä¡IJç■L'ç■L'çŽĎ

æIJňèŁĆåŘŮåŘęåď'ŰäÿĂäÿłåIJÍ `ActiveState` äÿ■åď'ĎçŘĘPythonå■ŮèŁĆçăĄçŽĎçńäèŁĆçŽĎåŘŕçc
ä¡£çŤÍASTæŸŕäÿĂäÿłæŻť'åŁäéńŸçžğçĆźçŽĎæŁĂæIJŕïijŇåźűäÿŤäź§æŻť'çőĂå■ŢäžŽãĂĆåŘĆèĂČäÿŃéİćäÿ

11.25 9.25 æŃĘèğčPythonå■ŮèŁĆçăĄ

éŮőécŸ

ä¡ăæČşéĂŽè£ĞåřĘä¡ăçŽĎäżčçăĄåŘ■çijŰèŕŚæĹŘä¡ŐçžğçŽĎå■ŮèŁĆçăĄæİěæ§ěçIJŃåőČåžŢåśĆçŽĎå

èğčåĘşæŰźæąĹ

`dis` æÍąåİŮåŘŕäżěèćńçŤÍæİěè¿ŞåĞžäżżä¡ŢPythonåĞ¡æŢřçŽĎåŘ■çijŰèŕŚçżŞæđIJãĂĆä¿ŃåęĆïijŽ

```
>>> def countdown(n):
... while n > 0:
...     print('T-minus', n)
...     n -= 1
... print('Blastoff!')
...
>>> import dis
>>> dis.dis(countdown)
...
>>>
```

èőÍèőž

å¡Şä¡ăæČşèęAç§ěéĄŞä¡ăçŽĎçÍŃåžŘåžŢåśÇçŽĎè£ŘèąŇæIJžåĹűçŽĎæŮűåĂŹïijŇdis æÍąåİŮæŸŕå¿ĹæIJL'çŤÍçŽĎãĂĆæŕŤåęĆåęĆæđIJä¡ăæČşèŕŢçİĂçŘĘèğčæĂğèČ¡çL'źå¿ĄãĂĆ èćń dis() åĞ¡æŢřèğčæđŘçŽĎåŐ§åğŃå■ŮèŁĆçăĄåęĆäÿŃæL'Ăçď'žïijŽ

```
>>> countdown.__code__.co_code
b"x
↪'\x00|\x00\x00d\x01\x00k\x04\x00r)\x00t\x00\x00d\x02\x00|\x00\x00\x83
\x02\x00\x01|\x00\x00d\x03\x008}
↪\x00\x00q\x03\x00Wt\x00\x00d\x04\x00\x83
\x01\x00\x01d\x00\x00S"
>>>
```

åęĆæđIJä¡ăæČşèĞłåůśèğčéĞŁè£Źæőţäžčçă ĄïijŇä¡ăéIJĂèęĄä¡£çŤÍäÿĂäžŻåIJÍ opcode æÍąåİŮäÿ■åőŽäźL'çŽĎåÿÿéĞŘăĂĆä¿ŃåęĆïijŽ

```
>>> c = countdown.__code__.co_code
>>> import opcode
>>> opcode.opname[c[0]]
>>> opcode.opname[c[0]]
'SETUP_LOOP'
>>> opcode.opname[c[3]]
'LOAD_FAST'
>>>
```

åěĞæĂłçŽĎæŸŕïijŇåIJÍ dis æÍąåİŮäÿ■åźűæšąæIJL'åĞ¡æŢřèől'ä¡ääžěçijŰçÍŃæŰźåijŘå¿ĹåőźæŸŞçŽĎ äÿ■è£ĞïijŇäÿŃéİćçŽĎçŤ§æĹŘåŹÍåĞ¡æŢřåŘŕäžěåřĘåŐ§åğŃå■ŮèŁĆçăĄåžŘåĹŮè¡ňæ■ćæĹŘ opcodes åŠŇåŘĆæŢřãĂĆ

```
import opcode

def generate_opcodes(codebytes):
    extended_arg = 0
    i = 0
    n = len(codebytes)
    while i < n:
```

(continues on next page)

(continued from previous page)

```
        op = codebytes[i]
        i += 1
        if op >= opcode.HAVE_ARGUMENT:
            oparg = codebytes[i] + codebytes[i+1]*256 + extended_arg
            extended_arg = 0
            i += 2
            if op == opcode.EXTENDED_ARG:
                extended_arg = oparg * 65536
                continue
        else:
            oparg = None
        yield (op, oparg)
```

ä¡£çŤÍæŰźæşŢåęĆäÿŃïijŽ

```
>>> for op, oparg in generate_opcodes(countdown.__code__.co_code):
...     print(op, opcode.opname[op], oparg)
```

è£Źçğ■æŰźåijŔå¿ĹåřŚæIJL'äžžç§ěéĄŞïijŇä¡ššŔŕäżěåĹľçŤÍåőČæŻ£æ■ćäżä¡Ţä¡äæČşèęĄæŻ£æ■ćçŽĎäÿŃéĺćæĹŚäżňçŤÍäÿĂäÿłçďžä¿ŃæĺěæijŤçďžæŢťäÿłè£ĞçÍŃïijŽ

```
>>> def add(x, y):
...     return x + y
...
>>> c = add.__code__
>>> c
<code object add at 0x1007beed0, file "<stdin>", line 1>
>>> c.co_code
b'|\x00\x00|\x01\x00\x17S'
>>>
>>> # Make a completely new code object with bogus byte code
>>> import types
>>> newbytecode = b'xxxxxxx'
>>> nc = types.CodeType(c.co_argcount, c.co_kwonlyargcount,
...     c.co_nlocals, c.co_stacksize, c.co_flags, newbytecode, c.co_
↪consts,
...     c.co_names, c.co_varnames, c.co_filename, c.co_name,
...     c.co_firstlineno, c.co_lnotab)
>>> nc
<code object add at 0x10069fe40, file "<stdin>", line 1>
>>> add.__code__ = nc
>>> add(2,3)
Segmentation fault
```

ä¡ššŔŕäżěåČŔè£ŹæŭůèĂ■åď ğæŃŻèőľèğčéĞŁåŹĺåěŤæžČãĂĆä¡ĘæŸŕïijŇåŕźäžŐçijŰåĘŹæŻť énŸçžğäi
äżŰäżňåŔŕèČ¡çIJ§çŽĎéIJĂèęĄéĞ■åĘŹå■ŮèŁĆçăĄãĂĆæIJňèŁĆæIJĂåŘŐçŽĎéČĺåĹĘæijŤçďžäžĘè£ŹäÿłæŸ
this code on ActiveState

12 çňňå■ĄçńäïijŽæĺąåİŮäÿŐåŇĚ

æĺąåİŮäÿŐåŇĚæŸŕäżżä¡Ţåď ğåđŇçĺŃåžŘçŽĎæäÿå£ČïijŇåřśè£đPythonåőL'èčĚçĺŃåžŘæIJňèžńäź§æŸŕ

Contents:

12.1 10.1 æđĎåżžäÿĂäÿłæĺąåİŮçŽĎåśĆçžğåŇĚ

éŮőécŸ

ä¡äæČşåřĘä¡äçŽĎäżčçäĄçżĎçżĞæĹŘçŤśå¿Ĺåď ŽåĹĘåśĆæĺąåİŮæđĎæĹŘçŽĎåŇĚãĂĆ

èğčåĘşæŰźæąĹ

åřĄèčĚæĹŘåŇĚæŸŕå¿ĹçőĂå■ŢçŽĎãĂĆåIJĺæŰĞäżűçşżçż§äÿŁçżĎçżĞä¡äçŽĎäżčçäĄïijŇåźűçąőä£İæŕ
ä¿ŃåęĆïijŽ

```
graphics/
    __init__.py
    primitive/
        __init__.py
        line.py
        fill.py
        text.py
    formats/
        __init__.py
        png.py
        jpg.py
```

äÿĂæŮęä¡äåĄŽåĹřäžĘè£ŹäÿĂçĆźïijŇä¡äåžŤèŕěèČ¡åď §æL'ğèąŇåŘĎçğ■importèŕ■åŘěïijŇåęĆäÿŃïijŽ

```
import graphics.primitive.line
from graphics.primitive import line
import graphics.formats.jpg as jpg
```

èőĺèőž

åőŽäźL'æĺąåİŮçŽĎåśĆæňąçżŞæđĎåřśåČŘåIJĺæŰĞäżűçşżçż§äÿŁäżžçńŃçŻőå¡ŢçżŞæđĎäÿĂæäůäőźæŸ
æŰĞäżű__init__.pyçŽĎçŻőçŽĎæŸŕèęĄåŇĚåŘńäÿ■åŘŇè£ŘèąŇçžğåĹńçŽĎåŇĚçŽĎåŘŕéĂL'çŽĎåĹİåğŃåŇ
äÿ¿äÿłä¿Ńå■ŘïijŇåęĆæđIJä¡äæL'ğèąŇäžĘèŕ■åŘěimport graphicsïijŇ æŰĞäżűgraph-
ics/__init__.pyåřĘèćńåŕijåĚě,äżžçńŃgraphicsåŚ¡åŘ■çl'žéŮť çŽĎåĘĚåőźãĂĆåČŘimport
graphics.format.jpgè£ŹæäůåŕijåĚěïijŇæŰĞäżűgraphics/__init__.pyåŠŇæŰĞäżűgraphics/formats/__init__.py

çżİåď ğéČĺåĹĘæŮűůåĂŹèől'__init__.pyçl'žçİĂåřśåě¡ãĂĆä¡ĘæŸŕæIJL'äžŻæČĚåĘţäÿŃåŘŕèČ¡åŇĚåŘńäżč
äÿ¿äÿłä¿Ńå■ŘïijŇ__init__.pyèČ¡åď §çŤĺæĺěèĞłåŁĺåŁäè¡¡å■ŘæĺąåİŮ:

```
# graphics/formats/__init__.py
from . import jpg
from . import png
```

åČŔè£ŹæăůäÿĂäÿłæŰĞäżű,çŤĺæĹůåŔŕäżěäżĚäżĚéĂŽè£Ğimport grahpics.formatsæİěäżčæŻ£import graphics.formats.jpgäżěåŔŁimport graphics.formats.pngãĂĆ

__init__.pyçŽĎåĚűäżŰåÿÿçŤĺçŤĺæşŢåŇĚæŃňåřĘåď'ŽäÿłæŰĞäżűåŘĹåźűåĹřäÿĂäÿłéĂżè¿ŚåŚ¡åŘ■çľžé

æŢŔéŤŘçŽĎçĺŃåžŔåŚŸäijŽåŔŚçŐřïijŇå■şä¡£æšąæIJĽ__init__.pyæŰĞäżűå■ŸåIJĺïijŇpythonäż■çĎűä

12.2 10.2 æŐğåĹűæĺąåİŮèćńåĚĺéČĺåŕijåĚěçŽĎåĘĚåőź

éŮőécŸ

å¡Şä¡£çŤĺâĂŹfrom module import *âĂŹ èŕ■åŔěæŮűïijŇåÿŇæIJŻåŕźäżŐæĺąåİŮæĹŰåŇĚåŕijåĞžçŽĎçňę

èğčåĘşæŰźæąĹ

åIJĺä¡äçŽĎæĺąåİŮäÿ■åőŽäźĽäÿĂäÿłåŔŸéĞŔ __all__ æİěæŸŐçąőåIJřåĹŮåĞžéIJĂèęĄåŕijåĞžçŽĎåĘĚå

äÿ¿äÿłä¿Ňå■Ř:

```
# somemodule.py
def spam():
    pass

def grok():
    pass

blah = 42
# Only export 'spam' and 'grok'
__all__ = ['spam', 'grok']
```

èőĺèőž

åř¡çőąåijžçČĹåŔ■åŕźä¡£çŤĺ âĂŸfrom module import *âĂŹ, ä¡ĘæŸŕåIJĺåőŽäźĽäžĘåď'ğéĞŔåŔŸéĞŔåŘ■çŽĎæĺąåİŮäÿ■éćŚçźĄä¡£çŤĺãĂĆ åęĆæđIJä¡ääÿ■åĄŽäżżä¡ŢäžŃ, è£ŹæăůçŽĎåŕijåĚěåřĘäijŽåŕijåĚěæĽĂæIJĽäÿ■äżěäÿŃåĹŠçž£åijĂåď'ť'çŽĎã åŘęäÿĂæŰźéİć,åęĆæđIJåőŽäźĽäžĘ __all__ , éĆčäźĹåŘłæIJĽèćńåĹŮäÿ¿åĞžçŽĎäÿIJèě£äijŽèćńåŕijåĞžãĂĆ

åęĆæđIJä¡äářĘ __all__ åőŽäźĽæĹŘäÿĂäÿłçľžåĹŮèąĺ, æšąæIJĽäÿIJèě£åřĘèćńåŕijåĚěãĂĆ åęĆæđIJ __all__ åŇĚåŘńæIJłåőŽäźĽçŽĎåŘ■å■Ů, åIJĺåŕijåĚěæŮűåijŢèţůAttributeErrorãĂĆ

12.3 10.3 ä¡£çŤÍçŻÿåŕźèůŕå¿ĎåŘ■åŕijåĚěåŇĚäÿ■å■ŘæÍąåİŮ

éŮőéćŸ

åřĘäžčçăĄçżĎçżĞæĹŘåŇĚ,æČşçŤÍimportèŕ■åŘěäžŐåŘęäÿĂäÿłåŇĚåŘ■æšąæIJL'çąňçijŰçăĄè£ĞçŽĎå

èğčåĘşæŰźæąĹ

ä¡£çŤÍåŇĚçŽĎçŻÿåŕźåŕijåĚěïijŇä¡£äÿĂäÿłæÍąåİŮåŕijåĚěåŘŇäÿĂäÿłåŇĚçŽĎåŘęäÿĂäÿłæÍąåİŮ
äÿ¿äÿłä¿Ňå■ŘïijŇåĄĞèő¿åIJÍä¡äçŽĎæŰĞäžűçşżçż§äÿŁæIJL'mypackageåŇĚïijŇçżĎçżĞåęĆäÿŇïijŽ

```
mypackage/
    __init__.py
    A/
        __init__.py
        spam.py
        grok.py
    B/
        __init__.py
        bar.py
```

åęĆæđIJæÍąåİŮmypackage.A.spamèęĄåŕijåĚěåŘŇçŻőå¡ŢäÿŇçŽĎæÍąåİŮgrokïijŇåőČåžŤèŕěåŇĚæŃňçŽ

```
# mypackage/A/spam.py
from . import grok
```

åęĆæđIJæÍąåİŮmypackage.A.spamèęĄåŕijåĚěäÿ■åŘŇçŻőå¡ŢäÿŇçŽĎæÍąåİŮB.barïijŇåőČåžŤèŕěä¡£çŤÍ

```
# mypackage/A/spam.py
from ..B import bar
```

äÿd'äÿłimportèŕ■åŘěéČ¡æšąåŇĚåŘńéąűåśĆåŇĚåŘ■ïijŇèĂŇæŸŕä¡£çŤÍäžĘspam.pyçŽĎçŻÿåŕźèůŕå¿ĎãĂ

èőÍèőž

åIJÍåŇĚåĘĚïijŇæŮćåŘŕäžěä¡£çŤÍçŻÿåŕźèůŕå¿Ďäź§åŘŕäžěä¡£çŤÍçżÍåŕźèůŕå¿ĎæÍěåŕijåĚěãĂĆ
äÿ¿äÿłä¿Ňå■ŘïijŽ

```
# mypackage/A/spam.py
from mypackage.A import grok # OK
from . import grok # OK
import grok # Error (not found)
```

åČŘmypackage.Aè£ŹæăůÿĄ¡£çŤÍçżÍåŕźèůŕå¿ĎåŘ■çŽĎäÿ■åĹl'äźŇåd'ĎæŸŕè£ŹåřĘéąűåśĆåŇĚåŘ■çąňçij
äÿ¿äÿłä¿Ňå■ŘïijŇåęĆæđIJä¡äæŤźåŘŸäžĘåŇĚåŘ■ïijŇä¡äåŕśå£ĚéążæčĂæ§ěæL'ĂæIJL'æŰĞäžűæÍěä£őæ■čæ
åŘŇæăůïijŇçąňçijŰçăĄçŽĎåŘ■çğřäijŽä¡£çğżåŁÍäžčçăĄåŘŸå¿ŮåŽŕéŽ¿ãĂĆäÿ¿äÿłä¿Ňå■ŘïijŇäź§èőÿæIJL'ä
åęĆæđIJä¡£çŤÍçŻÿåŕźåŕijåĚěïijŇéĆčäÿĂåĹĞéČ¡okïijŇçĎűèĂŇä¡£çŤÍçżÍåŕźèůŕå¿ĎåŘ■å¿ĹåŘŕèČ¡äijŽåĞžéŮ

importèŕ■åŘěçŽĎ . åŠŇ . . çIJŃèţůæİěå¿ĹæżŚçĺ¡, ä¡ĘåőČæŇĞåőŽçŻőå¡ŢåŘ■.äÿžå¡ŞåĽ■çŻőå¡ŢïijŇ..BäÿžçŻőå¡Ţ../BãĂĆè£Źçğŕèŕ■æşŢåŘłéĂĆçŤĺäžŐimport äÿ¿äÿłä¿Ňå■ŘïijŽ

```
from . import grok # OK
import .grok # ERROR
```

åř¡çőąä¡£çŤĺçŻÿåŕźåŕijåĚěçIJŃèţůæİěåČŘæŸŕæţŘèğĹæŰĞäżűçşżçż§ïijŇä¡ĘæŸŕäÿ■èČ¡åĹřåőŽäźĽåŇĚ

æIJĂåŘŐïijŇçŻÿåŕźåŕijåĚěåŘłéĂĆçŤĺäžŐåIJĺåŘĹéĂĆçŽĎåŇĚäÿ■çŽĎæĺąåİŮãĂĆåŕďåĚűæŸŕåIJĺéąűåş ä¿ŃåęĆïijŽ

```
% python3 mypackage/A/spam.py # Relative imports fail
```

åŘęäÿĂæŰźéİćïijŇåęĆæđIJä¡ăä¡£çŤĺPythonçŽĎ-méĂĽéąźæİěæĽğèąŇåĚĹåĽ■çŽĎèĎŽæIJňïijŇçŻÿåŕź ä¿ŃåęĆïijŽ

```
% python3 -m mypackage.A.spam # Relative imports work
```

æŻťåďŽçŽĎåŇĚçŽĎçŻÿåŕźåŕijåĚěçŽĎèČŇæŹŕç§ěèŕĘ,èŕůçIJŃ PEP 328 .

12.4 10.4 åřĘæĺąåİŮåĹĘåĽšæĹŘåďŽäÿłæŰĞäżű

éŮőécŸ

ä¡ăæČşåřĘäÿĂäÿłæĺąåİŮåĹĘåĽšæĹŘåďŽäÿłæŰĞäżűãĂĆä¡ĘæŸŕä¡ăäÿ■æČşåřĘåĹĘçężçŽĎæŰĞäżűçż§

èğčåĘşæŰźæąĹ

çĺŃåžŘæĺąåİŮåŘŕäżěéĂŽè£ĞåŘŸæĹŘåŇĚæİěåĹĘåĽšæĹŘåďŽäÿłçŃňçńŃçŽĎæŰĞäżűãĂĆèĂČèŹŚäÿ

```
# mymodule.py
class A:
    def spam(self):
        print('A.spam')

class B(A):
    def bar(self):
        print('B.bar')
```

åĄĞèő¿ä¡ăæČşmymodule.pyåĹĘäÿžäÿďäÿłæŰĞäżűïijŇæŕŘäÿłåőŽäźĽçŽĎäÿĂäÿłçśżãĂĆèęĄåĄŽåĹřè è£Źè£ŹäÿłçŻőå¡ŢäÿŃïijŇåĹŻåżžäżěäÿŃæŰĞäżűïijŽ

```
mymodule/
    __init__.py
    a.py
    b.py
```

åIJĺa.pyæŰĞäżűäÿ■æŘŠåĚěäżěäÿŃäżčçăĄïijŽ

```
# a.py
class A:
    def spam(self):
        print('A.spam')
```

åIJÍb.pyæŰĞäżűäÿ■æŘŠåĚěäżěäÿŃäżčçăĄïijŽ

```
# b.py
from .a import A
class B(A):
    def bar(self):
        print('B.bar')
```

æIJĂåŘŐïijŇåIJÍ __init__.py äÿ■ïijŇåřĘ2äÿłæŰĞäżűçšŸåŘĹåIJÍäÿĂèţůïijŽ

```
# __init__.py
from .a import A
from .b import B
```

åęĆæđIJæŇL'çĚğè£ŹäžŻæ■ěéłď'ïijŇæL'Ăäžğç Ť§çŽĎåŇĚMyModuleåřĘä¡IJäÿžäÿĂäÿłå■ŢäÿĂçŽĎéĂż

```
>>> import mymodule
>>> a = mymodule.A()
>>> a.spam()
A.spam
>>> b = mymodule.B()
>>> b.bar()
B.bar
>>>
```

èőÍèőž

åIJÍè£ŹäÿłçńăèŁĆäÿ■çŽĎäÿżèęĄéŮőéćŸæŸŕäÿĂäÿłèő¿èőąéŮőéćŸïijŇäÿ■çőąä¡ăæŸŕåŘęåÿŇæIJŻçŤÍæ

```
from mymodule.a import A
from mymodule.b import B
...
```

è£ŹæăůèČ¡åůëä¡IJïijŇä¡Ęè£ŹèőĺçŤÍæĹůæL'£åŘŮæŻť'åď'ŽçŽĎèť'§æŃĚïijŇçŤÍæĹůèęĄç§ěéĄŞäÿ■åŘŇ

```
from mymodule import A, B
```

åŕźåŘŐèĂĚèĂŇèÍĂïijŇèőĺ'mymoduleæĹŘäÿžäÿĂäÿłåď'ğçŽĎæžŘæŰĞäżűæŸŕæIJĂåÿÿèğĄçŽĎãĂĆä¡I
è£ŹæăůåĄŽçŽĎåĚşéŤőæŸŕåĹŻåżžäÿĂäÿłåŇĚçŽőå¡ŢïijŇä¡£çŤÍ __init__.py
æŰĞäżűæÍěåřĘæŕŘéČÍåĹĘçšŸåŘĹåIJÍäÿĂèţůãĂĆ

å¡ŞäÿĂäÿłæÍąåÍŮèćńåĹĘåL'šïijŇä¡äéIJĂèęĄçL'źåĹńæşÍæĎŘäžď'åŘL'åijŢçŤÍçŽĎæŰĞäżűåŘ■ãĂĆäÿ¿äÿ
from .a import A æÍěèŐůåŘŰãĂĆ

æŢť'äÿłçńăèŁĆéČ¡ä¡£çŤÍåŇĚçŽĎçŻÿåŕźåŕijåĚěæÍěéĄ£åĚ■åřĘéąűåśĆæÍąåÍŮåŘ■çąňçijŰçăĄåĹřæžŘäżč

ä¡IJäÿžè£ŹäÿĂçńăèŁĆçŽĎåżűäijÿïijŇåřĘäżŃçż■åżűè£§åŕijåĚěãĂĆåęĆåŻ¿æL'Ăçď'žïijŇ__init__.pyæŰ
èęĄåĄŽåĹřè£ŹäÿĂçĆźïijŇ__init__.pyæIJL'çżĘå¿őçŽĎåŘŸåŇŰïijŽ

```
# __init__.py
def A():
    from .a import A
    return A()

def B():
    from .b import B
    return B()
```

åIJÍè£ŹäÿłçL'ĹæIJňäÿ■ïijŇçśżAåŠŇçśżBèćńæŻ£æ■ćäÿžåIJÍçňňäÿĂæňąèő£éŮőæŮűåŁăè¡¡æL'ĂéIJĂçŽĎ
ä¿ŃåęĆïijŽ

```
>>> import mymodule
>>> a = mymodule.A()
>>> a.spam()
A.spam
>>>
```

åżűè£§åŁăè¡¡çŽĎäÿżèęAçijżçĆźæŸŕçżğæL'£åŠŇçśżåđŃæčĂæ§ěåŘŕèČ¡äijŽäÿ■æŰ■ãĂĆä¡ăåŘŕèČ¡äijŽ

```
if isinstance(x, mymodule.A): # Error
...

if isinstance(x, mymodule.a.A): # Ok
...
```

åżűè£§åŁăè¡¡çŽĎçIJ§åőđä¿Ńå■Ř, èğĄæăĞåĞĘåžŞ multiprocessing/__init__.py
çŽĎæžŘçăĄ.

12.5 10.5 åĹľçŤĺåŚ¡åŘ■çľžéŮťåŕijåĚěçŻőå¡ŢåĹĘæŢčçŽĎäżčçăĄ

éŮőécŸ

ä¡ăåŘŕèČ¡æIJL'åď'ğéĞŘçŽĎäżčçăĄïijŇçŤśäÿ■åŘŇçŽĎäžžæİěåĹĘæŢčåIJřçżť'æŁď'ãĂĆæŕŘäÿłéČĺåĹĘè

èğčåĘşæŰźæąĹ

äżŐæIJňèť'ĺäÿŁèőšïijŇä¡äèęĄåőŽäźL'äÿĂäÿłéąűçžğPythonåŇĚïijŇä¡IJäÿžäÿĂäÿłåď'ğéŻĘåŘĹåĹĘåijĂç

åIJÍçż§äÿĂäÿ■åŘŇçŽĎçŻőå¡ŢéĞŇçż§äÿĂçŻÿåŘŇçŽĎåŚ¡åŘ■çľžéŮť'ïijŇä¡ĘæŸŕèęĄåĹăăŐżçŤĺæİěåřĚ

```
foo-package/
    spam/
        blah.py
```

(continues on next page)

(continued from previous page)

```
bar-package/
    spam/
        grok.py
```

åIJÍè£Ź2äÿłçŻőå¡ŢéĞŇïijŇéČ¡æIJL'çİĂåĚśåŘŇçŽĎåŚ¡åŘ■çl'žéŮť'spamãĂĆåIJÍäżżä¡ŢäÿĂäÿłçŻőå¡ŢéĞ

èől'æĹŚäżňçIJŃçIJŃïijŇåęĆæđIJåřĘfoo-packageåŠŇbar-packageéČ¡åŁäåĹřpythonæĺąåİŮèůŕå¿ĎåźűåřİèŕŢåŕijåĚěäijŽåŘŚçŤ§äżĂäźĹ

```
>>> import sys
>>> sys.path.extend(['foo-package', 'bar-package'])
>>> import spam.blah
>>> import spam.grok
>>>
```

äÿd'äÿłäÿ■åŘŇçŽĎåŇĚçŻőå¡ŢèćńåŘĹåźűåĹřäÿĂèţůïijŇä¡äåŘŕäżěåŕijåĚěspam.blahåŠŇspam.grokïijŇå

èőÍèőž

åIJÍè£ŹéĞŇåůěä¡IJçŽĎæIJžåĹűèćńçğřäÿžâĂIJåŇĚåŚ¡åŘ■çl'žéŮť'âĂİçŽĎäÿĂäÿłçL'źå¿ĄãĂĆäżŐæIJňè

åŇĚåŚ¡åŘ■çl'žéŮť'çŽĎåĚşéŤőæŸŕçąőä£İéąűçžğçŻőå¡Ţäÿ■æšąæIJL'__init__.pyæŰĞäżűæİěä¡IJäÿžåĚśä
äÿ¿äÿłä¿Ňå■ŘïijŽ

```
>>> import spam
>>> spam.__path__
_NamespacePath(['foo-package/spam', 'bar-package/spam'])
>>>
```

åIJÍåőŽä¡■åŇĚçŽĎå■ŘçżĎäżűæŮűïijŇçŻőå¡Ţ__path__åřĘèćńçŤÍåĹř(ä¿ŃåęĆ,
å¡ŞåŕijåĚěspam.grokæĹŰèĂĚspam.blahçŽĎæŮűåĂŹ).

åŇĚåŚ¡åŘ■çl'žéŮť'çŽĎäÿĂäÿłéĞ■èęĄçL'źçĆźæŸŕäżżä¡ŢäżżéČ¡åŘŕäżěçŤÍèĞłåůśçŽĎäżčçăĄæİěæL'l'åś

```
my-package/
    spam/
        custom.py
```

åęĆæđIJä¡äåřĘä¡äçŽĎäżčçăĄçŻőå¡ŢåŠŇåĚűäżŰåŇĚäÿĂèţůæůżåŁäåĹřsys.pathïijŇè£ŹåřĘæŮäçijİåIJřå

```
>>> import spam.custom
>>> import spam.grok
>>> import spam.blah
>>>
```

äÿĂäÿłåŇĚæŸŕåŘęèćńä¡IJäÿžäÿĂäÿłåŇĚåŚ¡åŘ■çl'žéŮť'çŽĎäÿżèęĄæŰźæşŢæŸŕæčĂæ§ěåĚű__file__åś

```
>>> spam.__file__
Traceback (most recent call last):
```

(continues on next page)

(continued from previous page)

```
    File "<stdin>", line 1, in <module>
AttributeError: 'module' object has no attribute '__file__'
>>> spam
<module 'spam' (namespace)>
>>>
```

æŻť'åď'ŽçŽĎåŇĚåŚ¡åŘ■çľ'žéŮť'ä£ąæĄŕåŔŕäżěæ§ěçIJŃ PEP 420.

12.6 10.6 éĞ■æŰřåŁäè¡¡æĺąåİŮ

éŮőéćŸ

ä¡äæČşéĞ■æŰřåŁäè¡¡åůšçżŔåŁäè¡¡çŽĎæĺąåİŮïijŇåŻääÿžä¡äåŕźåĚűæžŘçäĄè£ŻèąŇäžĘä£őæŤźãĂĆ

èğčåĘşæŰźæąĹ

ä¡£çŤĺimp.reload()æİěéĞ■æŰřåŁäè¡¡åĚĹåĽ■åŁäè¡¡çŽĎæĺąåİŮãĂĆäÿ¿äÿłä¿Ňå■ŘïijŽ

```
>>> import spam
>>> import imp
>>> imp.reload(spam)
<module 'spam' from './spam.py'>
>>>
```

èőĺèőž

éĞ■æŰřåŁäè¡¡æĺąåİŮåIJĺåijĂåŔŚåŠŇèřČèŕŢè£ĞçĺŃäÿ■åÿÿåÿÿå¿ĹæIJĽçŤĺãĂĆä¡ĘåIJĺçŤ§äžğçŐŕåćČä

reload()æŞęéŹď'äžĘæĺąåİŮåžŢåśĆå■ŮåĚÿçŽĎåĘĚåőźïijŇåźűéĂŽè£ĞéĞ■æŰřæĽ'ğèąŇæĺąåİŮçŽĎæžŘ

åř¡çőąåęĆæ■ď'ïijŇreload()æšąæIJĽ'æŻť'æŰřåČŔâĂİfrom module import nameâĂIè£ŹæăůÄ¡£çŤĺimportèŕ■åŘěåŕijåĚěçŽĎåőŽäźĽ'ãĂĆäÿ¿äÿłä¿Ňå■ŘïijŽ

```
# spam.py
def bar():
    print('bar')

def grok():
    print('grok')
```

çŐřåIJĺåŘŕåŁĺäžď'äžŠåijŔäijŽèŕĬïijŽ

```
>>> import spam
>>> from spam import grok
>>> spam.bar()
```

(continues on next page)

(continued from previous page)

```
bar
>>> grok()
grok
>>>
```

äÿ■éĂĂåĞžPythonä£ŏæŤźspam.pyçŽĎæžŘçăĄïijŇåřĘgrok()åĞ¡æŢřæŤźæĹŘè£Źæăůïijž

```
def grok():
    print('New grok')
```

çŐřåIJĺåŻđåĹřäžď'äžŠåijŘäijŽèŕİïijŇéĞ■æŰřåŁäè¡¡æĺąåİŮïijŇåřİèŕŢäÿŃè£ŹäÿłåőđéłŇïijž

```
>>> import imp
>>> imp.reload(spam)
<module 'spam' from './spam.py'>
>>> spam.bar()
bar
>>> grok() # Notice old output
grok
>>> spam.grok() # Notice new output
New grok
>>>
```

åIJĺè£Źäÿłä¿Ňå■Řäÿ■ïijŇä¡äçIJŃåĹřæIJL'2äÿłçL'ĹæIJňçŽĎgrok()åĞ¡æŢřèćńåŁäè¡¡ãĂĆéĂŽåÿÿæİèèŕť'ï

åŻäæ■ď'ïijŇåIJĺçŤ§äžğçŐřåćČäÿ■åŘŕèČ¡éIJĂèęĄéĄ£åĚ■éĞ■æŰřåŁäè¡¡æĺąåİŮãĂĆåIJĺäžď'äžŠçŐřåćČ

12.7 10.7 è£ŘèąŇçŻőå¡ŢæĹŰåŐŃçijľæŰĞäżű

éŮőéćŸ

æĆĺæIJL'äÿĂäÿłåůšæĹŘéŢ£äÿžåŇĚåŘńåď'ŽäÿłæŰĞäżűçŽĎåžŤçŤĺïijŇåőČåůšè£IJäÿ■åĘ■æŸŕäÿĂäÿłçŐ

èğčåĘşæŰźæąĹ

åęĆæđIJä¡äçŽĎåžŤçŤĺçĺŃåžŘåůšçżŘæIJL'åď'ŽäÿłæŰĞäżűïijŇä¡ääŘŕäżěæŁŁä¡äçŽĎåžŤçŤĺçĺŃåžŘæŤ¿
äÿ¿äÿłä¿Ňå■ŘïijŇä¡ääŘŕäżěåČŘè£ŹæăůåĹŻåżžçŻőå¡ŢïijŽ

```
myapplication/
    spam.py
    bar.py
    grok.py
    __main__.py
```

åęĆæđIJ__main__.pyå■ŸåIJĺïijŇä¡ääŘŕäżěçőĂå■ŢåIJřåIJĺéąűçžğçŻőå¡Ţè£ŘèąŇPythonèğčéĞŁåŹĺïijŽ

```
bash % python3 myapplication
```

èğčéĞŁåŹĺåřĘæĽğèąŇ__main__.pyæŰĞäżűä¡IJäÿžäÿżçĺŃåžŘãĂĆ

åęĆæđIJä¡ššřĘä¡ăçŽĎäżčçăĄæĽŞåŇĚæĹŘzipæŰĞäżűïijŇè£Źçğ■æŁĂæIJŕåŘŇæăůäź§éĂĆçŤĺïijŇäÿ¿ä

```
bash % ls
spam.py bar.py grok.py __main__.py
bash % zip -r myapp.zip *.py
bash % python3 myapp.zip
... output from __main__.py ...
```

èőĺèőž

åĹŻåżžäÿĂäÿłçŻőå¡ŢæĹŰzipæŰĞäżűåźűæůżåŁă__main__.pyæŰĞäżűæİěåřĘäÿĂäÿłæŻť åď ğçŽĎPytho

çŤśäžŐçŻőå¡ŢåŠŇzipæŰĞäżűäÿŐæ■čåÿÿæŰĞäżűæIJĽäÿĂçĆźäÿ■åŘŇïijŇä¡ššŕřèČ¡è£ŸéIJĂèęĄåćđå

```
#!/usr/bin/env python3 /usr/local/bin/myapp.zip
```

12.8 10.8 èŕżåŘŰä¡■äžŐåŇĚäÿ■çŽĎæŢřæ■őæŰĞäżű

éŮőécŸ

ä¡ăçŽĎåŇĚäÿ■åŇĚåŘńäżčçăĄéIJĂèęĄåŐżèŕżåŘŰçŽĎæŢřæ■őæŰĞäżűãĂĆä¡ăéIJĂèęĄåř¡åŘřèČ¡åIJřçŤ

èğčåĘşæŰźæąĹ

åĄĞèő¿ä¡ăçŽĎåŇĚäÿ■çŽĎæŰĞäżűçżĎçżĞæĹŘåęĆäÿŃïijŽ

```
mypackage/
    __init__.py
    somedata.dat
    spam.py
```

çŐřåIJĺåĄĞèő¿spam.pyæŰĞäżűéIJĂèęĄèŕżåŘŰsomedata.datæŰĞäżűäÿ■çŽĎåĘĚåőźãĂĆä¡ššŘřäżěçŤĺä

```
# spam.py
import pkgutil
data = pkgutil.get_data(__package__, 'somedata.dat')
```

çŤśæ■ď äžğçŤ§çŽĎåŘŸéĞŘæŸŕåŇĚåŘńèřěæŰĞäżűçŽĎåŐ§åğŃåĘĚåőźçŽĎå■ŮèŁĆå■ŮçňęäÿšãĂĆ

èőÍèőž

èęĄèŕżåŘŰæŢřæ■őæŰĞäżűïijŇä¡ääŘŕèČ¡äijŽåĂ¿åŘŚäžŐçijŰåĘŹä¡£çŤÍåĘĚç¡őçŽĎI/OåŁ§èČ¡çŽĎäžčçăĄïijŇåęĆopen()ãĂĆä¡ĘæŸŕè£Źçğ■æŰźæşŢäź§æIJL'äÿĂäžŻéŮőéćŸãĂĆ

éęŰåĚĹïijŇäÿĂäÿłåŇĚåŕźèğčéĞŁåŹÍçŽĎå¡ŞåL'■åůëä¡IJçŻőå¡ŢåĞääźŐæšąæIJL'æŐğåĹűæİČãĂĆåŻäæ

çňňäžŇïijŇåŇĚéĂŽåÿÿäőL'èčĚä¡IJäÿž.zipæĹŰ.eggæŰĞäżűïijŇè£ŹäžŻæŰĞäżűäźűäÿ■åČŘåIJÍæŰĞäżŰ

pkgutil.get_data()åĞ¡æŢřæŸŕäÿĂäÿłèŕżåŘŰæŢřæ■őæŰĞäżűçŽĎéńŸçžğåůěåĚůïijŇäÿ■çŤÍçőąåŇĚæŸŕä

get_data()çŽĎçňňäÿĂäÿłåŘĆæŢřæŸŕåŇĚåŘńåŇĚåŘ■çŽĎå■ŮçňęäÿšãĂĆä¡ääŘŕäżěçŻť'æŐěä¡£çŤÍåŇĚ

12.9 10.9 åřĘæŰĞäżűåď'źåŁäåĚěåĹřsys.path

éŮőécŸ

ä¡äæŮäæşŢåŕijåĚěä¡äçŽĎPythonäżčçăĄåŻääÿżåőĆæL'ĂåIJÍçŽĎçŻőå¡Ţäÿ■åIJÍsys.pathéĞŇãĂĆä¡äæĆş

èğčåĘşæŰźæąĹ

æIJL'äÿď'çğ■åÿÿçŤÍçŽĎæŰźåijŘåřĘæŰřçŻőå¡ŢæůżåŁäåĹřsys.pathãĂĆçňňäÿĂçğ■ïijŇä¡ääŘŕäżěä¡£çŤÍ

```
bash % env PYTHONPATH=/some/dir:/other/dir python3
Python 3.3.0 (default, Oct 4 2012, 10:17:33)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more␣
↪information.
>>> import sys
>>> sys.path
['', '/some/dir', '/other/dir', ...]
>>>
```

åIJÍèĞłäőŽäźL'åžŤçŤÍçÍÑåžŘäÿ■ïijŇè£ŹæäůçŽĎçŐŕåćČåŘŸéĞŘåŘŕåIJÍçÍÑåžŘåŘŕåŁÍæŮűèő¿ç¡őæĹŰ

çňňäžŇçğ■æŰźæşŢæŸŕåĹŻåżžäÿĂäÿł.pthæŰĞäżűïijŇåřĘçŻőå¡ŢåĹŮäÿ¿åĞžæİěïijŇåČŘè£ŹæäůïijŽ

```
# myapplication.pth
/some/dir
/other/dir
```

è£Źäÿł.pthæŰĞäżűéIJĂèęĄæŤ¿åIJÍæ§ŘäÿłPythonçŽĎsite-packagesçŽőå¡ŢïijŇéĂŽåÿÿä¡■äžŐ/usr/local/lib/python3.3/site-packages æĹŰèĂĚ ~/.local/lib/python3.3/sitepackagesãĂĆå¡ŞèğčéĞŁåŹÍåŘŕåŁÍæŮűïijŇ.pthæŰĞäżűéĞŇåĹŮäÿ¿åĞžæİěçŽĎå■ŸåIJ

èőÍèőž

æŕŤèţůèť'źåŁŽåIJŕæL'¿æŰĞäżűïijŇä¡ääŘŕèČ¡äijŽåĂ¿åŘŚäžŐåĘŹäÿĂäÿłäżčçăĄæL'ŃåŁÍèřČèŁĆsys.pat

```
import sys
sys.path.insert(0, '/some/dir')
sys.path.insert(0, '/other/dir')
```

èŹ¡çĎűè£ŹèČ¡âĂIJåůěä¡IJâĂİïijŇä¡ĘæŸŕåIJÍåőđèůţäÿ■æđĄäÿžèĎĘåijśïijŇåžŤåř¡éĞŔéĄ£åĚ■ä¡£çŤÍãĂ

```
import sys
from os.path import abspath, join, dirname
sys.path.insert(0, join(abspath(dirname(__file__)), 'src'))
```

è£ŹåřĘsrccŻőå¡ŢæůżåŁäåĹřpathéĞŇïijŇåŠŇæĽğèąŇæŔŠåĚěæ■ěéłď çŽĎäżčçăĄåIJÍåŘŇäÿĂäÿłçŻőå¡

site-packagesçŻőå¡ŢæŸŕçňňäÿĽæŰźåŇĚåŠŇæÍąåİŮåőĽèčĚçŽĎçŻőå¡ŢãĂĆåęĆæđIJä¡ăæĽŃåŁÍåőĽèč packagesçŻőå¡ŢãĂĆèŹ¡çĎűçŤÍäžŐéĚ■ç¡őpathçŽĎ.pthæŰĞäżűå£ĚéążæŤ¿ç¡őåIJÍsite-packageséĞŇïijŇä¡ĘåőČéĚ■ç¡őçŽĎèůŕå¿ĎåŔŕäżěæŸŕçşżçż§äÿŁäżżä¡Ţä¡ăåÿŇæIJŻçŽĎçŻőå¡ŢãĂĆåŻăæ■d

12.10 10.10 éĂŽè£Ğå■Ůçňęäÿšå Ř■åŕijåĚěæÍąåİŮ

éŮőécŸ

ä¡ăæČşåŕijåĚěäÿĂäÿłæÍąåİŮïijŇä¡ĘæŸŕæÍąåİŮçŽĎåŘ■å■ŮåIJÍå■ŮçňęäÿšéĞŇãĂĆä¡ăæČşåŕźå■Ůçňęäÿ

èğčåĘşæŰźæąĹ

ä¡£çŤÍimportlib.import_module()åĞ¡æŢŕæİěæĽŃåŁÍåŕijåĚěåŘ■å■Ůäÿžå■ŮçňęäÿšçżŹåĞžçŽĎäÿĂäÿłæ

```
>>> import importlib
>>> math = importlib.import_module('math')
>>> math.sin(2)
0.9092974268256817
>>> mod = importlib.import_module('urllib.request')
>>> u = mod.urlopen('http://www.python.org')
>>>
```

import_moduleåŔłæŸŕçőĂå■ŢåIJŕæĽğèąŇåŠŇimportçŻÿåŘŇçŽĎæ■ěéłď ïijŇä¡ĘæŸŕè£ŤåŻđçŤ§æĹŘç

åęĆæđIJä¡ăæ■čåIJÍä¡£çŤÍçŽĎåŇĚïijŇimport_module()äź§åŔŕçŤÍäžŐçŻÿåŕźåŕijåĚěãĂĆä¡ĘæŸŕïijŇä¡ăé

```
import importlib
# Same as 'from . import b'
b = importlib.import_module('.b', __package__)
```

èőÍèőž

ä¡£çŤÍimport_module()æĽŃåŁÍåŕijåĚěæÍąåİŮçŽĎéŮőécŸéĂŽåÿÿåĞžçŐŕåIJÍäžěæ§Řçğ■æŰźåijŔçijŰå

åIJÍæŮğçŽĎäžčçăĄïijŇæIJL’æŮűä¡ăäijŽçIJŃåĹřçŤÍäžŐåŕijåĚěçŽĎåĘĚåžžåĞ¡æŢř__import__()ãĂĆåř¡ç
éĂŽåÿÿæŻť’åőźæŸŞä¡£çŤÍãĂĆ

èĞłåőŽäźL’åŕijåĚěè£ĞçÍŃçŽĎéńŸçžğåőđä¿ŃèğĄ10.11åřŔèŁĆ

12.11 10.11 éĂŽè£ĞéŠľå■Řè£IJçÍŃåŁäè¡¡æÍąåİŮ

éŮőéćŸ

ä¡ăæČşèĞłåőŽäźL’PythonçŽĎimportèŕ■åŔěïijŇä¡£å¿ŮåőČèČ¡äžŐè£IJçÍŃæIJžåŹÍäÿŁéİćéĂŔæŸŐçŽĎå

èğčåĘşæŰźæąĹ

éęŰåĚĹèęĄæŔŘåĞžæİěçŽĎæŸŕåőL’åĚĹéŮőéćŸãĂĆæIJňèŁĆèőĹèőžçŽĎæĂİæČşåęĆæđIJæšąæIJL’äÿĂ
äź§åřśæŸŕèŕť’ïijŇæĹŚäžňçŽĎäÿżèęĄçŽőçŽĎæŸŕæůśåĚěåĹĘæđŘPythonçŽĎimportèŕ■åŔěæIJžåĹűãĂĆ
åęĆæđIJä¡äçŘĘèğčäžĘæIJňèŁĆåĘĚéČĹåŐ§çŘĘïijŇä¡ääŕśèČ¡åď§äÿžåĚűäžŰäżżä¡ŢçŽőçŽĎèĂŇèĞłåőŽäźL’i
æIJL’äžĘè£ŹäžŻïijŇèőľæĹŚäžňçżğçż■åŘŚåL’■èţřãĂĆ

æIJňèŁĆæäÿå£ČæŸŕèő¿èőąåŕijåĚěèŕ■åŔěçŽĎæL’ľåśŢåŁ§èČ¡ãĂĆæIJL’å¿Ĺåď’Žçğ■æŰźæşŢåŔŕäżěåĄŽ
äÿ■è£ĞäÿžäžĘæijŤçď’žçŽĎæŰźä¿£ïijŇæĹŚäžňåijĂåğŃåĚĹæđĎéĂääÿŃéİćè£ŹäÿłPythonäżčçăĄçżŞæđĎïijŽ

```
testcode/
    spam.py
    fib.py
    grok/
        __init__.py
        blah.py
```

è£ŹäžŻæŰĞäżűçŽĎåĘĚåőźåźűäÿ■éĞ■èęĄïijŇäÿ■è£ĞæĹŚäžňåIJĺæŕŘäÿłæŰĞäżűäÿ■æŤ¿åĚěäžĘåřŚéĞ
è£Źæěůä¡ääŘŕäžěæţŃèŕŢåőČäžňåźűæ§ěçIJŃå¡ŞåőČäžňèćňåŕijåĚěæŮűçŽĎè¿ŞåĞžãĂĆä¿ŃåęĆïijŽ

```
# spam.py
print("I'm spam")

def hello(name):
    print('Hello %s' % name)

# fib.py
print("I'm fib")

def fib(n):
    if n < 2:
        return 1
    else:
        return fib(n-1) + fib(n-2)

# grok/__init__.py
print("I'm grok.__init__")
```

(continues on next page)

(continued from previous page)

```
# grok/blah.py
print("I'm grok.blah")
```

è£ŹéĞŇçŽĎçŻőçŽĎæŸŕåĚĄèőÿè£ŹäžŻæŰĞäżűä¡IJäÿžæÍąåİŮèćńè£IJçÍŃèő£éŮőãĂĆ
äź§èőÿæIJĂçőĂå■ŢçŽĎæŰźåijŘåřśæŸŕåřĘåőČäżňåŘŚåÿČåĹřäÿĂäÿłwebæIJ■åŁąåŹÍäÿŁéİćãĂĆåIJÍtestcode

```
bash % cd testcode
bash % python3 -m http.server 15000
Serving HTTP on 0.0.0.0 port 15000 ...
```

æIJ■åŁąåŹÍè£ŘèąŃèţůæİěåŘŐåĘ■åŘŕåŁÍäÿĂäÿłå■ŢçŃňçŽĎPythonèğčéĞŁåŹÍãĂĆ
çąőä£İä¡ăåŘŕäžěä¡£çŤÍ `urllib` èő£éŮőåĹřè£IJçÍŃæŰĞäżűãĂĆä¿ŃåęĆïijŽ

```
>>> from urllib.request import urlopen
>>> u = urlopen('http://localhost:15000/fib.py')
>>> data = u.read().decode('utf-8')
>>> print(data)
# fib.py
print("I'm fib")

def fib(n):
    if n < 2:
        return 1
    else:
        return fib(n-1) + fib(n-2)
>>>
```

äżŐè£ŹäÿłæIJ■åŁąåŹÍåŁäè¡¡æžŘäżčçăĄæŸŕæŐěäÿŃæİěæIJňèŁĆçŽĎå§žçąĂãĂĆ
äÿžäžĘæŻ£äżčæL'ŃåŁÍçŽĎéĂŽè£Ğ `urlopen()` æİěæŤűéŻĘæžŘæŰĞäżűïijŇ
æĹŚäžňéĂŽè£ĞèĞłåőŽäźL'importèŕ■åŘěæİěåIJÍåŘŐåŘŕèĞłåŁÍåÿőæĹŚäžňåĄŽåĹřãĂĆ

åŁäè¡¡è£IJçÍŃæÍąåİŮçŽĎçňňäÿĂçğ■æŰźæşŢæŸŕåĹŻåżžäÿĂäÿłæŸ¿åijŘçŽĎåŁäè¡¡åĞ¡æŢřæİěåőŃæĹŘ

```
import imp
import urllib.request
import sys

def load_module(url):
    u = urllib.request.urlopen(url)
    source = u.read().decode('utf-8')
    mod = sys.modules.setdefault(url, imp.new_module(url))
    code = compile(source, url, 'exec')
    mod.__file__ = url
    mod.__package__ = ''
    exec(code, mod.__dict__)
    return mod
```

è£ŹäÿłåĞ¡æŢřäijŽäÿŃè¡¡æžŘäżčçăĄïijŃåźűä¡£çŤÍ `compile()`
åřĘåĚűçijŰèŕŚåĹřäÿĂäÿłäżčçăĄåŕźèśąäÿ■ïijŃ çĎűåŘŐåIJÍäÿĂäÿłæŰřåĹŻåżžçŽĎæÍąåİŮåŕźèśąçŽĎå■ŮåĚÿä

```
>>> fib = load_module('http://localhost:15000/fib.py')
I'm fib
>>> fib.fib(10)
89
>>> spam = load_module('http://localhost:15000/spam.py')
I'm spam
>>> spam.hello('Guido')
Hello Guido
>>> fib
<module 'http://localhost:15000/fib.py' from 'http://
↪localhost:15000/fib.py'>
>>> spam
<module 'http://localhost:15000/spam.py' from 'http://
↪localhost:15000/spam.py'>
>>>
```

æ■čåęĆä¡ăæĽĂèğĄïijŇåŕźäžŐçőĂå■ŢçŽĎæÍąåÍŮè£ŹäÿłæŸŕèąŇå¿ŮéĂŽçŽĎãĂĆ
äÿ■è£ĞåőČåźűæšąæIJĽåţŇåĚěåĹřéĂŽåÿÿçŽĎimportèŕ■åŘěäÿ■ïijŇåęĆæđIJèęĄæŤŕæŇĄæŻťéńŸçžğçŽĎçż

äÿĂäÿłæŻťéĚůçŽĎåĄŽæşŢæŸŕåĹŻåżžäÿĂäÿłèĞłåőŽäźĽåŕijåĚěåŹĺãĂĆçňňäÿĂçğ■æŰźæşŢæŸŕåĹŻåż

```
# urlimport.py
import sys
import importlib.abc
import imp
from urllib.request import urlopen
from urllib.error import HTTPError, URLError
from html.parser import HTMLParser

# Debugging
import logging
log = logging.getLogger(__name__)

# Get links from a given URL
def _get_links(url):
    class LinkParser(HTMLParser):
        def handle_starttag(self, tag, attrs):
            if tag == 'a':
                attrs = dict(attrs)
                links.add(attrs.get('href').rstrip('/'))
    links = set()
    try:
        log.debug('Getting links from %s' % url)
        u = urlopen(url)
        parser = LinkParser()
        parser.feed(u.read().decode('utf-8'))
    except Exception as e:
        log.debug('Could not get links. %s', e)
    log.debug('links: %r', links)
    return links
```

(continues on next page)

(continued from previous page)

```
class UrlMetaFinder(importlib.abc.MetaPathFinder):
    def __init__(self, baseurl):
        self._baseurl = baseurl
        self._links = { }
        self._loaders = { baseurl : UrlModuleLoader(baseurl) }

    def find_module(self, fullname, path=None):
        log.debug('find_module: fullname=%r, path=%r', fullname,␣
↪path)
        if path is None:
            baseurl = self._baseurl
        else:
            if not path[0].startswith(self._baseurl):
                return None
            baseurl = path[0]
        parts = fullname.split('.')
        basename = parts[-1]
        log.debug('find_module: baseurl=%r, basename=%r', baseurl,␣
↪basename)

        # Check link cache
        if basename not in self._links:
            self._links[baseurl] = _get_links(baseurl)

        # Check if it's a package
        if basename in self._links[baseurl]:
            log.debug('find_module: trying package %r', fullname)
            fullurl = self._baseurl + '/' + basename
            # Attempt to load the package (which accesses __init__.
↪py)
            loader = UrlPackageLoader(fullurl)
            try:
                loader.load_module(fullname)
                self._links[fullurl] = _get_links(fullurl)
                self._loaders[fullurl] = UrlModuleLoader(fullurl)
                log.debug('find_module: package %r loaded',␣
↪fullname)
            except ImportError as e:
                log.debug('find_module: package failed. %s', e)
                loader = None
            return loader
        # A normal module
        filename = basename + '.py'
        if filename in self._links[baseurl]:
            log.debug('find_module: module %r found', fullname)
            return self._loaders[baseurl]
        else:
            log.debug('find_module: module %r not found', fullname)
```

(continues on next page)

(continued from previous page)

```
            return None

    def invalidate_caches(self):
        log.debug('invalidating link cache')
        self._links.clear()

# Module Loader for a URL
class UrlModuleLoader(importlib.abc.SourceLoader):
    def __init__(self, baseurl):
        self._baseurl = baseurl
        self._source_cache = {}

    def module_repr(self, module):
        return '<urlmodule %r from %r>' % (module.__name__, module._
↪_file__)

    # Required method
    def load_module(self, fullname):
        code = self.get_code(fullname)
        mod = sys.modules.setdefault(fullname, imp.new_
↪module(fullname))
        mod.__file__ = self.get_filename(fullname)
        mod.__loader__ = self
        mod.__package__ = fullname.rpartition('.')[0]
        exec(code, mod.__dict__)
        return mod

    # Optional extensions
    def get_code(self, fullname):
        src = self.get_source(fullname)
        return compile(src, self.get_filename(fullname), 'exec')

    def get_data(self, path):
        pass

    def get_filename(self, fullname):
        return self._baseurl + '/' + fullname.split('.')[-1] + '.py'

    def get_source(self, fullname):
        filename = self.get_filename(fullname)
        log.debug('loader: reading %r', filename)
        if filename in self._source_cache:
            log.debug('loader: cached %r', filename)
            return self._source_cache[filename]
        try:
            u = urlopen(filename)
            source = u.read().decode('utf-8')
            log.debug('loader: %r loaded', filename)
            self._source_cache[filename] = source
```

(continues on next page)

(continued from previous page)

```
            return source
        except (HTTPError, URLError) as e:
            log.debug('loader: %r failed. %s', filename, e)
            raise ImportError("Can't load %s" % filename)

    def is_package(self, fullname):
        return False

# Package loader for a URL
class UrlPackageLoader(UrlModuleLoader):
    def load_module(self, fullname):
        mod = super().load_module(fullname)
        mod.__path__ = [ self._baseurl ]
        mod.__package__ = fullname

    def get_filename(self, fullname):
        return self._baseurl + '/' + '__init__.py'

    def is_package(self, fullname):
        return True

# Utility functions for installing/uninstalling the loader
_installed_meta_cache = { }
def install_meta(address):
    if address not in _installed_meta_cache:
        finder = UrlMetaFinder(address)
        _installed_meta_cache[address] = finder
        sys.meta_path.append(finder)
        log.debug('%r installed on sys.meta_path', finder)

def remove_meta(address):
    if address in _installed_meta_cache:
        finder = _installed_meta_cache.pop(address)
        sys.meta_path.remove(finder)
        log.debug('%r removed from sys.meta_path', finder)
```

äÿŃéĬćæŸŕäÿĂäÿłäžď'äžŠäijŽèŕĬïijŃæijŤçď'žäžĘåęĆä¡Ţä¡£çŤĺåĽ■éĬćçŽĎäżčçăĄïijŽ

```
>>> # importing currently fails
>>> import fib
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
ImportError: No module named 'fib'
>>> # Load the importer and retry (it works)
>>> import urlimport
>>> urlimport.install_meta('http://localhost:15000')
>>> import fib
I'm fib
>>> import spam
```

(continues on next page)

(continued from previous page)

```
I'm spam
>>> import grok.blah
I'm grok.__init__
I'm grok.blah
>>> grok.blah.__file__
'http://localhost:15000/grok/blah.py'
>>>
```

è£ŹäÿłçĽźæőŁçŽĎæŰźæąĹäijŽåőĽèčĚäÿĂäÿłçĽźåĹńçŽĎæ§ěæĽ¿åŹÍ `UrlMetaFinder` åőđä¿ŃïijŃ ä¡IJäÿž `sys.meta_path` äÿ■æIJĂåŘŐçŽĎåőđä¡ŞãĂĆ å¡ŞæĺąåİŮèćńåŕijåĚěæŮűïijŃäijŽä¿İæ■ő `sys.meta_path` äÿ■çŽĎæ§ěæĽ¿åŹĺåőŽä¡■æĺąåİŮãĂĆ åIJÍè£ŹäÿłÄ¿Ńå■ŘäÿïijŃ`UrlMetaFinder` åőđä¿ŃæŸŕæIJĂåŘŐäÿĂäÿłæ§ěæĽ¿åŹĺæŰźæąĹïijŃ å¡ŞæĺąåİŮåIJĺäżżä¡ŢäÿĂäÿłæŹőéĂŽåIJřæŰźéČ¡æĽ¿äÿ

ä¡IJäÿžåÿÿèğĄçŽĎåőđçŐřæŰźæąĹïijŃ`UrlMetaFinder` çśżåŃĚèčĚåIJĺäÿĂäÿłçŤĺæĹůæŃĞåőŽçŽĎURLäÿŁãĂĆ åIJĺåĘĚéČĺïijŃæ§ěæĽ¿åŹĺéĂŽè£ĞæŁŞåŘŰæŃĞåč åŕijåĚěçŽĎæŮűåĂŹïijŃæĺąåİŮåŘ■äijŽèů§åůšæIJĽçŽĎéŞ¿æŐěä¡IJåŕźæŕŤãĂĆåęĆæđIJæĽ¿åĹřäžĘäÿĂäÿłåŃ äÿĂäÿłå■ŢçŃňçŽĎ `UrlModuleLoader` çśżèćńçŤĺæĺěäżŐè£IJçĺŃæIJžåŹĺäÿŁåŁäè¡¡æžŘäżčçăĄåźűåĹŻåżž è£ŹéĞŃçijŞå■ŸéŞ¿æŐěçŽĎäÿĂäÿłåŐ§åŻäæŸŕéĄ£åĚ■äÿ■å£ĚèęĄçŽĎHTTPèŕůæśĆéĞ■åď■åŕijåĚěãĂĆ

èĞłåőŽäźĽåŕijåĚěçŽĎçňňäžŃçğ■æŰźæşŢæŸŕçijŰåĘŹäÿĂäÿłéŠľå■ŘçŻť'æŐěåţŃåĚěåĹř `sys.path` åŘŸéĞŘäÿ■åŐżïijŃ èŕĘåĹńæ§ŘäžŻçŻőå¡ŢåŚ¡åŘ■æĺąåijŘãĂĆ åIJÍ `urlimport.py` äÿ■æůżåŁäåęĆäÿŃçŽĎçśżåŠŃæŤŕæŃĄåĞ¡æŢřïijŽ

```
# urlimport.py
# ... include previous code above ...
# Path finder class for a URL
class UrlPathFinder(importlib.abc.PathEntryFinder):
    def __init__(self, baseurl):
        self._links = None
        self._loader = UrlModuleLoader(baseurl)
        self._baseurl = baseurl

    def find_loader(self, fullname):
        log.debug('find_loader: %r', fullname)
        parts = fullname.split('.')
        basename = parts[-1]
        # Check link cache
        if self._links is None:
            self._links = [] # See discussion
            self._links = _get_links(self._baseurl)

        # Check if it's a package
        if basename in self._links:
            log.debug('find_loader: trying package %r', fullname)
            fullurl = self._baseurl + '/' + basename
            # Attempt to load the package (which accesses __init__.
↪py)
            loader = UrlPackageLoader(fullurl)
```

(continues on next page)

(continued from previous page)

```
            try:
                loader.load_module(fullname)
                log.debug('find_loader: package %r loaded',␣
↪fullname)
            except ImportError as e:
                log.debug('find_loader: %r is a namespace package',␣
↪fullname)
                loader = None
            return (loader, [fullurl])

        # A normal module
        filename = basename + '.py'
        if filename in self._links:
            log.debug('find_loader: module %r found', fullname)
            return (self._loader, [])
        else:
            log.debug('find_loader: module %r not found', fullname)
            return (None, [])

    def invalidate_caches(self):
        log.debug('invalidating link cache')
        self._links = None

# Check path to see if it looks like a URL
_url_path_cache = {}
def handle_url(path):
    if path.startswith(('http://', 'https://')):
        log.debug('Handle path? %s. [Yes]', path)
        if path in _url_path_cache:
            finder = _url_path_cache[path]
        else:
            finder = UrlPathFinder(path)
            _url_path_cache[path] = finder
        return finder
    else:
        log.debug('Handle path? %s. [No]', path)

def install_path_hook():
    sys.path_hooks.append(handle_url)
    sys.path_importer_cache.clear()
    log.debug('Installing handle_url')

def remove_path_hook():
    sys.path_hooks.remove(handle_url)
    sys.path_importer_cache.clear()
    log.debug('Removing handle_url')
```

èęĄä¡£çŤÍè£Źäÿłèůŕå¿ĎæŞěæL'¿åŹÍïijŇä¡ăåŘłéIJĂèęĄåIJÍ äÿ■åŁăåĚěURLéŞ¿æŐěãĂĆä¿ŃåęĆïijŽ `sys.path`

```
>>> # Initial import fails
>>> import fib
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
ImportError: No module named 'fib'

>>> # Install the path hook
>>> import urlimport
>>> urlimport.install_path_hook()

>>> # Imports still fail (not on path)
>>> import fib
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
ImportError: No module named 'fib'

>>> # Add an entry to sys.path and watch it work
>>> import sys
>>> sys.path.append('http://localhost:15000')
>>> import fib
I'm fib
>>> import grok.blah
I'm grok.__init__
I'm grok.blah
>>> grok.blah.__file__
'http://localhost:15000/grok/blah.py'
>>>
```

åĚşéŤőçĆźářśæŸŕ `handle_url()` åĞ¡æŢřïijŇåőČèćńæůżåŁäåĹřäžĘ `sys.path_hooks` åŘŸéĞŘäÿ■ãĂĆ å¡Ş `sys.path` çŽĎåőđä¡Şèćńåď ĎçŘĘæŮűïijŇäijŽèřČçŤÍ `sys.path_hooks` äÿ■çŽĎåĞ¡æŢřãĂĆ åęĆæđIJäżżä¡ŢäÿĂäÿłåĞ¡æŢřè£ŤåŻđäžĘäÿĂäÿłæ§ěæL'¿åŹĺåŕźèśą `sys.path` åőđä¡ŞåŁäè¡¡æĺąåİŮãĂĆ

è£IJçÍŃæĺąåİŮåŁäè¡¡èů§åĚűäżŰçŽĎåŁäè¡¡ä¡£çŤĺæŰźæşŢåĞääźŐæŸŕäÿĂæăůçŽĎãĂĆä¿ŃåęĆïijŽ

```
>>> fib
<urlmodule 'fib' from 'http://localhost:15000/fib.py'>
>>> fib.__name__
'fib'
>>> fib.__file__
'http://localhost:15000/fib.py'
>>> import inspect
>>> print(inspect.getsource(fib))
# fib.py
print("I'm fib")

def fib(n):
    if n < 2:
        return 1
    else:
```

(continues on next page)

(continued from previous page)

```
        return fib(n-1) + fib(n-2)
>>>
```

èőĺèőž

åIJĺèŕęçżĘèőĺèőžäźŃåĽ’■ïijŇæIJĽçĆźèęĄåijžèřČçŽĎæŸŕïijŇPythonçŽĎæĺąåİŮãĂĄåŇĚåŠŇåŕijåĚěæIJ
å■şä¡£çżŔéłŇäÿřåŕŇçŽĎPythonçĺŃåžŔåŚŸäź§å¿ĹåřŚèČ¡çš¿éĂŽåőČäżňãĂĆ
æĹŚåIJĺè£ŹéĞŇæŐĺè■ŘäÿĂäžŽåĂijçŽĎåŐżèřżçŽĎæŰĞæącåŠŇäźęçś■ïijŇåŇĚæŃň im-
portlib module åŠŇ PEP 302. æŰĞæącåĘĚåőźåIJĺè£ŹéĞŇäÿ■äijŽèćńéĞ■åď■æŔŘåĹřïijŇäÿ■è£ĞæĹŚåIJĺè£Ź

éęŰåĚĹïijŇåęĆæđIJä¡äæČşåĹŻåżžäÿĂäÿłæŰřçŽĎæĺąåİŮåŕźèśąïijŇä¡£çŤĺ imp.
new_module() åĞ¡æŢřïijŽ

```
>>> import imp
>>> m = imp.new_module('spam')
>>> m
<module 'spam'>
>>> m.__name__
'spam'
>>>
```

æĺąåİŮåŕźèśąéĂŽåÿÿæIJĽäÿĂäžŽæIJ§æIJŻåśđæĂğïijŇåŇĚæŃň __file__
ïijĹè£ŘèąŇæĺąåİŮåŁäè¡¡èŕ■åŘęçŽĎæŰĞäżűåŘ■ïijĽ åŠŇ __package__ (åŇĚåŘ■)ãĂĆ

åĚűæňąïijŇæĺąåİŮäijŽèćńèğčéĞŁåŹĺçijŞå■ŸèţůæİěãĂĆæĺąåİŮçijŞå■ŸåŔŕäżěåIJĺå■ŮåĚÿ
sys.modules äÿ■èćńæĽ¿åĹřãĂĆ åŽääÿžæIJĽäžĘè£ŹäÿłçijŞå■ŸæIJžåĹűïijŇéĂŽåÿÿåŔŕäżěåřEçijŞå■ŸåŠ

```
>>> import sys
>>> import imp
>>> m = sys.modules.setdefault('spam', imp.new_module('spam'))
>>> m
<module 'spam'>
>>>
```

åęĆæđIJçżŹåőŽæĺąåİŮåůšçżŔå■ŸåIJĺéĆčäźĹåřśäijŽçŻť’æŐěèŐůå¿ŮåůšçżŔèćńåĹŻåżžè£ĞçŽĎæĺąåİŮï

```
>>> import math
>>> m = sys.modules.setdefault('math', imp.new_module('math'))
>>> m
<module 'math' from '/usr/local/lib/python3.3/lib-dynload/math.so'>
>>> m.sin(2)
0.9092974268256817
>>> m.cos(2)
-0.4161468365471424
>>>
```

çŤśäžŐåĹŻåżžæĺąåİŮå¿ĹçőĂå■ŢïijŇå¿ĹåőźæŸŞçijŰåĘŹçőĂå■ŢåĞ¡æŢřæŕŤåęĆçňňäÿĂéČĺåĹĘçŽĎ
load_module() åĞ¡æŢřãĂĆ è£ŹäÿłæŰźæąĹçŽĎäÿĂäÿłçijžçĆźæŸŕå¿ĹéŽ¿åď ĎçŘĘåď■æİĆæČĚåĘţæŕŤ

äÿžäžĘåď ĎçŘĘäÿĂäÿłåŇĚïijŇä¡ăèęĄéĞ■æŰřåđçŐřæŹőéĂŽimportèŕ■åŘěçŽĎåžŢåśĆéĂżè¿ŚïijĹæŕŤåęĆa
æĽ ğèąŇéĆčäžŻæŰĞäżűïijŇèő¿ç¡őèůřå¿Ďç■Ľ ïijĽ ãĂĆè£Źäÿłåď ■æİĆæĂğåřśæŸŕäÿžäżĂäźĹæIJĂåě¡çŻť æŐ

æĽ ľ åśŢimportèŕ■åŘěå¿ĹçőĂå■ŢïijŇä¡ĘæŸŕäijŽæIJĽ å¿Ĺåď Žçğżå Ł ĺæŞ■ä¡IJãĂĆ
æIJĂéńŸåśĆäÿŁïijŇåŕijåĚěæŞ■ä¡IJèćńäÿĂäÿłä¡■äžŐsys.meta_pathåĹŮèąĺäÿ■çŽĎâĂIJåĚČèůŕå¿ĎâĂİæ§ěæ
åęĆæđIJä¡ăè¿ŞåĞžåőČçŽĎåĂijïijŇäijŽçIJŃåĹřäÿŃéİćè£ŹæăůïijŽ

```
>>> from pprint import pprint
>>> pprint(sys.meta_path)
[<class '_frozen_importlib.BuiltinImporter'>,
<class '_frozen_importlib.FrozenImporter'>,
<class '_frozen_importlib.PathFinder'>]
>>>
```

å¡ŞæĽ ğèąŇäÿĂäÿłèŕ■åŘěæŕŤåęĆ `import fib` æŮűïijŇèğčéĞŁåŹĺäijŽéĄ■åŐĘsys.mata_pathäÿ■çŽĎ
èřČçŤĺåőČäžňçŽĎ `find_module()` æŰźæşŢåőŽä¡■æ■čçąőçŽĎæĺąåİŮåŁäè¡¡åŹĺăĂĆ
åŘŕäżěéĂŽè£ĞåőđéłŇæİěçIJŃçIJŃïijŽ

```
>>> class Finder:
...     def find_module(self, fullname, path):
...         print('Looking for', fullname, path)
...         return None
...
>>> import sys
>>> sys.meta_path.insert(0, Finder()) # Insert as first entry
>>> import math
Looking for math None
>>> import types
Looking for types None
>>> import threading
Looking for threading None
Looking for time None
Looking for traceback None
Looking for linecache None
Looking for tokenize None
Looking for token None
>>>
```

æşĺæĎŘçIJŃ `find_module()` æŰźæşŢæŸŕæĂŐæăůåIJĺæŕŘäÿĂäÿłåŕijåĚěåřśèćńèğęåŘŚçŽĎãĂĆ
è£ŹäÿłæŰźæşŢäÿ■çŽĎpathåŘĆæŢřçŽĎä¡IJçŤĺæŸŕåď ĎçŘĘåŇĚãĂĆ
åď ŽäÿłåŇĚèćńåŕijåĚěïijŇåřśæŸŕäÿĂäÿłåŘŕåIJĺåŇĚçŽĎ `__path__`
åśđæĂğäÿ■æĽ ¿åĹřçŽĎèůŕå¿ĎåĹŮèąĺãĂĆ èęĄæĽ ¿åĹřåŇĚçŽĎå■ŘçżĎäżűåřśèęĄæčĂæ§ěè£ŹäżŻèůŕå¿ĎãĂ
æŕŤåęĆæşĺæĎŘåŕźäżŐ `xml.etree` åŠŇ `xml.etree.ElementTree`
çŽĎèůŕå¿ĎéĚ■ç¡ŏïijŽ

```
>>> import xml.etree.ElementTree
Looking for xml None
Looking for xml.etree ['/usr/local/lib/python3.3/xml']
Looking for xml.etree.ElementTree ['/usr/local/lib/python3.3/xml/
↪etree']
```

(continues on next page)

(continued from previous page)

```
Looking for warnings None
Looking for contextlib None
Looking for xml.etree.ElementPath ['/usr/local/lib/python3.3/xml/
↪etree']
Looking for _elementtree None
Looking for copy None
Looking for org None
Looking for pyexpat None
Looking for ElementC14N None
>>>
```

åIJÍ `sys.meta_path` äÿŁæ§ěæL'¿åŹÍçŽĎä¡■ç¡őå¿ÍéĞ■èęĄïijŇåřĘåőČäżŐéŸ§åď'ť'çğżåĹřéŸ§åř¿ïijŇ

```
>>> del sys.meta_path[0]
>>> sys.meta_path.append(Finder())
>>> import urllib.request
>>> import datetime
```

çŐřåIJÍä¡ăçIJŃäÿ■åĹřäżżä¡Ţè¿ŞåĞżäžĘïijŇåŻääÿżåŕijåĚěèćńsys.meta_pathäÿ■çŽĎåĚűäżŰåőđä¡Şåď'Ďç
è£ŹæŮűåĂŹïijŇä¡ääŘłæIJL'åIJÍåŕijåĚěäÿ■å■ŸåIJÍæÍąåİŮçŽĎæŮűåĂŹæL'■èČ¡çIJŃåĹřåőČèćńèğęåŘŚïijŽ

```
>>> import fib
Looking for fib None
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
ImportError: No module named 'fib'
>>> import xml.superfast
Looking for xml.superfast ['/usr/local/lib/python3.3/xml']
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
ImportError: No module named 'xml.superfast'
>>>
```

ä¡ääźŃåL'■åőL'èčĚè£ĞäÿĂäÿłæ■ŢèŐůæIJłç§ěæÍąåİŮçŽĎæ§ěæL'¿åŹÍïijŇè£ŹäÿłæŸŕ
`UrlMetaFinder` çśżçŽĎåĚşéŤőãĂĆ äÿĂäÿł `UrlMetaFinder` åőđä¿ŃèćńæůżåŁäåĹř
`sys.meta_path` çŽĎæIJńåř¿ïijŇä¡IJäÿżæIJĂåŘŐäÿĂäÿłæ§ěæL'¿åŹÍæŰźæąĹãĂĆ
åęĆæđIJèćńèřůæśĆçŽĎæÍąåİŮåŘ■äÿ■èČ¡åőŽä¡■ïijŇåřśäijŽèćńè£Źäÿłæ§ěæL'¿åŹÍåď'ĎçŘĘæŐL'ãĂĆ
åď'ĎçŘĘåŇĚçŽĎæŮűåĂŹéIJÄèęĄæşÍæĎŘïijŇåIJÍpathåŘĆæŢřäÿ■æŇĞåőŽçŽĎåĂijéIJÄèęĄèćńæčĂæ§ěïijŇ
åęĆæđIJäÿ■æŸŕïijŇèřěå■ŘæÍąåİŮå£Ěéążå¡ŠåśđäžŐåĚűäżŰæ§ěæL'¿åŹÍåźűèćńå£¡çŢěæŐL'ãĂĆ

åŕźäžŐåŇĚçŽĎåĚűäżŰåď'ĎçŘĘåŘŕåIJÍ `UrlPackageLoader`
çśżäÿ■èćńæL'¿åĹřãĂĆ è£Źäÿłçśżäÿ■äijŽåŕijåĚěåŇĚåŘ■ïijŇèĂŇæŸŕåŐżåŁäè¡¡åŕźåžŤçŽĎ
`__init__.py` æŰĞäżűãĂĆ åőČäź§äijŽèő¿ç¡őæÍąåİŮçŽĎ `__path__`
åśđæĂğïijŇè£ŹäÿĂæ■ěå¿ÍéĞ■èęĄïijŇ åŻääÿżåIJÍåŁäè¡¡åŇĚçŽĎå■ŘæÍąåİŮæŮűè£ŹäÿłåĂijäijŽèćńäijăçżŹå
`find_module()` èřČçŤÍãĂĆ å§žäžŐèůŕå¿ĎçŽĎåŕijåĚěéŠl'å■ŘæŸŕè£ŹäžŻæĂİæČşçŽĎäÿĂäÿłæL'l'åśŢïijŇ
æĹŚäżňéČ¡ç§ěéĄŞïijŇ`sys.path` æŸŕäÿĂäÿłPythonæ§ěæL'¿æÍąåİŮçŽĎçŻőå¡ŢåĹŮèąĺïijŇä¿ŃåęĆïijŽ

```
>>> from pprint import pprint
>>> import sys
```

(continues on next page)

(continued from previous page)

```
>>> pprint(sys.path)
['',
'/usr/local/lib/python33.zip',
'/usr/local/lib/python3.3',
'/usr/local/lib/python3.3/plat-darwin',
'/usr/local/lib/python3.3/lib-dynload',
'/usr/local/lib/...3.3/site-packages']
>>>
```

åIJÍ `sys.path` äÿ■çŽĎæŕŔäÿĂäÿłåŏđä¡ŞéČ¡äijŽèćńéćÌåď ŰçŽĎçżŚåŏŽåĹřäÿĂäÿłæ§ěæĽ¿åŹÍåŕźèśąäÿ ä¡ăåŔŕäžěéĂŽè£ĞæŞěçIJŃ `sys.path_importer_cache` åŐżçIJŃäÿŃè£ŹäžŻæŞěæĽ¿åŹÍïijŽ

```
>>> pprint(sys.path_importer_cache)
{'.': FileFinder('.'),
'/usr/local/lib/python3.3': FileFinder('/usr/local/lib/python3.3'),
'/usr/local/lib/python3.3/': FileFinder('/usr/local/lib/python3.3/
↪'),
'/usr/local/lib/python3.3/collections': FileFinder('...python3.3/
↪collections'),
'/usr/local/lib/python3.3/encodings': FileFinder('...python3.3/
↪encodings'),
'/usr/local/lib/python3.3/lib-dynload': FileFinder('...python3.3/
↪lib-dynload'),
'/usr/local/lib/python3.3/plat-darwin': FileFinder('...python3.3/
↪plat-darwin'),
'/usr/local/lib/python3.3/site-packages': FileFinder('...python3.3/
↪site-packages'),
'/usr/local/lib/python33.zip': None}
>>>
```

`sys.path_importer_cache` æŕŤ `sys.path` äijŽæŻť åď ğçĆźïijŃ åŻääÿžåőČäijŽäÿžæĽĂæIJĽèćńåŁäè¡¡äżčçăĄçŽĎçŻőå¡Ţèőŕå¡ŢåőČäżňçŽĎæ§ěæĽ¿åŹÍãĂĆ è£ŹåŃĚæŃňåŃĚçŽĎå■ŘçŻőå¡ŢïijŃè£ŹäžŻéĂŽåÿÿåIJÍ `sys.path` äÿ■æŸŕäÿ■å■ŸåIJÍçŽĎãĂĆ

èęĄæĽğèąŃ `import fib` ïijŃäijŽéąžåžŔæčĂæ§ě `sys.path` äÿ■çŽĎçŻőå¡ŢãĂĆ åŕźäžŐæŕŔäÿłçŻőå¡ŢïijŃåŘ■çğŕâĂIJfibâĂİäijŽèćńäijăçżŹçŻÿåžŤçŽĎ `sys.path_importer_cache` äÿ■çŽĎæ§ěæĽ¿åŹÍãĂĆ è£ŹäÿłåŔŕäżěèőľä¡ăåĹŻåżžèĞłåůśçŽĎæ§ěæĽ¿åŹÍåźűå

```
>>> class Finder:
... def find_loader(self, name):
...     print('Looking for', name)
...     return (None, [])
...
>>> import sys
>>> # Add a "debug" entry to the importer cache
>>> sys.path_importer_cache['debug'] = Finder()
>>> # Add a "debug" directory to sys.path
```

(continues on next page)

(continued from previous page)

```
>>> sys.path.insert(0, 'debug')
>>> import threading
Looking for threading
Looking for time
Looking for traceback
Looking for linecache
Looking for tokenize
Looking for token
>>>
```

åIJÍè£ŹéĞŇïijŇä¡ăåŘŕäżěäÿžåŘ■å■ŮâĂIJdebugâĂİåĹŻåżžäÿĂäÿłæŰřçŽĎçijŞå■Ÿåðđä¡Ş åźűåřĘåðČèð¿
sys.path äÿŁçŽĎçňňäÿĂäÿłãĂĆ åIJÍæĽĂæIJĽæŐěäÿŃæİěçŽĎåŕijåĚěäÿ■ïijŇä¡ääijŽçIJŃåĹřä¡äçŽĎæ§ěä
äÿ■è£ĞïijŇçŤśäžŐåðČè£ŤåŻđ (None, [])ïijŇéĆčäźĹåď ĎçŘĘè£ŻçÍŃäijŽçżğçż■åď ĎçŘĘäÿŃäÿĂäÿłåðđä¡ŞãĂ

sys.path_importer_cache çŽĎä¡£çŤÍèćńäÿĂäÿłå■ŸåĆÍåIJÍ sys.path_hooks
äÿ■çŽĎåĞ¡æŢřåĹŮèąĺæŐğåĹűãĂĆ èŕŢèŕŢäÿŃéİćçŽĎä¿Ńå■ŘïijŇåðČäijŽæÿĚéŹď çijŞå■Ÿåźűçżź
sys.path_hooks æůżåŁääÿĂäÿłæŰřçŽĎèůŕå¿ĎæčĂæ§ěåĞ¡æŢř

```
>>> sys.path_importer_cache.clear()
>>> def check_path(path):
...     print('Checking', path)
...     raise ImportError()
...
>>> sys.path_hooks.insert(0, check_path)
>>> import fib
Checked debug
Checking .
Checking /usr/local/lib/python33.zip
Checking /usr/local/lib/python3.3
Checking /usr/local/lib/python3.3/plat-darwin
Checking /usr/local/lib/python3.3/lib-dynload
Checking /Users/beazley/.local/lib/python3.3/site-packages
Checking /usr/local/lib/python3.3/site-packages
Looking for fib
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
ImportError: No module named 'fib'
>>>
```

æ■čåęĆä¡äæĽĂèğĄïijŇcheck_path() åĞ¡æŢřèćńæŕŘäÿł sys.path
äÿ■çŽĎåðđä¡ŞěřČçŤĺãĂĆ äÿ■éą¿ïijŇçŤśäžŐæŁŻåĞžäžĘ ImportError åijĆåÿÿïijŇ
åŢěéČ¡äÿ■äijŽåŘŚçŤ§äžĘïijĹäžĚäżĚåřĘæčĂæ§ěè¡ňçğżåĹřsys.path_hooksçŽĎäÿŃäÿĂäÿłåĞ¡æŢřïijĽãĂĆ

ç§ěéĄŞäžĘæĂŐæăůsys.pathæŸŕæĂŐæăůèćńåď ĎçŘĘçŽĎïijŇä¡äářśèČ¡æđĎåżžäÿĂäÿłèĞłåŐŽäźĽèůŕå¿

```
>>> def check_url(path):
...     if path.startswith('http://'):
...         return Finder()
...     else:
...         raise ImportError()
```

(continues on next page)

(continued from previous page)

```
...
>>> sys.path.append('http://localhost:15000')
>>> sys.path_hooks[0] = check_url
>>> import fib
Looking for fib # Finder output!
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
ImportError: No module named 'fib'

>>> # Notice installation of Finder in sys.path_importer_cache
>>> sys.path_importer_cache['http://localhost:15000']
<__main__.Finder object at 0x10064c850>
>>>
```

è£ŹåřśæŸŕæIJňèŁĆæIJĂåŘŐéČĺåĹĘçŽĎåĚşéŤőçĆźãĂĆäžŃåőđäÿŁïijŇäÿĂäÿłçŤĺæĺěåIJĺsys.pathäÿ■æ
å¡ŞåőČäżňèćńçćřåĹřçŽĎæŮűåĂŹïijŇäÿĂäÿłæŰřçŽĎ UrlPathFinder
åőđä¿ŃèćńåĹŻåżžåźűèćńæŤ¿åĚě sys.path_importer_cache.
äźŃåŘŐïijŇæĽĂæIJĽéIJĂèęĄæčĂæ§ě sys.path çŽĎåŕijåĚěèŕ■åŘěéČ¡äijŽä¡£çŤĺä¡ăçŽĎèĞłåőŽäźĽæ§ěa

å§žäžŐèůŕå¿ĎåŕijåĚěçŽĎåŇĚåď ĎçŘĘçĺ■å¿őæIJĽçĆźåď■æĺĆïijŇåźűäÿŤèů§
find_loader() æŰźæşŢè£ŤåŻđåĂijæIJĽåĚşãĂĆ åŕźäžŐçőĂå■ŢæĺąåĺŮïijŇfind_loader()
è£ŤåŻđäÿĂäÿłåĚČçżĎ(loader, None)ïijŇ åĚűäÿ■çŽĎload-
eræŸŕäÿĂäÿłçŤĺäžŐåŕijåĚěæĺąåĺŮçŽĎåŁäè¡¡åŹĺåőđä¿ŃãĂĆ

åŕźäžŐäÿĂäÿłæŹőéĂŽçŽĎåŇĚïijŇfind_loader() è£ŤåŻđäÿĂäÿłåĚČçżĎ(loader,
path)ïijŇ åĚűäÿ■çŽĎloaderæŸŕäÿĂäÿłçŤĺäžŐåŕijåĚěåŇĚïijĹåźűæĽğèąŇ__init__.pyïijĽçŽĎåŁäè¡¡åŹĺåőđä¿
pathæŸŕäÿĂäÿłäijŽåĹİåğŃåŇŰåŇĚçŽĎ __path__ åśđæĂğçŽĎçŻőå¡ŢåĹŮèąĺãĂĆ
ä¿ŃåęĆïijŇåęĆæđIJå§žçąĂURLæŸŕ http://localhost:15000 åźűäÿŤäÿĂäÿłçŤĺæĹůæĽğèąŇ
import grok , éĆčäźĹ find_loader() è£ŤåŻđçŽĎpathåřśäijŽæŸŕ [âĂŸhttp:
//localhost:15000/grokâĂŹ]

find_loader() è£ŸèęĄèČ¡åď ĎçŘĘäÿĂäÿłåŚ¡åŘ■çľžéŮť åŇĚäÿĂäÿłåŚ¡åŘ■çľžéŮť åŇĚä�ÿ■æIJĽäÿĂäÿłåŘĹæşŢçŽĎåŇĚçŻőå¡ŢåŘ■ïijŇä¡ĘæŸŕäÿ■å■ŸåIJĺ__init__.pyæŰ
è£ŹæăůçŽĎèŕĺïijŇfind_loader() å£Ĕéążè£ŤåŻđäÿĂäÿłåĚČçżĎ(None, path)ïijŇ
pathæŸŕäÿĂäÿłçŻőå¡ŢåĹŮèąĺïijŇçŤśåőČæĺĕæđĎåżžåŇĚçŽĎåőŽäźĽæIJĽ__init__.pyæŰĞäżűçŽĎ__path__
åŕźäžŐè£Źçğ■æČĚåĘţïijŇåŕijåĚěæIJžåĹűäijŽçżğçż■åĽ■èąŇåŐżæčĂæ§ěsys.pathäÿ■çŽĎçŻőå¡ŢăĂĆ
åęĆæđIJæĽ¿åĹřäžĘąåŚ¡åŘ■çľžéŮť åŇĚïijŇæĽĂæIJĽçŽĎçżŞæđIJèůŕå¿ĎèćńåŁäåĹřäÿĂèţůæĺĕæđĎåżžæIJĂ
åĚşäžŐåŚ¡åŘ■çľžéŮť åŇĚçŽĎæŻť åď Žä£ąæĄŕèŕůåŘĆèĂČ10.5åřŘèŁĆăĂĆ

æĽĂæIJĽçŽĎåŇĚéČ¡åŇĚåŘńäžĘäÿĂäÿłåĘĚéČĺèůŕå¿Ďèő¿ç¡őïijŇåŘŕäžěåIJĺ__path__åśđæĂğäÿ■çIJŃ

```
>>> import xml.etree.ElementTree
>>> xml.__path__
['/usr/local/lib/python3.3/xml']
>>> xml.etree.__path__
['/usr/local/lib/python3.3/xml/etree']
>>>
```

äźŃåĽ■æŘŘåĹřïijŇ__path__çŽĎèő¿ç¡őæŸŕéĂŽè£Ğ find_loader()
æŰźæşŢè£ŤåŻđåĂijæŐğåĹűçŽĎăĂĆ äÿ■è£ĞïijŇ__path__æŐěäÿŃæĺěäź§èćńsys.path_hooksäÿ■çŽĎåĞ¡æŢ

åŻžæ■ď'ïijŇä¡ĘåŇĚçŽĎå■ŘçżĎäžűèćńåŁäè¡¡åŘŐïijŇä¡■äžŐ__path__äÿ■çŽĎåőđä¡ŞäijŽèćń `handle_url()` åĞ¡æŢřæčĂæ§ěãĂĆ è£ŹäijŽåřijèĞť'æŰřçŽĎ `UrlPathFinder` åőđä¿ŃèćńåĹŻåżžåźűäÿŤèćńåŁäåĚěåĹř `sys.path_importer_cache` äÿ■ãĂĆ

è£ŸæIJL'äÿłéŽ¿çĆźåřśæŸŕ `handle_url()` åĞ¡æŢřäżěåŘŁåőČèů§åĘĚéČĹä¡£çŤĹçŽĎ `_get_links()` åĞ¡æŢřäźŃéŮť'çŽĎäžď'äžŠãĂĆ åęĆæđIJä¡äçŽĎæ§ěæL'¿åŹĹåőđçŐřéIJĂèęĄä¡£çŤĹåĹřåĚűæIJL'åŘřèČ¡è£ŹäžŻæĹąåİŮäijŽåIJĹæ§ěæL'¿åŹĹæŞ■ä¡IJæIJ§éŮť'è£ŻèąŇæŻť'åď'ŽçŽĎåřijåĚěãĂĆ åőČåŘřäżěåřijèĞť' `handle_url()` åŠŇåĚűäżŰæ§ěæL'¿åŹĹéČĹåĹĘéŹůåĚěäÿĂçğ■éĂŠå¡Šå¿łçŐřçŁűæĂĄä äÿžäžĘèğčéĞŁè£Źçğ■åŘřèČ¡æĂğïijŇåőđçŐřäÿ■æIJL'äÿĂäÿłèćńåĹŻåżžçŽĎæ§ěæL'¿åŹĹçijŞå■ŸïijĹæřŘäÿĂ åőČåŘřäżěéĄ£åĚ■åĹŻåżžéĞ■åď'■æ§ěæL'¿åŹĹçŽĎéŮőéćŸãĂĆ åŘęåď'ŰïijŇäÿŃéİćçŽĎäžčçăĄçL'Ğæőţ åŘřäżěçąőä£İæ§ěæL'¿åŹĹäÿ■äijŽåIJĹåĹİåğŃåŇŰéŞ¿æŐěéŻĘåŘĹçŽĎ

```
# Check link cache
if self._links is None:
    self._links = [] # See discussion
    self._links = _get_links(self._baseurl)
```

æIJĂåŘŐïijŇæ§ěæL'¿åŹĹçŽĎ `invalidate_caches()` æŰźæşŢæŸřäÿĂäÿłåůěåĚůæŰźæşŢïijŇçŤĹæİěæÿĚçŘĘåĘĚéČĹçijŞå■ŸãĂĆ è£ŹäÿłæŰźæşŢåĘ■çŤĹæĹůèřČçŤĹ `importlib.invalidate_caches()` çŽĎæŮůåĂŹèćńèğęåŘŚãĂĆ åęĆæđIJä¡äæČşèől'URLåřijåĚěèĂĚéĞ■æŰřèřżåŘŰéŞ¿æŐěåĹŮèąĹçŽĎèřĺåŘřä

åřźæřŤäÿŃäÿď'çğ■æŰźæąĹïijĹä£őæŤźsys.meta_pathæĹŰä¡£çŤĹäÿĂäÿłèůřå¿ĎéŠl'å■ŘïijL'ãĂĆ ä¡£çŤĹsys.meta_pathçŽĎåřijåĚěèĂĚåŘřäżěæŇL'çĚğèĞłåůśçŽĎéIJĂèęĄèĞłçŤśåď'ĎçŘĘæĹąåİŮãĂĆ ä¿ŃåęĆïijŇäőČäżňåŘřäżěäżŐæŢřæ■őäžŞäÿ■åřijåĚěæĹŰäżěäÿ■åŘŇäžŐäÿĂèĹňæĹąåİŮ/åŇĚåď'ĎçŘĘæŰźåi è£Źçğ■èĞłçŤśåŘŇæäůæĎŘåŚşçİĂåřijåĚěèĂĚéIJĂèęĄèĞłåůśè£ŻèąŇåĘĚéČĹçŽĎäÿĂäžŻçőąçŘĘãĂĆ åŘęåď'ŰïijŇå§žäžŐèůřå¿ĎçŽĎéŠl'å■ŘåŘłæŸŕéĂĆçŤĹäžŐåřźsys.pathçŽĎåď'ĎçŘĘãĂĆ éĂŽè£Ğè£Źçğ■æL'l'åśŢåŁäè¡¡çŽĎæĹąåİŮèů§æŹőéĂŽæŰźåijŘåŁäè¡¡çŽĎçL'źæĂğæŸŕäÿĂæäůçŽĎãĂĆ

åęĆæđIJåĹřçŐřåIJĹäÿžæ■ćä¡äè£ŸæŸŕäÿ■æŸŕå¿ĹæŸŐçŹ¡ïijŇéĆčäźĹåŘřäżěéĂŽè£ĞåćđåŁääÿĂäžŻæŮ

```
>>> import logging
>>> logging.basicConfig(level=logging.DEBUG)
>>> import urlimport
>>> urlimport.install_path_hook()
DEBUG:urlimport:Installing handle_url
>>> import fib
DEBUG:urlimport:Handle path? /usr/local/lib/python33.zip. [No]
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
ImportError: No module named 'fib'
>>> import sys
>>> sys.path.append('http://localhost:15000')
>>> import fib
DEBUG:urlimport:Handle path? http://localhost:15000. [Yes]
DEBUG:urlimport:Getting links from http://localhost:15000
DEBUG:urlimport:links: {'spam.py', 'fib.py', 'grok'}
DEBUG:urlimport:find_loader: 'fib'
DEBUG:urlimport:find_loader: module 'fib' found
DEBUG:urlimport:loader: reading 'http://localhost:15000/fib.py'
DEBUG:urlimport:loader: 'http://localhost:15000/fib.py' loaded
```

(continues on next page)

(continued from previous page)

```
I'm fib
>>>
```

æIJĂåŘŐïijŇåżżèőőä¡ăèŁśçĆźæŮűéŮť'çIJŃçIJŃ PEP 302 äżěåŘŁimportlibçŽĎæŰĞæąčãĂĆ

12.12 10.12 åŕijåĚěæĺąåİŮçŽĎåŘŇæŮűä£őæŤźæĺąåİŮ

éŮőécŸ

ä¡ăæČşçżŹæ§Řäÿłåůšå■ŸåIJĺæĺąåİŮäÿ■çŽĎåĞ¡æŢřæůżåŁăèčĚéěřåŹĺãĂĆäÿ■è£ĞïijŇåĽ■æŘŘæŸŕè£ŹäÿłæĺąåİŮåůšçżŘèćńåŕijåĚěåźűäÿŤèćńä¡£çŤĺè£ĞãĂĆ

èğčåĘşæŰźæąĹ

è£ŹéĞŇéŮőécŸçŽĎæIJňèť'ĺåŕśæŸŕä¡ăæČşåIJĺæĺąåİŮèćńåŁăè¡¡æŮűæĽğèąŇæ§ŘäÿłåŁĺä¡IJãĂĆåŘŕèČ¡æŸŕä¡ăæČşåIJĺäÿĂäÿłæĺąåİŮèćńåŁăè¡¡æŮűèğęåŘŚæ§ŘäÿłåŻđèřČåĞ¡æŢřæĺěéĂŽç§ěä¡ããĂĆ

è£ŹäÿłéŮőécŸåŘŕäżěä¡£çŤĺ10.11åřŘèŁĆäÿ■åŘŇæůçŽĎåŕijåĚěéŠľå■ŘæIJžåĹűæĺěåőđçŐřãĂĆäÿŃéĺć

```
# postimport.py
import importlib
import sys
from collections import defaultdict

_post_import_hooks = defaultdict(list)

class PostImportFinder:
    def __init__(self):
        self._skip = set()

    def find_module(self, fullname, path=None):
        if fullname in self._skip:
            return None
        self._skip.add(fullname)
        return PostImportLoader(self)

class PostImportLoader:
    def __init__(self, finder):
        self._finder = finder

    def load_module(self, fullname):
        importlib.import_module(fullname)
        module = sys.modules[fullname]
        for func in _post_import_hooks[fullname]:
            func(module)
```

(continues on next page)

(continued from previous page)

```
        self._finder._skip.remove(fullname)
        return module

def when_imported(fullname):
    def decorate(func):
        if fullname in sys.modules:
            func(sys.modules[fullname])
        else:
            _post_import_hooks[fullname].append(func)
        return func
    return decorate

sys.meta_path.insert(0, PostImportFinder())
```

è£ŹæăůïijŇä¡ăářśăŘŕäżěä¡£çŤÍ `when_imported()` èčĚéěřăŹÍäžĘïijŇä¿ŃăęĆïijŽ

```
>>> from postimport import when_imported
>>> @when_imported('threading')
... def warn_threads(mod):
...     print('Threads? Are you crazy?')
...
>>>
>>> import threading
Threads? Are you crazy?
>>>
```

ä¡IJäÿžäÿĂäÿłæŻť'ăőđéŹĚçŽĎä¿Ńă■ŘïijŇä¡ăăŘŕèČ¡æČşăIJÍăůšă■ŸăIJÍçŽĎăőŽäźĽäÿŁéİćæůżăŁăèčĚé

```
from functools import wraps
from postimport import when_imported

def logged(func):
    @wraps(func)
    def wrapper(*args, **kwargs):
        print('Calling', func.__name__, args, kwargs)
        return func(*args, **kwargs)
    return wrapper

# Example
@when_imported('math')
def add_logging(mod):
    mod.cos = logged(mod.cos)
    mod.sin = logged(mod.sin)
```

èőÍèőž

æIJňèŁĆæŁĂæIJŕä¿ÌètŰäžŐ10.11ářŘèŁĆäÿ■èőšè£řè£ĞçŽĎăŕijăĚěéŠľă■ŘïijŇăźűçÍ■ä¡IJä£őæŤźăĂĆ

`@when_imported` èčĚéěřăŹÍçŽĎä¡IJçŤÍæŸŕæşÍăĘŇăIJÍăŕijăĚěæŮűèćńæ£Ăæť'żçŽĎăď'ĎçŘĘăŹÍăĞ¡a

èŕěèčĚéěřåŹÍæčĂæ§ěsys.modulesæİěæ§ěçIJŃæĺąåİŮæŸŕåŘęçIJ§çŽĎåůšçżŔèćńåŁăè¡¡äžĘãĂĆ
åęĆæđIJæŸŕçŽĎèŕÍïijŇèŕěåď ĎçŘĘåŹÍèćńçńŃå■şèřČçŤÍãĂĆäÿ■çĎűïijŇåď ĎçŘĘåŹÍèćńæůżåŁăåĹř
`_post_import_hooks` å■ŮåĚÿäÿ■çŽĎäÿĂäÿłåĹŮèąĺäÿ■åŐżãĂĆ
`_post_import_hooks` çŽĎä¡IJçŤÍåŕśæŸŕæŤűéŻĘæĽĂæIJĽçŽĎäÿžæŕŔäÿłæĺąåİŮæşĺåĘŇçŽĎåď ĎçŘĘ
äÿĂäÿłæĺąåİŮåŘŕäžěæşĺåĘŇåď Žäÿłåď ĎçŘĘåŹÍãĂĆ

èęĄèőľæĺąåİŮåŕijåĚěåŘŐèğęåŘŚæůżåŁăçŽĎåŁĺä¡IJïijŇ`PostImportFinder`
çśżèćńèő¿ç¡őäÿžsys.meta_pathçňňäÿĂäÿłåĚČçť ăăĂĆ åőČäijŽæ■ŢèŐůæĽĂæIJĽæĺąåİŮåŕijåĚěæŞ■ä¡IJãĂĆ

æIJňèŁĆäÿ■çŽĎ `PostImportFinder` çŽĎä¡IJçŤÍåźűäÿ■æŸŕåŁăè¡¡æĺąåİŮïijŇèĂŇæŸŕèĞłåÿęåŕijåĚ
åőđéŹĚçŽĎåŕijåĚěèćńåğŤæť ¿çżŹä¡■äžŐsys.meta_pathäÿ■çŽĎåĚűäżŰæ§ěæĽ¿åŹÍãĂĆ
`PostImportLoader` çśżäÿ■çŽĎ `imp.import_module()`
åĞ¡æŢřèćńéĂŠå¡ŠçŽĎèřČçŤÍãĂĆ äÿžäžĘéĄ£åĚ■éŹůåĚěæŮăçž£å¿łçŐŕïijŇ`PostImportFinder`
ä£ÍæŇĄäžĘäÿĂäÿłæĽĂæIJĽèćńåŁăè¡¡è£ĞçŽĎæĺąåİŮéŻĘåŘĹãĂĆ
åęĆæđIJäÿĂäÿłæĺąåİŮåŘ■å■ŸåIJÍåŕśäijŽçŻť æŐěèćńå£¡çŢěæŐĽãĂĆ

å¡ŞäÿĂäÿłæĺąåİŮèćń `imp.import_module()` åŁăè¡¡åŘŐïijŇ
æĽĂæIJĽåIJÍ_post_import_hooksèćńæşĺåĘŇçŽĎåď ĎçŘĘåŹÍèćńèřČçŤÍïijŇä¡£çŤÍæŰřåŁăè¡¡æĺąåİŮä¡IJäÿžä

æIJĽäÿĂçĆźéIJĂèęĄæşĺæĎŘçŽĎæŸŕæIJňæIJžäÿ■éĂĆçŤÍäžŐéĆčäžŻéĂŽè£Ğ `imp.reload()` èćńæŸ¿åijŘåŁăè¡¡çŽĎæĺąåİŮãĂĆ äź§åŕśæŸŕèŕť ïijŇåęĆæđIJä¡ăåŁăè¡¡äÿĂäÿłäźŃåĽ■åůšèćńåŁă
åŘęåď ŰïijŇèęĄæŸŕä¡ăäžŐsys.modulesäÿ■åĹăéŹď æĺąåİŮçĎűåŘŐåĘ■éĞ■æŰřåŕijåĚěïijŇåď ĎçŘĘåŹÍåŘĹä

æŻť åď ŽåĚşäžŐåŕijåĚěåŘŐéŠľ å■Ŕä£ąæĄŕèŕůåŘĆèĂČ PEP 369.

12.13 10.13 åőĽèčĚçğĄæIJĽçŽĎåŇĚ

éŮőéćŸ

ä¡ăæČşèęĄåőĽèčĚäÿĂäÿłçňňäÿĽæŰźåŇĚïijŇä¡ĘæŸŕæšąæIJĽæİČéŹŘåřĘåőČåőĽèčĚåĹřçşżçż§Python
æĹŰèĂĚïijŇä¡ăåŘŕèČ¡æČşèęĄåőĽèčĚäÿĂäÿłä¿ŻèĞłåůśä¡£çŤÍçŽĎåŇĚïijŇèĂŇäÿ■æŸŕçşżçż§äÿŁéĺćæĽĂæ
åŘęåď ŰïijŇèęĄæŸŕä¡ăäžŐsys.modulesäÿ■åĹăéŹď æĺąåİŮçĎűåŘŐåĘ■éĞ■æŰřåŕijåĚěïijŇåď ĎçŘĘåŹÍåŘĹä

æŻť åď ŽåĚşäžŐåŕijåĚěåŘŐéŠľ å■Ŕä£ąæĄŕèŕůåŘĆèĂČ PEP 369.

èğčåĘşæŰźæąĹ

PythonæIJĽäÿĂäÿłçŤÍæĹůåőĽèčĚçŽőå¡ŢïijŇéĂŽåÿÿçśżäijijâĂİ~/.local/lib/python3.3/site-packagesâĂİãĂĆ èęĄåijžåĹűåIJÍè£ŹäÿłçŽőå¡Ţäÿ■åőĽèčĚåŇĚïijŇåŘŕä¡£çŤÍåőĽèčĚéĂĽéąźâĂIJ–userâĂİãĂ

```
python3 setup.py install --user
```

æĹŰèĂĚ

```
pip install --user packagename
```

åIJÍsys.pathäÿ■çŤÍæĹůçŽĎâĂIJsite-packagesâĂİçŽőå¡Ţä¡■äžŐçşżçż§çŽĎâĂIJsite-packagesâĂİçŽőå¡ŢäźŃåĽ■ãĂĆ åŽäæ■ď ïijŇä¡ăåőĽèčĚåIJÍéĞŇéİćçŽĎåŇĚåřśæŕŤçşżçż§åůšåőĽèčĚçŽĎåŇ
ïijĹåř¡çőąåźűäÿ■æĂżæŸŕè£ŹæăůïijŇèęĄåŘŰåĘşäžŐçňňäÿĽæŰźåŇĚçőąçŘĘåŹÍïijŇæŕŤåęĆdistributeæĹŰpi

èőÍèőž

éĂŽåÿÿåŇĚäijŽèćńåőL'èčĚåĹřçşżçż§çŽĎsite-packagesçŻőå¡Ţäÿ■åŐżïijŇèůŕå¿ĎçśżäijijâĂIJ/usr/local/l
packagesâĂİãĂĆ äÿ■è£ĞïijŇè£ŹæăůåĄŽéIJĂèęĄæIJL'çőąçŘĘåŚŸæİČéŹŘåźűäÿŤä¡£çŤĺsudoåŚ¡äżd'ãĂĆ
ářśçőŮä¡äæIJL'è£ŹæăůçŽĎæİČéŹŘåŐżæL'ğèąŇåŚ¡äżd'ïijŇä¡£çŤĺsudoåŐżåőL'èčĚäÿĂäÿłæŰřçŽĎïijŇåŘŕèČ

åőL'èčĚåŇĚåĹřçŤĺæĹůçŻőå¡Ţäÿ■éĂŽåÿÿæŸŕäÿĂäÿłæIJL'æŢĹçŽĎæŰźæąĹïijŇåőČåĚĄèőÿä¡äåĹŻåżżä

åŘęåď'ŰïijŇä¡äè£ŸåŘŕäżěåĹŻåżżäÿĂäÿłèŹŽæŃ§çŐŕåćČïijŇè£ŹäÿłæĹŚäżňåIJĺäÿŃäÿĂèŁĆäijŽèőšåĹřä

12.14 10.14 åĹŻåżżæŰřçŽĎPythonçŐŕåćČ

éŮőécŸ

ä¡äæČşåĹŻåżżäÿĂäÿłæŰřçŽĎPythonçŐŕåćČïijŇçŤĺæİěåőL'èčĚæĺąåİŮåŠŇåŇĚãĂĆ
äÿ■è£ĞïijŇä¡ääÿ■æČşåőL'èčĚäÿĂäÿłæŰřçŽĎPythonåĚŃéŽĘïijŇäź§äÿ■æČşåŕźçşżçż§PythonçŐŕåćČäžğçŤ§

èğčåĘşæŰźæąĹ

ä¡ääŘŕäżěä¡£çŤĺ `pyvenv` åŚ¡äżd'åĹŻåżżäÿĂäÿłæŰřçŽĎâĂIJèŹŽæŃ§âĂİçŐŕåćČãĂĆ
è£ŹäÿłåŚ¡äżd'èćńåőL'èčĚåIJĺPythonèğčéĞŁåŹĺåŘŇäÿĂçŻőå¡ŢïijŇæĹŰWindowsäÿŁéİćçŽĎScriptsçŻőå¡Ţäÿ

```
bash % pyvenv Spam
bash %
```

äijăçżŹ `pyvenv` åŚ¡äżd'çŽĎåŘ■å■ŮæŸŕåřĘèęĄèćńåĹŻåżżçŽĎçŻőå¡ŢåŘ■ãĂĆå¡ŞèćńåĹŻåżżåŘŐïijŇS

```
bash % cd Spam
bash % ls
bin include lib pyvenv.cfg
bash %
```

åIJĺbinçŻőå¡Ţäÿ■ïijŇä¡ääijŽæL'¿åĹřäÿĂäÿłåŘŕäżěä¡£çŤĺçŽĎPythonèğčéĞŁåŹĺïijŽ

```
bash % Spam/bin/python3
Python 3.3.0 (default, Oct 6 2012, 15:45:22)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more␣
↪information.
>>> from pprint import pprint
>>> import sys
>>> pprint(sys.path)
['',
'/usr/local/lib/python33.zip',
'/usr/local/lib/python3.3',
'/usr/local/lib/python3.3/plat-darwin',
'/usr/local/lib/python3.3/lib-dynload',
```

(continues on next page)

(continued from previous page)

```
'/Users/beazley/Spam/lib/python3.3/site-packages']
>>>
```

è£ŹäÿłèğčéĞŁåŹÍçŽĎçL'źçĆźåřśæŸŕäżŰçŽĎsite-packagesçŻőå¡Ţèćńèő¿ç¡őäÿžæŰřåĹŻåżžçŽĎçŐřåćČ
åęĆæđIJä¡äèęĄåőL'èčĚçňňäÿL'æŰźåŇĚïijŇåőČäžňäijŽèćńåőL'èčĚåIJÍéĆčéĞŇïijŇèĂŇäÿ■æŸŕéĂŽåÿÿçşżçż
packagesçŻőå¡ŢãĂĆ

èőÍèőž

åĹŻåżžèŹŽæŃ§çŐŕåćČéĂŽåÿÿæŸŕäÿžäžĘåőL'èčĚåŠŇçőąçŘĘçňňäÿL'æŰźåŇĚãĂĆ
æ■čåęĆä¡ääIJÍä¿Ńå■Řäÿ■çIJŃåĹřçŽĎéĆčæăůïijŇsys.path
åŘŸéĞŘåŇĚåŘńæÍěèĞłäžŐçşżçż§PythonçŽĎçŻőå¡ŢïijŇ èĂŇ site-
packagesçŻőå¡ŢåůšçżŕèćńéĞ■åőŽä¡■åĹřäÿĂäÿłæŰřçŽĎçŻőå¡ŢãĂĆ

æIJL'äžĘäÿĂäÿłæŰřçŽĎèŹŽæŃ§çŐŕåćČïijŇäÿŃäÿĂæ■ěåŕśæŸŕåőL'èčĚäÿĂäÿłåŇĚçőąçŘĘåŹÍïijŇæŕŤå
ä¡ĘåőL'èčĚè£ŹæăůçŽĎåůěåĚůåŠŇåŇĚçŽĎæŮűåĂŹïijŇä¡äéIJĂèęĄçąőä£Íä¡ää¡£çŤÍçŽĎæŸŕèŹŽæŃ§çŐŕåćČ
åőČäijŽåřĘåŇĚåőL'èčĚåĹřæŰřåĹŻåżžçŽĎsite-packagesçŻőå¡Ţäÿ■åŐżãĂĆ

åř¡çőąäÿĂäÿłèŹŽæŃ§çŐŕåćČçIJŃäÿŁåŐżæŸŕPythonåőL'èčĚçŽĎäÿĂäÿłåd'■åĹűïijŇ
äÿ■è£ĞåőČåőđéŹĚäÿŁåŘłåŇĚåŘńäžĘåřŚéĞŘåĞääÿłæŰĞäżűåŠŇäÿĂäžŻçňęåŘůéŞ¿æŐěãĂĆ
æL'ĂæIJL'æăĞåĞĘåžŞåĞ¡æŰĞäżűåŠŇåŘŕæL'ğèąŇèğčéĞŁåŹÍéČ¡æÍěèĞłåŐ§æÍěçŽĎPythonåőL'èčĚãĂĆ
åŽăæ■d'ïijŇåĹŻåżžè£ŹæăůçŽĎçŐřåćČæŸŕå¿Ĺåőźæ ŸŞçŽĎïijŇåźűäÿŤåĞääźŐäÿ■äijŽæűĹèĂŮæIJżåŹÍèţĎæ

éżŸèőd'æČĚåĘţäÿŃïijŇèŹŽæŃ§çŐŕåćČæŸŕçl'žçŽĎïijŇäÿ■åŇĚåŘńäżżä¡ŢéćÍåd'ŰçŽĎçňňäÿL'æŰźåżŞ
åŘŕäżěä¡£çŤÍâĂIJ–system-site-packagesâĂÍéĂL'éąźæÍěåĹŻåżžèŹŽæŃ§çŐŕåćČïijŇä¿ŃåęĆïijŽ

```
bash % pyvenv --system-site-packages Spam
bash %
```

èů§åd'ŽåĚşäžŐ `pyvenv` åŠŇèŹŽæŃ§çŐŕåćČçŽĎä£ąæĄŕåŘŕäżěåŘĆèĂČ PEP 405.

12.15 10.15 åĹĘåŘŚåŇĚ

éŮőéćŸ

ä¡ääůšçżŘçijŰåĘŹäžĘäÿĂäÿłæIJL'çŤÍçŽĎäžŞïijŇæČşåřĘåőČåĹĘäžńçżŹåĚűäżŰäžžãĂĆ

èğčåĘşæŰźæąĹ

åęĆæđIJä¡äæČşåĹĘåŘŚä¡äçŽĎäżčçăĄïijŇçňňäÿĂäżűäžŃåřśæŸŕçżŹåőČäÿĂäÿłåŤŕäÿĂçŽĎåŘ■å■ŮïijŇ
ä¿ŃåęĆïijŇäÿĂäÿłåĚÿåđŃçŽĎåĞ¡æŢřåžŞåŇĚäijŽçśżäijijäÿŃéÍćè£Źæăůïij Ž

```
projectname/
    README.txt
    Doc/
        documentation.txt
```

(continues on next page)

(continued from previous page)

```
    projectname/
        __init__.py
        foo.py
        bar.py
        utils/
            __init__.py
            spam.py
            grok.py
    examples/
        helloworld.py
        ...
```

èęĄèőľä¡ăçŽĎåŇĚåŘŕäżěåŘŚåÿČåĞžåŐżïijŇéęŰåĚĹä¡ăèęĄçijŰåĘŹäÿĂäÿł `setup.py` ïijŇçśżäijijäÿŃéİćè£ŹæăůïijŽ

```
# setup.py
from distutils.core import setup

setup(name='projectname',
    version='1.0',
    author='Your Name',
    author_email='you@youraddress.com',
    url='http://www.you.com/projectname',
    packages=['projectname', 'projectname.utils'],
)
```

äÿŃäÿĂæ■ěïijŇåřśæŸŕåĹŻåżžäÿĂäÿł `MANIFEST.in` æŰĞäżűïijŇåĹŮåĞžæĽĂæIJĽåIJĺä¡ăçŽĎåŇĚäÿ

```
# MANIFEST.in
include *.txt
recursive-include examples *
recursive-include Doc *
```

çąőä£İ `setup.py` åŠŇ `MANIFEST.in` æŰĞäżűæŤ¿åIJĺä¡ăçŽĎåŇĚçŽĎæIJĂéąűçžğçŻőå¡Ţäÿ■ãĂĆäÿĂæŮęä¡ăåůšçżŘåĄŽäžĘè£ŹäžŻïijŇä¡ăåřśåŘŕäżěåČŘäÿŃéİćè£ŹæăůæĽğèąŇåŚ¡äżďæİěåĹŻåżžäÿĂäÿłæžŘ

```
% bash python3 setup.py sdist
```

åőČäijŽåĹŻåżžäÿĂäÿłæŰĞäżűæŕŤåęĆâĂİprojectname-1.0.zipâĂİ æĹŰâĂIJprojectname-1.0.tar.gzâĂİ, åĚůä¡Şä¿ÌèţŰäžŐä¡ăçŽĎçşżçż§åźşåŘřãĂĆåęĆæđIJäÿĂåĹĞæ■čåÿÿïijŇè£ŹäÿłæŰĞäżűåřśåŘŕäżěåŘŚéĂĄçżŹåĹńäžžä¡£çŤĺæĹŰèĂĚäÿŁäijäèĞş Python Package Index.

èőÍèőž

åŕźäžŐçžŕPythonäżčçăĄïijŇçijŰåĘŹäÿĂäÿłæŹőéĂŽçŽĎ `setup.py` æŰĞäżűéĂŽåÿÿå¿ĹçőĂå■ŢãĂĆ äÿĂäÿłåŘŕèČ¡çŽĎéŮőéćŸæŸŕä¡ăå£ĚéążæĽŃåŁĺåĹŮåĞžæĽĂæIJĽæđĎæäÿĂäÿłåÿÿèğĄéŤŹèŕŕåřśæŸŕäżĚäżĚåŘłåĹŮåĞžäÿĂäÿłåŇĚçŽĎæIJĂéąűçžğçŻőå¡ŢïijŇå£ŸèőřäžĘåŇĚåŘńåŇ

è£ŹäźŐæŸŕäÿžäżĂäźĹåIJÍ `setup.py` äÿ■åŕźäžŐåŇĚçŽĎèŕť æŸŐåŇĚåŘńäžĘåĹŮèąÍ
`packages=['projectname', 'projectname.utils']`

åď ğéČÍåĹĘPythonçÍŃåžŘåŚŸéČ¡ç§ěéĄŞïijŇæIJL'å¿Ĺåď Žçňňäÿ L'æŰźåŇĚçőąçŘĘåŹÍä¿ŻéĂL'æŃl'ïijŇ
æIJL'äžŽæŸŕäÿžäžĘæŻ£äżčæăĞåĞĘåžŞäÿ■çŽĎdistutilsãĂĆæşĺæĎŘåęĆæđIJä¡ăä¿ÌèţŰè£ŹäžŻåŇĚïijŇ
çŤĺæĹůåŘŕèČ¡äÿ■èČ¡åőL'èčĚä¡ăçŽĎè¡ŕäżűïijŇéŹď éİđäżŰäżňåůšçżŘäžŃåĚĹåőL'èčĚè£ĞæL'ĂéIJĂèęĄçŽĎ
æ■čåŻäåęĆæ■d'ïijŇä¡ăæŻť åžŤèŕěæŮűåĹżèőŕä¡ŘèűŁçőĂå■ŢèűŁåě¡çŽĎéĄŞçŘĘãĂĆ
æIJĂåě¡èől'ä¡ăçŽĎäžčçăĄä¡£çŤĺæăĞåĞĘçŽĎPython 3åőL'èčĚãĂĆ
åęĆæđIJåĚűäżŰåŇĚäź§éIJĂèęĄçŽĎèŕÍïijŇåŘŕäżěéĂŽè£ĞäÿĂäÿłåŘŕéĂL'éąźæÍěæŤŕæŇĄãĂĆ

åŕźäžŐæűL'åŘŁåĹřCæL'l'åśŢçŽĎäżčçăĄæL'ŞåŇĚäÿŐåĹĘåŘŚåŕśæŻť åď ■æÍĆçĆźäžĘãĂĆ
çňň15çňäåŕźåĚşäžŐCæL'l'åśŢçŽĎè£ŹæŰźéÍćç§ěèŕĘæIJL'äÿĂäžŻèŕęçżĘèőšèğčïijŇçL'źåĹńæŸŕåIJÍ15.2åŕŘèł

13 çňňå■ĄäÿĂçńäïijŽç¡ŚçżIJäÿŐWebçijŰçÍŃ

æIJňçńäæŸŕåĚşäžŐåIJÍç¡ŚçżIJåžŤçŤĺåŠŇåĹĘåÿČåijŘåžŤçŤĺäÿ■ä¡£çŤĺçŽĎåŘĎçğ■äÿżécŸãĂĆäÿżécŸã

Contents:

13.1 11.1 ä¡IJäÿžäőćæĹůçńŕäÿŐHTTPæIJ■åŁąäžď äžŠ

éŮőécŸ

ä¡äéIJĂèęĄéĂŽè£ĞHTTPå■ŔèőäżěäőćæĹůçńŕçŽĎæŰźäijŘèő£éŮőäď Žçğ■æIJ■åŁąãĂĆä¿ŃäęĆïijŇäÿ

èğčåĘşæŰźæąĹ

åŕźäžŐçőĂå■ŢçŽĎäžŃæČĚæÍěèŕť ïijŇéĂŽäÿÿä¡£çŤÍ `urllib.request` æĹąåÍŮåŕšäď §äžĘãĂĆä¿ŃäęĆïijŇåŘŚéĂĄäÿĂäÿłçőĂå■ŢçŽĎHTTP GETèŕůæśĆåĹřè£IJçÍŃçŽĎæIJ■åŁąäÿŁïijŇåŘŕäżěè£ŹæăůåĄŽïijŽ

```
from urllib import request, parse

# Base URL being accessed
url = 'http://httpbin.org/get'

# Dictionary of query parameters (if any)
parms = {
   'name1' : 'value1',
   'name2' : 'value2'
}

# Encode the query string
querystring = parse.urlencode(parms)

# Make a GET request and read the response
```

(continues on next page)

(continued from previous page)

```
u = request.urlopen(url+'?' + querystring)
resp = u.read()
```

åęĆæđIJä¡ăéIJĂèęĄä¡£çŤÍPOSTæŰźæşŢåIJÍèŕůæśĆäÿżä¡Şäÿ■åŘŚéĂĄæ§ěèŕćåŘĆæŢřïijŇåŘŕäżěåřĘåŘ
`urlopen()` åĞ¡æŢřïijŇåřśåČŘè£ŹæăůïijŽ

```
from urllib import request, parse

# Base URL being accessed
url = 'http://httpbin.org/post'

# Dictionary of query parameters (if any)
parms = {
   'name1' : 'value1',
   'name2' : 'value2'
}

# Encode the query string
querystring = parse.urlencode(parms)

# Make a POST request and read the response
u = request.urlopen(url, querystring.encode('ascii'))
resp = u.read()
```

åęĆæđIJä¡ăéIJĂèęĄåIJÍåŘŚåĞžçŽĎèŕůæśĆäÿ■æŘŘä¿ŻäÿĂäžŻèĞłåőŽäźL'çŽĎHTTPåď't'ïijŇä¿ŃåęĆäł
`user-agent` å■Ůæőţ,åŘŕäżěåĹŻåżžäÿĂäÿłåŇĚåŘńå■Ůæőţ åĂijçŽĎå■ŮåĚÿïijŇåźűåĹŻåżžäÿĂäÿłRequestå
`urlopen()` ïijŇåęĆäÿŃïijŽ

```
from urllib import request, parse
...

# Extra headers
headers = {
    'User-agent' : 'none/ofyourbusiness',
    'Spam' : 'Eggs'
}

req = request.Request(url, querystring.encode('ascii'),␣
↪headers=headers)

# Make a request and read the response
u = request.urlopen(req)
resp = u.read()
```

åęĆæđIJéIJĂèęĄäžď'äžŠçŽĎæIJ■åŁąæŕŤäÿŁéİćçŽĎä¿Ńå■ŘéČ¡èęĄåď■æİĆïijŇäź§èőÿåžŤèŕěåŐżçIJŃç
requests åžŞïijĹhttps://pypi.python.org/pypi/requestsïijL'ãĂĆä¿ŃåęĆïijŇäÿŃéİćè£Źäÿłçď'žä¿ŃéĞĞçŤÍreques

```
import requests
```

(continues on next page)

(continued from previous page)

```
# Base URL being accessed
url = 'http://httpbin.org/post'

# Dictionary of query parameters (if any)
parms = {
   'name1' : 'value1',
   'name2' : 'value2'
}

# Extra headers
headers = {
    'User-agent' : 'none/ofyourbusiness',
    'Spam' : 'Eggs'
}

resp = requests.post(url, data=parms, headers=headers)

# Decoded text returned by the request
text = resp.text
```

åĚşäžŐrequestsåžŞïijŇäÿĂäÿłåĂijå¿ŮäÿĂæŘŘçŽĎçĽźæĂğåřśæŸŕåőČèČ¡äžěåď Žçğ■æŰźåijŘäžŐèřůæ
`resp.text` åÿęçżŹæĹŚäžňçŽĎæŸŕäžěUnicodeèğčçăĄçŽĎåŞ■åžŤæŰĞæIJňãĂĆä¡ĘæŸŕïijŇåęĆæđIJåŐżèč
`resp.content` ïijŇåřśäijŽå¿ŮåĹřåŐ§åğŃçŽĎäžŇè£ŻåĹűæŢřæ■õãĂĆåŘęäÿĂæŰźéİćïijŇåęĆæđIJèő£éŮć
`resp.json` ïijŇéĆčäźĹåřśäijŽå¿ŮåĹřJSONæäijåijŘçŽĎåŞ■åžŤåĘĚåőźãĂĆ

äÿŃéİćè£Źäÿłçďžä¿ŃåĹľçŤÍ `requests` åžŞåŘŚèţůäÿĂäÿłHEAD-èřůæśĆïijŇåźűäžŐåŞ■åžŤäÿ■æŘŘåŘŰåĞžäÿĂäžŻHTTPåď ťæŢřæ■õçŽĎå■Ůæőţïijž

```
import requests

resp = requests.head('http://www.python.org/index.html')

status = resp.status_code
last_modified = resp.headers['last-modified']
content_type = resp.headers['content-type']
content_length = resp.headers['content-length']
```

äÿŃéİćæŸŕäÿĂäÿłåĹľçŤÍrequestséĂŽè£Ğå§žæIJňèőď èřĄçŹżå¡ŢPypiçŽĎä¿Ńå■ŘïijŽ

```
import requests

resp = requests.get('http://pypi.python.org/pypi?:action=login',
                    auth=('user','password'))
```

äÿŃéİćæŸŕäÿĂäÿłåĹľçŤÍrequestsåřĘHTTP cookiesäžŐäÿĂäÿłèřůæśĆäijăéĂŠåĹřåŘęäÿĂäÿłçŽĎä¿Ńå■

```
import requests

# First request
```

(continues on next page)

(continued from previous page)

```
resp1 = requests.get(url)
...

# Second requests with cookies received on first requests
resp2 = requests.get(url, cookies=resp1.cookies)
```

æIJĂåŘŐä¡Ęåźűéİđæ IJĂäÿ■éĞ■èęAçŽĎäÿĂäÿłä¿Ńå■ŘæŸŕçŤÍrequestsäÿŁäijĂåĘĚåőźïijŽ

```
import requests
url = 'http://httpbin.org/post'
files = { 'file': ('data.csv', open('data.csv', 'rb')) }

r = requests.post(url, files=files)
```

èőÍèőž

åŕźäžŐçIJ§çŽĎå¿ÍçőĂå■ŢHTTPåőćæĹůçńŕäžčçăĄïijŇçŤÍåĘĚç¡őçŽĎ urllib æÍąåİŮéĂŽåÿÿåŕśèűşåď§äžĘãĂĆä¡ĘæŸŕïijŇåęĆæđIJä¡äèęĄåĄŽçŽĎäÿ■äżĚäżĚåŘłæŸŕçőĂå■ŢçŽĎGETæĹ requests åď ğæŸ¿èžńæĽŃçŽĎæŮűåĂŹäžĘãĂĆ

ä¿ŃåęĆïijŇåęĆæđIJä¡ääĘşäőŽåİŽæŇĄä¡£çŤÍæăĞåĞĘçŽĎçÍŃåžŘåžŞèĂŇäÿ■èĂČèŹŚåČŔ requests è£ŹæăůçŽĎçňňäÿĽæŰźåžŞïijŇéĆčäźĹäź§èőÿåŕśäÿ■å¿Ůäÿ■ä¡£çŤÍåžŢåśĆçŽĎ http.client æÍąåİŮæİěäőđçŐřèĞłäůśçŽĎäžčçăĄãĂĆæŕŤæŰźèŕť ïijŇäÿŃéİćçŽĎäžčçăĄäśŢçď žäžĘäęĆä

```
from http.client import HTTPConnection
from urllib import parse

c = HTTPConnection('www.python.org', 80)
c.request('HEAD', '/index.html')
resp = c.getresponse()

print('Status', resp.status)
for name, value in resp.getheaders():
    print(name, value)
```

åŘŇæăůåIJřïijŇåęĆæđIJå£ĚéążçijŰåĘŹæűĽ åŔŁäžčçŘĘãĂĄèőď èŕĄãĂĄcookiesäžěåŔŁåĚűäżŰäÿĂäžĂ urllib åŕśæŸ¿å¿ŮçĽźåĹńåĹńæĽ■åŠŇåŢřåŮęãĂĆæŕŤæŰźèŕť ïijŇäÿŃéİćè£Źäÿłçď žä¿ŃäőđçŐřå IJÍPython

```
import urllib.request

auth = urllib.request.HTTPBasicAuthHandler()
auth.add_password('pypi','http://pypi.python.org','username',
↪'password')
opener = urllib.request.build_opener(auth)

r = urllib.request.Request('http://pypi.python.org/pypi?
↪:action=login')
u = opener.open(r)
```

(continues on next page)

(continued from previous page)

```
resp = u.read()

# From here. You can access more pages using opener
...
```

åİęçŹ¡èŕť'ïijŇæL'ĂæIJL'çŽĎè£ŹäžŻæŞ■ä¡IJåIJÍ requests åžŞäÿ■éČ¡åŘŸå¿ŮçőĂå■ŢçŽĎåď'ŽãĂĆ

åIJÍåijĂåŘŚè£ĞçÍŃäÿ■æţŃèŕŢHTTPåőćæĹůçńŕäžčçăĄåÿÿåÿÿæŸŕå¿Ĺäžď'äžžæšőäÿğçŽĎïijŇåŻääÿžæL //httpbin.orgïijL'ãĂĆè£ŹäÿłçńŹçĆźäijŽæŐěæŤůåŘŚåĞžçŽĎèŕůæśĆïijŇçĎůåŘŐäžěJSONçŽĎå¡ćåijŘåřĘçŻÿ

```
>>> import requests
>>> r = requests.get('http://httpbin.org/get?name=Dave&n=37',
...     headers = { 'User-agent': 'goaway/1.0' })
>>> resp = r.json
>>> resp['headers']
{'User-Agent': 'goaway/1.0', 'Content-Length': '', 'Content-Type': '
↪',
'Accept-Encoding': 'gzip, deflate, compress', 'Connection':
'keep-alive', 'Host': 'httpbin.org', 'Accept': '*/*'}
>>> resp['args']
{'name': 'Dave', 'n': '37'}
>>>
```

åIJÍèęĄåŘŇäÿĂäÿłçIJ§æ■čçŽĎçńŹçĆźè£ŻèąŇäžď'äžŠåL'■ïijŇåĚĹåIJÍ httpbin.org è£ŹæăůçŽĎç¡ŚçńŹäÿŁåĄŽåőđéłŇåÿÿåÿÿæŸŕåŘŕåŘŰçŽĎåŁđæşŢãĂĆåŕď'åĚűæŸŕå¡ŞæĹŚäžňéĺćåŕź3æňąçŹ

åř¡çőąæIJňèŁĆæšąæIJL'æűL'åŘŁïijŇ request åžŞè£Ÿåŕźèőÿåď'ŽéńŸçžğçŽĎHTTPåőćæĹůçńŕå■Ŕèőő requests æĺąåİŮçŽĎæŰĞæąčïijĹhttp://docs.python-requests.org)èť'ÍéĞŘå¿ĹéńŸïijĹåİęçŹ¡èŕť'æŕŤåIJÍè£Źç§■ç§■çŽĎäÿĂèŁĆçŽĎçŕĞåźĚäÿ■æL'ĂæŘŘä¿ŻçŽĎäżżä¡Ţä£ą

13.2 11.2 åĹŻåżžTCPæIJ■åŁąåŹÍ

éŮőécŸ

ä¡ăæČşåőđçŐřäÿĂäÿłæIJ■åŁąåŹÍïijŇéĂŽè£ĞTCPå■ŔèőőåŠŇåőćæĹůçńŕéĂŽä£ąãĂĆ

èğčåĘşæŰźæąĹ

åĹŻåżžäÿĂäÿłTCPæIJ■åŁąåŹÍçŽĎäÿĂäÿłçőĂå■ŢæŰźæşŢæŸŕä¡£çŤÍ socketserver åžŞãĂĆä¿ŃåęĆïijŇäÿŃéĺćæŸŕäÿĂäÿłçőĂå■ŢçŽĎåžŤç■ŤæIJ■åŁąåŹÍïijŽ

```
from socketserver import BaseRequestHandler, TCPServer

class EchoHandler(BaseRequestHandler):
    def handle(self):
        print('Got connection from', self.client_address)
```

(continues on next page)

(continued from previous page)

```
        while True:

            msg = self.request.recv(8192)
            if not msg:
                break
            self.request.send(msg)

if __name__ == '__main__':
    serv = TCPServer(('', 20000), EchoHandler)
    serv.serve_forever()
```

åIJÍè£Źæőţäžčçă Ąäÿ■ïijŇä¡ăăőŽäźL'äžĘäÿĂäÿłçL'źæőŁçŽĎåď'ĎçŘĘçśżïijŇåőđçŐřäžĘäÿĂäÿł handle() æŰźæşŢïijŇçŤÍæİěäÿžăőćæÉŮçńŕè£đæŐěæIJ■åŁąãĂĆ request åśđæĂğæŸŕăőćæÉŮçńŕsocketïijŇclient_address æIJL'ăőćæÉŮçńŕåIJřåİĂãĂĆ äÿžäžĘæţŃèŕŢè£ŹäÿłæIJ■åŁąåŹÍïijŇè£ŘèąŇăőČăźűæL'ŞåijĂåŘęåď'ŰäÿĂäÿłPythonè£ŻçÍŃè£đæŐěè£Źäÿłæ

```
>>> from socket import socket, AF_INET, SOCK_STREAM
>>> s = socket(AF_INET, SOCK_STREAM)
>>> s.connect(('localhost', 20000))
>>> s.send(b'Hello')
5
>>> s.recv(8192)
b'Hello'
>>>
```

å¿Ĺåď'ŽæŮűåĂŹïijŇåŕŕäżěå¿Ĺăőźæ ŸŞçŽĎăőŽäźL'äÿĂäÿłäÿ■åŘŇçŽĎåď'ĎçŘĘåŹÍãĂĆäÿŃéİćæŸŕäÿĂ StreamRequestHandler å§žçśżăřĘäÿĂäÿłçśżæŰĞäżűæŐěåŘčæŤ¿ç¡ăőåIJÍåžŢåśĆsocketäÿŁçŽĎä¿Ňå■Ř

```
from socketserver import StreamRequestHandler, TCPServer

class EchoHandler(StreamRequestHandler):
    def handle(self):
        print('Got connection from', self.client_address)
        # self.rfile is a file-like object for reading
        for line in self.rfile:
            # self.wfile is a file-like object for writing
            self.wfile.write(line)

if __name__ == '__main__':
    serv = TCPServer(('', 20000), EchoHandler)
    serv.serve_forever()
```

èőÍèőž

socketserver åŕŕäżěèőľæĹŚäżňå¿Ĺăőźæ ŸŞçŽĎåĹŻäżžçőĂå■ŢçŽĎTCPæIJ■åŁąåŹÍãĂĆ ä¡ĘæŸŕïijŇä¡äéIJĂèęĄæşÍæĎŘçŽĎæŸŕïijŇéżŸèőď'æČĚåĘţäÿŃè£Źçğ■æIJ■åŁąåŹÍæŸŕå■Ţçž£çÍŃçŽĎïijŇäÿ åęĆæđIJä¡äæČşåď'ĎçŘĘåď'ŽäÿłăőćæĹŮçńŕïijŇåŕŕäżěåĹİåğŃåŇŰäÿĂäÿł ForkingTCPServer æĹŰèĂĚæŸŕ ThreadingTCPServer åŕźèśąãĂĆä¿ŃåęĆïijŽ

```
from socketserver import ThreadingTCPServer


if __name__ == '__main__':
    serv = ThreadingTCPServer(('', 20000), EchoHandler)
    serv.serve_forever()
```

ä¡£çŤÍforkæĹŰçž£çÍŃæIJ■åŁąåŹÍæIJL'äÿłæ¡IJåIJÍéŮőéćŸåřśæŸŕåőČäżňäijŽäÿžæŕŔäÿłåőćæĹůçńŕè£đæ
çŤśäžŐåőćæĹůçńŕè£đæŐěæŢřæŸŕæšąæIJL'éŹŘåĹűçŽĎïijŇåŻäæ■ď'äÿĂäÿłæĄűæĎŔçŽĎéżŚåőćåŘŕäżěåŘŇ

åęĆæđIJä¡äæŃĚå£Čè£ŹäÿłéŮőéćŸïijŇä¡äåŘŕäżěåĹŻåżžäÿĂäÿłéćĎåĚĹåĹĘéĚ■åď'gåřŘçŽĎåůěä¡IJçž£ç
ä¡äåĚĹåĹŻåżžäÿĂäÿłæŹőéĂŽçŽĎéİđçž£çÍŃæIJ■åŁąåŹÍïijŇçĎűåŘŐåIJÍäÿĂäÿłçž£çÍŃæśääÿ■ä¡£çŤÍ
`serve_forever()` æŰźæşŢæİěåŘŕåŁÍåőČäżňäĂĆ

```
if __name__ == '__main__':
    from threading import Thread
    NWORKERS = 16
    serv = TCPServer(('', 20000), EchoHandler)
    for n in range(NWORKERS):
        t = Thread(target=serv.serve_forever)
        t.daemon = True
        t.start()
    serv.serve_forever()
```

äÿĂèĹňæİěèőšïijŇäÿĂäÿł `TCPServer` åIJÍåőđä¿ŃåŇŰçŽĎæŮűåĂŹäijŽçżŚåőŽåźűæ£Ăæť'żçŻÿåžŤçŽ
`socket` ãĂĆ äÿ■è£ĞïijŇæIJL'æŮűåĂŹä¡äæČşéĂŽè£Ğèő¿ç¡őæ§ŘäžŻéĂL'éąźåŐżèřČæŢť'åžŢäÿŃçŽĎ
socket' ïijŇåŘŕäżěèő¿ç¡őåŘĆæŢř `bind_and_activate=False` ãĂĆåęĆäÿŃïijŽ

```
if __name__ == '__main__':
    serv = TCPServer(('', 20000), EchoHandler, bind_and_
↪activate=False)
    # Set up various socket options
    serv.socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR,␣
↪True)
    # Bind and activate
    serv.server_bind()
    serv.server_activate()
    serv.serve_forever()
```

äÿŁéİćçŽĎ `socket` éĂL'éąźæŸŕäÿĂäÿłéİđåÿÿæŹőéĄ■çŽĎéĚ■ç¡őéąźïijŇåőČåĚĄèőÿæIJ■åŁąåŹÍéĞ■æ
çŤśäžŐèęĄèćńçżŘåÿÿä¡£çŤÍåĹřïijŇåőČèćńæŤ¿ç¡őåĹřçśżåŘŸéĞŘäÿ■ïijŇåŘŕäżěçŻť'æŐěåIJÍ
`TCPServer` äÿŁéİćèő¿ç¡õăĂĆ åIJÍåőđä¿ŃåŇŰæIJ■åŁąåŹÍçŽĎæŮűåĂŹåŐżèő¿ç¡őåőČçŽĎåĂijïijŇåęĆäÿŃ

```
if __name__ == '__main__':
    TCPServer.allow_reuse_address = True
    serv = TCPServer(('', 20000), EchoHandler)
    serv.serve_forever()
```

åIJÍäÿŁéİćçď'žä¿Ńäÿ■ïijŇæĹŚäżňæijŤçď'žäžĘäÿď'çğ■äÿ■åŘŇçŽĎåď'ĎçŘĘåŹÍå§žçśżïijĹ
`BaseRequestHandler` åŠŇ `StreamRequestHandler` ïijL'ãĂĆ
`StreamRequestHandler` æŻť'åŁăçĄţæť'żçĆźïijŇèČ¡éĂŽè£Ğèő¿ç¡õåĚűäżŰçŽĎçśżåŘŸéĞŘæİěæŤŕæŇ

```
import socket

class EchoHandler(StreamRequestHandler):
    # Optional settings (defaults shown)
    timeout = 5                         # Timeout on all socket␣
↪operations
    rbufsize = -1                       # Read buffer size
    wbufsize = 0                        # Write buffer size
    disable_nagle_algorithm = False     # Sets TCP_NODELAY socket␣
↪option
    def handle(self):
        print('Got connection from', self.client_address)
        try:
            for line in self.rfile:
                # self.wfile is a file-like object for writing
                self.wfile.write(line)
        except socket.timeout:
            print('Timed out!')
```

æIJĂåŘŐïijŇè£ŸéIJĂèęĄæşÍæĎŘçŽĎæŸŕåůÍåď ğéČÍåĹĘPythonçŽĎéńŸåśÇç¡ŚçżIJæĺąåİŮïijĹæŕŤåęĆI
RPCç■L'ïijL'éČ¡æŸŕåżžçńŇåIJÍ socketserver åŁ§èČ¡äźŇäÿŁãĂĆ
äź§åŕśæŸŕèŕť'ïijŇçŻť'æŐěä¡£çŤÍ `socket` åžŞæİěåőđçŐřæIJ■åŁąåŹÍäź§åźűäÿ■æŸŕå¿ĹéŽ¿ãĂĆ
äÿŇéİćæŸŕäÿĂäÿłä¡£çŤÍ `socket` çŻť'æŐěçijŰçÍŇåőđçŐřçŽĎäÿĂäÿłæIJ■åŁąåŹÍçőĂå■Ţä¿Ňå■ŘïijŽ

```
from socket import socket, AF_INET, SOCK_STREAM

def echo_handler(address, client_sock):
    print('Got connection from {}'.format(address))
    while True:
        msg = client_sock.recv(8192)
        if not msg:
            break
        client_sock.sendall(msg)
    client_sock.close()

def echo_server(address, backlog=5):
    sock = socket(AF_INET, SOCK_STREAM)
    sock.bind(address)
    sock.listen(backlog)
    while True:
        client_sock, client_addr = sock.accept()
        echo_handler(client_addr, client_sock)

if __name__ == '__main__':
    echo_server(('', 20000))
```

13.3 11.3 åĹŻåżžUDPæIJ■åŁąåŹÍ

éŮőécŸ

ä¡ăæČşåőđçŐřäÿĂäÿłå§žäžŐUDPå■ŘèőőçŽĎæIJ■åŁąåŹÍæĺěäÿŐåőćæĹůçńŕéĂŽä£ąãĂĆ

èğčåĘşæŰźæąĹ

èů§TCPäÿĂæăůïijŇUDPæIJ■åŁąåŹÍäź§åŘŕäżěéĂŽè£Ğä¡£çŤÍ `socketserver` åžŞå¿Ĺåőźæ ŸŞçŽĎèćńåĹŻåżžãĂĆ ä¿ŃåęĆïijŇäÿŃéĺćæŸŕäÿĂäÿłçőĂå■ŢçŽĎæŮűéŮťæIJ■åŁąåŹÍïijŽ

```
from socketserver import BaseRequestHandler, UDPServer
import time

class TimeHandler(BaseRequestHandler):
    def handle(self):
        print('Got connection from', self.client_address)
        # Get message and client socket
        msg, sock = self.request
        resp = time.ctime()
        sock.sendto(resp.encode('ascii'), self.client_address)

if __name__ == '__main__':
    serv = UDPServer(('', 20000), TimeHandler)
    serv.serve_forever()
```

èů§äźŃåĽ■äÿĂæăůïijŇä¡ăåĚĹåőŽäźĽäÿĂäÿłåőđçŐř `handle()` çĽźæőŁæŰźæşŢçŽĎçśżïijŇäÿžåőćæĹůçńŕè£đæŐěæIJ■åŁąãĂĆ è£ŹäÿłçśżçŽĎ `request` åśđæĂğæŸŕäÿĂäÿłåŇĚåŘńäžĘæŢŕæ■őæŁěåŠŇåžŢåśĆsocketåŕźèśąçŽĎåĚČçżĎãĂĆ`client_address` åŇĚåŘńäžĘåőćæĹůçńŕåIJŕåĺĂãĂĆ

æĹŚäżňæĺěæţŃèŕŢäÿŃè£ŹäÿłæIJ■åŁąåŹÍïijŇéęŰåĚĹè£ŘèąŇåőČïijŇçĎűåŘŐæĽŞåijĂåŘęåď ŰäÿĂäÿłl

```
>>> from socket import socket, AF_INET, SOCK_DGRAM
>>> s = socket(AF_INET, SOCK_DGRAM)
>>> s.sendto(b'', ('localhost', 20000))
0
>>> s.recvfrom(8192)
(b'Wed Aug 15 20:35:08 2012', ('127.0.0.1', 20000))
>>>
```

èőÍèőž

äÿĂäÿłåĚÿåđŃçŽĎUDPæIJ■åŁąåŹÍæŐěæŤűåĹřè¿¿çŽĎæŢŕæ■őæŁě(æűĹæĄŕ)åŠŇåőćæĹůçńŕåIJŕåĺĂãĂ åőČèęĄçżŹåőćæĹůçńŕåŻđåŘŚäÿĂäÿłæŢŕæ■őæŁěãĂĆåŕźäžŐæŢŕæ■őæŁěçŽĎäijăéĂĄïijŇ ä¡ăåžŤèŕěä¡£çŤÍsocketçŽĎ `sendto()` åŠŇ `recvfrom()` æŰźæşŢãĂĆ åř¡çőąäijăçż§çŽĎ `send()` åŠŇ `recv()` äź§åŘŕäżěè¿¿åĹřåŘŇæăůçŽĎæŢĹæđIJïijŇ ä¡ĘæŸŕåĽ■éĺćçŽĎäÿď äÿłæŰźæşŢåŕźäžŐUDPè£đæŐěèĂŇèĺĂæŻť æŹőéĄ■ãĂĆ

çŤśäžŐæšąæIJL'åžŢåśĆçŽĎè£đæŐěïijŇUPDæIJ■åŁąåŹÍçŻÿåŕźäžŐTCPæIJ■åŁąåŹÍæÌěèőšåőđçŐřèţůæÌ
äÿ■è£ĞïijŇUDPåď'l'çŤ§æŸŕäÿ■åŘŕéÌäçŽĎïijŁåŻääÿžéĂŽä£ąæšąæIJL'åżžçńNè£đæŐěïijŇæűŁæĄŕåŘŕèČ¡äÿ
åŻäæ■ď'éIJĂèęĄçŤśä¡äèĞłåůśæÌěåEşåőŽèŕěæĂŐæăůåď'ĎçŘĘäÿćåď'śæűŁæĄŕçŽĎæČĚåEţãĂĆè£Źäÿłåůšçżl
äÿ■è£ĞéĂŽåÿÿæÌěèŕť'ïijŇåęĆæđIJåŘŕéÌäæĂğåŕźäžŐä¡äçÍŃåžŘå¿ŁéĞ■èęĄïijŇä¡äéIJĂèęĄåĂ§åŁľäžŐåžŘåÍ
UDPéĂŽåÿÿèćńçŤÍåIJÍéĆčäžŽåŕźäžŐåŘŕéÌääijäè¿ŞèęĄæŚĆäÿ■æŸŕå¿ŁéńŸçŽĎåIJžåŘŁãĂĆä¿ŃåęĆïijŇåIJÍå
æŮäéIJĂè£ŤåŻđæĄćåď'■äÿćåď'śçŽĎæŢŕæ■őåŇĚïijŁçÍŃåžŘåŘłéIJĂçőĂå■ŢçŽĎå£¡çŢěåőČåźűçżğçż■åŘŚåL

`UDPServer` çśżæŸŕå■Ţçž£çÍŃçŽĎïijŇäź§åŕśæŸŕèŕť'äÿĂæňąåŘłèČ¡äÿżäÿĂäÿłåőćæŁůçńŕè£đæŐěæIJ■
åőđéŹĚä¡£çŤÍäÿ■ïijŇè£ŹäÿłæŮäèőžæŸŕåŕźäžŐUDPè£ŸæŸŕTCPéČ¡äÿ■æŸŕäżĂäźŁåď'ğéŮőéćŸãĂĆ
åęĆæđIJä¡äæČşèęĄåźűåŘŚæŞ■ä¡IJïijŇåŘŕäżěåőđä¿ŃåŇŰäÿĂäÿł `ForkingUDPServer`
æŁŰ `ThreadingUDPServer` åŕźèśąïijŽ

```
from socketserver import ThreadingUDPServer

   if __name__ == '__main__':
    serv = ThreadingUDPServer(('',20000), TimeHandler)
    serv.serve_forever()
```

çŻť'æŐěä¡£çŤÍ `socket` æÌěåőđçŐřäÿĂäÿłUDPæIJ■åŁąåŹÍäź§äÿ■éŽ¿ïijŇäÿŃéÌćæŸŕäÿĂäÿłä¿Ńå■ŘïijŽ

```
from socket import socket, AF_INET, SOCK_DGRAM
import time

def time_server(address):
    sock = socket(AF_INET, SOCK_DGRAM)
    sock.bind(address)
    while True:
        msg, addr = sock.recvfrom(8192)
        print('Got message from', addr)
        resp = time.ctime()
        sock.sendto(resp.encode('ascii'), addr)

if __name__ == '__main__':
    time_server(('', 20000))
```

13.4 11.4 éĂŽè£ĞCIDRåIJřåÌĂçŤ§æŁŘåŕźåžŤçŽĎIPåIJřåÌĂéŻĘ

éŮőéćŸ

ä¡äæIJL'äÿĂäÿłCIDRç¡ŚçżIJåIJřåÌĂæŕŤåęĆâĂIJ123.45.67.89/27âĂÌïijŇä¡äæČşåřĘåĚűè¡ňæ■ćæŁŘåőČa
ïijŁæŕŤåęĆïijŇâĂIJ123.45.67.64âĂÌ, âĂIJ123.45.67.65âĂÌ, âĂę, âĂIJ123.45.67.95âĂÌ)ïijL'

èğčåEşæŰźæąŁ

åŘŕäżěä¡£çŤÍ `ipaddress` æÍąåÌŮå¿ŁåőźæŸŞçŽĎåőđçŐřè£ŹæăůçŽĎèőąçőŮãĂĆä¿ŃåęĆïijŽ

```
>>> import ipaddress
>>> net = ipaddress.ip_network('123.45.67.64/27')
>>> net
IPv4Network('123.45.67.64/27')
>>> for a in net:
...     print(a)
...
123.45.67.64
123.45.67.65
123.45.67.66
123.45.67.67
123.45.67.68
...
123.45.67.95
>>>

>>> net6 = ipaddress.ip_network('12:3456:78:90ab:cd:ef01:23:30/125')
>>> net6
IPv6Network('12:3456:78:90ab:cd:ef01:23:30/125')
>>> for a in net6:
...     print(a)
...
12:3456:78:90ab:cd:ef01:23:30
12:3456:78:90ab:cd:ef01:23:31
12:3456:78:90ab:cd:ef01:23:32
12:3456:78:90ab:cd:ef01:23:33
12:3456:78:90ab:cd:ef01:23:34
12:3456:78:90ab:cd:ef01:23:35
12:3456:78:90ab:cd:ef01:23:36
12:3456:78:90ab:cd:ef01:23:37
>>>
```

`Network` äź§åĚĄèőÿåČŕæŢřçżĎäÿĂæăůçŽĎçť ćåijŢåŕŰåĂijïijŇä¿ŃåęĆïijŽ

```
>>> net.num_addresses
32
>>> net[0]

IPv4Address('123.45.67.64')
>>> net[1]
IPv4Address('123.45.67.65')
>>> net[-1]
IPv4Address('123.45.67.95')
>>> net[-2]
IPv4Address('123.45.67.94')
>>>
```

åŕęåď Ůïij Ňä¡ăè£ŸåŕŕäżěæL'ğèąŇç¡ŚçżIJæĹŘåŚŸæčĂæ§ěäźŃçśżçŽĎæŞ■ä¡IJïijŽ

```
>>> a = ipaddress.ip_address('123.45.67.69')
```

(continues on next page)

(continued from previous page)

```
>>> a in net
True
>>> b = ipaddress.ip_address('123.45.67.123')
>>> b in net
False
>>>
```

äÿĂäÿłIPåIJřåİĂåŠŇç¡ŚçżIJåIJřåİĂèČ¡éĂŽè£ĞäÿĂäÿłIPæŐěåŘčæİěæŇĞåőŽïijŇä¿ŃåęĆïijŽ

```
>>> inet = ipaddress.ip_interface('123.45.67.73/27')
>>> inet.network
IPv4Network('123.45.67.64/27')
>>> inet.ip
IPv4Address('123.45.67.73')
>>>
```

èőÍèőž

`ipaddress` æĹąåİŮæIJL'å¿Ĺåď'ŽçśżåŘŕäżěèąĺçď'žIPåIJřåİĂãĂĄç¡ŚçżIJåŠŇæŐěåŘčãĂĆ å¡Şä¡ăéIJĂèęĄæŞ■ä¡IJç¡ŚçżIJåIJřåİĂïijĹæŕŤåęĆèğčæđŘãĂĄæL'Şå■řãĂĄéłŇèřĄç■L'ïijL'çŽĎæŮűåĂŹäijŽå

èęĄæşĺæĎŘçŽĎæŸŕïijŇ`ipaddress` æĹąåİŮèů§åĚűäżŰäÿĂäżŽåŠŇç¡ŚçżIJçŻÿåĚşçŽĎæĹąåİŮæŕŤåęĆ `socket` åžŞäžď'éŻĘå¿ĹåřŚãĂĆ æL'ÄäżěïijŇä¡ääÿ■èČ¡ä¡£çŤÍ `IPv4Address` çŽĎåőđä¿ŃæİěäżčæŻ£äÿĂäÿłåIJřåİĂå■ŮçňęäÿšïijŇä¡äéęŰåĚĹå¿ŮæŸ¿åijŘçŽĎä¡£çŤÍ `str()` è¡ňæ■ćåőČãĂĆä¿ŃåęĆïijŽ

```
>>> a = ipaddress.ip_address('127.0.0.1')
>>> from socket import socket, AF_INET, SOCK_STREAM
>>> s = socket(AF_INET, SOCK_STREAM)
>>> s.connect((a, 8080))
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: Can't convert 'IPv4Address' object to str implicitly
>>> s.connect((str(a), 8080))
>>>
```

æŻť'åď'ŽçŻÿåĚşåĘĚåőźïijŇèřůåŘĆèĂČ An Introduction to the ipaddress Module

13.5 11.5 åĹŻåżżäÿĂäÿłçőĂå■ŢçŽĎRESTæŐěåŘč

éŮőécŸ

ä¡äæČşä¡£çŤÍäÿĂäÿłçőĂå■ŢçŽĎRESTæŐěåŘčéĂŽè£Ğç¡ŚçżIJè£IJçÍŃæŐğåĹűæĹŰèő£éŮőä¡äçŽĎåžŤ

èğčåĘşæŰźæąĹ

æđĎåżžäÿĂäÿłRESTéčŐæăijçŽĎæŐěåŘčæIJĂçőĂå■ŢçŽĎæŰźæşŢæŸŕåĹŻåżžäÿĂäÿłå§žäžŐWSGIæă 3333ïijĽçŽĎå¿ĹåřŘçŽĎåžŞïijŇäÿŃéĺćæŸŕäÿĂäÿłä¿Ńå■ŘïijŽ

```
# resty.py

import cgi

def notfound_404(environ, start_response):
    start_response('404 Not Found', [ ('Content-type', 'text/plain
↪') ])
    return [b'Not Found']

class PathDispatcher:
    def __init__(self):
        self.pathmap = { }

    def __call__(self, environ, start_response):
        path = environ['PATH_INFO']
        params = cgi.FieldStorage(environ['wsgi.input'],
                                  environ=environ)
        method = environ['REQUEST_METHOD'].lower()
        environ['params'] = { key: params.getvalue(key) for key in␣
↪params }
        handler = self.pathmap.get((method,path), notfound_404)
        return handler(environ, start_response)

    def register(self, method, path, function):
        self.pathmap[method.lower(), path] = function
        return function
```

äÿžäžĘä¡£çŤĺè£ŹäÿłèřČåžęåŹĺïijŇä¡ääŘłéIJĂèęĄçijŰåĘŹäÿ■åŘŇçŽĎåď ĎçŘĘåŹĺïijŇåřśåČŘäÿŃéĺćè£

```
import time

_hello_resp = '''\
<html>
  <head>
     <title>Hello {name}</title>
   </head>
   <body>
     <h1>Hello {name}!</h1>
   </body>
</html>'''

def hello_world(environ, start_response):
    start_response('200 OK', [ ('Content-type','text/html')])
    params = environ['params']
    resp = _hello_resp.format(name=params.get('name'))
```

(continues on next page)

(continued from previous page)

```
    yield resp.encode('utf-8')

_localtime_resp = '''\
<?xml version="1.0"?>
<time>
  <year>{t.tm_year}</year>
  <month>{t.tm_mon}</month>
  <day>{t.tm_mday}</day>
  <hour>{t.tm_hour}</hour>
  <minute>{t.tm_min}</minute>
  <second>{t.tm_sec}</second>
</time>'''

def localtime(environ, start_response):
    start_response('200 OK', [ ('Content-type', 'application/xml')␣
↪])
    resp = _localtime_resp.format(t=time.localtime())
    yield resp.encode('utf-8')

if __name__ == '__main__':
    from resty import PathDispatcher
    from wsgiref.simple_server import make_server

    # Create the dispatcher and register functions
    dispatcher = PathDispatcher()
    dispatcher.register('GET', '/hello', hello_world)
    dispatcher.register('GET', '/localtime', localtime)

    # Launch a basic server
    httpd = make_server('', 8080, dispatcher)
    print('Serving on port 8080...')
    httpd.serve_forever()
```

èęĄæţŃèŕŢäÿŃè£ŹäÿłæIJ■åŁąåŹĺïijŇä¡ăåŘŕäżěä¡£çŤĺäÿĂäÿłæţŘèğĹåŹĺæĹŰ `urllib` åŠŇåőČäžď äžŠãĂĆä¿ŃåęĆïijŽ

```
>>> u = urlopen('http://localhost:8080/hello?name=Guido')
>>> print(u.read().decode('utf-8'))
<html>
  <head>
     <title>Hello Guido</title>
   </head>
   <body>
     <h1>Hello Guido!</h1>
   </body>
</html>

>>> u = urlopen('http://localhost:8080/localtime')
>>> print(u.read().decode('utf-8'))
```

(continues on next page)

(continued from previous page)

```
<?xml version="1.0"?>
<time>
  <year>2012</year>
  <month>11</month>
  <day>24</day>
  <hour>14</hour>
  <minute>49</minute>
  <second>17</second>
</time>
>>>
```

èőÍèőž

åIJÍçijŰåĘŹRESTæŐěåŘčæŮűïijŇéĂŽåÿÿéČ¡æŸŕæIJ■åŁąäžŐæŹőéĂŽçŽĎHTTPèŕůæśĆãĂĆä¡ĘæŸŕèu
è£ŹäžŻæŢřæ■őäžěåŘĎçğ■æăĞåĞĘæäijåijŘçijŰçăĄïijŇæŕŤåęĆXMLãĂĄJSONæŁŰCSVãĂĆ
åř¡çőąçÍŃåžŘçIJŃäÿŁåŐżå¿ŁçőĂå■ŢïijŇä¡ĘæŸŕäżěè£Źçğ■æŰźåijŘæŘŘä¿ŻçŽĎAPIåŕźäžŐå¿Łåď ŽåžŤçŤÍç

ä¿ŃåęĆïijŇéŢ£æIJ§è£ŘèąŇçŽĎçÍŃåžŘåŘŕèČ¡äijŽä¡£çŤÍäÿĂäÿłREST
APIæÌěåőđçŐřçŻŚæŐğæŁŰèŕŁæŰ■ãĂĆ åď ğæŢřæ■őåžŤçŤÍçÍŃåžŘåŘŕäżěä¡£çŤÍRESTæÌěæđĎåżžäÿĂäÿłæ
RESTè£ŸèČ¡çŤÍæÌěæŐğåŁűçąňäżűèő¿åď ĞæŕŤåęĆæIJžåŹÍäžžãĂĄäijäæĎ§åŹÍãĂĄåůěåŐĆæŁŰçĄŕæşąãĂĆ
æŻť éĞ■èęĄçŽĎæŸŕïijŇREST APIåůšçżŘèćńåď ğéĞŘåőćæŁůçńŕçijŰçÍŃçŐŕåćĆæL’ĂæŤŕæŇĄïijŇæŕŤåęĆ-
Javascript, Android, iOSç■L’ãĂĆ åŻäæ■ď’ïijŇåĹľçŤÍè£Źçğ■æŐěåŘčåŘŕäżěèőľä¡äåijĂåŘŚåĞžæŻť åŁäåď■æ

äÿžäžĘåőđçŐřäÿĂäÿłçőĂå■ŢçŽĎRESTæŐěåŘčïijŇä¡äåŘłéIJĂèőľä¡äçŽĎçÍŃåžŘäżčçăĄæżąèűşPython
WSGIèćńæăĞåĞĘåžŞæŤŕæŇĄïijŇåŘŇæŮűäź§èćńçżÌåď ğéČÍåŁĘçňňäÿL’æŰźwebæąĘæđűæŤŕæŇĄãĂĆ
åŻäæ■ď’ïijŇåęĆæđIJä¡äçŽĎäżčçăĄéĄţå¿łè£ŹäÿłæăĞåĞĘïijŇåIJÍåŘŐéÌćçŽĎä¡£çŤÍè£ĞçÍŃäÿ■åŕśäijŽæŻť åŁ

åIJÍWSGIäÿ■ïijŇä¡äåŘŕäżěåČŘäÿŃéÌćè£ŹæäůçżęäőŽçŽĎæŰźåijŘäżěäÿĂäÿłåŘŕèřČçŤÍåŕźèśąå¡ćåijŘæÌ

```
import cgi

def wsgi_app(environ, start_response):
    pass
```

environ åśđæĂğæŸŕäÿĂäÿłå■ŮåĚÿïijŇåŇĚåŘńäžĘäżŐwebæIJ■åŁąåŹÍåęĆA-
pache[åŘĆèĂČInternet RFC 3875]æŘŘä¿ŻçŽĎCGIæŐěåŘčäÿ■èŐůåŘŰçŽĎåĂijãĂĆ
èęĄåřĘè£ŹäžŻäÿ■åŘŇçŽĎåĂijæŘŘåŘŰåĞžæÌěïijŇä¡äåŘŕäżěåČŘè£ŹäźĹè£ŹæäůåĘŹïijŽ

```
def wsgi_app(environ, start_response):
    method = environ['REQUEST_METHOD']
    path = environ['PATH_INFO']
    # Parse the query parameters
    params = cgi.FieldStorage(environ['wsgi.input'],␣
↪environ=environ)
```

æĹŚäžňåśŢçď žäžĘäÿĂäžŻåÿÿèğĄçŽĎåĂijãĂĆ`environ['REQUEST_METHOD']`
äżčèąÍèřůæśĆçśżåđŃåęĆGETãĂĄPOSTãĂĄHEADç■L’ãĂĆ `environ['PATH_INFO']`
èąÍçď žèćńèřůæśĆèţĎæžŘçŽĎèůřå¿ĎãĂĆ èřČçŤÍ `cgi.FieldStorage()`
åŘŕäżěäżŐèřůæśĆäÿ■æŘŘåŘŰæ§ěèŕćåŘĆæŢřåźůåřĘåőČäżňæŤ¿åĚěäÿĂäÿłçśżå■ŮåĚÿåŕźèśąäÿ■äżěä¿£åŘŐ

`start_response` åŔĆæŢřæŸŕäÿĂäÿłäÿžäžĘåĹİåğŃåŇŰäÿĂäÿłèŕůæśĆåŕźèśąèĂŇå£Ěéążèćńèř ČçŤĺ çňňäÿĂäÿłåŔĆæŢřæŸŕè£ŤåŻđçŽĎHTTPçŁűæĂĄåĂijïijŇçňňäžŇäÿłåŔĆæŢřæŸŕäÿĂäÿł(åŘ■,åĂij)åĚČçżĎ

```python
def wsgi_app(environ, start_response):
    pass
    start_response('200 OK', [('Content-type', 'text/plain')])
```

äÿžäžĘè£ŤåŻđæŢřæ■őïijŇäÿĂäÿłWSGIçĺŃåžŔå£Ěéążè£ŤåŻđäÿĂäÿłå■ŮèŁĆå■ŮçňęäÿšåžŔåĹŮãĂĆåŔ

```python
def wsgi_app(environ, start_response):
    pass
    start_response('200 OK', [('Content-type', 'text/plain')])
    resp = []
    resp.append(b'Hello World\n')
    resp.append(b'Goodbye!\n')
    return resp
```

æĹŰèĂĚïijŇä¡ăè£ŸåŔŕäżĕä¡£çŤĺ `yield` ïijŽ

```python
def wsgi_app(environ, start_response):
    pass
    start_response('200 OK', [('Content-type', 'text/plain')])
    yield b'Hello World\n'
    yield b'Goodbye!\n'
```

è£ŹéĞŇèęĄåijžèřČçŽĎäÿĂçĆźæŸŕæIJĂåŘŐè£ŤåŻđçŽĎå£ĚéążæŸŕå■ŮèŁĆå■ŮçňęäÿšãĂĆåęĆæđIJè£ å¡ŞçĎűïijŇåźűæšąæIJĽèęĄæśĆä¡ăè£ŤåŻđçŽĎäÿĂåőŽæŸŕæŰĞæIJňïijŇä¡ăåŔŕäżĕå¿Ĺè¡żæİ¿çŽĎçijŰåĘŹäÿ

åř¡çőąWSGIçĺŃåžŔéĂŽåÿÿèćńåőŽäźĽæĹŘäÿĂäÿłåĞ¡æŢřïijŇäÿ■è£Ğä¡ăäź§åŔŕäżĕä¡£çŤĺçśżåőđä¿ŃæĬ `__call__()` æŰźæşŢãĂĆä¿ŃåęĆïijŽ

```python
class WSGIApplication:
    def __init__(self):
        ...
    def __call__(self, environ, start_response)
       ...
```

æĹŚäżňåůšçżŔåIJĺäÿŁéİćä¡£çŤĺè£Źçğ■æŁĂæIJŕåĹŻåżž `PathDispatcher` çśżãĂĆ è£ŹäÿłåĹĘåŔŚåŹĺäžĚäžĚåŔłæŸŕçőąçŘĘäÿĂäÿłå■ŮåĚÿïijŇåřĘ(æŰźæşŢ,èůŕå¿Ď)åŕźæŸăåřĎåĹřåď ĎçŘĘåŹ å¡ŞäÿĂäÿłèŕůæśĆåĹřæİěæŮűïijŇåőČçŽĎæŰźæşŢåŠŇèůŕå¿ĎèćńæŔŘåŔŰåĞžæİěïijŇçĎűåŘŐèćńåĹĘåŔŚåĹ åŔęåď ŰïijŇäżżä¡Ţæ§ěèŕćåŔŸéĞŔäijŽèćńèğčæđŘåŘŐæŤ¿åĹřäÿĂäÿłå■ŮåĚÿäÿ■ïijŇäžě `environ['params']` å¡ćåijŔå■ŸåĆĺãĂĆ åŘŐéİćè£Źäÿłæ■ěéłď åď łåÿÿèğĄïijŇæĽĂäżĕåżžèőőä¡ăåIJĺåĹĘ ä¡£çŤĺåĹĘåŔŚåŹĺçŽĎæŰűåĂŹïijŇä¡ăåŔłéIJĂçőĂå■ŢçŽĎåĹŻåżžäÿĂäÿłåőđä¿ŃïijŇçĎűåŘŐéĂŽè£ĞåőČæş çijŰåĘŹè£ŹäžŻåĞ¡æŢřåžŤèŕěèűĚçžğçőĂå■ŢäžĘïijŇåŔłèęĄä¡ăéĄţå¿ł `start_response()` åĞ¡æŢřçŽĎçijŰåĘŹèğĎåĹŹïijŇåźűäÿŤæIJĂåŘŐè£ŤåŻđå■ŮèŁĆå■Ůçňęäÿšå■şåŔŕã

å¡ŞçijŰåĘŹè£Źçğ■åĞ¡æŢřçŽĎæŰűåĂŹè£ŸéIJĂæşĺæĎŘçŽĎäÿĂçĆźåřśæŸŕåŕźäžŐå■Ůçňęäÿšæĺąæİ£çŽ æšąäžžæĎ£æĎŘåĘŹéĆčçğ■åĹřåď ĎæůůåŘĹçİĂ `print()` åĞ¡æŢř ãĂĄXM-LåŠŇåď ğéĞŔæăijåijŔåŇŰæŞ■ä¡IJçŽĎäżčçăĄãĂĆ æĹŚäżňäÿŁéİćä¡£çŤĺäžĘäÿĽåijŢåŔůåŇĚåŔńçŽĎéćĎåĚ è£Źçğ■æŰźåijŔçŽĎåŔŕäżĕèőľæĹŚäżňå¿Ĺåőźæ ŸŞçŽĎåIJĺäżĕåŘŐä£őæŤźè¿ŞåĞžæăijåijŔ(åŔłéIJĂèęĄä£őæ

æIJĂåŘŐïijŇä¡£çŤĺWSGIè£ŸæIJĽäÿĂäÿłå¿ĹéĞ■èęĄçŽĎéČĺåĹĘåřśæŸŕæšąæIJĽäżĂäźĹåIJřæŰźæŸŕé

åŻžääÿžæžĞåĞĘåŕźäžŐæIJ■åŁąåŹĺåŠŇæąĘæđűæŸŕäÿ■çńŇçŽĎïijŇä¡žåŔŕäżěåřĘä¡žçŽĎçĺŃåžŔæŤ¿åĚěäżżä
æĹŚäżňä¡£çŤĺäÿŃéĺćçŽĎäżčçžĄæţŃèŕŢæţŃèŕŢæIJňèŁĆäżčçžĄïijŽ

```
if __name__ == '__main__':
    from wsgiref.simple_server import make_server

    # Create the dispatcher and register functions
    dispatcher = PathDispatcher()
    pass

    # Launch a basic server
    httpd = make_server('', 8080, dispatcher)
    print('Serving on port 8080...')
    httpd.serve_forever()
```

äÿŁéĺćäżčçžĄåĹŻåżżäžĘäÿŽäÿłçőŽåå■ŢçŽĎæIJ■åŁąåŹĺïijŇçĎůåŘŐä¡žåřśåŔŕäżěæĺěæţŃèŕŢäÿŃä¡žçŽĎ
æIJŽåŘŐïijŇå¡Şä¡žåĞĘåď'Ğè£ŻäÿŽæ■ěæĽľåśŢä¡žçŽĎçĺŃåžŔçŽĎæŮűåŽĹïijŇä¡žåŔŕäżěä£őæŤźè£Źäÿłäżĕ

WSGIæIJňèžńæŸŕäÿŽäÿłå¿ĹåřŔçŽĎæžĞåĞĘãŽĆåŻžæ■ď'åőČåźűæšąæIJĽæŘŘä¿ŻäÿŽäžŻénŸçžğçŽĎ
è£ŹäžŻä¡žèĞłåůśåőđçŐřèţůæĺěäź§äÿ■éŽ¿ãŽĆäÿ■è£ĞåęĆæđIJä¡žæČşèęĄæŻť'åď'ŽçŽĎæŤŕæŇĄïijŇåŔŕäżěè
`WebOb` æĹŰèŽĚ `Paste`

13.6 11.6 éŽè£ĞXML-RPCåőđçŐřçőŽåå■ŢçŽĎè£IJçĺŃèřČçŤĺ

éŮőéćŸ

ä¡žæČşæĽ'¿åĹřäÿŽäÿłçőŽåå■ŢçŽĎæŰźåijŕåŐżæĽ'ğèąŇè£ŘèąŇåIJĺè£IJçĺŃæIJżåŹĺäÿŁéĺćçŽĎPythonçĺ

èğčåĘşæŰźæąĹ

åőđçŐřäÿŽäÿłè£IJçĺŃæŰźæşŢèřČçŤĺçŽĎæIJĂçőŽåå■ŢæŰźåijŕæŸŕä¡£çŤĺXML-RPCãĂĆäÿŃéĺćæĹŚäżňæijŤçď'žäÿŽäÿŃäÿŽäÿłåőđçŐřäžĘéŤő-åŽijå■ŸåĆĺåŁ§èČ¡çŽĎçőŽåå■ŢæIJ■åŁąåŹĺïijŽ

```
from xmlrpc.server import SimpleXMLRPCServer

class KeyValueServer:
    _rpc_methods_ = ['get', 'set', 'delete', 'exists', 'keys']
    def __init__(self, address):
        self._data = {}
        self._serv = SimpleXMLRPCServer(address, allow_none=True)
        for name in self._rpc_methods_:
            self._serv.register_function(getattr(self, name))

    def get(self, name):
        return self._data[name]

    def set(self, name, value):
```

(continues on next page)

(continued from previous page)

```
        self._data[name] = value

    def delete(self, name):
        del self._data[name]

    def exists(self, name):
        return name in self._data

    def keys(self):
        return list(self._data)

    def serve_forever(self):
        self._serv.serve_forever()

# Example
if __name__ == '__main__':
    kvserv = KeyValueServer(('', 15000))
    kvserv.serve_forever()
```

äÿŃéİćæĹŚäżňäżŐäÿĂäÿłåőćæĹůçńŕæIJžåŹĺäÿŁéİćæİěèő£éŮőæIJ■åŁąåŹĺïijŽ

```
>>> from xmlrpc.client import ServerProxy
>>> s = ServerProxy('http://localhost:15000', allow_none=True)
>>> s.set('foo', 'bar')
>>> s.set('spam', [1, 2, 3])
>>> s.keys()
['spam', 'foo']
>>> s.get('foo')
'bar'
>>> s.get('spam')
[1, 2, 3]
>>> s.delete('spam')
>>> s.exists('spam')
False
>>>
```

èőĺèőž

XML-RPC åŘŕäżěèőľæĹŚäżňå¿Ĺåőźæ ŸŞçŽĎæđĎéĂääÿĂäÿłçőĂå■ŢçŽĎè£IJçĺŃèřČçŤĺæIJ■åŁąãĂĆä¡éĂŽè£ĞåőČçŽĎæŰźæşŢ `register_function()` æİěæşĺåĘŇåĞ¡æŢřïijŇçĎűåŘŐä¡£çŤĺæŰźæşŢ `serve_forever()` åŘŕåŁĺåőČãĂĆ åIJĺäÿŁéİćæĹŚäżňåřĘè£ŹäžŻæ■ěéłďæŤ¿åIJĺäÿĂèţůåĘŹåĹřäÿĂäÿłçśż

```
from xmlrpc.server import SimpleXMLRPCServer
def add(x,y):
    return x+y

serv = SimpleXMLRPCServer(('', 15000))
```

(continues on next page)

(continued from previous page)

```
serv.register_function(add)
serv.serve_forever()
```

XML-RPCæŽť éIJšåĞžæİěçŽĎåĞ¡æŢřåŘłèČ¡éĂĆçŤÍäžŐéČÍåĹĘæŢřæ■őçśżåđŃïijŇærŤåęĆå■Ůçňęäÿš
åŕźäžŐåĚűäżŰçśżåđŃářśå¿ŮéIJĂèęĄåĄŽäžŻéćİåď ŰçŽĎåŁ§èŕ¿äžĘãĂĆ
ä¿ŃåęĆïijŇåęĆæđIJä¡äæČşéĂŽè£Ğ XML-RPC äijäéĂŠäÿĂäÿłåŕźèśąåőđä¿ŃïijŇåőđéŹĚäÿŁåŘłæIJĹäżŰçŽĎ

```
>>> class Point:
...     def __init__(self, x, y):
...             self.x = x
...             self.y = y
...
>>> p = Point(2, 3)
>>> s.set('foo', p)
>>> s.get('foo')
{'x': 2, 'y': 3}
>>>
```

çśżäijijçŽĎïijŇåŕźäžŐäžŇè£ŻåĹűæŢřæ■őçŽĎåď ĎçŘĘäź§èů§ä¡äæČşèśąçŽĎäÿ■åď łäÿĂæäůïijŽ

```
>>> s.set('foo', b'Hello World')
>>> s.get('foo')
<xmlrpc.client.Binary object at 0x10131d410>

>>> _.data
b'Hello World'
>>>
```

äÿĂèĹňæİěèőšïijŇä¡ääÿ■åžŤèŕěåřĘ XML-RPC æIJ■åŁąäżěåĚňåĚśAPIçŽĎæŰźåijŘæŽť éIJšåĞžæİěãĂĆ
åŕźäžŐè£Źçğ■æČĚåĘţïijŇéĂŽåÿÿåĹĘåÿČåijŘåžŤçŤÍçÍŃåžŘäijŽæŸŕäÿĂäÿłæŽť åě¡çŽĎéĂĹ'æŃľãĂĆ

XML-RPCçŽĎäÿĂäÿłçijžçĆźæŸŕåőČçŽĎæĂğèČ¡ãĂĆ`SimpleXMLRPCServer`
çŽĎåőđçŐřæŸŕå■Ţçž£çÍŃçŽĎïijŇ æĹ'Ääżěåőčäÿ■éĂĆåŘĹäžŐåď ğåđŃçÍŃåžŘïijŇåř¡çőąæĹŚäżňåIJÍ11.2åřĺ
åŘęåď ŰïijŇçŤśäžŐ XML-RPC åřĘæĹ'ĂæIJĹ'æŢřæ■őéČ¡åžŘåĹŮåŇŰäÿžXMLæäijåijŘïijŇæĹ'ÄäżěåőčäijŽ
ä¡ĘæŸŕåőČäź§æIJĹ'äijŸçĆźïijŇè£Źçğ■æŰźåijŘçŽĎçijŰçăĄåŘŕäżěèćńçżİåď ğéČÍåĹĘåĚűäżŰçijŰçÍŃèŕ■èÍĂ
éĂŽè£Ğä¡£çŤÍè£Źçğ■æŰźåijŘïijŇåĚűäżŰèŕ■èÍĂçŽĎåőćæĹůçńŕçÍŃåžŘéČ¡èČ¡èő£éŮőä¡äçŽĎæIJ■åŁąäĂĆ

èŹ¡çĎűXML-RPCæIJĹ'å¿Ĺåď ŽçijžçĆźïijŇä¡ĘæŸŕåęĆæđIJä¡äéIJĂèęĄå£ńéĂ§æđĎåżžäÿĂäÿłçőĂå■Ţè£
æIJĹ'æŮűåĂŹïijŇçőĂå■ŢçŽĎæŰźæąĹåŕśåůšçżŘèűşåď §äžĘãĂĆ

13.7 11.7 åIJÍäÿ■åŘŇçŽĎPythonèğčéĞŁåŹÍäźŃéŮťäžď äžŠ

éŮőéćŸ

ä¡äåIJÍäÿ■åŘŇçŽĎæIJžåŹÍäÿŁéİćè£ŘèąŇçİĂåď Žäÿł Pythonèğčé ĞŁåŹÍäźŃéŮťäžď äžŠ

èğčåĘşæŰźæąĹ

éĂŽè£Ğä¡£çŤÍ `multiprocessing.connection` æÍąåİŮåŘŕäžěå¿Ĺåőźæ ŸŞçŽĎåőđçŐřèğčéĞŁåŹÍ
äÿŃéÍćæŸŕäÿĂäÿłçőĂå■ŢçŽĎåžŤç■ŤæIJ■åŁąåŹÍä¿Ńå■ŘïijŽ

```
from multiprocessing.connection import Listener
import traceback

def echo_client(conn):
    try:
        while True:
            msg = conn.recv()
            conn.send(msg)
    except EOFError:
        print('Connection closed')

def echo_server(address, authkey):
    serv = Listener(address, authkey=authkey)
    while True:
        try:
            client = serv.accept()

            echo_client(client)
        except Exception:
            traceback.print_exc()

echo_server(('', 25000), authkey=b'peekaboo')
```

çĎűåŘŐåőćæĹůçńŕè£đæŐěæIJ■åŁąåŹÍåźűåŘŚéĂĄæűĹæĄŕçŽĎçőĂå■Ţçď žä¿ŃïijŽ

```
>>> from multiprocessing.connection import Client
>>> c = Client(('localhost', 25000), authkey=b'peekaboo')
>>> c.send('hello')
>>> c.recv()
'hello'
>>> c.send(42)
>>> c.recv()
42
>>> c.send([1, 2, 3, 4, 5])
>>> c.recv()
[1, 2, 3, 4, 5]
>>>
```

èů§åžŢåśĆsocketäÿ■åŘŃçŽĎæŸŕïijŃæŕŘäÿłæűĹæĄŕäijŽåőŃæŢť ä£Ìå■ŸïijĹæŕŘäÿĂäÿłéĂŽè£Ğsend()
åŘęåď ŰïijŃæĽĂæIJĽåŕźèśąäijŽéĂŽè£ĞpickleåžŘåĹŮåŃŰãĂĆåŻäæ■ď ïijŃäżżä¡ŢåĚijåőźpickleçŽĎåŕźèśą

èőÍèőž

çŻőåĽ■æIJĽå¿Ĺåď ŽçŤÍæÌěåőđçŐřåŘĎçğ■æűĹæĄŕäijäè¿ŞçŽĎåŃĚåŠŃåĞ¡æŢřåžŞïijŃæŕŤåęĆZeroMQ
ä¡äè£ŸæIJĽåŘęåď ŰäÿĂçğ■éĂĽæŃĽåŕśæŸŕèĞłåůśåIJĺåžŢåśĆsocketå§žçąĂäźŃäÿŁæÌěåőđçŐřäÿĂäÿłæűĹæ

ä¡ĘæŸŕä¡ăæČşèęĄçőĂå■ŢäÿĂçĆźçŽĎæŰźæąĹïijŇéĆčäźĹè£ŹæŮűåĂŹ multiprocessing.connection åřśæť¿äÿŁçŤĺåIJžäžĘãĂĆ äżĚäżĚä¡£çŤĺäÿĂäžŻçőĂå■ŢçŽĎèŕ■åŘěå■şåŘŕåőđçŐřåď'ŽäÿłèğčéĞŁåŹĺäźŇéŮť'çŽĎæűĹæĄŕéĂŽä£ąãĂĆ

åęĆæđIJä¡ăçŽĎèğčéĞŁåŹĺè£ŘèąŇåIJĺåŘŇäÿĂåŘŕæIJžåŹĺäÿŁéİćïijŇéĆčäźĹä¡ăåŘŕäżěä¡£çŤĺåŘęåď'ŰçŽ
èęĄæČşä¡£çŤĺUNIXå§§åěŮæŐěå■ŮæİěåĹŻåżžäÿĂäÿłè£đæŐěïijŇåŘłéIJĂçőĂå■ŢçŽĎåřĘåIJřåİĂæŤźåĘŹäÿ

```
s = Listener('/tmp/myconn', authkey=b'peekaboo')
```

èęĄæČşä¡£çŤĺWindowsåŚ¡åŘ■çőąéĄŞæİěåĹŻåżžè£đæŐěïijŇåŘłéIJĂåČŘäÿŇéİćè£ŹæăůäÿĄçŤĺäÿĂäÿłä

```
s = Listener(r'\\.\pipe\myconn', authkey=b'peekaboo')
```

äÿĂäÿłéĂŽçŤĺåĞĘåĹŹæŸŕïijŇä¡ääÿ■èęĄä¡£çŤĺ multiprocessing æİěåőđçŐřäÿĂäÿłåŕźåď'ŰçŽĎåĚňåĚśæIJ■åŁąãĂĆ Client() åŠŇ Listener() äÿ■çŽĎ authkey åŘĆæŢřçŤĺæİěèőď'èŕĄåŘŚèţůè£đæŐěçŽĎçżĹçńŕçŤĺæĹůãĂĆ
åęĆæđIJåŕĘéŠěäÿ■åŕźäijŽäžğçŤ§äÿĂäÿłåijĆåÿÿãĂĆæ■ď'åď'ŰïijŇèŕěæĺąåİŮæIJĂéĂĆåŘĹçŤĺæİěåżžçńŇéŢ£
ä¿ŇåęĆïijŇäÿď'äÿłèğčéĞŁåŹĺäźŇéŮť'åŘŕåŁĺåŘŐåŕśåijĂåğŇåżžçńŇè£đæŐěåźűåIJĺåď'ĎçŘĘæ§ŘäÿłéŮőéćŸ

åęĆæđIJä¡ăéIJĂèęĄåŕźåžŢåśĆè£đæŐěåĄŽæŻť'åď'ŽçŽĎæŐğåĹűïijŇæŕŤåęĆéIJĂèęĄæŤŕæŇĄèűĚæŮűã
ä¡ăæIJĂåě¡ä¡£çŤĺåŘęåď'ŰçŽĎåžŞæĹŰèĂĚæŸŕåIJĺéńŸåśĆsocketäÿŁæİěåőđçŐřè£ŹäžŻçL'źæĂğãĂĆ

13.8 11.8 åőđçŐřè£IJçĺŃæŰźæşŢèřČçŤĺ

éŮőéćŸ

ä¡ăæČşåIJĺäÿĂäÿłæűĹæĄŕäijăè¿ŞåśĆåęĆ sockets ãĂĄmultiprocessing connections æĹŰ ZeroMQ çŽĎå§žçąĂäźŇäÿŁåőđçŐřäÿĂäÿłçőĂå■ŢçŽĎè£IJçĺŃè£ĞçĺŃèřČçŤĺïijĹRPC

èğčåĘşæŰźæąĹ

åřĘåĞ¡æŢřèřůæśĆãĂĄåŘĆæŢřåŠŇè£ŤåŻđåĂijä¡£çŤĺpickleçijŰçăĄåŘŐïijŇåIJĺäÿ■åŘŇçŽĎèğčéĞŁåŹ
äÿŇéİćæŸŕäÿĂäÿłçőĂå■ŢçŽĎPRCåď'ĎçŘĘåŹĺïijŇåŘŕäżěèćńæŢť'åŘĹåĹřäÿĂäÿłæIJ■åŁąåŹĺäÿ■åŐżïijŽ

```
# rpcserver.py

import pickle
class RPCHandler:
    def __init__(self):
        self._functions = { }

    def register_function(self, func):
        self._functions[func.__name__] = func

    def handle_connection(self, connection):
        try:
            while True:
                # Receive a message
```

(continues on next page)

(continued from previous page)

```
                func_name, args, kwargs = pickle.loads(connection.
↪recv())
                # Run the RPC and send a response
                try:
                    r = self._functions[func_name](*args,**kwargs)
                    connection.send(pickle.dumps(r))
                except Exception as e:
                    connection.send(pickle.dumps(e))
        except EOFError:
            pass
```

èęĄä¡£çŤÍè£ŹäÿłåďĎçŘĘåŹÍïijŇä¡ăéIJĂèęĄåřĘåőČåŁăåĚěåĹřäÿĂäÿłæűĹæĄŕæIJ■åŁąåŹÍäÿ■ãĂĆä¡ăæä¡ĘæŸŕä¡£çŤÍ multiprocessing åžŞæŸŕæIJĂçőĂå■ŢçŽĎãĂĆäÿŃéİćæŸŕäÿĂäÿłR-PCæIJ■åŁąåŹÍä¿Ńå■ŘïijŽ

```
from multiprocessing.connection import Listener
from threading import Thread

def rpc_server(handler, address, authkey):
    sock = Listener(address, authkey=authkey)
    while True:
        client = sock.accept()
        t = Thread(target=handler.handle_connection, args=(client,))
        t.daemon = True
        t.start()

# Some remote functions
def add(x, y):
    return x + y

def sub(x, y):
    return x - y

# Register with a handler
handler = RPCHandler()
handler.register_function(add)
handler.register_function(sub)

# Run the server
rpc_server(handler, ('localhost', 17000), authkey=b'peekaboo')
```

äÿžäžĘäżŐäÿĂäÿłè£IJçÍŃåőćæĹůçńŕèő£éŮőæIJ■åŁąåŹÍïijŇä¡ăéIJĂèęĄåĹŻåżžäÿĂäÿłåŕźåžŢçŽĎçŤÍæĺ

```
import pickle

class RPCProxy:
    def __init__(self, connection):
        self._connection = connection
    def __getattr__(self, name):
```

(continues on next page)

(continued from previous page)

```
        def do_rpc(*args, **kwargs):
            self._connection.send(pickle.dumps((name, args,␣
↪kwargs)))
            result = pickle.loads(self._connection.recv())
            if isinstance(result, Exception):
                raise result
            return result
        return do_rpc
```

èęĄä¡£çŤÍè£ŹäÿłäžčçŘĘçśżïijŇä¡äéIJĂèęĄåřĘåĚűåŇĚèčĚåĹřäÿĂäÿłæIJ■åŁąåŹÍçŽĎè£đæŐěäÿŁéİćïijŇ

```
>>> from multiprocessing.connection import Client
>>> c = Client(('localhost', 17000), authkey=b'peekaboo')
>>> proxy = RPCProxy(c)
>>> proxy.add(2, 3)

5
>>> proxy.sub(2, 3)
-1
>>> proxy.sub([1, 2], 4)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "rpcserver.py", line 37, in do_rpc
    raise result
TypeError: unsupported operand type(s) for -: 'list' and 'int'
>>>
```

èęĄæşÍæĎŘçŽĎæŸŕå¿Ĺåď'ŽæŭĹæĄŕåśĆïijĹæŕŤåęĆ `multiprocessing` ïijĽåůšçżŘä¡£çŤÍpickleåžŘåĹŮåŇŰäžĘæŢřæ■őãĂĆ åęĆæđIJæŸŕè£ŹæăůçŽĎèŕÍïijŇåŕź `pickle.dumps()` åŠŇ `pickle.loads()` çŽĎèřĎçŤÍèęĄåŐżæŐĽ'ãĂĆ

èőÍèőž

`RPCHandler` åŠŇ `RPCProxy` çŽĎå§žæIJňæĂÍèůŕæŸŕå¿ĹæŕŤè¿ČçőĂå■ŢçŽĎãĂĆ åęĆæđIJäÿĂäÿłåőćæĹůçńŕæČşèęĄèřĎçŤÍäÿĂäÿłè£IJçÍŇåĞ¡æŢřïijŇæŕŤåęĆ `foo(1, 2, z=3)` ,äžčçŘĘçśżåĹŻåżžäÿĂäÿłåŇĚåŘńäžĘåĞ¡æŢřåŘ■åŠŇåŘĆæŢřçŽĎåĚČçżĎ `('foo', (1, 2), {'z': 3})` ãĂĆ è£ŹäÿłåĚČçżĎèćńpickleåžŘåĹŮåŇŰåŘŐéĂŽè£ĞÇ¡ŚçżIJè£đæŐěåŘŚçŤ§åČ è£ŹäÿĂæ■ěåIJÍ `RPCProxy` çŽĎ `__getattr__()` æŰźæşŢè£ŤåŽđçŽĎ `do_rpc()` éŮ■åŇĚäÿ■åőŇæĹŘãĂĆ æIJ■åŁąåŹÍæŐěæŤűåŘŐéĂŽè£ĞpickleåŘ■åžŘåĹŮåŇŰæűĹæĄŕïijŇæ§ěæL'¿åĞ¡æ æL'ğèąŇçżŞæđIJ(æĹŰåijĆåÿÿ)èćńpickleåžŘåĹŮåŇŰåŘŐè£ŤåŽđåŘŚéĂĄçżŹåőćæĹůçńŕãĂĆæĹŚäžňçŽĎåő `multiprocessing` è£ŻèąŇéĂŽä£ąãĂĆ äÿ■è£ĞïijŇè£Źçğ■æŰźåijŘåŘŕäžěéĂĆçŤÍäžŐåĚűäžŰäżżä¡ŢæűĹ äžĚäžĚåŘłéIJĂèęĄåřĘè£đæŐěåŕźèśąæ■ćæĹŘåŘĹéĂĆçŽĎZeroMQçŽĎsocketåŕźèśąå■şåŘŕãĂĆ

çŤśäžŐåžŢåśĆéIJĂèęĄä¿ÌètŰpickleïijŇéĆčäźĹåőL'åĚĺéŮőécŸåřśéIJĂèęĄèĂČèŹŚäžĘ ïijĹåŽääÿžäÿĂäÿłèĄłæŸŐçŽĎéżŚåőćåŘŕäžěåĹŻåżžçL'źåőŽçŽĎæűĹæĄŕïijŇèČ¡åď'§èőľäżżæĎŘåĞ¡æŢřéĂŽ åŻäæ■ď'ä¡äæřÿè£IJäÿ■èęĄåĚĄèőÿæÌěèĞłäÿ■ä£ąäżżæĹŰæIJłèőď'èŕĄçŽĎåőćæĹůçńŕçŽĎRPCãĂĆçL'źåĹńæ è£Źçğ■åŘłèČ¡åIJÍåĘĚéČÍèćńä¡£çŤÍïijŇä¡■äžŐéŸšçĄńåćŹåŘŐéÌćåźűäÿŤäÿ■èęĄåŕźåď'ŰæŽť'éIJšãĂĆ

ä¡IJäÿžpickleçŽĎæŻ£äżčïijŇä¡ääź§èőÿåŘŕäžěèĂČèŹŚä¡£çŤÍJSONãĂĄXMLæĹŰäÿĂäžŻåĚűäžŰçŽĎçi

ä¿ŃåęĆïijŇæIJňæIJžåőđä¿ŃåŘŕäžěå¿Ĺåőźæŷ§çŽĎæŤźåĘźæĹŘJSONçijŰçăĄæŰźæąĹãĂĆè£ŸéIJĂèęĄåřĘ
`pickle.loads()` åŠŇ `pickle.dumps()` æŻ£æ■ćæĹŘ `json.loads()` åŠŇ `json.dumps()` å■şåŘŕïijŽ

```
# jsonrpcserver.py
import json

class RPCHandler:
    def __init__(self):
        self._functions = { }

    def register_function(self, func):
        self._functions[func.__name__] = func

    def handle_connection(self, connection):
        try:
            while True:
                # Receive a message
                func_name, args, kwargs = json.loads(connection.
↪recv())
                # Run the RPC and send a response
                try:
                    r = self._functions[func_name](*args,**kwargs)
                    connection.send(json.dumps(r))
                except Exception as e:
                    connection.send(json.dumps(str(e)))
        except EOFError:
             pass

# jsonrpcclient.py
import json

class RPCProxy:
    def __init__(self, connection):
        self._connection = connection
    def __getattr__(self, name):
        def do_rpc(*args, **kwargs):
            self._connection.send(json.dumps((name, args, kwargs)))
            result = json.loads(self._connection.recv())
            return result
        return do_rpc
```

åőđçŐřRPCçŽĎäÿĂäÿłæŕŤè¿Čåď■æĬĆçŽĎéŮőéćŸæŸŕåęĆä¡ŢåŐżåď ĎçŘĘåijĆåÿÿãĂĆèĞşåřŚïijŇå¡Ş
åŻăæ■ď ïijŇè£ŤåŻđçżŹåőćæĹůçńŕçŽĎåijĆåÿÿæĽĂäžčèąĺçŽĎåŘńäźĽ åřśèęĄåě¡åě¡èő¿èőąäžĘãĂĆ
åęĆæđIJä¡ää¡£çŤĺpickleïijŇåijĆåÿÿåŕźèśąåőđä¿ŃåIJĺåőćæĹůçńŕèČ¡èćńåŘ■åžŘåĹŮåŇŰåźűæŁŻåĞžãĂĆåęĆ
äÿ■è£ĞèĞşåřŚïijŇä¡äåžŤèŕěåIJĺåŞ■åžŤäÿ■è£ŤåŻđåijĆåÿÿå■ŮçňęäÿšäĂĆæĹŚäžňåIJĺJSONçŽĎä¿Ňå■Řäÿ■å

åŕźäžŐåĚűäżŰçŽĎRPCåőđçŐřä¿Ňå■ŘïijŇæĹŚæŐĺè■Řä¡äçIJŃçIJŃåIJĺXML-RPCäÿ■ä¡£çŤĺçŽĎ `SimpleXMLRPCServer` åŠŇ `ServerProxy` çŽĎåőđçŐřïijŇ äź§åřśæŸŕ11.6åřŘèŁĆäÿ■çŽĎåĘĚåőźãĂĆ

13.9 11.9 çőĂå■ŢçŽĎåőćæĹůçńŕèőďèŕĄ

éŮőéćŸ

ä¡ăæČşåIJĺåĹĘåÿČåijŘçşżçż§äÿ■åőđçŐřäÿĂäÿłçőĂå■ŢçŽĎåőćæĹůçńŕè£đæŐěèőď'èŕĄåŁ§èČ¡ïijŇåŘĹä

èğčåĘşæŰźæąĹ

åŘŕäžěåĹľçŤĺ `hmac` æĺąåİŮåőđçŐřäÿĂäÿłè£đæŐěæŘąæĽŃïijŇäżŐèĂŇåőđçŐřäÿĂäÿłçőĂå■ŢèĂŇéńŸ

```
import hmac
import os

def client_authenticate(connection, secret_key):
    '''
    Authenticate client to a remote service.
    connection represents a network connection.
    secret_key is a key known only to both client/server.
    '''
    message = connection.recv(32)
    hash = hmac.new(secret_key, message)
    digest = hash.digest()
    connection.send(digest)

def server_authenticate(connection, secret_key):
    '''
    Request client authentication.
    '''
    message = os.urandom(32)
    connection.send(message)
    hash = hmac.new(secret_key, message)
    digest = hash.digest()
    response = connection.recv(len(digest))
    return hmac.compare_digest(digest,response)
```

å§žæIJňåŐ§çŘĘæŸŕå¡Şè£đæŐěåżžçńŃåŘŐïijŇæIJ■åŁąåŹĺçżŹåőćæĹůçńŕåŘŚéĂĄäÿĂäÿłéŽŘæIJžçŽĎ
`os.urandom()` è£ŤåŻđåĂijïijĽãĂĆ åőćæĹůçńŕåŠŇæIJ■åŁąåŹĺåŘŇæŮűåĹľçŤĺh-
macåŠŇäÿĂäÿłåŘłæIJĽåŘŇæŰźç§ěéĄŞçŽĎåŕĘéŠěæİěèőąçőŮåĞžäÿĂäÿłåŁăåŕĘåŞĹåÿŇåĂijãĂĆçĎűåŘŐå
æIJ■åŁąåŹĺéĂŽè£ĞæŕŤè¿Čè£ŹäÿłåĂijåŠŇèĞłåůśèőąçőŮçŽĎæŸŕåŘęäÿĂèĞť'æİěåĘşåőŽæŐěåŘŮæĹŰæŃŠ
`hmac.compare_digest()` åĞ¡æŢřãĂĆ ä¡£çŤĺè£ŹäÿłåĞ¡æŢřåŘŕäžěéĄ£åĚ■éĄ■åĹřæŮűéŮť'åĹĘæđŘæŤ
äÿžäžĘä¡£çŤĺè£ŹäžŻåĞ¡æŢřïijŇä¡ăéIJĂèęĄåřĘåőČéŻĘæĹŘåĹřåůšæIJĽçŽĎç¡ŚçżIJæĹŰæűĹæĄŕäžčçăĄäÿ■

```
from socket import socket, AF_INET, SOCK_STREAM

secret_key = b'peekaboo'
def echo_handler(client_sock):
    if not server_authenticate(client_sock, secret_key):
        client_sock.close()
```

(continues on next page)

(continued from previous page)

```
        return
    while True:

        msg = client_sock.recv(8192)
        if not msg:
            break
        client_sock.sendall(msg)

def echo_server(address):
    s = socket(AF_INET, SOCK_STREAM)
    s.bind(address)
    s.listen(5)
    while True:
        c,a = s.accept()
        echo_handler(c)

echo_server(('', 18000))

Within a client, you would do this:

from socket import socket, AF_INET, SOCK_STREAM

secret_key = b'peekaboo'

s = socket(AF_INET, SOCK_STREAM)
s.connect(('localhost', 18000))
client_authenticate(s, secret_key)
s.send(b'Hello World')
resp = s.recv(1024)
```

èőÍèőž

hmac èőď èřĄçŽĎäÿĂäÿłåÿÿèğĄä¡£çŤÍåIJžæŹŕæŸŕåĘĚéČÍæűĹæĄŕéĂŽä£ąçşżçż§åŠŇè£ŻçÍŃéŮť éĂŽä¿ŃåęĆïijŇåęĆæđIJä¡ăçijŰåĘŹçŽĎçşżçż§æűL'åŔŁåĹřäÿĂäÿłéŻĘç¿ď äÿ■åď Žäÿłåď ĎçŘĘåŹÍäźŃéŮť çŽĎéĂäźäåŔŕäžěä¡£çŤÍæIJňèŁĆæŰźæąĹæĺěçąőä£ÍåŔłæIJL'èćńåĚĄèőÿçŽĎè£ŻçÍŃäźŃéŮť æL'■èČ¡å¡ijæ■ď éĂŽä£ąäžŃåőđäÿŁïijŇå§žäžŐ hmac çŽĎèőď èřĄèćń multiprocessing æÍąåÍŮä¡£çŤÍæĺěåőđçŐřå■Řè£ŻçÍŃçŻť æŐěçŽĎéĂŽä£ąãĂĆ

è£ŸæIJL'äÿĂçĆźéIJĂèęĄåijžěřČçŽĎæŸŕè£đæŐěèőď èřĄåŠŇåŁäåŕĘæŸŕäÿď çăĄäžŃãĂĆ èőď èřĄæĹŘåŁ§äźŃåŘŐçŽĎéĂŽä£ąæűĹæĄŕæŸŕäžěæŸŐæŰĞå¡ćåijŔåŘŚéĂĄçŽĎïijŇäżä¡Ţäžžå ŔłèęĄæč

hmacèőď èřĄçőŮæşŢå§žäžŐåŞĹåÿŇåĞ¡æŢřåęĆMD5åŠŇSHA-1ïijŇåĚşäžŐè£ŹäÿłåIJÍIETF RFC 2104äÿ■æIJL'èŕęçżĘäżŃçż■ãĂĆ

13.10 11.10 ålJÍç¡ŚçżlJælJ■åŁąäÿ■åŁššĚěSSL

éŮőécŸ

ä¡äæČşåőđçŐřäÿĂäÿłå§žäžŐsocketsçŽĎç¡ŚçżIJæIJ■åŁąïijŇåőćæĹůçńřåŠŇæIJ■åŁąåŹÍéĂŽè£ĞSSLå■Ř

èğčåĘşæŰźæąĹ

ssl æÍąåİŮèČ¡äÿžåžŢåśĆsocketè£đæŐěæůżåŁăSSLçŽĎæŤŕæŇĄăĂĆ ssl.wrap_socket() åĞ¡æŢřæŐěåŘŮäÿĂäÿłåůšå■ŸåIJÍçŽĎsocketä¡IJäÿžåŘĆæŢřåźűä¡£çŤÍSSLåśĆæİěåŇĚèč ä¿ŃåęĆïijŇäÿŃéİćæŸŕäÿĂäÿłçőĂå■ŢçŽĎåžŤç■ŤæIJ■åŁąåŹÍïijŇèČ¡åIJÍæIJ■åŁąåŹÍçńřäÿžæĽĂæIJĽåőćæĹ

```
from socket import socket, AF_INET, SOCK_STREAM
import ssl

KEYFILE = 'server_key.pem'   # Private key of the server
CERTFILE = 'server_cert.pem' # Server certificate (given to client)

def echo_client(s):
    while True:
        data = s.recv(8192)
        if data == b'':
            break
        s.send(data)
    s.close()
    print('Connection closed')

def echo_server(address):
    s = socket(AF_INET, SOCK_STREAM)
    s.bind(address)
    s.listen(1)

    # Wrap with an SSL layer requiring client certs
    s_ssl = ssl.wrap_socket(s,
                            keyfile=KEYFILE,
                            certfile=CERTFILE,
                            server_side=True
                            )
    # Wait for connections
    while True:
        try:
            c,a = s_ssl.accept()
            print('Got connection', c, a)
            echo_client(c)
        except Exception as e:
            print('{}: {}'.format(e.__class__.__name__, e))

echo_server(('', 20000))
```

äÿŃéİćæĹŚäžňæijŤçď žäÿĂäÿłáőćæĹůçńŕè£đæŐěæIJ■åŁąåŹĺçŽĎäžď äžŠä¿Ńå■ŘãĂĆáőćæĹůçńŕäijŽèŕ

```
>>> from socket import socket, AF_INET, SOCK_STREAM
>>> import ssl
>>> s = socket(AF_INET, SOCK_STREAM)
>>> s_ssl = ssl.wrap_socket(s,
                 cert_reqs=ssl.CERT_REQUIRED,
                 ca_certs = 'server_cert.pem')
>>> s_ssl.connect(('localhost', 20000))
>>> s_ssl.send(b'Hello World?')
12
>>> s_ssl.recv(8192)
b'Hello World?'
>>>
```

è£Źçğ■çŻť æŐěåď ĎçŘĘåžŢåśĆsocketæŰźåijŕæIJĽäÿłéŮőéćŸåŕśæŸŕáőČäÿ■èČ¡å¿Ĺåě¡çŽĎèů§æăĞåČ
ä¿ŃåęĆïijŇçżIåď ğéČĺåĹĘæIJ■åŁąåŹĺäžčçăĄïijĹHTTPãĂĄXML-
RPCç■L'ïijL'åőđéŹĚäÿŁæŸŕå§žäžŐ socketserver ažŞçŽĎãĂĆ
áőćæĹůçńŕäžčçăĄåIJĺäÿĂäÿłè¿ČéńŸåśĆäÿŁáőđçŐřãĂĆæĹŚäžňéIJĂèęĄåŘęåď ŰäÿĂçğ■çĺ■å¿őäÿ■åŘŇçŽĎ

éęŰåĚĹïijŇåŕźäžŐæIJ■åŁąåŹĺèĂŇèĺĂïijŇåŘŕäžěéĂŽè£ĞåČŘäÿŃéİćè£Źæăůä¡£çŤĺäÿĂäÿłmixinçśżæİě

```
import ssl

class SSLMixin:
    '''
    Mixin class that adds support for SSL to existing servers based
    on the socketserver module.
    '''
    def __init__(self, *args,
                 keyfile=None, certfile=None, ca_certs=None,
                 cert_reqs=ssl.CERT_NONE,
                 **kwargs):
        self._keyfile = keyfile
        self._certfile = certfile
        self._ca_certs = ca_certs
        self._cert_reqs = cert_reqs
        super().__init__(*args, **kwargs)

    def get_request(self):
        client, addr = super().get_request()
        client_ssl = ssl.wrap_socket(client,
                                     keyfile = self._keyfile,
                                     certfile = self._certfile,
                                     ca_certs = self._ca_certs,
                                     cert_reqs = self._cert_reqs,
                                     server_side = True)
        return client_ssl, addr
```

äÿžäžĘä¡£çŤĺè£ŹäÿłmixinçśżïijŇä¡ääŕäžěářĘáőČèů§åĚűäžŰæIJ■åŁąåŹĺçśżæůůåŘĹãĂĆä¿ŃåęĆïijŇäÿ
RPCæIJ■åŁąåŹĺä¿Ńå■ŘïijŽ

```
# XML-RPC server with SSL

from xmlrpc.server import SimpleXMLRPCServer

class SSLSimpleXMLRPCServer(SSLMixin, SimpleXMLRPCServer):
    pass

Here's the XML-RPC server from Recipe 11.6 modified only slightly␣
↪to use SSL:

import ssl
from xmlrpc.server import SimpleXMLRPCServer
from sslmixin import SSLMixin

class SSLSimpleXMLRPCServer(SSLMixin, SimpleXMLRPCServer):
    pass

class KeyValueServer:
    _rpc_methods_ = ['get', 'set', 'delete', 'exists', 'keys']
    def __init__(self, *args, **kwargs):
        self._data = {}
        self._serv = SSLSimpleXMLRPCServer(*args, allow_none=True,␣
↪**kwargs)
        for name in self._rpc_methods_:
            self._serv.register_function(getattr(self, name))

    def get(self, name):
        return self._data[name]

    def set(self, name, value):
        self._data[name] = value

    def delete(self, name):
        del self._data[name]

    def exists(self, name):
        return name in self._data

    def keys(self):
        return list(self._data)

    def serve_forever(self):
        self._serv.serve_forever()

if __name__ == '__main__':
    KEYFILE='server_key.pem'     # Private key of the server
    CERTFILE='server_cert.pem'   # Server certificate
    kvserv = KeyValueServer(('', 15000),
                            keyfile=KEYFILE,
                            certfile=CERTFILE)
```

(continues on next page)

(continued from previous page)

```
    kvserv.serve_forever()
```

ä¡£çŤÍè£ŹäÿłæIJ■åŁąåŹÍæŮűïijÑä¡ăåŘŕäżěä¡£çŤÍæŹőéĂŽçŽĎ `xmlrpc.client` æÍąåÌŮæÌěè£đæŐěåőČãĂĆ åŘłéIJĂèęĄåIJÍURLäÿ■æŇĞåőŽ `https:` å■ţåŘŕïijÑä¿ŃåęĆïijŽ

```
>>> from xmlrpc.client import ServerProxy
>>> s = ServerProxy('https://localhost:15000', allow_none=True)
>>> s.set('foo','bar')
>>> s.set('spam', [1, 2, 3])
>>> s.keys()
['spam', 'foo']
>>> s.get('foo')
'bar'
>>> s.get('spam')
[1, 2, 3]
>>> s.delete('spam')
>>> s.exists('spam')
False
>>>
```

åŕźäžŐSSLåőćæĹůçńŕæÌěèőšäÿĂäÿłæŕŤè¿Čåď■æÌĆçŽĎéŮőéćŸæŸŕåęĆä¡Ţçąőèőď æIJ■åŁąåŹÍèŕĄäźęä
äÿ■åźÿçŽĎæŸŕïijÑæŽĆæŮűè£ŸæšąæIJĽäÿĂäÿłæăĞåĞĘæŰźæşŢæÌěèğčåĘşè£ŹäÿłéŮőéćŸïijÑéIJĂèęĄèĞł
äÿ■è£ĞïijÑäÿŃéÌćçżŹåĞžäÿĂäÿłä¿Ńå■ŘïijÑçŤÍæÌěåżžçńŃäÿĂäÿłåőĽåĚÍçŽĎXML-RPCè£đæŐěæÌěçąőèőď æIJ■åŁąåŹÍèŕĄäźęïijŽ

```
from xmlrpc.client import SafeTransport, ServerProxy
import ssl

class VerifyCertSafeTransport(SafeTransport):
    def __init__(self, cafile, certfile=None, keyfile=None):
        SafeTransport.__init__(self)
        self._ssl_context = ssl.SSLContext(ssl.PROTOCOL_TLSv1)
        self._ssl_context.load_verify_locations(cafile)
        if certfile:
            self._ssl_context.load_cert_chain(certfile, keyfile)
        self._ssl_context.verify_mode = ssl.CERT_REQUIRED

    def make_connection(self, host):
        # Items in the passed dictionary are passed as keyword
        # arguments to the http.client.HTTPSConnection()␣
↪constructor.
        # The context argument allows an ssl.SSLContext instance to
        # be passed with information about the SSL configuration
        s = super().make_connection((host, {'context': self._ssl_
↪context}))

        return s

# Create the client proxy
```

(continues on next page)

(continued from previous page)

```
s = ServerProxy('https://localhost:15000',
                transport=VerifyCertSafeTransport('server_cert.pem
↪'),
                allow_none=True)
```

æIJ■åŁąåŹÍåřĘèŕĄäźęåŘŚéĂĄçżŹåőćæĹůçńŕïijŇåőćæĹůçńŕæİěçąőèőď åőČçŽĎåŘĹæşŢæĂğãĂĆè£ŹçğĂ
åęĆæđIJæIJ■åŁąåŹÍæČşèęĄçąőèőď åőćæĹůçńŕïijŇåŘŕäžěåřĘæIJ■åŁąåŹÍåŘŕåŁÍäžčçăĄä£őæŤźåęĆäÿŃïijŽ

```
if __name__ == '__main__':
    KEYFILE='server_key.pem'    # Private key of the server
    CERTFILE='server_cert.pem' # Server certificate
    CA_CERTS='client_cert.pem' # Certificates of accepted clients

    kvserv = KeyValueServer(('', 15000),
                            keyfile=KEYFILE,
                            certfile=CERTFILE,
                            ca_certs=CA_CERTS,
                            cert_reqs=ssl.CERT_REQUIRED,
                            )
    kvserv.serve_forever()
```

äÿžäžĘèőľ XML-RPCåőćæĹůçńŕåŘŚéĂĄèŕĄäźęïijŇä£őæŤź `ServerProxy` çŽĎåĹİåğŃåŇŰäžčçăĄäęĆäÿŃïijŽ

```
# Create the client proxy
s = ServerProxy('https://localhost:15000',
                transport=VerifyCertSafeTransport('server_cert.pem',
                                                  'client_cert.pem',
                                                  'client_key.pem'),
                allow_none=True)
```

èőĺèőž

èŕŢçİĂåŐżè£ŘèąŇæIJňèŁĆçŽĎäžčçăĄèČ¡æţŃèŕŢä¡ăçŽĎçşżçż§éĚ■ç¡őèČ¡åŁŻåŠŇçŘĘèğčSSLãĂĆ
åŘŕèČ¡æIJĂåď ğçŽĎæŇŚæĹŸæŸŕåęĆä¡ŢäÿĂæ■ěæ■ěçŽĎèŐůåŘŰůĹİåğŃéĚ■ç¡őkeyãĂĄèŕĄäźęåŠŇåĚűäžŰ

æĹŚèğčéĞŁäÿŃåĹřåžŢéIJĂèęĄåŢěïijŇæŕŘäÿĂäÿłSSLè£đæŐěçżĹçńŕäÿĂèĹňéČ¡äijŽæIJL'äÿĂäÿłçğĄé
è£ŹäÿłèŕĄäźęåŇĚåŘňäžĘåĚňéŠěåźůåIJĺæŕŘäÿĂæňąè£đæŐěçŽĎæŮůåĂŹéČ¡äijŽåŘŚéĂĄçżŹåŕźæŰźãĂĆ
åŕźäžŐåĚňåĚśæIJ■åŁąåŹĺïijŇåőČäžňçŽĎèŕĄäźęéĂŽåÿÿæŸŕèćńæİČåĺĄèŕĄäźęæIJžæđĎæŕŤåęĆVerisignãĂA
äÿžäžĘçąőèőď æIJ■åŁąåŹĺç■¿åŘ■ïijŇåőćæĹůçńŕåŻđä£İå■ŸäÿĂäż¡åŇĚåŘňäžĘä£ąäżżæŐĹæİČæIJžæđĎçŽĎ
ä¿ŃåęĆïijŇwebæţŘèğĹåŹĺä£İå■ŸäžĘäÿżèęĄçŽĎèőď èŕĄæIJžæđĎçŽĎèŕĄäźęïijŇåźűä¡£çŤĺåőČæİěäÿžæŕŘä
åŕźæIJňåřŘèŁĆçď žä¿ŃèĂŇèĺĂïijŇåŘłæŸŕäÿžäžĘæţŃèŕŢïijŇæĹŚäżňåŘŕäžěåĹŻåżżèĞłç■¿åŘ■çŽĎèŕĄäźęïij

::

bash % openssl req -new -x509 -days 365 -nodes -out server_cert.pem -keyout server_key.pem

Generating a 1024 bit RSA private key âĂęâĂęâĂęâĂęâĂęâĂęâĂęâĂęâĂęâĂęâĂęâĂęâĂęâĂęâĂęâĂęâĂę++++++
âĂę++++++

writing new private key to âĂŸserver_key.pemâĂŹ

You are about to be asked to enter information that will be incorporated into your certificate request. What you are about to enter is what is called a Distinguished Name or a DN. There are quite a few fields but you can leave some blank For some fields there will be a default value, If you enter âĂŸ.âĂŹ, the field will be left blank.

Country Name (2 letter code) [AU]:US State or Province Name (full name) [SomeState]:Illinois Locality Name (eg, city) []:Chicago Organization Name (eg, company) [Internet Widgits Pty Ltd]:Dabeaz, LLC Organizational Unit Name (eg, section) []: Common Name (eg, YOUR name) []:localhost Email Address []: bash %

åIJÍåĹŻåżžèŕĄäźęçŽĎæŮűåĂŹïijŇåŘĎäÿłåĂijçŽĎèő¿åőŽåŕŕäżěæŸŕäżżæĎŕçŽĎïijŇä¡ĘæŸŕâĂİCom
NameâĂIJçŽĎåĂijéĂŽåÿÿèęĄåŇĚåŘńæIJ■åŁąåŹĺçŽĎDNSäÿżæIJżåŘ■ãĂĆ
åęĆæđIJä¡ääŕłæŸŕåIJĺæIJňæIJżæţŃèŕŢïijŇéĆčäźĹåŕśä¡£çŤĺâĂİlocalhostâĂIJïijŇåŘęåĹŹä¡£çŤĺæIJ■åŁąåŹĺç

:: âĂŤ–BEGIN RSA PRIVATE KEYâĂŤ– MIICXQIBAAKBgQCZrCNLoEyAKF+f9UNcFaz5Osa6jf7qkbUl8si5xQrY3ZYC7juu
nL1dZLn/VbEFIITaUOgvBtPv1qUWTJGwga62VSG1oFE0ODIx3g2Nh4sRf+rySsx2
L4442nx0z4O5vJQ7k6eRNHAZUUnCL50+YvjyLyt7ryLSjSuKhCcJsbZgPwIDAQAB
AoGAB5evrr7eyL4160tM5rHTeATlaLY3UBOe5Z8XN8Z6gLiB/ucSX9AysviVD/6F
3oD6z2aL8jbeJc1vHqjt0dC2dwwm32vVl8mRdyoAsQpWmiqXrkvP4Bsl04VpBeHw
Qt8xNSW9SFhceL3LEvw9M8i9MV39viih1ILyH8OuHdvJyFECQQDLEjl2d2ppxND9
PoLqVFAirDfX2JnLTdWbc+M11a9Jdn3hKF8TcxfEnFVs5Gav1MusicY5KB0ylYPb
YbTvqKc7AkEAwbnRBO2VYEZsJZp2X0IZqP9ovWokkpYx+PE4+c6MySDgaMcigL7v
WDIHJG1CHudD09GbqENasDzyb2HAIW4CzQJBAKDdkv+xoW6gJx42Auc2WzTcUHCA
eXR/+BLpPrhKykzbvOQ8YvS5W764SUO1u1LWs3G+wnRMvrRvlMCZKgggBjkCQQCG
Jewto2+a+WkOKQXrNNScCDE5aPTmZQc5waCYq4UmCZQcOjkUOiN3ST1U5iuxRqfb
V/yX6fw0qh+fLWtkOs/JAkA+okMSxZwqRtfgOFGBfwQ8/iKrnizeanTQ3L6scFXI
CHZXdJ3XQ6qUmNxNn7iJ7S/LDawo1QfWkCfD9FYoxBlg âĂŤ–END RSA PRIVATE KEYâĂŤ–

æIJ■åŁąåŹĺèŕĄäźęæŰĞäżűserver_cert.pemåĘĚåőźçśżäijijäÿŃéİćè£ŹæăůïijŽ

:: âĂŤ–BEGIN CERTIFICATEâĂŤ– MIIC+DCCAmGgAwIBAgIJAPMd+vi45js3MA0GCSqGSIb3DQEBB
BAYTAlVTMREwDwYDVQQIEwhJbGxpbm9pczEQMA4GA1UEBxMHQ2hpY2FnbzEUMBIG
A1UEChMLRGFiZWF6LCBMTEMxEjAQBgNVBAMTCWxvY2FsaG9zdDAeFw0xMzAxMTEx
ODQyMjdaFw0xNDAxMTExODQyMjdaMFwxCzAJBgNVBAYTAlVTMREwDwYDVQQIEwhJ
bGxpbm9pczEQMA4GA1UEBxMHQ2hpY2FnbzEUMBIGA1UEChMLRGFiZWF6LCBMTEMx
EjAQBgNVBAMTCWxvY2FsaG9zdDCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA
mawjS6BMgChfn/VDXBWs+TrGuo3+6pG1JfLIucUK2N2WAu47rpy9XWS5/1WxBSCE
2lDoLwbT79alFkyRsIGutlUhtaBRNDgyMd4NjYeLEX/q8krMdi+OONp8dM+DubyU

O5OnkTRwGVFJwi+dPmL48i8re68i0o0rioQnCbG2YD8CAwEAAaOBwTCBvjAdBgNV
HQ4EFgQUrtoLHHgXiDZTr26NMmgKJLJLFtIwgY4GA1UdIwSBhjCBg4AUrtoLHHgX
iDZTr26NMmgKJLJLFtKhYKReMFwxCzAJBgNVBAYTAlVTMREwDwYDVQQIEwhJbGxp

bm9pczEQMA4GA1UEBxMHQ2hpY2FnbzEUMBIGA1UEChMLRGFiZWF6LCBMTEMxEjAQ
BgNVBAMTCWxvY2FsaG9zdIIJAPMd+vi45js3MAwGA1UdEwQFMAMBAf8wDQYJKoZI
hvcNAQEFBQADgYEAFci+dqvMG4xF8UTnbGVvZJPIzJDRee6Nbt6AHQo9pOdAIMAu
WsGCplSOaDNdKKzl+b2UT2Zp3AIW4Qd51bouSNnR4M/gnr9ZD1ZctFd3jS+C5XRp
D3vvcW5lAnCCC80P6rXy7d7hTeFu5EYKtRGXNvVNd/06NALGDflrrOwxF3Y=
âĂŤ–END CERTIFICATEâĂŤ–

åIJÍæIJ■åŁąåŹÍçńŕäżčçăĄäÿ■ïijŇçğĄéŠěåŠŇèŕĄäźęæŰĞäżűäijŽèćńäijăçżŹSSLçŻÿåĚşçŽĎåŇĚèčĚåĞ¡
çğĄéŠěåžŤèŕěåIJÍä£Ìå■ŸåIJÍæIJ■åŁąåŹÍäÿ■ïijŇåźűåŁääżěåőL'åĚÍä£ÌæŁď'ãĂĆ

åIJÍåőćæĹůçńŕäżčçăĄäÿ■ïijŇéIJĂèęĄä£Ìå■ŸäÿĂäÿłåŘĹæşŢèŕĄäźęæŐĹæÌČæŰĞäżűæÌěçąőèőď'æIJ■åŁ
åęĆæđIJä¡äæšąæIJL'è£ŹäÿłæŰĞäżűïijŇä¡ääŘŕäżěåIJÍåőćæĹůçńŕåď'■åĹűäÿĂäż¡æIJ■åŁąåŹÍçŽĎèŕĄäźęåźűä¡
è£đæŐěåżžçńŇåŘŐïijŇæIJ■åŁąåŹÍäijŽæŘŘä¿ŻåőČçŽĎèŕĄäźęïijŇçĎűåŘŐä¡ääŕśèČ¡ä¡£çŤÍåůšçżŘä£Ìå■ŸçŽ

æIJ■åŁąåŹÍäź§èČ¡éĂL'æŃl'æŸŕåŘęèęĄçąőèőď'åőćæĹůçńŕçŽĎèžńäż¡ãĂĆåęĆæđIJèęĄè£ŹæăůåĄŽçŽĎ
æIJ■åŁąåŹÍäź§éIJĂèęĄä£Ìå■ŸäÿĂäÿłèćńä£ąäżżèŕĄäźęæŐĹæÌČæŰĞäżűæÌěçąőèőď'åőćæĹůçńŕèŕĄäźęăĂĆ

åęĆæđIJä¡äèęĄåIJÍçIJ§åőđçŐŕåćČäÿ■äÿžä¡äçŽĎç¡ŚçżIJæIJ■åŁąåŁääÿŁSSLçŽĎæŤŕæŇĄïijŇè£ŹåŕŘèŁ
ä¡äè£ŸåžŤèŕěåŘĆèĂČåĚűäżŰçŽĎæŰĞæąčïijŇåĄŽåě¡èŁśèť'źäÿ■åŕŚæŮűéŮť'æÌěæţŃèŕŢåőČæ■čåÿÿåůěä¡IJ
^_^

13.11 11.11 è£ŻçÍŃéŮť'äijăéĂŠSocketæŰĞäżűæŘŘè£řçňę

éŮőéćŸ

ä¡äæIJL'åď'Žäÿł Pythonèğčé ĞŁåŹÍè£ŻçÍŃåIJÍåŘŇæŮűè£ŘèąŇïijŇä¡äæČşåřĘæ§ŘäÿłæL'ŞåijĂçŽĎæŰĞä
æŕŤåęĆïijŇåĄĞèő¿æIJL'äÿłæIJ■åŁąåŹÍè£ŻçÍŃçŻÿåžŤè£đæŐěèŕůæśĆïijŇä¡ĘæŸŕåőđéŹĚçŽĎçŻÿåžŤéĂżè¿Ş

èğčåĘşæŰźæąĹ

äÿžäžĘåIJÍåď'Žäÿłè£ŻçÍŃäÿ■äijăéĂŠæŰĞäżűæŘŘè£řçňęïijŇä¡äéęŰåĚĹéIJĂèęĄåřĘåőČäżňè£đæŐěåĹřä
èĂŇåIJÍwindowsäÿŁéÌćä¡äéIJĂèęĄä¡£çŤÍåŚ¡åŘ■çőąéĄŞãĂĆäÿ■è£Ğä¡äæŮäéIJĂçIJ§çŽĎéIJĂèęĄåŐżæŞ■ä¡
éĂŽåÿÿä¡£çŤÍ `multiprocessing` æÍąåÌŮæÌěåĹŻåżžè£ŹæäůçŽĎè£đæŐěäijŽæŻť'åőźæŸŞäÿĂäżŽãĂĆ

äÿĂæŮęäÿĂäÿłè£đæŐěèćńåĹŻåżžïijŇä¡ääŘŕäżěä¡£çŤÍ `multiprocessing.reduction` äÿ■çŽĎ `send_handle()` åŠŇ `recv_handle()` åĞ¡æŢŕåIJÍäÿ■åŘŇçŽĎåď'ĎçŘĘåŹÍçŻť'æŐěäijăéĂŠæŰĞäżűæŘŘè£řçňęãĂĆ
äÿŃéÌćçŽĎä¿Ńå■ŘæijŤçď'žäžĘæIJĂå§žæIJňçŽĎçŤÍæşŢïijŽ

```
import multiprocessing
from multiprocessing.reduction import recv_handle, send_handle
import socket

def worker(in_p, out_p):
    out_p.close()
    while True:
        fd = recv_handle(in_p)
        print('CHILD: GOT FD', fd)
```

(continues on next page)

(continued from previous page)

```
        with socket.socket(socket.AF_INET, socket.SOCK_STREAM,␣
↪fileno=fd) as s:
            while True:
                msg = s.recv(1024)
                if not msg:
                    break
                print('CHILD: RECV {!r}'.format(msg))
                s.send(msg)

def server(address, in_p, out_p, worker_pid):
    in_p.close()
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, True)
    s.bind(address)
    s.listen(1)
    while True:
        client, addr = s.accept()
        print('SERVER: Got connection from', addr)
        send_handle(out_p, client.fileno(), worker_pid)
        client.close()

if __name__ == '__main__':
    c1, c2 = multiprocessing.Pipe()
    worker_p = multiprocessing.Process(target=worker, args=(c1,c2))
    worker_p.start()

    server_p = multiprocessing.Process(target=server,
                  args=(('', 15000), c1, c2, worker_p.pid))
    server_p.start()

    c1.close()
    c2.close()
```

åIJÍè£Źäÿłä¿Ńå■Řäÿ■ïijŇäÿď äÿłè£ŻçÍŃèćńåĹŻåżżåźűéĂŽè£ĞäÿĂäÿł multiprocessing çőąéĄŞè£đæŐěęţůæİěãĂĆ æIJ■åŁąåŹÍè£ŻçÍŃæĽŞåijĂäÿĂäÿłsocketåźűç■Ľ å¿Ěåőćæĺ åůěä¡IJè£ŻçÍŃäżĚäżĚä¡£çŤÍ recv_handle() åIJÍçőąéĄŞäÿŁéÍćç■Ľ å¿ĚæŐěæŤűäÿĂäÿłæŰĞäżűæŘŘè£řç å¡ŞæIJ■åŁąåŹÍæŐěæŤűåĹřäÿĂäÿłè£đæŐěïijŇåőČåřĘäžğçŤ§çŽĎsocketæŰĞäżűæŘŘè£řçňęéĂŽè£Ğ send_handle() äijäéĂŠçżŹåůěä¡IJè£ŻçÍŃãĂĆ åůěä¡IJè£ŻçÍŃæŐěæŤűåĹřsocketåŘŐåŘŚåőćæĹůçńřåŻđå

åęĆæđIJä¡ăä¡£çŤÍTelnetæĹŰçśżäijijåůěåĚůè£đæŐěåĹřæIJ■åŁąåŹÍïijŇäÿŃéÍćæŸřäÿĂäÿłæijŤçď žä¿Ńå■

bash % python3 passfd.py SERVER: Got connection from (âĂŸ127.0.0.1âĂŹ, 55543) CHILD: GOT FD 7 CHILD: RECV bâĂŹHellornâĂŹ CHILD: RECV bâĂŹWorldrnâĂŹ

æ■ď ä¿ŃæIJĂéĞ■èęĄçŽĎéČÍåĹĘæŸřæIJ■åŁąåŹÍæŐěæŤűåĹřçŽĎåőćæĹůçńřsocketåőđéŹĚäÿŁèćńåŘęå æIJ■åŁąåŹÍäżĚäżĚåŘłæŸřåřĘåĚűè¡ňæĽŃåźűåĚşéŮ■æ■ď è£đæŐěïijŇçĎűåŘŐç■Ľ å¿ĚäÿŃäÿĂäÿłè£đæŐěãĂ

èőÍèőž

åŕźäžŐåď ğéČÍåĹEçÍŃåžŘåŚŸæİěèőšåIJÍäÿ■åŘŇè£ŻçÍŃäźŃéŮť äijäéĂŠæŰĞäžűæŘŕè£řçňęåě¡åČŕæšą
ä¡ĘæŸŕïijŇæIJĹ æŮűåĂŹåőČæŸŕæđĎåžžäÿĂäÿłåŘŕæĹ ľ åśŢçşżçż§çŽĎå¿ĹæIJĹ çŤÍçŽĎåůěåĚůãĂĆä¿ŃåęĆ
ä¡ääŘŕäžěæIJĹ åď ŽäÿłPythonèğčéĞŁåŹÍåőđä¿ŃïijŇåřĘæŰĞäžűæŘŕè£řçňęäijäéĂŠçżŹåĚűåőČèğčéĞŁåŹÍæ

send_handle() åŠŇ recv_handle() åĞ¡æŢřåŘłèČ¡åď §çŤÍäžŐ multiprocessing è£đæŐěãĂĆ ä¡£çŤÍåőČäžňæİěäżčæŻ£çőąéĄŞçŽĎä¡£çŤÍïijĹåŘĆèĂČ11.7èŁĆïijĹ ïijŇ
ä¿ŃåęĆïijŇä¡ääŘŕäžěèőľ æIJ■åŁąåŹÍåŠŇåůěä¡IJèĂĚåŘĎèĞłäžěå■ŢçŃňçŽĎçÍŃåžŘæİěåŘŕåŁÍãĂĆäÿŃéİćæŸ

```
# servermp.py
from multiprocessing.connection import Listener
from multiprocessing.reduction import send_handle
import socket

def server(work_address, port):
    # Wait for the worker to connect
    work_serv = Listener(work_address, authkey=b'peekaboo')
    worker = work_serv.accept()
    worker_pid = worker.recv()

    # Now run a TCP/IP server and send clients to worker
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, True)
    s.bind(('', port))
    s.listen(1)
    while True:
        client, addr = s.accept()
        print('SERVER: Got connection from', addr)

        send_handle(worker, client.fileno(), worker_pid)
        client.close()

if __name__ == '__main__':
    import sys
    if len(sys.argv) != 3:
        print('Usage: server.py server_address port', file=sys.
↪stderr)
        raise SystemExit(1)

    server(sys.argv[1], int(sys.argv[2]))
```

è£ŘèąŇè£ŹäÿłæIJ■åŁąåŹÍïijŇåŘłéIJĂèęĄæĹ ğèąŇ *python3 servermp.py /tmp/servconn 15000* ïijŇäÿŃéİćæŸŕçŻÿåžŢçŽĎåůěä¡IJèĂĚäžčçăĄïijŽ

```
# workermp.py

from multiprocessing.connection import Client
from multiprocessing.reduction import recv_handle
import os
```

(continues on next page)

(continued from previous page)

```
from socket import socket, AF_INET, SOCK_STREAM

def worker(server_address):
    serv = Client(server_address, authkey=b'peekaboo')
    serv.send(os.getpid())
    while True:
        fd = recv_handle(serv)
        print('WORKER: GOT FD', fd)
        with socket(AF_INET, SOCK_STREAM, fileno=fd) as client:
            while True:
                msg = client.recv(1024)
                if not msg:
                    break
                print('WORKER: RECV {!r}'.format(msg))
                client.send(msg)

if __name__ == '__main__':
    import sys
    if len(sys.argv) != 2:
        print('Usage: worker.py server_address', file=sys.stderr)
        raise SystemExit(1)

    worker(sys.argv[1])
```

èęĄè£ŘèąŇåůëä¡IJèĂĚïijŇæL'ğèąŇæL'ğèąŇåŚ¡äżď *python3 workermp.py /tmp/servconn* . æŢĹæđIJèů§ä¡£çŤÍPipe()ä¿Ńå■ŘæŸŕåőŇåĚÍäÿĂæăůçŽĎãĂĆ æŰĞäżűæŔŘè£řçňęçŽĎäijäéĂŠäijŽæűL'åŔŁåĹřUNIXå§§åěŮæŐěå■ŮçŽĎåĹŻåżžåŠŇåěŮæŐěå■ŮçŽĎ sendmsg() æŰźæşŢãĂĆ äÿ■è£Ğè£ŹçğĘæŁĂæIJŕåźűäÿ■åÿÿèğĄïijŇäÿŃéĺćæŸŕä¡£çŤĺåěŮæŐěå■Ůæĺěäijăĕ

```
# server.py
import socket

import struct

def send_fd(sock, fd):
    '''
    Send a single file descriptor.
    '''
    sock.sendmsg([b'x'],
                 [(socket.SOL_SOCKET, socket.SCM_RIGHTS, struct.
↪pack('i', fd))])
    ack = sock.recv(2)
    assert ack == b'OK'

def server(work_address, port):
    # Wait for the worker to connect
    work_serv = socket.socket(socket.AF_UNIX, socket.SOCK_STREAM)
    work_serv.bind(work_address)
    work_serv.listen(1)
```

(continues on next page)

(continued from previous page)

```
    worker, addr = work_serv.accept()

    # Now run a TCP/IP server and send clients to worker
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, True)
    s.bind(('',port))
    s.listen(1)
    while True:
        client, addr = s.accept()
        print('SERVER: Got connection from', addr)
        send_fd(worker, client.fileno())
        client.close()

if __name__ == '__main__':
    import sys
    if len(sys.argv) != 3:
        print('Usage: server.py server_address port', file=sys.
↪stderr)
        raise SystemExit(1)

    server(sys.argv[1], int(sys.argv[2]))
```

äÿŃéĬćæŸŕä¡£çŤĺåěŮæŐěå■ŮçŽĎåůěä¡IJèĂĚåőđçŐřïijŽ

```
# worker.py
import socket
import struct

def recv_fd(sock):
    '''
    Receive a single file descriptor
    '''
    msg, ancdata, flags, addr = sock.recvmsg(1,
                                        socket.CMSG_LEN(struct.
↪calcsize('i')))

    cmsg_level, cmsg_type, cmsg_data = ancdata[0]
    assert cmsg_level == socket.SOL_SOCKET and cmsg_type == socket.
↪SCM_RIGHTS
    sock.sendall(b'OK')

    return struct.unpack('i', cmsg_data)[0]

def worker(server_address):
    serv = socket.socket(socket.AF_UNIX, socket.SOCK_STREAM)
    serv.connect(server_address)
    while True:
        fd = recv_fd(serv)
        print('WORKER: GOT FD', fd)
```

(continues on next page)

(continued from previous page)

```
        with socket.socket(socket.AF_INET, socket.SOCK_STREAM,␣
↪fileno=fd) as client:
            while True:
                msg = client.recv(1024)
                if not msg:
                    break
                print('WORKER: RECV {!r}'.format(msg))
                client.send(msg)

if __name__ == '__main__':
    import sys
    if len(sys.argv) != 2:
        print('Usage: worker.py server_address', file=sys.stderr)
        raise SystemExit(1)

    worker(sys.argv[1])
```

åęĆæđIJä¡ăæČşåIJÍä¡ăçŽĎçÍŃåžŘäÿ■äijăéĂŠæŰĞäżűæŘŘè£řçňęïijŃåżżèőőä¡ăåŘĆéŸĚåĚűäżŰäÿĂäžŽ
æŕŤåęĆ `Unix Network Programming by W. Richard Stevens (Prentice Hall, 1990)`. åIJÍWindowsäÿŁäijăéĂŠæŰĞäżűæŘŘè£řçňęèů§UnixæŸŕäÿ■äÿĂæăůçŽĎïijŃåżżèőőä¡ăçăŤ
`multiprocessing.reduction` äÿ■çŽĎæžŘäżčçăĄçIJŃçIJŃåĚűåůěä¡IJåŐ§çŘĘãĂĆ

13.12 11.12 çŘĘèğčäžŃäżűéľśåŁÍçŽĎIO

éŮőécŸ

ä¡ăåžŤèŕěåůšçżŘåŘňè£Ğå§žäžŐäžŃäżűéľśåŁÍæĹŰåijĆæ■ěI/OçŽĎåŇĚïijŃä¡ĘæŸŕä¡ăè£Ÿäÿ■èČ¡åőŃåŁ
æĹŰèĂĚæŸŕåęĆæđIJä¡£çŤÍåőČçŽĎèŕÍäijŽåŕźä¡ăçŽĎçÍŃåžŘäžğçŤ§äżĂäźĹå¡śåŞ■ãĂĆ

èğčåĘşæŰźæąĹ

äžŃäżűéľśåŁÍI/OæIJňèťÍäÿŁæÍěèőšåřśæŸŕåřĘå§žæIJňI/OæŞ■ä¡IJïijĹæŕŤåęĆèŕżåŠŇåĘŹïijL'è¡ňåŇŰäÿž
ä¿ŃåęĆïijŃå¡ŞæŢřæ■őåIJÍæ§ŘäÿłsocketäÿŁèćńæŐěåŘŮåŘŐïijŃåőČäijŽè¡ňæ■ćæĹŘäÿĂäÿł
`receive` äžŃäżűïijŃçĎŰůåŘŐèćńä¡ăőŽäźL'çŽĎåŽđèřČæŰźæşŢæĹŰåĞ¡æŢřæÍěåď'ĎçŘĘãĂĆ
ä¡IJäÿžäÿĂäÿłåŘŕèČ¡çŽĎèţůåğŃçĆźïijŃäÿĂäÿłäžŃäżűéľśåŁÍçŽĎæąĘæđűåŘŕèČ¡äijŽäżěäÿĂäÿłăőđçŐřäžĘä

```
class EventHandler:
    def fileno(self):
        'Return the associated file descriptor'
        raise NotImplemented('must implement')

    def wants_to_receive(self):
        'Return True if receiving is allowed'
        return False

    def handle_receive(self):
```

(continues on next page)

(continued from previous page)

```
        'Perform the receive operation'
        pass

    def wants_to_send(self):
        'Return True if sending is requested'
        return False

    def handle_send(self):
        'Send outgoing data'
        pass
```

è£ŹäÿłçśżçŽĎåŏđä¿Ńä¡IJäÿžæŘŠäżűèćńæŤ¿åĚěçśżäijijäÿŃéİćè£ŹæăůçŽĎäžŃäżűå¿łçŐŕäÿ■ïijŽ

```
import select

def event_loop(handlers):
    while True:
        wants_recv = [h for h in handlers if h.wants_to_receive()]
        wants_send = [h for h in handlers if h.wants_to_send()]
        can_recv, can_send, _ = select.select(wants_recv, wants_
↪send, [])
        for h in can_recv:
            h.handle_receive()
        for h in can_send:
            h.handle_send()
```

äžŃäżűå¿łçŐŕçŽĎåĚşéŤőéČĺåĹĘæŸŕ `select()` èřČçŤĺïijŇåőČäijŽäÿ■æŰ■è¡ŏèŕćæŰĞäżűæŘŘè£řçňę
åIJĺèřČçŤĺ `select()` äźŃåĽ■ïijŇäžŃäżűå¿łçŐŕäijŽèŕćéŮőæĽĂæIJĽçŽĎåď ĎçŘĘåŹĺæİěåĘşåőŽåŞłäÿĂäÿ
çĎűåŘŐåőČåřĘçżŞæđIJåĹŮèąĺæŘŘä¿ŻçżŹ `select()` ãĂĆçĎűåŘŐ `select()`
è£ŤåŽđåĞĘåď ĞæŐěåŘŮæĹŰåŘŚéĂĄçŽĎåŕźèśąçżĎæĹŘçŽĎåĹŮèąĺãĂĆ
çĎűåŘŐçŻÿåžŤçŽĎ `handle_receive()` æĹŰ `handle_send()`
æŰźæşŢèćńèğęåŘŚãĂĆ

çijŰåĘŹåžŤçŤĺçĺŃåžŘçŽĎæŮűåĂŹïijŇEventHandler
çŽĎåŏđä¿ŃäijŽèćńåĹŻåżžãĂĆä¿ŃåęĆïijŇäÿŃéİćæŸŕäÿď äÿłçŐĂå■ŢçŽĎå§žäžŐUDPç¡ŚçżIJæIJ■åŁąçŽĎåď

```
import socket
import time

class UDPServer(EventHandler):
    def __init__(self, address):
        self.sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
        self.sock.bind(address)

    def fileno(self):
        return self.sock.fileno()

    def wants_to_receive(self):
        return True
```

(continues on next page)

(continued from previous page)

```
class UDPTimeServer(UDPServer):
    def handle_receive(self):
        msg, addr = self.sock.recvfrom(1)
        self.sock.sendto(time.ctime().encode('ascii'), addr)

class UDPEchoServer(UDPServer):
    def handle_receive(self):
        msg, addr = self.sock.recvfrom(8192)
        self.sock.sendto(msg, addr)

if __name__ == '__main__':
    handlers = [ UDPTimeServer(('',14000)), UDPEchoServer(('',
↪15000))  ]
    event_loop(handlers)
```

æţŃèŕŢè£ŹæőţäżčçăĄïijŇèŕŢçİĂäżŐåŔęåď'ŰäÿĂäÿłPythonèğčéĞŁåŹÍè£đæŐěåőČïijŽ

```
>>> from socket import *
>>> s = socket(AF_INET, SOCK_DGRAM)
>>> s.sendto(b'',('localhost',14000))
0
>>> s.recvfrom(128)
(b'Tue Sep 18 14:29:23 2012', ('127.0.0.1', 14000))
>>> s.sendto(b'Hello',('localhost',15000))
5
>>> s.recvfrom(128)
(b'Hello', ('127.0.0.1', 15000))
>>>
```

åőđçŐřäÿĂäÿłTCPæIJ■åŁąåŹÍäijŽæŻť'åŁåď'■æİĆäÿĂçĆźïijŇåŻääÿžæŕŔäÿĂäÿłåőćæĹůçńŕéČ¡èęĄåĹİäÿŃéİćæŸŕäÿĂäÿłTCPåžŤç■ŤåőćæĹůçńŕä¿Ńå■ŘïijŽ

```
class TCPServer(EventHandler):
    def __init__(self, address, client_handler, handler_list):
        self.sock = socket.socket(socket.AF_INET, socket.SOCK_
↪STREAM)
        self.sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR,
↪ True)
        self.sock.bind(address)
        self.sock.listen(1)
        self.client_handler = client_handler
        self.handler_list = handler_list

    def fileno(self):
        return self.sock.fileno()

    def wants_to_receive(self):
        return True
```

(continues on next page)

(continued from previous page)

```
    def handle_receive(self):
        client, addr = self.sock.accept()
        # Add the client to the event loop's handler list
        self.handler_list.append(self.client_handler(client, self.
↪handler_list))

class TCPClient(EventHandler):
    def __init__(self, sock, handler_list):
        self.sock = sock
        self.handler_list = handler_list
        self.outgoing = bytearray()

    def fileno(self):
        return self.sock.fileno()

    def close(self):
        self.sock.close()
        # Remove myself from the event loop's handler list
        self.handler_list.remove(self)

    def wants_to_send(self):
        return True if self.outgoing else False

    def handle_send(self):
        nsent = self.sock.send(self.outgoing)
        self.outgoing = self.outgoing[nsent:]

class TCPEchoClient(TCPClient):
    def wants_to_receive(self):
        return True

    def handle_receive(self):
        data = self.sock.recv(8192)
        if not data:
            self.close()
        else:
            self.outgoing.extend(data)

if __name__ == '__main__':
   handlers = []
   handlers.append(TCPServer(('',16000), TCPEchoClient, handlers))
   event_loop(handlers)
```

TCPä¿Ńå■ŘçŽĎåĚşéŤőçĆźæŸŕäżŐåď ĎçŘĘåŹÍäÿ■åĹŮèąĺåćđåŁäåŠŇåĹäéŹď åőćæĹůçńŕçŽĎæŞ■ä¡IJã
åŕźæŕŘäÿĂäÿłè£đæŐěïijŇäÿĂäÿłæŰřçŽĎåď ĎçŘĘåŹÍèćńåĹŻåżžåźůåŁäåĹřåĹŮèąĺäÿ■ãĂĆå¡Şè£đæŐěèćńåĚ
åęĆæđIJä¡äè£ŘèąŇçÍŃåžŘåźűèŕŢçİĂçŤÍTelnetæĹŰçśżäijijåůěåĚůè£đæŐěïijŇåőČäijŽåřĘä¡äåŘŚéĂĄçŽĎæŭ

èőÍèőž

åőđéŹĚäÿŁæĽĂæIJĽçŽĎäžŃäžűéľśåŁÍæąĘæđűåŐ§çŘĘèů§äÿŁéÍćçŽĎä¿Ńå■ŘçŻÿåůőæŮäåĞäãĂĆåőć
ä¡ĘæŸŕåIJÍæIJĂæäÿå£ČçŽĎéČÍåŁĘïijŇéČ¡äijŽæIJĽäÿĂäÿłè¡őèŕćçŽĎå¿łçŐŕæÍěæčĂæ§ěæť żåŁÍsocketïijŇäz

äžŃäžűéľśåŁÍI/OçŽĎäÿĂäÿłåŘŕèČ¡åě¡åď ĎæŸŕåőČèČ¡åď ĎçŘĘéÍđåÿÿåď ğçŽĎåźűåŘŚè£đæŐěïijŇèĂŃ
äź§åřśæŸŕèŕť'ïijŇselect() èřČçŤÍïijĹæĹŰåĚűäžŰç■Ľ'æŢĹçŽĎïijĽ'èČ¡çŽŚåŘňåď ğéĞŘçŽĎsocketåźűåŞ■
åIJÍå¿łçŐŕäÿ■äÿĂæňąåď ĎçŘĘäÿĂäÿłäžŃäžűïijŇåźűäÿ■éIJĂèęĄåĚűäžŰçŽĎåźűåŘŚæIJžåĹűãĂĆ

äžŃäžűéľśåŁÍI/OçŽĎçijžçĆźæŸŕæšąæIJĽçIJ§æ■čçŽĎåŘŇæ■ěæIJžåĹűãĂĆ
åęĆæđIJäżżä¡ŢäžŃäžűåď ĎçŘĘåŹÍæŰźæşŢéŸżåąđæĹŰæĽ'ğèąŃäÿĂäÿłèĂŮæŮűěőąçőŮïijŇåőČäijŽéŸżåąđa
èřČçŤÍéĆčäžŻåźűäÿ■æŸŕäžŃäžűéľśåŁÍéčŐæäijçŽĎåžŞåĞ¡æŢŕäź§äijŽæIJĽéŮőéćŸïijŇåŘŇæäůèęĄæŸŕæ§I

åŕźäžŐéŸżåąđæĹŰèĂŮæŮűěőąçőŮçŽĎéŮőéćŸåŘŕäżěéĂŽè£ĞåřĘäžŃäžűåŘŚéĂĄäÿłåĚűäžŰå■ŢçŃňç
äÿ■è£ĞïijŇåIJÍäžŃäžűå¿łçŐŕäÿ■åijŢåĚěåď Žçž£çÍŃåŠŇåď Žè£ŻçÍŃæŸŕæŕŤè¿ČæčŸæĽ'ŃçŽĎïijŇ
äÿŃéÍćçŽĎä¿Ńå■ŘæijŤçď žäžĘåęĆä¡Ţä¡£çŤÍ concurrent.futures
æÍąåÍŮæÍěåőđçŐŕïijŽ

```
from concurrent.futures import ThreadPoolExecutor
import os

class ThreadPoolHandler(EventHandler):
    def __init__(self, nworkers):
        if os.name == 'posix':
            self.signal_done_sock, self.done_sock = socket.
↪socketpair()
        else:
            server = socket.socket(socket.AF_INET, socket.SOCK_
↪STREAM)
            server.bind(('127.0.0.1', 0))
            server.listen(1)
            self.signal_done_sock = socket.socket(socket.AF_INET,
                                                  socket.SOCK_
↪STREAM)
            self.signal_done_sock.connect(server.getsockname())
            self.done_sock, _ = server.accept()
            server.close()

        self.pending = []
        self.pool = ThreadPoolExecutor(nworkers)

    def fileno(self):
        return self.done_sock.fileno()

    # Callback that executes when the thread is done
    def _complete(self, callback, r):

        self.pending.append((callback, r.result()))
        self.signal_done_sock.send(b'x')

    # Run a function in a thread pool
```

(continues on next page)

(continued from previous page)

```
    def run(self, func, args=(), kwargs={},*,callback):
        r = self.pool.submit(func, *args, **kwargs)
        r.add_done_callback(lambda r: self._complete(callback, r))

    def wants_to_receive(self):
        return True

    # Run callback functions of completed work
    def handle_receive(self):
        # Invoke all pending callback functions
        for callback, result in self.pending:
            callback(result)
            self.done_sock.recv(1)
        self.pending = []
```

åIJÍäžčçăĄäÿ■ïijŇrun() æŰźæşŢèćńçŤÍæĺěåřĘåůëä¡IJæŔŘäžď'çżŹåŻđèřČåĞ¡æŢřæśäïijŇåď'ĎçŘĘåőŇ
åőđéŹĚåůëä¡IJèćńæŔŘäžď'çżŹ ThreadPoolExecutor åőđä¿ŃãĂĆ
äÿ■è£ĞäÿĂäÿłéŽ¿çĆźæŸŕå■ŔèřČèőąçőŮçżŞæđIJåŠŇäžŃäžűå¿łçŐŕïijŇäÿžäžĘèğčåĘşåőČïijŇæĹŚäżňåĹŻåż
å¡Şçž£çÍŃæśäåőŇæĹŘåůëä¡IJåŘŐïijŇåőČäijŽæL'ğèąŇçśżäÿ■çŽĎ _complete()
æŰźæşŢãĂĆ è£ŹäÿłæŰźæşŢåĘ■æ§ŔäÿłsocketäÿŁåĘŹåĚěå■ŮèŁĆäźŃåL'■äijŽèőšæŇĆèţůçŽĎåŻđèřČåĞ¡æŢ
fileno() æŰźæşŢè£ŤåŻđåŔęåď'ŰçŽĎéĆčäÿłsocketãĂĆ åŻäæ■ď'ïijŇè£Źäÿłå■ŮèŁĆèćńåĘŹåĚěæŮűïijŇå
çĎűåŘŐ handle_receive() æŰźæşŢèćńæ£Ăæť'żåźűäÿžæL'ĂæIJL'äźŃåL'■æŔŘäžď'çŽĎåůëä¡IJæL'ğèąŇ
åĺęçŹ¡èőšïijŇèŕť'äžĘè£ŹäźĹåď'Žè£đæĹŚèĞłåůšéČ¡æŹŢäžĘãĂĆ
äÿŃéĺćæŸŕäÿĂäÿłçőĂå■ŢçŽĎæIJ■åŁąåŹĺïijŇæijŤçď'žäžĘåęĆä¡Ţä¡£çŤĺçž£çÍŃæśäæĺěåőđçŐřèĂŮæŮűçŽĎè

```
# A really bad Fibonacci implementation
def fib(n):
    if n < 2:
        return 1
    else:
        return fib(n - 1) + fib(n - 2)

class UDPFibServer(UDPServer):
    def handle_receive(self):
        msg, addr = self.sock.recvfrom(128)
        n = int(msg)
        pool.run(fib, (n,), callback=lambda r: self.respond(r,␣
↪addr))

    def respond(self, result, addr):
        self.sock.sendto(str(result).encode('ascii'), addr)

if __name__ == '__main__':
    pool = ThreadPoolHandler(16)
    handlers = [ pool, UDPFibServer(('',16000))]
    event_loop(handlers)
```

è£ŘèąŇè£ŹäÿłæIJ■åŁąåŹĺïijŇçĎűåŘŐèŕŢçĺĂçŤĺåĚűåőČPythonçÍŃåžŔæĺěæţŃèŕŢåőČïijŽ

```
from socket import *
sock = socket(AF_INET, SOCK_DGRAM)
for x in range(40):
    sock.sendto(str(x).encode('ascii'), ('localhost', 16000))
    resp = sock.recvfrom(8192)
    print(resp[0])
```

ä¡ăăžŤèŕěèČ¡åIJÍäÿ■åŘŇçłŮåŘčäÿ■éĞ■åď■çŽĎæĽğèąŇè£ŹäÿłçÍŃåžŘïijŇåźűäÿŤäÿ■äijŽå¡śåŞ■åĹřåĚ

åůšçżŘéŸĚèŕżåőŇäžĘè£ŹäÿĂåřŘèŁĆïijŇéĆčäźĹä¡ăăžŤèŕěä¡£çŤÍè£ŹéĞŇçŽĎäżčçăĄåŘŮïij§äź§èőÿäÿ■
äÿ■è£ĞïijŇåęĆæđIJä¡ăçŘĘèğčäžĘå§žæIJňåŐ§çŘĘïijŇä¡ăářśèČ¡çŘĘèğčè£ŹäžŻæąĘæđűæĽĂä¡£çŤÍçŽĎæăÿ
ä¡IJäÿžářźåŻđèřČåĞ¡æŢřçijŰçÍŃçŽĎæŻ£äżčïijŇäžŃäżűéľśåŁÍçijŰçăĄæIJĽæŮűåĂŹäijŽä¡£çŤÍåĹřå■ŘçÍŃïi

13.13 11.13 åŘŚéĂĄäÿŐæŐěæŤűåď ğåđŃæŢřçżĎ

éŮőécŸ

ä¡ăèęĄéĂŽè£Ğç¡ŚçżIJè£đæŐěåŘŚéĂĄåŠŇæŐěåŘŮè£đçż■æŢřæ■őçŽĎåď ğåđŃæŢřçżĎïijŇåźűåř¡éĞŘ

èğčåĘşæŰźæąĹ

äÿŃéİćçŽĎåĞ¡æŢřåĹľçŤÍ `memoryviews` æİěåŘŚéĂĄåŠŇæŐěåŘŮåď ğæŢřçżĎïijŽ

```
# zerocopy.py

def send_from(arr, dest):
    view = memoryview(arr).cast('B')
    while len(view):
        nsent = dest.send(view)
        view = view[nsent:]

def recv_into(arr, source):
    view = memoryview(arr).cast('B')
    while len(view):
        nrecv = source.recv_into(view)
        view = view[nrecv:]
```

äÿžäžĘæţŃèřŢçÍŃåžŘïijŇéęŮåĚĹåĹŻåżžäÿĂäÿłéĂŽè£Ğsocketè£đæŐěçŽĎæIJ■åŁąåŹĺåŠŇåőćæĹůçńřç

```
>>> from socket import *
>>> s = socket(AF_INET, SOCK_STREAM)
>>> s.bind(('', 25000))
>>> s.listen(1)
>>> c,a = s.accept()
>>>
```

åIJĺåőćæĹůçńřïijĹåŘęåď ŰäÿĂäÿłèğčéĞŁåŹĺäÿ■ïijĽïijŽ

```
>>> from socket import *
>>> c = socket(AF_INET, SOCK_STREAM)
>>> c.connect(('localhost', 25000))
>>>
```

æIJňèŁĆçŽĎçŻőæăĞæŸŕä¡ăèČ¡éĂŽè£Ğè£đæŐěäijăè¿ŞäÿĂäÿłèűĚåď ğæŢřçżĎãĂĆè£Źçğ■æČĚåĘţçŽĎ
`array` æĺąåİŮæĹŰ `numpy` æĺąåİŮæİěåĹŻåżžæŢřçżĎïijŽ

```
# Server
>>> import numpy
>>> a = numpy.arange(0.0, 50000000.0)
>>> send_from(a, c)
>>>

# Client
>>> import numpy
>>> a = numpy.zeros(shape=50000000, dtype=float)
>>> a[0:10]
array([ 0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.])
>>> recv_into(a, c)
>>> a[0:10]
array([ 0.,  1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9.])
>>>
```

èőĺèőž

åIJÍæŢřæ■őåŕĘéŻĘåđŃåĹĘåÿČåijŔèőąçőŮåŠŇåźşèąŇèőąçőŮçÍŃåžŔäÿ■ïijŇèĞłåůśåĘŹçÍŃåžŔæİěåőđç
äÿ■è£ĞïijŇèęĄæŸŕä¡ăçąőåőđæČşè£Źæăůå ĄŽïijŇä¡ăåŔŕèČ¡éIJĂèęĄåřĘä¡ăçŽĎæŢřæ■őè¡ňæ■ćæĹŘåŐ§åğŃ
ä¡ăåŔŕèČ¡è£ŸéIJĂèęĄåřĘæŢřæ■őåĹĞåL'šæĹŘåď'ŽäÿłåİŮïijŇåŽääÿžåď'ğéČĺåĹĘåŠŇç¡ŚçżIJçŻÿåĚşçŽĎåĞ¡

äÿĂçğ■æŰźæşŢæŸŕä¡£çŤĺæ§Řçğ■æIJžåĹűåžŔåĹŮåŇŰæŢřæ■őâĂŤâĂŤåŔŕèČ¡åřĘåĚúè¡ňæ■ćæĹŘäÿĂ
äÿ■è£ĞïijŇè£ŹæăůæIJĂçżĹäijŽåĹŻåżžæŢřæ■őçŽĎäÿĂäÿłåď'■åĹűãĂĆ
åřśçőŮä¡ăåŔłæŸŕéŻűçćŐçŽĎåĄŽè£ŹäžŻïijŇä¡ăçŽĎäżčçăĄæIJĂçżĹè£ŸæŸŕäijŽæIJL'åď'ğéĞŔçŽĎåřŔåđŃ

æIJňèŁĆéĂŽè£Ğä¡£çŤĺåĘĚå■ŸèğĘåŻ¿åśŢçď'žäžĘäÿĂäžŻé■ŤæşŢæŞ■ä¡IJãĂĆ
æIJňèť'ĺäÿŁïijŇäÿĂäÿłåĘĚå■ŸèğĘåŻ¿åřśæŸŕäÿĂäÿłåůšå■ŸåIJÍæŢřçżĎçŽĎèęĘçŻŰåśĆãĂĆäÿ■äżĚäżĚæŸŕé
åĘĚå■ŸèğĘåŻ¿è£ŸèČ¡äżěäÿ■åŘŇçŽĎæŰźåijŔè¡ňæ■ćæĹŘäÿ■åŘŇçśżåđŃæIěèąĺçŐřæŢřæ■őãĂĆ
è£ŹäÿłåřśæŸŕäÿŇéĺćè£Źäÿłèŕ■åŘěçŽĎçŻőçŽĎïijŽ

```
view = memoryview(arr).cast('B')
```

åőČæŐěåŘŮäÿĂäÿłæŢřçżĎ arråźűåřĘåĚúè¡ňæ■ćäÿžäÿĂäÿłæŮäçňęåŘůå■ŮèŁĆçŽĎåĘĚå■ŸèğĘåŻ¿ãĂĆ
æŕŤåęĆ `socket.send()` æĹŰ `send.recv_into()` ãĂĆ
åIJĺåĘĚéČĺïijŇè£ŹäżŻæŰźæşŢèČ¡åď'§çŻť'æŐěæŞ■ä¡IJè£ŹäÿłåĘĚå■ŸåŇžå§§ãĂĆä¿ŃåęĆïijŇ`sock.send()` çŻť'æŐěäżŐåĘĚå■Ÿäÿ■åŘŚçŤ§æŢřæ■őèĂŇäÿ■éIJĂèęĄåď'■åĹűãĂĆ `send.recv_into()` ä¡£çŤĺè£ŹäÿłåĘĚå■ŸåŇžå§§ä¡IJäÿžæŐěåŘŮæŞ■ä¡IJçŽĎè¿ŞåĚěçijŞåĘšåŇžãĂĆ

åL'l'äÿŃçŽĎäÿĂäÿłéŽ¿çĆźåřśæŸŕsocketåĞ¡æŢřåŔŕèČ¡åŔłæŞ■ä¡IJéČĺåĹĘæŢřæ■őãĂĆ
éĂŽåÿÿæIěèőšïijŇæĹŚäżňå¿Ůä¡£çŤĺå¿Ĺåď'Žäÿ■åŘŇçŽĎ `send()` åŠŇ `recv_into()`

æİĕäijäè¿ŞæŢť äÿłæŢřçżĎãĂĆ äÿ■çŤĺæŃĚå£ČïijŇæŕŘæňąæŞ■ä¡IJåŘŐïijŇèğĘåŻ¿äijŽéĂŽè£ĞåŘŚéĂĄæĹŰ
æŰřçŽĎèğĘåŻ¿åŘŇæăůäź§æŸŕåĘĚå■ŸèęĘçŻŰåśĆãĂĆåŽäæ■ď ïijŇè£ŸæŸŕæšąæIJĽäżżä¡ŢçŽĎåď■åĹűæŞ

è£ŹéĞŇæIJĽäÿłéŮőéćŸåřśæŸŕæŐěåŘŮèĂĚå£ĚéążäžŇåĚĹç§ěéĄŞæIJĽåď ŽåřŚæŢřæ■őèęĄèćňåŘŚéĂ
äżěä¿£åőČèČ¡éćĎåĹĘéĚ■äÿĂäÿłæŢřçżĎæĹŰèĂĚçąőä£ÍåőČèČ¡åřĘæŐěåŘŮçŽĎæŢřæ■őæŤ¿åĚěäÿĂäÿłåůš
åęĆæđIJæšąåŁđæşŢç§ěéĄŞçŽĎèŕÍïijŇåŘŚéĂĄèĂĚåřśå¿ŮåĚĹåřĘæŢřæ■őåď ğåřŘåŘŚéĂĄè£ĞæÍěïijŇçĎűåł

14 çňňå■ĄäžŇçńäïijŽåźűåŘŚçijŰçĺŃ

åŕźäžŐåźűåŘŚçijŰçĺŃ, PythonæIJĽåď Žçğ■éŢ£æIJ§æŤŕæŇĄçŽĎæŰźæşŢ, åŇĚæŃňåď Žçž£çĺŃ, èřČçŤĺå■Řè£ŻçĺŃ, äžěåŘŁåŘĎçğ■åŘĎæăůçŽĎåĚşäžŐçŤ§æĹŘåŹĺåĞ¡æŢřçŽĎæŁĂåů
è£ŹäÿĂçńäåřĘäijŽçżŹåĞžåźűåŘŚçijŰçĺŃåŘĎçğ■æŰźéÍćçŽĎæŁĂåůğ,
åŇĚæŃňéĂŽçŤĺçŽĎåď Žçž£çĺŃæŁĂæIJŕäžěåŘŁåźűèąŇèőąçőŮçŽĎåőđçŐřæŰźæşŢ.

åČŘçżŘéłŇäÿřåŕŇçŽĎçĺŃåžŘåŚŸæĽĂç§ěéĄŞçŽĎéĆčæăů,
åď ğåőűæŃĚå£ČåźűåŘŚçŽĎçĺŃåžŘæIJĽæ¡IJåIJĺçŽĎå■śéŹľ. åŻäæ■ď,
æIJňçńäçŽĎäÿżèęĄçŻőæăĞäźŃäÿĂæŸŕçżŹåĞžæŻť åŁååŘŕä£ąèţŰåŠŇæŸŞèřČèŕŢçŽĎäżčçăĄ.

Contents:

14.1 12.1 åŘŕåŁĺäÿŐåĄIJæ■ćçž£çĺŃ

éŮőéćŸ

ä¡äèęĄäÿžéIJĂèęĄåźűåŘŚæĽğèąŇçŽĎäżčçăĄåĹŻåżž/éŤĂæŕĄçž£çĺŃ

èğčåĘşæŰźæąĹ

threading åžŞåŘŕäżěåIJĺå■ŢçŃňçŽĎçž£çĺŃäÿ■æĽğèąŇäżżä¡ŢçŽĎåIJĺ Python äÿ■åŘŕäżěèřČçŤĺçŽĎåŕźèśąãĂĆä¡ååŘŕäżěåĹŻåżžäÿĂäÿł Thread åŕźèśąåźűåřĘä¡äèęĄæĽğèąŇçŽĎåŕźèśąäžě target åŘĆæŢřçŽĎå¡ćåijŘæŘŘä¿ŻçżŹèřěåŕźèśąãĂĆ äÿŃéÍćæŸŕäÿĂäÿłçőĂå■ŢçŽĎä¿Ňå■ŘïijŽ

```
# Code to execute in an independent thread
import time
def countdown(n):
    while n > 0:
        print('T-minus', n)
        n -= 1
        time.sleep(5)

# Create and launch a thread
from threading import Thread
t = Thread(target=countdown, args=(10,))
t.start()
```

å¡Şä¡ăåĹŻåżžåě¡äÿĂäÿłçž£çÍŃåŕźèśąåŘŐïijŇèřěåŕźèśąåźűäÿ■äijŽçńŃå■şæL'ğèąŇïijŇéŹď'éİđä¡ăèřČçŤĺä
`start()` æŰźæşŢïijĹå¡Şä¡ăèřČçŤĺ `start()` æŰźæşŢæŮűïijŇåőČäijŽèřČçŤĺä¡ăäijăéĂŠè£ŻæİěçŽĎåĞ¡æŢ
POSIX çž£çÍŃæĹŰèĂĚäÿĂäÿł Windows çž£çÍŃïijL'ïijŇè£ŹäžŻçž£çÍŃåřĘçŤśæŞ■ä¡IJçşżçż§æİěåĚÍæİČçőąçl

```
if t.is_alive():
    print('Still running')
else:
    print('Completed')
```

ä¡ăäź§åŔŕäżěåřĘäÿĂäÿłçž£çÍŃåŁăåĚěåĹřå¡ŞåL'■çž£çÍŃïijŇåźűç■L'å¿ĚåőČçżĹæ■ćïijŽ

```
t.join()
```

PythonèğčéĞŁåŹÍçŻť'åĹřæL'ĂæIJL'çž£çÍŃéČ¡çżĹæ■ćåL'■äż■ä£ÍæŇĄè£ŘèąŇãĂĆåŕźäžŐéIJĂèęĄéŢ£æ
ä¿ŃåęĆïijŽ

```
t = Thread(target=countdown, args=(10,), daemon=True)
t.start()
```

åŘŐåŘřçž£çÍŃæŮăæşŢç■L'å¿ĚïijŇäÿ■è£ĞïijŇè£ŹäžŻçž£çÍŃäijŽåIJÍäÿżçž£çÍŃçżĹæ■ćæŮűèĞłåŁéŤĂa
éŹď'äžĘåęĆäÿŁæL'Ăçď'žçŽĎäÿď'äÿłæŞ■ä¡IJïijŇåźűæšąæIJL'åď'łåď'ŽåŘŕäżěåŕźçž£çÍŃåĄŽçŽĎäžŃæČĚãĂĆ

```
class CountdownTask:
    def __init__(self):
        self._running = True

    def terminate(self):
        self._running = False

    def run(self, n):
        while self._running and n > 0:
            print('T-minus', n)
            n -= 1
            time.sleep(5)

c = CountdownTask()
t = Thread(target=c.run, args=(10,))
t.start()
c.terminate() # Signal termination
t.join()      # Wait for actual termination (if needed)
```

åęĆæđIJçž£çÍŃæL'ğèąŇäÿĂäžŻåČŘI/Oè£ŹæăůçŽĎéŸżåąđæŞ■ä¡IJïijŇéĆčäźĹéĂŽè£Ğè¡ŏèŕćæİěçżĹæ■
ä¿Ňå■ŘåęĆäÿŃïijŽ

```
class IOTask:
    def terminate(self):
        self._running = False

    def run(self, sock):
        # sock is a socket
        sock.settimeout(5)         # Set timeout period
```

(continues on next page)

(continued from previous page)

```
        while self._running:
            # Perform a blocking I/O operation w/ timeout
            try:
                data = sock.recv(8192)
                break
            except socket.timeout:
                continue
            # Continued processing
            ...
        # Terminated
        return
```

èőÍèőž

çŤśäžŐåĚÍåśĂèğčéĞŁéŤĄïijĹGILïijĽçŽĎåŐ§åŻăïijŇPython çŽĎçž£çÍŃèćnéŹŘåĹűåĹřåŘŇäÿĂæŮűåĹżåŘłåĚĄèőÿäÿĂäÿłçž£çÍŃæĽğèąŇè£ŹæăůäÿĂäÿłæĽğèąŇæÍąåđŃ çŽĎçž£çÍŃæŻť'éĂĆçŤÍäžŐåď'ĎçŘĘI/OåŠŇåĚűäżŰéIJĂèęĄåźűåŘŚæĽğèąŇçŽĎéŸżåąđæŞ■ä¡IJïijĹærŤåęĆ

æIJĽæŮűä¡ääijŽçIJŃåĹřäÿŃè¿źè£Źçğ■éĂŽè£ĞçżğæĽ£ `Thread` çśżæİěåőđçŐřçŽĎçž£çÍŃïijŽ

```
from threading import Thread

class CountdownThread(Thread):
    def __init__(self, n):
        super().__init__()
        self.n = n
    def run(self):
        while self.n > 0:

            print('T-minus', self.n)
            self.n -= 1
            time.sleep(5)

c = CountdownThread(5)
c.start()
```

ář¡çőąè£ŹæăůäźŞåŘŕäżěåůěä¡IJïijŇä¡Ęè£Źä¡£å¿Ůä¡äçŽĎäżčçăĄä¿İèţŰäžŐ `threading` åžŞïijŇæĽĂäżěä¡äçŽĎè£ŹäžŻäżčçăĄåŘłèČ¡åIJÍçž£çÍŃäÿŁäÿŃæŰĞäÿ■ä¡£çŤÍãĂĆäÿŁæŰĞæ `threading` åžŞæŮäåĚşçŽĎïijŇè£ŹæăůåřŚä¡£å¿Ůè£ŹäžŻäżčçăĄåŘŕäżěèćńçŤÍåIJÍåĚűäżŰçŽĎäÿŁäÿŃæŰĞ `multiprocessing` æÍąåİŮåIJÍäÿĂäÿłå■ŢçŃňçŽĎè£ŻçÍŃäÿ■æĽğèąŇä¡äçŽĎäżčçăĄïijŽ

```
import multiprocessing
c = CountdownTask(5)
p = multiprocessing.Process(target=c.run)
p.start()
```

åĘ■æňąéĞ■çŤşïijŇè£ŹæőţäżčçăĄäżĚéĂĆçŤÍäžŐ CountdownTask

çśżæŸŕäżěçŃňçńŃäžŐåőđéŹĚçŽĎåźůåŘŚæĽŃæőţïijĹåď'Žçž£çĺŃãĂĄåď'Žè£ŻçĺŃç■Ľ'ç■Ľ'ïijĽ'åőđçŐřçŽĎæ

14.2 12.2 åĹď'æŰ■çž£çĺŃæŸŕåŘęåůšçżŘåŘŕåŁĺ

éŮőécŸ

ä¡ăåůšçżŘåŘŕåŁĺäžĘäÿĂäÿłçž£çĺŃïijŃä¡ĘæŸŕä¡ăæČşç§ěéĄŞåőČæŸŕäÿ■æŸŕçIJ§çŽĎåůšçżŘåijĂåğŃè£

èğčåĘşæŰźæąĹ

çž£çĺŃçŽĎäÿĂäÿłåĚşéŤőçĽźæĂğæŸŕæŕŘäÿłçž£çĺŃéČ¡æŸŕçŃňçńŃè£ŘèąŃäÿŤçŁűæĂĄäÿ■åŘŕéćĎæţ
threading åžŞäÿ■çŽĎ Event åŕźèśąãĂĆ Event åŕźèśąåŃĚåŘńäÿĂäÿłåŘŕçŤśçž£çĺŃèő¿ç¡őçŽĎä£ąåŘůæ
åŕźèśąäÿ■çŽĎä£ąåŘůæăĞå£Ůèćńèő¿ç¡őäÿžåĄĞãĂĆåęĆæđIJæIJĽçž£çĺŃç■Ľ'å¿ĚäÿĂäÿł
event åŕźèśąïijŃèĂŃè£Źäÿł event åŕźèśąçŽĎæăĞå£ŮäÿžåĄĞïijŃéĆčäźĹè£Źäÿłçž£çĺŃåřĘäijŽèćńäÿĂçŻť'éŸ
event åŕźèśąçŽĎä£ąåŘůæăĞå£Ůèő¿ç¡őäÿžçIJ§ïijŃåőČåřĘåŤď'éĘŞæĽ'ĂæIJĽç■Ľ'å¿Ěè£Źäÿł
event åŕźèśąçŽĎçž£çĺŃãĂĆåęĆæđIJäÿĂäÿłçž£çĺŃç■Ľ'å¿ĚäÿĂäÿłåůšçżŘèćńèő¿ç¡őäÿžçIJ§çŽĎ
event åŕźèśąïijŃéĆčäźĹåőČåřĘå£¡çŢěè£ŹäÿłäžŃäżűïijŃçżğçż■æĽ'ğèąŃãĂĆ
äÿŃè¿źçŽĎäżčçăĄåśŢçď'žäžĘåęĆä¡Ţä¡£çŤĺ Event æĺěå■ŘèřČçž£çĺŃçŽĎåŘŕåŁĺïijŽ

```
from threading import Thread, Event
import time

# Code to execute in an independent thread
def countdown(n, started_evt):
    print('countdown starting')
    started_evt.set()
    while n > 0:
        print('T-minus', n)
        n -= 1
        time.sleep(5)

# Create the event object that will be used to signal startup
started_evt = Event()

# Launch the thread and pass the startup event
print('Launching countdown')
t = Thread(target=countdown, args=(10,started_evt))
t.start()

# Wait for the thread to start
started_evt.wait()
print('countdown is running')
```

å¡Şä¡æĽ'ğèąŃè£ŹæőţäżčçăĄïijŃâĂIJcountdown is runningâĂİ æĂżæŸŕæŸ¿çď'žåIJĺ
âĂIJcountdown startingâĂİ äźŃåŘŐæŸ¿çď'žãĂĆè£ŹæŸŕçŤśäžŐä¡£çŤĺ
event æĺěå■ŘèřČçž£çĺŃïijŃä¡£å¿Ůäÿżçž£çĺŃèęĄç■Ľ'åĹř countdown()
åĞ¡æŢřè¿ŞåĞžåŘŕåŁĺä£ąæĄŕåŘŐïijŃæĽ'■èČ¡çżğçż■æĽ'ğèąŃãĂĆ

èőÍèőž

event åŕźèśąæIJĂåě¡å■Ţæňąä¡£çŤÍïijŇåŕśæŸŕèŕť'ïijŇä¡ăåĹŻåżžäÿĂäÿł event åŕźèśąïijŇèőľæ§Řäÿłçž£çÍŃç■L'å¿Ěè£ŹäÿłåŕźèśąïijŇäÿĂæŮęè£Źäÿłåŕźèśąèćńèő¿ç¡őäÿžçIJ§ïijŇä¡ăåŕśåžŤèŕè clear() æŰźæşŢæĺěéĞ■ç¡ő event åŕźèśąïijŇä¡ĘæŸŕå¿ĹéŽ¿çąőä£ĺåőL'åĚĺåIJřæÿĚçŘĘ event åŕźèśąåźűåŕźåőČéĞ■æŰřèţŃåĂijãĂĆå¿ĹåŘŕèČ¡äijŽåŘŚçŤ§éŤŹè£ĞäžŃäżűãĂĄæ■żéŤĄæĹŰèĂĚåĚűä event åŕźèśąçŽĎäżčçăĄäijŽåIJĺçž£çÍŃåĘ■æňąç■L'å¿Ěè£ŹäÿŁ event åŕźèśąäźŃåL'■æL'ğèąŇïijL'ãĂĆåęĆæđIJäÿĂäÿłçž£çÍŃéIJĂèęĄäÿ■åĄIJåIJřéĞ■åď■ä¡£çŤÍ event åŕźèśąïijŇä¡ăæIJĂåě¡ä¡£çŤÍ Condition åŕźèśąæĺěäżčæŻ£ãĂĆäÿŃéĺćçŽĎäżčçăĄä¡£çŤÍ Condition åŕźèśąåőđçŐřäžĘäÿĂäÿłåŚÍæIJ§åőŽæŮűåŹÍïijŇæŕŘå¡ŞåőŽæŮűåŹÍèűĚæŮűçŽĎæŮűåĂŹïijŇåĺ

```
import threading
import time

class PeriodicTimer:
    def __init__(self, interval):
        self._interval = interval
        self._flag = 0
        self._cv = threading.Condition()

    def start(self):
        t = threading.Thread(target=self.run)
        t.daemon = True

        t.start()

    def run(self):
        '''
        Run the timer and notify waiting threads after each interval
        '''
        while True:
            time.sleep(self._interval)
            with self._cv:
                 self._flag ^= 1
                 self._cv.notify_all()

    def wait_for_tick(self):
        '''
        Wait for the next tick of the timer
        '''
        with self._cv:
            last_flag = self._flag
            while last_flag == self._flag:
                self._cv.wait()

# Example use of the timer
ptimer = PeriodicTimer(5)
ptimer.start()

# Two threads that synchronize on the timer
```

(continues on next page)

(continued from previous page)

```
def countdown(nticks):
    while nticks > 0:
        ptimer.wait_for_tick()
        print('T-minus', nticks)
        nticks -= 1

def countup(last):
    n = 0
    while n < last:
        ptimer.wait_for_tick()
        print('Counting', n)
        n += 1

threading.Thread(target=countdown, args=(10,)).start()
threading.Thread(target=countup, args=(5,)).start()
```

eventåŕźèśąçŽĎäÿĂäÿłéĞ■èęĄçĽźçĆźæŸŕå¦ŞåőČèćńèő¿ç¦őäÿžçIJ§æŮűäijŽåŤď'éĘŠæĽ'ĂæIJĽ'ç■Ľ'å¿Ě
`Condition` åŕźèśąæİěæŻ£äżčãĂĆèĂČèŹŚäÿĂäÿŃè£Źæőţä¦£çŤĺä£ąåŘůéĞŘåőđçŐřçŽĎäżčçăĄïijŽ

```
# Worker thread
def worker(n, sema):
    # Wait to be signaled
    sema.acquire()

    # Do some work
    print('Working', n)

# Create some threads
sema = threading.Semaphore(0)
nworkers = 10
for n in range(nworkers):
    t = threading.Thread(target=worker, args=(n, sema,))
    t.start()
```

è£ŘèąŇäÿŁè¿źçŽĎäżčçăĄåřĘäijŽåŘŕåŁĺäÿĂäÿłçž£çĺŃæśäïijŇä¦ĘæŸŕåźűæšąæIJĽäżĂäźĹäžŃæČĚåŔŚ

```
>>> sema.release()
Working 0
>>> sema.release()
Working 1
>>>
```

çijŰåĘŹæűĽ'åŔŁåĹřåď'ğéĞŘçŽĎçž£çĺŃéŮť'åŘŇæ■ěéŮőéćŸçŽĎäżčçăĄäijŽèőľä¦ăçŮŻäÿ■æňšçŤ§ãĂĆ

14.3 12.3 çž£çÍŃéŮťéĂŽä£ą

éŮőéćŸ

ä¡äçŽĎçÍŃåžŘäÿ■æIJL'åď'Žäÿłçž£çÍŃïijŇä¡äéIJĂèęĄåIJÍè£ŹäžŻçž£çÍŃäźŃéŮť'åőL'åĚÍåIJřäžď'æ■ćä£ąa

èğčåĘşæŰźæąĹ

äżŐäÿĂäÿłçž£çÍŃåŘŚåŘęäÿĂäÿłçž£çÍŃåŘŚéĂĄæŢřæ■őæIJĂåőL'åĚÍçŽĎæŰźåijŘåŘŕèČ¡åřśæŸŕä¡£çŤÍ queue åžŞäÿ■çŽĎéŸ§åĹŮäžĘãĂĆåĹŻåżžäÿĂäÿłèćńåď'Žäÿłçž£çÍŃåĚśäžńçŽĎ Queue åŕźèśąïijŇè£ŹäžŻçž£çÍŃéĂŽè£Ğä¡£çŤÍ put() åŠŇ get() æŞ■ä¡IJæİěåŘŚéŸ§åĹŮäÿ■æůżåŁăæĹŰèĂĚåĹăéŹď'åĚČçť'ăăĂĆ ä¿ŃåęĆïijŽ

```
from queue import Queue
from threading import Thread

# A thread that produces data
def producer(out_q):
    while True:
        # Produce some data
        ...
        out_q.put(data)

# A thread that consumes data
def consumer(in_q):
    while True:
# Get some data
        data = in_q.get()
        # Process the data
        ...

# Create the shared queue and launch both threads
q = Queue()
t1 = Thread(target=consumer, args=(q,))
t2 = Thread(target=producer, args=(q,))
t1.start()
t2.start()
```

Queue åŕźèśąåůšçżŘåŇĚåŘńäžĘå£ĚèęĄçŽĎéŤĄïijŇæL'Ääžěä¡ääŘŕäžěéĂŽè£ĞäőČåIJÍåď'Žäÿłçž£çÍŃé å¡Şä¡£çŤÍéŸ§åĹŮæŮűïijŇå■ŘèřČçŤ§äžğèĂĚåŠŇæűĹèť'źèĂĚçŽĎåĚşéŮ■éŮőéćŸåŘŕèČ¡äijŽæIJL'äÿĂäžŻé

```
from queue import Queue
from threading import Thread

# Object that signals shutdown
_sentinel = object()

# A thread that produces data
```

(continues on next page)

(continued from previous page)

```
def producer(out_q):
    while running:
        # Produce some data
        ...
        out_q.put(data)

    # Put the sentinel on the queue to indicate completion
    out_q.put(_sentinel)

# A thread that consumes data
def consumer(in_q):
    while True:
        # Get some data
        data = in_q.get()

        # Check for termination
        if data is _sentinel:
            in_q.put(_sentinel)
            break

        # Process the data
        ...
```

æIJňä¿Ńäÿ■æIJL'äÿĂäÿłçL'źæőŁçŽĎåIJřæŰźïijŽæűĹèt'źèĂĚåIJĺèŕżåĹřè£Źäÿłç L'źæőŁåĂijäźŃåŘÓçńŃ
åř¡çőąéŸ§åĹŮæŸŕæIJĂåÿÿèğĄçŽĎçž£çĺŃéŮt'éĂŽä£ąæIJžåĹűïijŃä¡ĘæŸŕäż■çĎűåŘŕäżěèĞłåůśéĂŽè£ĞåĹŻ
`Condition` åŘŸéĞŘæĬěåŇĚèčĚä¡ăçŽĎæŢřæ■őçżŞæđĎãĂĆäÿŃè¿źè£Źäÿłä¿Ńå■ŘæijŤçď'žäžĘåęĆä¡ŢåĹŻ

```
import heapq
import threading

class PriorityQueue:
    def __init__(self):
        self._queue = []
        self._count = 0
        self._cv = threading.Condition()
    def put(self, item, priority):
        with self._cv:
            heapq.heappush(self._queue, (-priority, self._count,␣
↪item))
            self._count += 1
            self._cv.notify()

    def get(self):
        with self._cv:
            while len(self._queue) == 0:
                self._cv.wait()
            return heapq.heappop(self._queue)[-1]
```

ä¡£çŤĺéŸ§åĹŮæĬěè£ŻèąŇçž£çĺŃéŮt'éĂŽä£ąæŸŕäÿĂäÿłå■ŢåŘŚãĂĄäÿ■çąőåőŽçŽĎè£ĞçĺŃãĂĆéĂŽåÿ
`task_done()` åŠŇ `join()` ïijŽ

```
from queue import Queue
from threading import Thread

# A thread that produces data
def producer(out_q):
    while running:
        # Produce some data
        ...
        out_q.put(data)

# A thread that consumes data
def consumer(in_q):
    while True:
        # Get some data
        data = in_q.get()

        # Process the data
        ...
        # Indicate completion
        in_q.task_done()

# Create the shared queue and launch both threads
q = Queue()
t1 = Thread(target=consumer, args=(q,))
t2 = Thread(target=producer, args=(q,))
t1.start()
t2.start()

# Wait for all produced items to be consumed
q.join()
```

åęĆæđIJäÿĂäÿłçž£çÍŃéIJĂèęĄåIJÍäÿĂäÿłâĂIJæűĹèť źèĂĚâĂİçž£çÍŃåď ĎçŘĘåőŇçĽźåőŽçŽĎæŢřæ■ő
`Event` æŤ¿åĹřäÿĂèţůä¡£çŤÍïijŇè£ŹæăůâĂIJçŤ§äžğèĂĚâĂİåřśåŘŕäżěéĂŽè£Ğè£ŹäÿłEventåŕźèśąæİěçŻŚæţl

```
from queue import Queue
from threading import Thread, Event

# A thread that produces data
def producer(out_q):
    while running:
        # Produce some data
        ...
        # Make an (data, event) pair and hand it to the consumer
        evt = Event()
        out_q.put((data, evt))
        ...
        # Wait for the consumer to process the item
        evt.wait()

# A thread that consumes data
```

(continues on next page)

(continued from previous page)

```
def consumer(in_q):
    while True:
        # Get some data
        data, evt = in_q.get()
        # Process the data
        ...
        # Indicate completion
        evt.set()
```

èőÍèőž

å§žäžŐçőĂå■ŢéŸ§åĹŮçijŰåĘŹåď ŽçžÍçÍÑçÍÑåžŘåIJÍåď ŽæŢřæČĚåĘţäÿÑæŸŕäÿĂäÿłæŕŤè¿ČæŸŐæŹż
ä¡£çŤÍçž£çÍÑéŸ§åĹŮæIJL'äÿĂäÿłèęĄæşÍæĎŘçŽĎéŮőéćŸæŸŕïijÑåŘŚéŸ§åĹŮäÿ■æůžåŁäæŢřæ■őéąźæŮűů

```
from queue import Queue
from threading import Thread
import copy

# A thread that produces data
def producer(out_q):
    while True:
        # Produce some data
        ...
        out_q.put(copy.deepcopy(data))

# A thread that consumes data
def consumer(in_q):
    while True:
        # Get some data
        data = in_q.get()
        # Process the data
        ...
```

Queue åŕźèśąæŘŘä¿ŻäÿĂäžŻåIJÍå¡ŞåL'■äÿŁäÿÑæŰĞå¿ĹæIJL'çŤÍçŽĎéŹĎåŁäçL'źæĂğãĂĆæŕŤåęĆåIJ
Queue åŕźèśąæŮűæŘŘä¿ŻåŘŕéĂL'çŽĎ size åŘĆæŢřæİěéŹŘåĹűåŘŕäžěæůżåŁäåĹřéŸ§åĹŮäÿ■çŽĎåĚĆçť ä
âĂIJæűĹèť źâĂİçŽĎéĂ§åžęå£ńïijÑéĆčäźĹä¡£çŤÍåŻžåőŽåď ğåřŘçŽĎéŸ§åĹŮåřśåŘŕäžěåIJÍéŸ§åĹŮåůšæżąçŽ
get() åŠŇ put() æŰźæşŢéČ¡æŤŕæŇĄéİđéŸżåąđæŰźåijŘåŠŇèő¿åőŽèűĚæŮűïijÑä¿ŃåęĆïijŽ

```
import queue
q = queue.Queue()

try:
    data = q.get(block=False)
except queue.Empty:
    ...

try:
```

(continues on next page)

(continued from previous page)

```
    q.put(item, block=False)
except queue.Full:
    ...

try:
    data = q.get(timeout=5.0)
except queue.Empty:
    ...
```

è£ŹäžŻæŞ■ä¡IJéČ¡åŘŕäżěçŤÍæİěéĄ£åĚ■å¡ŞæĽ'ğèąŇæ§ŘäžŻçĽ'źåőŽéŸ§åĹŮæŞ■ä¡IJæŮűåŘŚçŤ§æŮăé
put() æŰźæşŢåŠŇäÿĂäÿłåŻžåőŽåď'ğåřŘçŽĎéŸ§åĹŮäÿĂèţůä¡£çŤÍïijŇè£Źæăůå¡ŞéŸ§åĹŮåůšæżąæŮűåřšá

```
def producer(q):
    ...
    try:
        q.put(item, block=False)
    except queue.Full:
        log.warning('queued item %r discarded!', item)
```

åęĆæđIJä¡ăèŕŢåŻ¿èőľ'æűĹèť'źèĂĚçž£çÍŃåIJÍæĽ'ğèąŇåČŘ q.get() è£ŹæăůçŽĎæŞ■ä¡IJæŮűïijŇèűĚæŮűèĞłåŁÍçżĹæ■ćäžěä¿£æčĂæ§ěçżĹæ■ćæăĞå£ŮïijŇä¡ăåžŤèŕěä¡£çŤÍ q.get() çŽĎåŘŕéĂĽ'åŘĆæŢř timeout ïijŇåęĆäÿŃïijŽ

```
_running = True

def consumer(q):
    while _running:
        try:
            item = q.get(timeout=5.0)
            # Process item
            ...
        except queue.Empty:
            pass
```

æIJĂåŘŐïijŇæIJĽ q.qsize() ïijŇ q.full() ïijŇ q.empty() ç■Ľ'åőđçŤÍæŰźæşŢåŘŕäżěèŐůåŘŰäÿĂäÿłéŸ§åĹŮçŽĎå¡ŞåĽ'■åď'ğåřŘåŠŇçŁűæĂĄãĂĆä¡ĘèęĄæşÍæĎŘïijŇè empty() åĹď'æŰ■åĞžè£ŹäÿłéŸ§åĹŮäÿžçľ'žïijŇä¡ĘåŘŇæŮűåŘęåď'ŰäÿĂäÿłçž£çÍŃåŘŕèČ¡åůšçżŘåŘŚè£Źä

14.4 12.4 çżŹåĚşéŤőéČÍåĹĘåŁăéŤĄ

éŮőécŸ

ä¡ăéIJĂèęĄåŕźåď'Žçž£çÍŃçÍŃåžŘäÿ■çŽĎäÿť'çŢŇåŇžåŁăéŤĄäżěéĄ£åĚ■çńđäžĽ'æİąäżűãĂĆ

èğčåĘşæŰźæąĹ

èęĄåIJĺåď Žçž£çĺŃçĺŃåžŘäÿ■åőĽ'åĚĺä¡£çŤĺåŘŕåŘŸåŕźèśąïijŇä¡ăéIJĂèęĄä¡£çŤĺ threading åžŞäÿ■çŽĎ `Lock` åŕźèśąïijŇåŕśåČŘäÿŃè¿źè£ŹäÿłäĺŃå■Řè£ŹæăůïijŽ

```
import threading

class SharedCounter:
    '''
    A counter object that can be shared by multiple threads.
    '''
    def __init__(self, initial_value = 0):
        self._value = initial_value
        self._value_lock = threading.Lock()

    def incr(self,delta=1):
        '''
        Increment the counter with locking
        '''
        with self._value_lock:
             self._value += delta

    def decr(self,delta=1):
        '''
        Decrement the counter with locking
        '''
        with self._value_lock:
             self._value -= delta
```

`Lock` åŕźèśąåŠŇ `with` èŕ■åŘěåĺŮäÿĂèţůä¡£çŤĺåŘŕäžěä£ĺèŕĄäžŠæŰěæĽ'ğèąŇïijŇåŕśæŸŕæŕŘæňąåŘłæI
with èŕ■åŘěåŇĚåŘńçŽĎäžčçăĄåĺŮãĂĆwith èŕ■åŘěäijŽåIJĺè£ŹäÿłäžčçăĄåĺŮæĽ'ğèąŇåĽ'■èĞłåŁĺèŐůåŘŰéŤ

èőĺèőž

çž£çĺŃèřČåžęæIJňèť'ĺäÿŁæŸŕäÿ■çąőåőŽçŽĎïijŇåŻäæ■ď'ïijŇåIJĺåď Žçž£çĺŃçĺŃåžŘäÿ■éŤŹèŕŕåIJŕä¡£çŤ
åIJĺäÿĂäžŻâĂIJèĂĄçŽĎâĂİ Python äžčçăĄäÿ■ïijŇæŸ¿åijŘèŐůåŘŰåŠŇéĞŁæŤ¿éŤĄæŸŕå¿Łåÿÿèğ ĄçŽĎãĂ

```
import threading

class SharedCounter:
    '''
    A counter object that can be shared by multiple threads.
    '''
    def __init__(self, initial_value = 0):
        self._value = initial_value
        self._value_lock = threading.Lock()

    def incr(self,delta=1):
        '''
```

(continues on next page)

(continued from previous page)

```
        Increment the counter with locking
        '''
        self._value_lock.acquire()
        self._value += delta
        self._value_lock.release()

    def decr(self,delta=1):
        '''
        Decrement the counter with locking
        '''
        self._value_lock.acquire()
        self._value -= delta
        self._value_lock.release()
```

çŻÿæŕŤäžŐè£Źçğ■æŸ¿åijŔèřČçŤĺçŽĎæŰźæşŢïijŇwith èŕ■åŔěæŻť åŁääijŸéŻĚïijŇäź§æŻť äÿ■åőźæŸŞ
release() æŰźæşŢæĹŰèĂĚçĺŃåžŔåIJĺèŐůå¿ŮéŤĄäźŃåŘŐäžğçŤ§åijĆåÿÿè£Źäÿď çğ■æČĚåĘţïijĹä¡£çŤĺ
with èŕ■åŔěåŔŕäžěä£ÌèŕĄåIJĺè£Źäÿď çğ■æČĚåĘţäÿŃäż■èČ¡æ■čçąőéĞŁæŤ¿éŤĄïijĽãĂĆ
äÿžäžĘéĄ£åĚ■åĞžçŐřæ■żéŤĄçŽĎæČĚåĘţïijŇä¡£çŤĺéŤĄæIJžåĹűçŽĎçĺŃåžŔåžŤèŕěèő¿åőŽäÿžæŕŘäÿłçž£çĺ
åIJĺ threading åžŞäÿ■è£ŸæŔŘä¿ŻäžĘåĚűäżŰçŽĎåŘŃæ■ěåŐ§èŕ■ïijŇæŕŤåęĆ RLock
åŠŃ Semaphore åŕźèśąãĂĆä¡ĘæŸŕæäźæ■őäżěå¿ĂçżŘéłŃïijŇè£ŹäžŻåŐ§èŕ■æŸŕçŤĺäžŐäÿĂäžŻçĽźæőŁçŽ
RLock ïijĹåŔŕéĞ■åĚěéŤĄïijĽåŔŕäžěèćńåŘŃäÿĂäÿłçž£çĺŃåď Žæňąè ŐůåŔŰïijŇäÿżèęĄçŤĺæĺěåőđçŐřå§žäžĆ
SharedCounter çśżïijŽ

```
import threading

class SharedCounter:
    '''
    A counter object that can be shared by multiple threads.
    '''
    _lock = threading.RLock()
    def __init__(self, initial_value = 0):
        self._value = initial_value

    def incr(self,delta=1):
        '''
        Increment the counter with locking
        '''
        with SharedCounter._lock:
            self._value += delta

    def decr(self,delta=1):
        '''
        Decrement the counter with locking
        '''
        with SharedCounter._lock:
             self.incr(-delta)
```

åIJĺäÿŁè¿źè£Źäÿłä¿Ńå■Řäÿ■ïijŇæšąæIJĽåŕźæŕŘäÿĂäÿłåőđä¿Ńäÿ■çŽĎåŔŕåŘŸåŕźèśąåŁäéŤĄïijŇåŘŰè
decr æŰźæşŢãĂĆ è£Źçğ■åőđçŐřæŰźåijŘçŽĎäÿĂäÿłçĽźçĆźæŸŕïijŇæŮäèőžè£Źäÿłçśżæ IJĽåď ŽåřŚäÿłåőđä¿

ä£ąåŘůéĞŘåŕźèśąæŸŕäÿĂäÿłåżžçńŇåIJĺåĚśäžńèőąæŢřåŹĺå§žçąĂäÿŁçŽĎåŘŇæ■ěåŐ§èŕ■ãĂĆåęĆæđIJèőąæŢ
èŕ■åŘěåřĘèőąæŢřåŹĺåĞŘ1ïijŇçž£çĺŃèćńåĚĄèőÿæĽğèąŇãĂĆwith
èŕ■åŘěæĽğèąŇçżŞæİ§åŘŐïijŇèőąæŢřåŹĺåŁäïijŚãĂĆåęĆæđIJèőąæŢřåŹĺäÿž0ïijŇçž£çĺŃåřĘèćnéŸżåąđïijŇçŽ

```
from threading import Semaphore
import urllib.request

# At most, five threads allowed to run at once
_fetch_url_sema = Semaphore(5)

def fetch_url(url):
    with _fetch_url_sema:
        return urllib.request.urlopen(url)
```

åęĆæđIJä¡ăåŕźçž£çĺŃåŘŇæ■ěåŐ§èŕ■çŽĎåžŢåśĆçŘĘèőžåŠŇåőđçŐřæĎ§åĚť èűčïijŇåŘŕäżěåŘĆèĂČæŞ

14.5 12.5 éŸšæ■ćæ■żéŤĄçŽĎåŁäéŤĄæIJžåĹű

éŮőéćŸ

ä¡äæ■čåIJĺåĘŹäÿĂäÿłåď Žçž£çĺŃçĺŃåžŘïijŇåĚűäÿ■çž£çĺŃéIJĂèęĄäÿĂæňąèŐůåŘŮåď ŽäÿłéŤĄïijŇæ■d

èğčåĘşæŰźæąĹ

åIJĺåď Žçž£çĺŃçĺŃåžŘäÿ■ïijŇæ■żéŤĄéŮőéćŸå¿Ĺåď ğäÿĂéČĺåĹĘæŸŕçŤśäžŐçž£çĺŃåŘŇæŮűèŐůåŘŮå
æŮűůåĂŹåŘŚçŤ§éŸżåąđïijŇéĆčäźĹè£Źäÿłçž£çĺŃåřśåŘŕèČ¡éŸżåąđåĚűäżŰçž£çĺŃçŽĎæĽğèąŇïijŇäżŐèĂŇ
èğčåĘşæ■żéŤĄéŮőéćŸçŽĎäÿĂçğ■æŰźæąĹæŸŕäÿžçĺŃåžŘäÿ■çŽĎæŕŘäÿĂäÿłéŤĄåĹĘéĚ■äÿĂäÿłåŤŕäÿĂçŽ
æŸŕéİđåÿÿåőźæŸŞåőđçŐřçŽĎïijŇçď žä¿ŃåęĆäÿŃïijŽ

```
import threading
from contextlib import contextmanager

# Thread-local state to stored information on locks already acquired
_local = threading.local()

@contextmanager
def acquire(*locks):
    # Sort locks by object identifier
    locks = sorted(locks, key=lambda x: id(x))

    # Make sure lock order of previously acquired locks is not␣
↪violated
    acquired = getattr(_local,'acquired',[])
    if acquired and max(id(lock) for lock in acquired) >=␣
↪id(locks[0]):
        raise RuntimeError('Lock Order Violation')
```

(continues on next page)

(continued from previous page)

```
    # Acquire all of the locks
    acquired.extend(locks)
    _local.acquired = acquired

    try:
        for lock in locks:
            lock.acquire()
        yield
    finally:
        # Release locks in reverse order of acquisition
        for lock in reversed(locks):
            lock.release()
        del acquired[-len(locks):]
```

åęĆä¡Ţä¡£çŤÍè£ŹäÿłäÿŁäÿŃæŰĞçőąçŘĘåŹÍåŚćïij§ä¡ăåŘŕäżěæŃL'çĚğæ■čåÿÿéĂŤå¿ĎåĹŻåżżäÿĂäÿłéŤ acquire() åĞ¡æŢřæİěçŤşèřůéŤĄïijŃ çď'žä¿ŃåęĆäÿŃïijŽ

```
import threading
x_lock = threading.Lock()
y_lock = threading.Lock()

def thread_1():
    while True:
        with acquire(x_lock, y_lock):
            print('Thread-1')

def thread_2():
    while True:
        with acquire(y_lock, x_lock):
            print('Thread-2')

t1 = threading.Thread(target=thread_1)
t1.daemon = True
t1.start()

t2 = threading.Thread(target=thread_2)
t2.daemon = True
t2.start()
```

åęĆæđIJä¡ăæL'ğèąŃè£Źæőţäżčçă Ąïij Ňä¡ăäijŽåŘŚçŐřåőČå■şä¡£åIJÍäÿ■åŘŃçŽĎåĞ¡æŢřäÿ■äżěäÿ■åŘŃç åĚűåĚşéŤőåIJÍäžŐïijŃåIJÍçňňäÿĂæőţäżčçă Ąäÿ■ïijŃæĹŚäżňářżè£ŹäžŻéŤĄè£ŻèąŃäžĘæŐŠäžŘãĂĆéĂŽè£Ğ åęĆæđIJæIJL'åď'Žäÿł acquire() æŞ■ä¡IJèćňåţŃåěŮèřČçŤÍïijŃåŘŕäżěéĂŽè£Ğçž£çÍŃæIJňåIJřå■ŸåĆÍïijĹT åĄĞèő¿ä¡ăçŽĎäżčçă ĄæŸŕè£ŹæăůåĘŹçŽĎïijŽ

```
import threading
x_lock = threading.Lock()
y_lock = threading.Lock()

def thread_1():
```

(continues on next page)

(continued from previous page)

```
    while True:
        with acquire(x_lock):
            with acquire(y_lock):
                print('Thread-1')

def thread_2():
    while True:
        with acquire(y_lock):
            with acquire(x_lock):
                print('Thread-2')

t1 = threading.Thread(target=thread_1)
t1.daemon = True
t1.start()

t2 = threading.Thread(target=thread_2)
t2.daemon = True
t2.start()
```

åęĆæđIJä¡ăè£ŘèąŇè£ŹäÿłçL'ĹæIJňçŽĎäżčçăĄïijŇå£ĚåőŽäijŽæIJL'äÿĂäÿłçž£çÍŃåŘŚçŤ§åť'ľæžČïijŇåi

```
Exception in thread Thread-1:
Traceback (most recent call last):
  File "/usr/local/lib/python3.3/threading.py", line 639, in _
↪bootstrap_inner
    self.run()
  File "/usr/local/lib/python3.3/threading.py", line 596, in run
    self._target(*self._args, **self._kwargs)
  File "deadlock.py", line 49, in thread_1
    with acquire(y_lock):
  File "/usr/local/lib/python3.3/contextlib.py", line 48, in __
↪enter__
    return next(self.gen)
  File "deadlock.py", line 15, in acquire
    raise RuntimeError("Lock Order Violation")
RuntimeError: Lock Order Violation
>>>
```

åŘŚçŤ§åť'ľæžČçŽĎåŐ§åŻäåIJÍäžŐïijŇæŕŘäÿłçž£çÍŃéČ¡èőřå¡ŢçĺĂèĞłåůśåůšçżŘèŐůåŘŰåĹřçŽĎéŤĄã
`acquire()` åĞ¡æŢřäijŽæčĂæ§ěäźŃåL'■åůšçżŘèŐůåŘŰçŽĎéŤĄåĹŮèąÍïijŇ
çŤśäžŐéŤĄæŸŕæŇL'çĚğå■ĞåžŘæŐŠåĹŮèŐůåŘŰçŽĎïijŇæL'ÄäżěåĞ¡æŢřäijŽèőď'äÿžäźŃåL'■åůšèŐůåŘŰçŽ

èőÍèőž

æ■żéŤĄæŸŕæŕŘäÿĂäÿłåď'Žçž£çÍŃçÍŃåžŘéČ¡äijŽéÍćäÿť'çŽĎäÿĂäÿłéŮőécŸïijĹåŕśåČŘåőČæŸŕæŕŘäÿĂ
çž£çÍŃåŘłèČ¡åŘŇæŮűä£ÍæŇĄäÿĂäÿłéŤĄïijŇè£ŹæăůçÍŃåžŘåřśäÿ■äijŽèćńæ■żéŤĄéŮőécŸæL'ĂåŽřæL'řãĂ

æ■żéŤĄçŽĎæčĂæţŃäÿŐæĄćåď'■æŸŕäÿĂäÿłåĞäźŐæšąæIJL'äijŸéŽĚçŽĎèğčåĘşæŰźæąĹçŽĎæL'ľåśŢ

è£ŘèąŇçŽĎæŮűåĂŹäijŽæŕŘéŽŤäÿĂæőţæŮűéŮť'éĞ■ç¡őèőąæŢřåŹĺïijŇåIJÍæšąæIJL'åŘŚçŤ§æ■żéŤĄçŽĎæČ
èűĚæŮűïijŇè£ŹæŮűçÍŇåžŘäijŽéĂŽè£ĞéĞ■åŘŕèĞłèžńæĄćåď■åĹřæ■čåÿÿçŁűæĂĄãĂĆ

éĄ£åĚ■æ■żéŤĄæŸŕåŘęåď'ŰäÿĂçğ■èğčåĘşæ■żéŤĄéŮőécŸçŽĎæŰźåijŘïijŇåIJÍè£ŻçÍŇèŐůåŘŰéŤĄçŽ
æ■żéŤĄçŁűæĂĄãĂĆèŕĄæŸŐåřśçŢŹçżŹèŕżèĂĚä¡IJäÿžçżČäźäážĘãĂĆéĄ£åĚ■æ■żéŤĄçŽĎäÿżèęĄæĂİæČşa
æ■żéŤĄçŽĎäÿĂäÿłå£ĚèęĄæİąäžűïijŇäżŐèĂŇéĄ£åĚ■çÍŇåžŘè£ŻåĚěæ■żéŤĄçŁűæĂĄãĂĆ

äÿŃéİćäžěäÿĂäÿłåĚşäžŐçž£çÍŇæ■żéŤĄçŽĎçżŘåĚÿéŮőécŸïijŽâĂIJåŞšå■ęåőűåřśéď'ŘéŮőécŸâĂİïijŇä
éİćåĹ'■æIJL'äÿĂçćŮéě■åŠŇäÿĂåŘłç■ůå■ŘãĂĆåIJÍè£ŹéĞŇæŕŘäÿłåŞšå■ęåőűåŘŕäžěçIJŃåĄŽæŸŕäÿĂäÿłçŃ
æĂİèĂČãĂĄåŘČéě■äÿĹ'çğ■çŁűæĂĄäÿ■çŽĎäÿĂäÿłãĂĆéIJĂèęĄæşÍæĎŘçŽĎæŸŕïijŇæŕŘäÿłåŞšå■ęåőűåŘČ
éĆčäźĹäžŰäžňäžŤäÿłéČ¡åŘłèČ¡æŃ£çİĂäÿĂåŘłç■ůå■ŘåİŘåIJÍéĆčåĎ£ïijŇçŻť'åĹřéě£æ■żãĂĆæ■ď'æŮűäžŰä
äÿŃéİćæŸŕäÿĂäÿłçőĂå■ŢçŽĎä¡£çŤÍæ■żéŤĄéĄ£åĚ■æIJžåĹûèğčåĘşâĂIJåŞšå■ęåőűåřśéď'ŘéŮőécŸâĂİçŽĎ

```
import threading

# The philosopher thread
def philosopher(left, right):
    while True:
        with acquire(left,right):
             print(threading.currentThread(), 'eating')

# The chopsticks (represented by locks)
NSTICKS = 5
chopsticks = [threading.Lock() for n in range(NSTICKS)]

# Create all of the philosophers
for n in range(NSTICKS):
    t = threading.Thread(target=philosopher,
                         args=(chopsticks[n],chopsticks[(n+1) %␣
↪NSTICKS]))
    t.start()
```

æIJĂåŘŐïijŇèęĄçL'źåĹńæşÍæĎŘåĹřïijŇäÿžäžĘéĄ£åĚ■æ■żéŤĄïijŇæL'ĂæIJL'çŽĎåŁäéŤĄæŞ■ä¡IJå£Ĕé
`acquire()` åĞ¡æŢřãĂĆåęĆæđIJäżčçăĄäÿ■çŽĎæ§ŘéČĺåĹĘçżŢè£Ğacquire
åĞ¡æŢřçŻť'æŐěçŤşèŕůéŤĄïijŇéĆčäźĹæŢť'äÿłæ■żéŤĄéĄ£åĚ■æIJžåĹůåřśäÿ■ètůä¡IJçŤÍäžĘãĂĆ

14.6 12.6 ä£İå■Ÿçž£çÍŇçŽĎçŁűæĂĄä£ąæĄŕ

éŮőécŸ

ä¡äéIJĂèęĄä£İå■Ÿæ■čåIJÍè£ŘèąŇçž£çÍŇçŽĎçŁűæĂĄïijŇè£ŹäÿłçŁűæĂĄåŕźäžŐåĚüäžŰçŽĎçž£çÍŇæŸ

èğčåĘşæŰźæąĹ

æIJL'æŮűåIJÍåď'Žçž£çÍŇçijŰçÍŇäÿ■ïijŇä¡äéIJĂèęĄåŘłä£İå■Ÿå¡ŞåL'■è£ŘèąŇçž£çÍŇçŽĎçŁűæĂĄãĂĆ
èęĄè£ŹäźĹåĄŽïijŇåŘŕä¡£çŤÍ `thread.local()` åĹŻåžžäÿĂäÿłæIJňåIJřçž£çÍŇå■ŸåĆÍåŕźèśąąãĂĆ
åŕźè£ŹäÿłåŕźèśąçŽĎåśđæĂğçŽĎä£İå■ŸåŠŇèŕżåŘŰæŞ■ä¡IJéČ¡åŘłäijŽåŕźæL'ğèąŇçž£çÍŇåŘŕèğĄïijŇèĂŇåĚ

ä¡IJäÿžä¡£çŤÍæIJňåIJřå■ŸåĆÍçŽĎäÿĂäÿłæIJL'èűčçŽĎåőđéŹĚä¿Ňå■ŘïijŇ èĂČèŹŚåIJÍ8.3åřŘèŁĆåőŽäźL'è£ĞçŽĎ `LazyConnection` äÿŁäÿŇæŰĞçőąçŘĘåŹÍçśžãĂĆ äÿŇéİćæĹŚäżňåŕźåőČè£ŻèąŇäÿĂäžŻåřŘçŽĎä£őæŤźä¡£å¿ŮåőČåŘŕäżěéĂĆçŤÍäžŐåď'Žçž£çÍŃïijŽ

```
from socket import socket, AF_INET, SOCK_STREAM
import threading

class LazyConnection:
    def __init__(self, address, family=AF_INET, type=SOCK_STREAM):
        self.address = address
        self.family = AF_INET
        self.type = SOCK_STREAM
        self.local = threading.local()

    def __enter__(self):
        if hasattr(self.local, 'sock'):
            raise RuntimeError('Already connected')
        self.local.sock = socket(self.family, self.type)
        self.local.sock.connect(self.address)
        return self.local.sock

    def __exit__(self, exc_ty, exc_val, tb):
        self.local.sock.close()
        del self.local.sock
```

äżčçăĄäÿ■ïijŇèĞłåůśèğĆåŕ§åŕźäžŐ `self.local` åśđæĂğçŽĎä¡£çŤÍãĂĆ åőČèćńåĹİåğŇåŇŰäÿžäÿĂäÿł `threading.local()` åőđä¿ŇãĂĆ åĚűäżŰæŰźæşŢæŞ■ä¡IJèćńå■ŸåĆÍäÿž `self.local.sock` çŽĎåěŮæŐěå■ŮåŕźèśąãĂĆ æIJL'äžĘè£ŹäžŻåřśåŘŕäżěåIJÍåď'Žçž£çÍŃäÿ■åőL'åĚÍçŽĎä¡£çŤÍ `LazyConnection` åőđä¿ŇäžĘãĂĆä¿ŇåęĆïijŽ

```
from functools import partial
def test(conn):
    with conn as s:
        s.send(b'GET /index.html HTTP/1.0\r\n')
        s.send(b'Host: www.python.org\r\n')

        s.send(b'\r\n')
        resp = b''.join(iter(partial(s.recv, 8192), b''))

    print('Got {} bytes'.format(len(resp)))

if __name__ == '__main__':
    conn = LazyConnection(('www.python.org', 80))

    t1 = threading.Thread(target=test, args=(conn,))
    t2 = threading.Thread(target=test, args=(conn,))
    t1.start()
    t2.start()
    t1.join()
    t2.join()
```

åőČäźŃæL’ÄažěèąŇå¿ŮéĂŽçŽĎåŐ§åŻäæŸŕæŕŘäÿłçž£çĺŃäijŽåĹŻåžžäÿĂäÿłèĞłåůśäÿŞåśđçŽĎåěŮæŐ
åŻäæ■d’ïijŇå¡Şäÿ■åŘŇçŽĎçž£çĺŃæL’ğèąŇåěŮæŐěå■ŮæŞ■ä¡IJæŮűïijŇçŤśažŐæŞ■ä¡IJçŽĎæŸŕäÿ■åŘŇçŽ

èőĺèőž

åIJĺåď’ğéČĺåĹEçĺŃåžŘäÿ■åĹŻåžžåŠŇæŞ■ä¡IJçž£çĺŃçL’źåőŽçŁűæĂĄåźűäÿ■äijŽæIJL’äžĂäźĹéŮőéćŸã
äÿ■è£ĞïijŇå¡ŞåĞžäžEéŮőéćŸçŽĎæŮűåĂŹïijŇéĂŽåÿÿæŸŕåŻääÿžæ§Řäÿłåŕźèśąèćńåď’Žäÿłçž£çĺŃä¡£çŤĺåĹ
æŕŤåęĆäÿĂäÿłåěŮæŐěå■ŮæĹŮæŰĞäžűãĂĆä¡ääÿ■èČ¡èőľ’æL’ĂæIJL’çž£çĺŃåĚśäžńäÿĂäÿłå■ŢçŃňåŕźèśąïijŇ
åŻääÿžåď’Žäÿłçž£çĺŃåŘŇæŮűèŕżåŠŇåEŹçŽĎæŮűåĂŹäijŽäžğçŤ§æůůäźśãĂĆ
æIJňåIJřçž£çĺŃå■ŸåĆĺéĂŽè£Ğèőľ’è£ŹäžŻèţĎæžŘåŘłèČ¡åIJĺèćńä¡£çŤĺçŽĎçž£çĺŃäÿ■åŘŕèğĄæĺèèğčåEşè£Ź

æIJňèŁĆäÿ■ïijŇä¡£çŤĺ `thread.local()` åŘŕäžěèőľ’ `LazyConnection` çśżæŤŕæŇĄäÿĂäÿłçž£çĺŃäÿĂäÿłè£đæŐěïijŇ
èĂŇäÿ■æŸŕåŕźäžŐæL’ĂæIJL’çŽĎè£ŻçĺŃéČ¡åŘłæIJL’äÿĂäÿłè£đæŐěãĂĆ

åĚűåŐ§çŘEæŸŕïijŇæŕŘäÿł `threading.local()` åőđä¿ŃäÿžæŕŘäÿłçž£çĺŃçżť’æŁď’çİĂäÿĂäÿłå■Ţç
æL’ĂæIJL’æŹőéĂŽåőđä¿ŃæŞ■ä¡IJæŕŤåęĆèŐůåŘŰãĂĄä£őæŤźåŠŇåĹäéŹď’åĂijäžĚäžĚæŞ■ä¡IJè£ŹäÿłåůÛå
æŕŘäÿłçž£çĺŃä¡£çŤĺäÿĂäÿłçŃňçńŃçŽĎå■ŮåĚÿåŕśåŘŕäžěä£ÌèŕĄæŢřæ■őçŽĎéŽŤçężäžEąĂĆ

14.7 12.7 åĹŻåžžäÿĂäÿłçž£çĺŃæśă

éŮőéćŸ

ä¡ăåĹŻåžžäÿĂäÿłåůěä¡IJèĂĚçž£çĺŃæśăïijŇçŤĺæĺěåŞ■ážŤåőćæĹůçńŕèŕůæśĆæĹŰæL’ğèąŇåĚűäžŰçŽĎå

èğčåEşæŰźæąĹ

`concurrent.futures` åĞ¡æŢřåžŞæIJL’äÿĂäÿł `ThreadPoolExecutor` çśżåŘŕäžěèćńçŤĺæĺěåőŇæĹŘè£ŹäÿłäžżåŁąãĂĆ äÿŃéĺćæŸŕäÿĂäÿłçőĂå■ŢçŽĎTCPæIJ■åŁąåŹĺïijŇä¡£çŤĺäžĚ

```
from socket import AF_INET, SOCK_STREAM, socket
from concurrent.futures import ThreadPoolExecutor

def echo_client(sock, client_addr):
    '''
    Handle a client connection
    '''
    print('Got connection from', client_addr)
    while True:
        msg = sock.recv(65536)
        if not msg:
            break
        sock.sendall(msg)
    print('Client closed connection')
    sock.close()

def echo_server(addr):
```

(continues on next page)

(continued from previous page)

```
    pool = ThreadPoolExecutor(128)
    sock = socket(AF_INET, SOCK_STREAM)
    sock.bind(addr)
    sock.listen(5)
    while True:
        client_sock, client_addr = sock.accept()
        pool.submit(echo_client, client_sock, client_addr)

echo_server(('',15000))
```

åęĆæđIJä¡ăæČşæL'ŃåŁÍåĹŻåżžä¡äèĞłåůśçŽĎçž£çÍŃæśäïijŇ
éĂŽåÿÿåŘŕäżěä¡£çŤÍäÿĂäÿłQueueæİěè¡żæİ¿åőđçŐřãĂĆäÿŃéİćæŸŕäÿĂäÿłçÍ■å¿őäÿ■åŘŇä¡ĘæŸŕæL'ŃåŁÍåő

```
from socket import socket, AF_INET, SOCK_STREAM
from threading import Thread
from queue import Queue

def echo_client(q):
    '''
    Handle a client connection
    '''
    sock, client_addr = q.get()
    print('Got connection from', client_addr)
    while True:
        msg = sock.recv(65536)
        if not msg:
            break
        sock.sendall(msg)
    print('Client closed connection')

    sock.close()

def echo_server(addr, nworkers):
    # Launch the client workers
    q = Queue()
    for n in range(nworkers):
        t = Thread(target=echo_client, args=(q,))
        t.daemon = True
        t.start()

    # Run the server
    sock = socket(AF_INET, SOCK_STREAM)
    sock.bind(addr)
    sock.listen(5)
    while True:
        client_sock, client_addr = sock.accept()
        q.put((client_sock, client_addr))

echo_server(('',15000), 128)
```

ä¡£çŤÍ `ThreadPoolExecutor` çŻÿåŕźäžŐæĽŃåŁÍåŏđçŐřçŽĎäÿĂäÿłåě¡åď ĎåIJÍäžŐåŏČä¡£å¿Ů äżżåŁąæŘŘäžď èĂĚæŻť æŰźä¿£çŽĎäżŐèćńèřČçŤÍåĞ¡æŢřäÿ■èŐůåŘŰè£ŤåŻđåĂijãĂĆä¿ŃåęĆïijŇä¡ääŘŕèČ

```
from concurrent.futures import ThreadPoolExecutor
import urllib.request

def fetch_url(url):
    u = urllib.request.urlopen(url)
    data = u.read()
    return data

pool = ThreadPoolExecutor(10)
# Submit work to the pool
a = pool.submit(fetch_url, 'http://www.python.org')
b = pool.submit(fetch_url, 'http://www.pypy.org')

# Get the results back
x = a.result()
y = b.result()
```

ä¿Ńå■Řäÿ■è£ŤåŻđçŽĎhandleåŕźèśąäijŽåÿőä¡ääď ĎçŘĘæĽ ĂæIJĽçŽĎéŸżåąđäÿŐå■Řä¡IJïijŇçĎűåŘŐä çĽźåĹńçŽĎïijŇ`a.result()` æŞ■ä¡IJäijŽéŸżåąđè£ŻçÍŃçŻť åĹřåŕźåžŤçŽĎåĞ¡æŢřæĽ ğèąŇäőŇæĹŘåźűè£Ť

èőÍèőž

éĂŽåÿÿæİěèőšïijŇä¡ääžŤèŕěéĄ£åĚ■çijŰåĘŹçž£çÍŃæŢřéĞŘåŘŕäžěæŮäéŹŘåĹűäćđéŢ£çŽĎçÍŃäžŘãĂĆ

```
from threading import Thread
from socket import socket, AF_INET, SOCK_STREAM

def echo_client(sock, client_addr):
    '''
    Handle a client connection
    '''
    print('Got connection from', client_addr)
    while True:
        msg = sock.recv(65536)
        if not msg:
            break
        sock.sendall(msg)
    print('Client closed connection')
    sock.close()

def echo_server(addr, nworkers):
    # Run the server
    sock = socket(AF_INET, SOCK_STREAM)
    sock.bind(addr)
    sock.listen(5)
    while True:
```

(continues on next page)

(continued from previous page)

```
        client_sock, client_addr = sock.accept()
        t = Thread(target=echo_client, args=(client_sock, client_
↪addr))
        t.daemon = True
        t.start()

echo_server(('',15000))
```

ář¡çőąè£Źäÿłäź§åŘŕäżěåůěä¡IJïijŇ ä¡ĘæŸŕåőČäÿ■èČ¡æŁţå¿ąæIJL'äžžèŕŢåŻ¿éĂŽè£ĞåĹŻåżžåď'ğéĞŘçž
éĂŽè£Ğä¡£çŤĺécĎåĚĹåĹİåğŇåŇŰçŽĎçž£çĺŃæśäïijŇä¡ääŘŕäżěèő¿ç¡őåŘŇæŮűè£ŘèąŇçž£çĺŃçŽĎäÿŁéŹŘa

ä¡ääŘŕèČ¡äijŽåĚşå£ČåĹŻåżžåď'ğéĞŘçž£çĺŃäijŽæIJL'äżĂäźĹåŘŐæđIJãĂĆ
çŐräżčæŞ■ä¡IJçşżçż§åŘŕäżěå¿Ĺè¡żæĺ¿çŽĎåĹŻåżžåĞää■Čäÿłçž£çĺŃçŽĎçž£çĺŃæśäãĂĆ
çŤŽèĞşïijŇåŘŇæŮűåĞää■Čäÿłçž£çĺŃç■L'å¿Ĕåůěä¡IJáźűäÿ■äijŽåŕźåĚűäżŰäżčçăĄäžğçŢ§æĂğèČ¡å¡śåŞ■ãĂ
å¡ŞçĎűäžĘïijŇåęĆæđIJæL'ĂæIJL'çž£çĺŃåŘŇæŮűèćńåŤd'éĘŠåźűçńŃå■şåIJĺCPUäÿŁæL'ğèąŇïijŇéĆčåřśäÿ■
éĂŽåÿÿïijŇä¡ääžŤèŕěåŘłåIJĺI/Oåď'ĎçŘĘçŻÿåĚşäżčçăĄäÿ■ä¡£çŤĺçž£çĺŃæśäãĂĆ

åĹŻåżžåď'ğçŽĎçž£çĺŃæśäçŽĎäÿĂäÿłåŘŕèČ¡éIJĂèęĄåĚşæşĺçŽĎéŮőécŸæŸŕåĘĚå■ŸçŽĎä¡£çŤĺãĂĆ
ä¿ŃåęĆïijŇåęĆæđIJä¡ääIJĺOS Xçşżçż§äÿŁéĺćåĹŻåżž2000äÿłçž£çĺŃïijŇçşżçż§æŸ¿çď'žPythonè£ŻçĺŃä¡£çŤĺã
äÿ■è£ĞïijŇè£Źäÿłèőąçő̊ŮéĂŽåÿÿæŸŕæIJL'èŕŕåůőçŽĎãĂĆå¡ŞåĹŻåżžäÿĂäÿłçž£çĺŃæŮűïijŇæŞ■ä¡IJçşżçż§äi
æŤ¿ç¡őçž£çĺŃçŽĎæL'ğèąŇæĂĹïijĹéĂŽåÿÿæŸŕ8MBåď'ğåřŘïijL'ãĂĆä¡ĘæŸŕè£ŹäÿłåĘĚå■ŸåŘłæIJL'äÿĂåřŘ
åŻăæ■d'ïijŇPythonè£ŻçĺŃä¡£çŤĺåĹřçŽĎçIJ§åőđåĘĚå■ŸåĚűåőđå¿ĹåřŘ
ïijĹæŕŤåęĆïijŇåŕźäžŐ2000äÿłçž£çĺŃæIèèőšïijŇåŘłä¡£çŤĺåĹřäžĘ70MBçŽĎçIJ§åőđåĘĚå■ŸïijŇèĂŇäÿ■æŸŕ
åęĆæđIJä¡äæŃĚå£ČèŹŽæŃ§åĘĚå■Ÿåď'ğåřŘïijŇåŘŕäżěä¡£çŤĺ threading.
stack_size() åĞ¡æŢřæIěéŹ■ä¡ŐåőČãĂĆä¿ŃåęĆïijŽ

```
import threading
threading.stack_size(65536)
```

åęĆæđIJä¡ääŁääÿŁè£ŹæIąèŕ■åŘěåźűåĘ■æňąè£ŘèąŇåL'■éIćçŽĎåĹŻåżž2000äÿłçž£çĺŃèŕŢéłŇïijŇ
ä¡ääijŽåŘŚçŐŕPythonè£ŻçĺŃåŘłä¡£çŤĺåĹřäžĘåď'ğæęĆ210MBçŽĎèŹŽæŃ§åĘĚå■ŸïijŇèĂŇçIJ§åőđåĘĚå■Ÿ
æşĺæĎŘçž£çĺŃæäĹåď'ğåřŘå£ĚéążèĞşåřŚäÿž32768å■ŮèŁĆïijŇéĂŽåÿÿæŸŕçşżçż§åĘĚå■Ÿéąţåď'ğåřŘïijĹ409

14.8 12.8 çőĂå■ŢçŽĎåźűèąŇçijŰçĺŃ

éŮőécŸ

ä¡äæIJL'äÿłçĺŃåžŘèęĄæL'ğèąŇCPUåŕĘéŻĘåđŃåůěä¡IJïijŇä¡äæČşèőľäżŰåĹľçŤĺåď'ŽæäÿCPUçŽĎäijŸ

èğčåĘşæŰźæąĹ

concurrent.futures åžŞæŘŘä¿ŻäžĘäÿĂäÿł ProcessPoolExecutor çśżïijŇ
åŘŕèćńçŤĺæĚěåIJĺäÿĂäÿłå■ŢçŃňçŽĎPythonèğčéĞŁåŹĺäÿ■æL'ğèąŇèőąçőŮåŕĘéŻĘåđŃåĞ¡æŢřãĂĆ
äÿ■è£ĞïijŇèęĄä¡£çŤĺåőČïijŇä¡äéęŰåĚĹèęĄæIJL'äÿĂäžŻèőąçőŮåŕĘéŻĘåđŃçŽĎäżżåŁąãĂĆ
æĹŚäżňéĂŽè£ĞäÿĂäÿłçőĂå■ŢèĂŇåőđéŹĚçŽĎä¿Ňå■ŘæĚěæijŤçď'žåőČãĂĆåĄĞåőŽä¡äæIJL'äÿłApache
webæIJ■åŁąåŹĺæŮěå£ŮçŻőå¡ŢçŽĎgzipåŐŃçijľåŇĚïijŽ

```
logs/
   20120701.log.gz
   20120702.log.gz
   20120703.log.gz
   20120704.log.gz
   20120705.log.gz
   20120706.log.gz
   ...
```

è£ŻäÿĂæ■ěåĄĞèő¿æŕŔäÿłæŮěå£ŮæŰĞäżűåĘĚåőźçśżäijijäÿŃéĺćè£Źæăůïij Ž

```
124.115.6.12 - - [10/Jul/2012:00:18:50 -0500] "GET /robots.txt ..."␣
↪200 71
210.212.209.67 - - [10/Jul/2012:00:18:51 -0500] "GET /ply/ ..." 200␣
↪11875
210.212.209.67 - - [10/Jul/2012:00:18:51 -0500] "GET /favicon.ico ..
↪." 404 369
61.135.216.105 - - [10/Jul/2012:00:20:04 -0500] "GET /blog/atom.xml␣
↪..." 304 -
...
```

äÿŃéĺćæŸŕäÿĂäÿłèĎŽæIJňïijŇåIJÍè£ŹäžŻæŮěå£ŮæŰĞäżűäÿ■æ§ěæĽ¿åĞžæĽĂæIJĽèő£éŮőè£Ğrobot

```python
# findrobots.py

import gzip
import io
import glob

def find_robots(filename):
    '''
    Find all of the hosts that access robots.txt in a single log␣
↪file
    '''
    robots = set()
    with gzip.open(filename) as f:
        for line in io.TextIOWrapper(f,encoding='ascii'):
            fields = line.split()
            if fields[6] == '/robots.txt':
                robots.add(fields[0])
    return robots

def find_all_robots(logdir):
    '''
    Find all hosts across and entire sequence of files
    '''
    files = glob.glob(logdir+'/*.log.gz')
    all_robots = set()
    for robots in map(find_robots, files):
        all_robots.update(robots)
```

(continues on next page)

(continued from previous page)

```
    return all_robots

if __name__ == '__main__':
    robots = find_all_robots('logs')
    for ipaddr in robots:
        print(ipaddr)
```

åL'■éİćçŽĎçÍŃåžŘä¡£çŤÍäžĘéĂŽåÿÿçŽĎmap-reduceéčŐæăijæİěçijŰåĘŹãĂĆ åĞ¡æŢř find_robots() åIJÍäÿĂäÿłæŰĞäżűåŘ■éŻĘåŘĹäÿŁåĄŽmapæŞ■ä¡IJïijŇåźűåřĘçżŞæđIJæśĞæĂżäÿžäÿĂä äź§åřśæŸŕ find_all_robots() åĞ¡æŢřäÿ■çŽĎ all_robots éŻĘåŘĹãĂĆ çŐřåIJÍïijŇåĄĞèő¿ä¡ăæČşèęĄä£őæŤźè£ŹäÿłçÍŃåžŘèőľ'åőČä¡£çŤÍåď'ŽæăÿCPUãĂĆ å¿ĹçőĂå■ŢâĂŤâĂŤåŘłéIJĂèęĄåřĘmap()æŞ■ä¡IJæŻ£æ■ćäÿžäÿĂäÿł concurrent.futures åžŞäÿ■çŤ§æĹŘçŽĎçśżäijijæŞ■ä¡IJå■şåŘŕãĂĆ äÿŃéİćæŸŕäÿĂäÿłçőĂå■Ţä£őæŤźçL'ĹæIJňïijŽ

```
# findrobots.py

import gzip
import io
import glob
from concurrent import futures

def find_robots(filename):
    '''
    Find all of the hosts that access robots.txt in a single log␣
↪file

    '''
    robots = set()
    with gzip.open(filename) as f:
        for line in io.TextIOWrapper(f,encoding='ascii'):
            fields = line.split()
            if fields[6] == '/robots.txt':
                robots.add(fields[0])
    return robots

def find_all_robots(logdir):
    '''
    Find all hosts across and entire sequence of files
    '''
    files = glob.glob(logdir+'/*.log.gz')
    all_robots = set()
    with futures.ProcessPoolExecutor() as pool:
        for robots in pool.map(find_robots, files):
            all_robots.update(robots)
    return all_robots

if __name__ == '__main__':
    robots = find_all_robots('logs')
```

(continues on next page)

(continued from previous page)

```
    for ipaddr in robots:
        print(ipaddr)
```

éĂŽè£Ğè£ŹäÿłäŁőæŤźåŘŐïijŇè£ŘèąŇè£ŹäÿłèĎŽæIJňäžğçŤ§åŘŇæăůçŽĎçżŞæđIJïijŇä¡ĘæŸŕåIJÍåŻŻ
åőđéŹĚçŽĎæĂğèČ¡äijŸåŇŰæŢĹæđIJæăźæ■őä¡ăçŽĎæIJžåŹÍCPUæŢřéĞŘçŽĎäÿ■åŘŇèĂŇäÿ■åŘŇãĂĆ

èőÍèőž

`ProcessPoolExecutor` çŽĎåĚÿåđŃçŤÍæşŢåęĆäÿŃïijŽ

```
from concurrent.futures import ProcessPoolExecutor

with ProcessPoolExecutor() as pool:
    ...
    do work in parallel using pool
    ...
```

åĚűåŐ§çŘĘæŸŕïijŇäÿĂäÿł `ProcessPoolExecutor`
åĹŻåżžNäÿłçŃňçńŃçŽĎPythonèğčéĞŁåŹÍïijŇ NæŸŕçşżçż§äÿŁéÍćåŘŕçŤÍCPUçŽĎäÿłæŢřãĂĆä¡ăåŘŕäżěéĂŽ
`ProcessPoolExecutor(N)` æÍěä£őæŤź åď ĎçŘĘåŹÍæŢřéĞŘãĂĆè£Źäÿłåď ĎçŘĘæśäăijŽäÿĂçŻť è£Řèą
çĎűåŘŐåď ĎçŘĘæśäèćńåĚşéŮ■ãĂĆäÿ■è£ĞïijŇçÍŃåžŘäijŽäÿĂçŻť ç■L'å¿ĚçŻť åĹřæL'ĂæIJL'æŘŘäžď çŽĎå

èćńæŘŘäžď åĹřæśääÿ■çŽĎåůěä¡IJå£ĚéążèćńåőŽäźL'äÿžäÿĂäÿłåĞ¡æŢřãĂĆæIJL'äÿď çğ■æŰźæşŢåŐżæ
åęĆæđIJä¡äæČşèőľäÿĂäÿłåĹŮèąÍæŐÍåŕijæĹŰäÿĂäÿł `map()`
æŞ■ä¡IJåźűèąŇæL'ğèąŇçŽĎèŕİïijŇåŘŕä¡£çŤÍ `pool.map()` :

```
# A function that performs a lot of work
def work(x):
    ...
    return result

# Nonparallel code
results = map(work, data)

# Parallel implementation
with ProcessPoolExecutor() as pool:
    results = pool.map(work, data)
```

åŘęåď ŰïijŇä¡äåŘŕäżěä¡£çŤÍ `pool.submit()` æÍěæL'ŃåŁÍçŽĎæŘŘäžď å■ŢäÿłäżżåŁąïijŽ

```
# Some function
def work(x):
    ...
    return result

with ProcessPoolExecutor() as pool:
    ...
    # Example of submitting work to the pool
```

(continues on next page)

(continued from previous page)

```
    future_result = pool.submit(work, arg)

    # Obtaining the result (blocks until done)
    r = future_result.result()
    ...
```

åęĆæđIJä¡ăæL'ŃåŁÍæŘŘäžď'äÿĂäÿłäżżåŁąïijŇçżŞæđIJæŸŕäÿĂäÿł `Future` åőđä¿ŃãĂĆ èęĄèŐůåŘŰæIJĂçżĹçżŞæđIJïijŇä¡ăéIJĂèęĄěřĆçŤĺåőĆçŽĎ `result()` æŰźæşŢãĂĆ åőČäijŽéŸżåąđè£ŽçĺŃçŽť'åĹřçżŞæđIJèćńè£ŤåŽđæĺěãĂĆ

åęĆæđIJäÿ■æČşéŸżåąđïijŇä¡ăè£ŸåŘŕäžěä¡£çŤĺäÿĂäÿłåŽđèřČåĞ¡æŢřïijŇä¿ŃåęĆïijŽ

```
def when_done(r):
    print('Got:', r.result())

with ProcessPoolExecutor() as pool:
     future_result = pool.submit(work, arg)
     future_result.add_done_callback(when_done)
```

åŽđèřČåĞ¡æŢřæŐěåŘŮäÿĂäÿł `Future` åőđä¿ŃïijŇèćńçŤĺæĺěèŐůåŘŰæIJĂçżĹçŽĎçżŞæđIJïijĹæŕŤåęĆ
ář¡çőąåď'ĎçŘĘæśäå¿Ĺåőźæ ŸŞä¡£çŤĺïijŇåIJĺèő¿èőąåď'ğçĺŃåžŘçŽĎæŮűåĂŹè£ŸæŸŕæIJL'å¿Ĺåď'ŽéIJĂèęĄæ

- è£Źçğ■åźűèąŇåď'ĎçŘĘæŁĂæIJŕåŘłéĂĆçŤĺäžŐéĆčäžŻåŘŕäžěèćńåĹĘèğčäÿžäžŠçŻÿçŃňçńŃéČĺåĹĘçŽ
- èćńæŘŘäžď'çŽĎäżżåŁąå£Ěéążæ ŸŕçőĂå■ŢåĞ¡æŢřå¡ćåijŘãĂĆåŕźäžŐæŰźæşŢãĂĄéŮ■åŇĚåŠŇåĚűäżŮ
- åĞ¡æŢřåŘĆæŢřåŠŇè£ŤåŽđåĂijå£ĚéążåĚijåőźpickleïijŇåŽääÿžèęĄä¡£çŤĺåĹřè£ŽçĺŃéŮť'çŽĎéĂŽä£ąï
- èćńæŘŘäžď'çŽĎäżżåŁąåĞ¡æŢřäÿ■åžŤä£ĺçŢŹçŁűæĂĄæĹŰæIJL'åL'ŕä¡IJçŤĺãĂĆéŹď'äžĘæL'Ş å■řæŮěå

  äÿĂæŮęåŘŕåŁĺä¡ääÿ■èČ¡æŐğåĹűå■Řè£ŽçĺŃçŽĎäżżä¡ŢèąŇäÿžïijŇåŽäæ■ď'æIJĂåě¡ä£ĺæŇĄçőĂå■ŢåŠ

- åIJĺUnixäÿŁè£ŽçĺŃæśäéĂŽè£ĞèřČçŤĺ `fork()` çşżçż§èřČçŤĺèćńåĹŽåżžïijŇ

åőČäijŽåĚŃéŽĘPythonèğčéĞŁåŹĺïijŇåŇĚæŃňforkæŮűçŽĎæL'ĂæIJL'çĺŃåžŘçŁűæĂĄãĂĆ
èĂŇåIJĺWindowsäÿŁïijŇåĚŃéŽĘèğčéĞŁåŹĺæŮűäÿ■äijŽåĚŃéŽĘçŁűæĂĄãĂĆ åőđéŹĚçŽĎ-
forkæŞ■ä¡IJäijŽåIJĺçňňäÿĂæňąèřČçŤĺ `pool.map()` æĹŰ `pool.submit()`
åŘŐåŘŚçŤ§ãĂĆ

- å¡Şä¡ăæůůåŘĹä¡£çŤĺè£ŽçĺŃæśäåŠŇåď'Žçž£çĺŃçŽĎæŮűåĂŹèęĄçL'źåĹńåřŘå£ČãĂĆ

  ä¡ăåžŤèřěåIJĺåĹŽåżžäżżä¡Ţçž£çĺŃäźŃåL'■åĚĹåĹŽåżżåźűæ£Ăæť'żè£ŽçĺŃæśäïijĹæŕŤåęĆåIJĺçĺŃåžŘåŘŕ

14.9 12.9 PythonçŽĎåĚĺåśĂéŤĄéŮőéćŸ

éŮőéćŸ

ä¡ăåůšçżŘåŘňèŕť'è£ĞåĚĺåśĂèğčéĞŁåŹĺéŤĄGILïijŇæŃĚå£ČåőČäijŽå¡śåŞ■åĹřåď'Žçž£çĺŃçĺŃåžŘçŽĎæ

èğčåĘşæŰźæąĹ

åř¡çőąPythonåőŇåĚĺæŤŕæŇĄåď'Žçž£çĺŃçijŰçĺŃïijŇ ä¡ĘæŸŕèğčéĞŁåŹĺçŽĎCèŕ■èĺĂåőđçŐŕéČĺåĹĘåIJĺ
åőđéŹĚäÿŁïijŇèğčéĞŁåŹĺèćńäÿĂäÿłåĚĺåśĂèğčéĞŁåŹĺéŤĄä£ĺæŁď'çĺĂïijŇåőČçąőä£ĺäżżä¡ŢæŰűåĂŹéČ¡åŔ
GILæIJĂåď'ğçŽĎéŰőéćŸåřśæŸŕPythonçŽĎåď'Žçž£çĺŃçĺŃåžŘåźűäÿ■èČ¡åĹl'çŤĺåď'ŽæäÿCPUçŽĎäijŸåŁ£
ïijĹæŕŤåęĆäÿĂäÿłä¡£çŤĺäžĘåď'Žäÿłçž£çĺŃçŽĎèőąçőŮåŕĘéŻĘåđŃçĺŃåžŘåŘłäijŽåIJĺäÿĂäÿłå■ŢCPUäÿŁéĺćé

åIJĺèőĺèőžæŹőéĂŽçŽĎGILäźŃåL'■ïijŇæIJL'äÿĂçĆźèęĄåijžèřČçŽĎæŸŕGILåŘłäijŽå¡śåŞ■åĹřéĆčäžŻäÿ
åęĆæđIJä¡äçŽĎçĺŃåžŘåď'ğéČĺåĹĘåŘłäijŽæűL'åŘŁåĹřI/OïijŇæŕŤåęĆç¡ŚçżIJäžď'äžŠïijŇéĆčäźĹä¡£çŤĺåď'Žç
åŻääÿžåőČäžňåď'ğéČĺåĹĘæŮűéŮť'éČ¡åIJĺç■L'å¿ĚãĂĆåőđéŹĚäÿŁïijŇä¡äåőŇåĚĺåŘŕäżěæŤ¿å£ČçŽĎåĹŻåżżå
çŐřäżčæŞ■ä¡IJçşżçż§è£ŘèąŇè£ŹäźĹåď'Žçž£çĺŃæšąæIJL'äżżä¡ŢåŐŃåŁŻïijŇæšąåŢěåŘŕæŃĚå£ČçŽĎãĂĆ

èĂŇåŕźäžŐä¿ĺèţŰCPUçŽĎçĺŃåžŘïijŇä¡äéIJĂèęĄåijĎæÿĚæěŽæL'ğèąŇçŽĎèőąçőŮçŽĎçL'źçĆźãĂĆ
ä¿ŃåęĆïijŇäijŸåŇŰåžŢåśĆçőŮæşŢèęĄæŕŤä¡£çŤĺåď'Žçž£çĺŃè£ŘèąŇå£ńå¿Ůåď'ŽãĂĆ
çśżäijijçŽĎïijŇçŤśäžŐPythonæŸŕèğčéĞŁæL'ğèąŇçŽĎïijŇåęĆæđIJä¡äåřĘéĆčäžŻæĂğèČ¡çŞűéćĹäżčçäĄçğżå
éĂ§åžęäź§äijŽæŘŘå■ĞçŽĎå¿Ĺå£ńäĂĆåęĆæđIJä¡äèęĄæŞ■ä¡IJæŢřçżĎïijŇéĆčäźĹä¡£çŤĺNumPyè£ŹæäůçŽĎ
æIJĂåŘŐïijŇä¡äè£ŸåŕäżěèĂČèŹŚäÿŇåĚűäżŰåŘŕéĂL'åőđçŐřæŰźæąĹïijŇæŕŤåęĆPyPyïijŇåőČéĂŽè£ĞäÿĂ
ïijĹäÿ■è£ĞåIJĺåĘŹè£ŹæIJňäźęçŽĎæŮűåĂŹåőČè£Ÿäÿ■èČ¡æŤŕæŇĄPython 3ïijL'ãĂĆ

è£ŸæIJL'äÿĂçĆźèęĄæşĺæĎŘçŽĎæŸŕïijŇçž£çĺŃäÿ■æŸŕäÿŞéŮĺçŤĺæĭèäijŸåŇŰæĂğèČ¡çŽĎãĂĆ
äÿĂäÿłCPUä¿ĺèţŰåđŃçĺŃåžŘåŘŕèČ¡äijŽä¡£çŤĺçž£çĺŃæĭèçőąçŘĘäÿĂäÿłåŻ¿å¡ćçŤĺæĹůçŢŇéĺćãĂĄäÿĂäÿłç¡
è£ŹæŮűåĂŹïijŇGILäijŽäžğçŤ§äÿĂäžŻéŮőéćŸïijŇåŽääÿžåęĆæđIJäÿĂäÿłçž£çĺŃéŢ£æIJ§æŇĄæIJL'GILçŽĎ
äžŃåőđäÿŁïijŇäÿĂäÿłåĘŹçŽĎäÿ■åě¡çŽĎCèŕ■èĺĂæL'l'åŞţäijŽåŕijèĞť'è£ŹäÿłéŮőéćŸæŻť'åŁääÿěéĞ■ïijŇ
åř¡çőąäžčçăĄçŽĎèőąçőŮéČĺåĹĘäijŽæŕŤäźŃåL'■è£ŘèąŇçŽĎæŻť'å£ńäžŻãĂĆ

èŕť'äžĘè£ŹäźĹåď'ŽïijŇçŐřåIJĺæČşèŕť'çŽĎæŸŕæĹŚäžňæIJL'äÿď'çğ■ç■ŰçŢěæĭèèğčåĘşGILçŽĎçijžçĆźãĂ
éęŰåĚĹïijŇåęĆæđIJä¡äåőŇåĚĺåůěä¡IJäžŐPythonçŐŕåćČäÿ■ïijŇä¡äåŘŕäżěä¡£çŤĺ
multiprocessing æĺąåĭŮæĭěåĹŻåżżäÿĂäÿłè£ŻçĺŃæśäïijŇ
åźűåČŘå■ŘåŘŇåď'ĎçŘĘåŹĺäÿĂæăůçŽĎä¡£çŤĺåőČãĂĆä¿ŃåęĆïijŇåĄĞåęĆä¡äæIJL'åęĆäÿŃçŽĎçž£çĺŃäžčç

```
# Performs a large calculation (CPU bound)
def some_work(args):
    ...
    return result

# A thread that calls the above function
def some_thread():
    while True:
        ...
        r = some_work(args)
    ...
```

ä£őæŤźäžčçăĄïijŇä¡£çŤĺè£ŻçĺŃæśäïijŽ

```
# Processing pool (see below for initiazation)
pool = None

# Performs a large calculation (CPU bound)
def some_work(args):
    ...
    return result
```

(continues on next page)

(continued from previous page)

```
# A thread that calls the above function
def some_thread():
    while True:
        ...
        r = pool.apply(some_work, (args))
        ...

# Initiaze the pool
if __name__ == '__main__':
    import multiprocessing
    pool = multiprocessing.Pool()
```

è£ŹäÿłéĂŽè£Ğä¡£çŤÍäÿĂäÿłæŁĂåůğåĹľçŤÍè£ŻçÍŃæśăèğčåĘşäžĘGILçŽĎéŮőéćŸãĂĆ
å¡ŞäÿĂäÿłçž£çÍŃæČşèęĄæĽğèąŇCPUåŕĘéŻĘåđŃÅůěä¡IJæŮűïijŇäijŽåřĘäżżåŁąåŚŚçżŹè£ŻçÍŃæśăĂĆ
çĎűåŘŐè£ŻçÍŃæśääijŽåIJÍåŘęåď ŰäÿĂäÿłè£ŻçÍŃäÿ■åŘŕåŁÍäÿĂäÿłå■ŢçŃňçŽĎPythonèğčéĞŁåŹÍæÍěåůěä¡I
å¡Ş£çÍŃç■L'å¿ĚçżŞæđIJçŽĎæŮűåĂŹäijŽéĞŁæŤ¿GILãĂĆ åźűäÿŤïijŇçŤśäžŐèőąçőŮäżżåŁąåIJÍå■ŢçŃňèğ
åIJÍäÿĂäÿłåď Žæäÿçşżçż§äÿŁéÍćïijŇä¡ääijŽåŚŚçŐřè£ŹäÿłæŁĂæIJŕåŘŕäżèèőľä¡åå¿Łåě¡çŽĎåĹľçŤÍåď ŽCPU

åŘęåď ŰäÿĂäÿłèğčåĘşGILçŽĎç■ŰçŢěæŸŕä¡£çŤÍCæL'ľåśŢçijŰçÍŃæŁĂæIJŕãĂĆ
äÿżèęĄæĂÍæČşæŸŕåřĘèőąçőŮåŕĘéŻĘåđŃäżżåŁąè¡ňçğżçżŹCïijŇèů§PythonçŃňçńŃïijŇåIJÍåůěä¡IJçŽĎæŮűä
è£ŹåŘŕäżěéĂŽè£ĞåIJÍCäżčçăĄäÿ■æŚŠåĚěäÿŃéÍćè£ŹæăůçŽĎçL'źæőŁåőŘæÍěåőŇæĹŘïijŽ

```
#include "Python.h"
...

PyObject *pyfunc(PyObject *self, PyObject *args) {
   ...
   Py_BEGIN_ALLOW_THREADS
   // Threaded C code
   ...
   Py_END_ALLOW_THREADS
   ...
}
```

åęĆæđIJä¡ăä¡£çŤÍåĚűäżŰåůěåĚůèő£éŮőCèŕ■èÍĂïijŇæŕŤåęĆåŕźäžŐCythonçŽĎctypesåžŞïijŇä¡ăäÿ■éIJĂ
ä¿ŃåęĆïijŇctypesåIJÍèřČçŤÍCæŮűäijŽèĞłåŁÍéĞŁæŤ¿GILãĂĆ

èőÍèőž

èőÿåď ŽçÍŃåžŘåŚŸåIJÍéÍćåŕźçž£çÍŃæĂğèČ¡éŮőéćŸçŽĎæŮűåĂŹïijŇél'ňäÿŁåřśäijŽæĂłç¡łGILïijŇäżĂ
åĚűåőđè£Źæăůå■Řåď łäÿ■åŐŽéĄŞäź§åď łåď l'çIJ§äžĘçĆźãĂĆ
ä¡IJäÿžäÿĂäÿłçIJ§åőđçŽĎä¿Ńå■ŘïijŇåIJÍåď Žçž£çÍŃçŽĎç¡ŚçżIJçijŰçÍŃäÿ■çěđçğŸçŽĎ
`stalls` åŘŕèČ¡æŸŕåŻääÿžåĚűäżŰåŐ§åŻäæŕŤåęĆäÿĂäÿłDNSæ§ěæL'¿åźűæŮűïijŇèĂŇèů§GILæŕńæŮäåĚş
æIJĂåŘŐä¡äçIJ§çŽĎéIJĂèęĄåĚĹåŐżæŘđæĞĆä¡äçŽĎäżčçăĄæŸŕåŘęçIJ§çŽĎèćńGILå¡śåŞ■åĹŕãĂĆ
åŘŇæŮűè£ŸèęĄæŸŐçŹ¡GILåď ğéČĺåĹĘéČ¡åžŤèŕěåŘłåĚşæşÍCPUçŽĎåď ĎçŘĘèĂŇäÿ■æŸŕI/O.

åęĆæđIJä¡ääĞĘåď Ğä¡£çŤÍäÿĂäÿłåď ĎçŘĘåŹÍæśäïijŇæşÍæĎŘçŽĎæŸŕè£Źæăůå ĄŽæűĽ'åŘŁåĹŕæŢřæ■ĉ
èćńæL'ğèąŇçŽĎæŞ■ä¡IJéIJĂèęĄæŤ¿åIJÍäÿĂäÿłéĂŽè£Ğdefèŕ■åŘěåőŽäźL'çŽĎPythonåĞ¡æŢřäÿ■ïijŇäÿ■èČ¡

åźűäÿŤåĞ¡æŢřåŘĆæŢřåŠŇè£ŤåŻđåĂijå£ĚéążèęĄåĚijåőźpickleãĂĆ
åŘŇæăůïijŇèęĄæĽğèąŇçŽĎäżżåŁąéĞŘå£Ěéążèűşåď§åď ğäżěåijěèąěéćÍåď ŰçŽĎéĂŽä£ąåijĂéŤĂãĂĆ

åŘęåď ŰäÿĂäÿłéŽ¿çĆźæŸŕå¡Şæůůå ŘĹä¡£çŤÍçž£çÍŃåŠŇè£ŻçÍŃæśăçŽĎæŮűůåĂŹäijŽèőĺ ä¡ăå¿Ĺåď ť çŰi
åęĆæđIJä¡ăèęĄåŘŇæŮűä¡£çŤÍäÿď èĂĚïijŇæIJĂåě¡åIJÍçÍŃåžŘåŘŕåŁÍæŮűïijŇåĹŻåżżäżżä¡Ţçž£çÍŃäźŃåĽ■å
çĎűåŘŐçž£çÍŃä¡£çŤÍåŘŇæăůçŽĎè£ŻçÍŃæśăæÍěè£ŻèąŇåőČäżňçŽĎèőąçőŮåŕĘéŻĘąđŃåůěä¡IJãĂĆ

CæĽ ľ åśŢæIJĂéĞ■èęĄçŽĎçĽźå¿ĄæŸŕåőČäżňåŠŇPythonèğčéĞŁåŹÍæŸŕä£ÍæŇĄçŃňçńŃçŽĎãĂĆ
äź§åŕśæŸŕèŕť ïijŇåęĆæđIJä¡ăăĞĘąđ ĞåŕĘPythonäÿ■çŽĎäżżåŁąåĹĘéĚ■åĹřCäÿ■åŐżæĽ ğèąŇïijŇ
ä¡ăéIJĂèęĄçąőä£ÍCäżčçăĄçŽĎæŞ■ä¡IJèů§Pythonä£ÍæŇĄçŃňçńŃïijŇ
è£ŹåŕśæĎŘåŚşçÍĂäÿ■èęĄä¡£çŤÍPythonæŢŕæ■őçżŞæđĎäżěåŘŁäÿ■èęĄěŕĆçŤÍPythonçŽĎC
APIãĂĆ åŘęåď ŰäÿĂäÿłåŕśæŸŕä¡ăèęĄçąőä£ÍCæĽ ľ åśŢæĽ ĂåĄŽçŽĎåůěä¡IJæŸŕèŭşåď §çŽĎïijŇåĂijå¿Ůä¡ăè
äź§åŕśæŸŕèŕť CæĽ ľ åśŢæŃĚèť §èţůäžĘąđ ğéĞŘçŽĎèőąçőŮäżżåŁąïijŇèĂŇäÿ■æŸŕåŕŚæŢŕåĞääÿłèőąçőŮãĂĆ

è£ŹäžŻèğčåĘşGILçŽĎæŰźæąĹäźűäÿ■èČ¡éĂĆçŤÍäžŐæĽ ĂæIJĽéŮőécŸãĂĆ
ä¿ŃåęĆïijŇæ§ŘäžŻçśżåđŃçŽĎåžŤçŤÍçÍŃåžŘåęĆæđIJèćńåĹĘèğčäÿžåď Žäÿłè£ŻçÍŃåď ĎçŘĘçŽĎèŕÍäźűäÿ■è
äź§äÿ■èČ¡åŕĘąőČçŽĎéČÍåĹĘäžčçăĄæŤźæĹŘCèŕ■èÍĂæĽ ğèąŇãĂĆ
åŕźäžŐè£ŹäžŻåžŤçŤÍçÍŃåžŘïijŇä¡ăåŕśèęĄèĞłåůśéIJĂæśĆèğčåĘşæŰźæąĹäžĘ
ïijĹæŕŤåęĆåď Žè£ŻçÍŃèő£éŮőåĚśäžńåĘĚå■ŸåŇžïijŇåď ŽèğčæđŘåźÍè£ŘèąŇäžŐåŘŇäÿĂäÿłè£ŻçÍŃç■L ïijL
æĹŰèĂĚïijŇä¡ăè£ŸåŘŕäżěèĂČèŹŚäÿŃåĚűäżŰçŽĎèğčéĞŁåźÍłőđçŐŕïijŇæŕŤåęĆPyPyãĂĆ

äžĘèğčæŻť åď ŽåĚşäžŐåIJÍCæĽ ľ åśŢäÿ■éĞŁæŤ¿GILïijŇèŕůåŘĆèĂČ15.7åŠŇ15.10åŕŘèŁĆãĂĆ

14.10 12.10 åőŽäźĽäÿĂäÿłActoräżżåŁą

éŮőécŸ

ä¡ăæČşåőŽäźĽèů§actoræĺąåijŘäÿ■çśżäijijâĂIJactorsâĂIèğŠèĽšçŽĎäżżåŁą

èğčåĘşæŰźæąĹ

actoræĺąåijŘæŸŕäÿĂçğ■æIJĂåŘď èĂĄçŽĎäź§æŸŕæIJĂçőĂå■ŢçŽĎåźűèąŇåŠŇåĹĘåÿČåijŘèőąçőŮèğčä
äžŇåőđäÿŁïijŇåőČåď ľ çŤ§çŽĎçőĂå■ŢæĂğæŸŕåőČåęĆæ■ď åŘŮæňćè£ŐçŽĎéĞ■èęĄåŐ§åŻäźŃäÿĂãĂĆ
çőĂå■ŢæÍěèőšïijŇäÿĂäÿłactoråŕśæŸŕäÿĂäÿłåźűåŘŚæĽ ğèąŇçŽĎäżżåŁąïijŇåŘłæŸŕçőĂå■ŢçŽĎæĽ ğèąŇåŘ
åŞ■åžŤè£ŹäžŻæűĹæĄŕæŮűïijŇåőČåŘŕèČ¡è£ŸäijŽçżŹåĚűäżŰactoråŘŚéĂĄæŻť è£ŻäÿĂæ■ěçŽĎæűĹæĄŕã
actoräźŃéŮť çŽĎéĂŽä£ąæŸŕå■ŢåŘŚåŠŇåijĆæ■ěçŽĎãĂĆåŻäæ■ď ïijŇæűĹæĄŕåŘŚéĂĄèĂĚäÿ■ç§ěéĄŞæűĹ
äź§äÿ■äijŽæŐěæŤűåĹřäÿĂäÿłæűĹæĄŕåůšèćńåď ĎçŘĘçŽĎåŻđåžŤæĹŰéĂŽç§ěãĂĆ

çżŞåŘĹä¡£çŤÍäÿĂäÿłçž£çÍŃåŠŇäÿĂäÿłéŸ§åĹŮåŘŕäżěå¿Ĺåőźæ ŸŞçŽĎåőŽäźĽactorïijŇä¿ŃåęĆïijŽ

```
from queue import Queue
from threading import Thread, Event

# Sentinel used for shutdown
class ActorExit(Exception):
    pass

class Actor:
```

(continues on next page)

(continued from previous page)

```
    def __init__(self):
        self._mailbox = Queue()

    def send(self, msg):
        '''
        Send a message to the actor
        '''
        self._mailbox.put(msg)

    def recv(self):
        '''
        Receive an incoming message
        '''
        msg = self._mailbox.get()
        if msg is ActorExit:
            raise ActorExit()
        return msg

    def close(self):
        '''
        Close the actor, thus shutting it down
        '''
        self.send(ActorExit)

    def start(self):
        '''
        Start concurrent execution
        '''
        self._terminated = Event()
        t = Thread(target=self._bootstrap)

        t.daemon = True
        t.start()

    def _bootstrap(self):
        try:
            self.run()
        except ActorExit:
            pass
        finally:
            self._terminated.set()

    def join(self):
        self._terminated.wait()

    def run(self):
        '''
        Run method to be implemented by the user
        '''
```

(continues on next page)

(continued from previous page)

```
        while True:
            msg = self.recv()

# Sample ActorTask
class PrintActor(Actor):
    def run(self):
        while True:
            msg = self.recv()
            print('Got:', msg)

# Sample use
p = PrintActor()
p.start()
p.send('Hello')
p.send('World')
p.close()
p.join()
```

è£Źäÿłä¿Ńå■Řäÿ■ïijŇä¡ăä¡£çŤĺactoråőđä¿ŃçŽĎ `send()` æŰźæşŢåŘŚéĂĄæűĹæĄŕçżŹåőČäżňăĂĆ åĚűæIJžåĹűæŸŕïijŇè£ŹäÿłæŰźæşŢäijŽåřĘæűĹæĄŕæŤ¿åĚěäÿĂäÿçĎűåŘŐåřĘåĚűě¡ňäžď'çżŹåď'ĎçŘĘèćńæŐěåŘŮæűĹæĄŕçŽĎäÿĂäÿłåĘĚéČĺçž£çĺŃãĂĆ `close()` æŰźæşŢéĂŽè£ĞåIJĺéŸ§åĹŮäÿ■æŤ¿åĚěäÿĂäÿłçĽźæőŁçŽĎåŞĺåĚţåĂijïijĹActorExitïijĽæĺěåĚşéÜçŤĺæĹůåŘŕäżěéĂŽè£ĞçżğæĽ£ActoråźűåőŽäźĽåőđçŐřèĞłåůšåď'ĎçŘĘéĂżè¿Śrun()æŰźæşŢæĺěåőŽäźĽæŰřĺ `ActorExit` åijĆåÿÿçŽĎä¡£çŤĺåřśæŸŕçŤĺæĹůèĞłåőŽäźĽäžčçăĄåŘŕäżěåIJĺéIJĂèęĄçŽĎæŮűåĂŹæĺěæ■ŢèĆïijĹåijĆåÿÿèćńget()æŰźæşŢæŁŻåĞžåźűäijăæŠ■åĞžåŐżïijĽãĂĆ

åęĆæđIJä¡ăæŤ¿åő¡åŕźäžŐåŘŇæ■ěåŠŇåijĆæ■ěæűĹæĄŕåŚéĂĄçŽĎèęĄæśĆïijŇ çśżactoråŕźèśąè£ŸåŘŕäżěéĂŽè£ĞçŤ§æĹŘåŹĺæĺěçőĂåŇŰåőŽäźĽãĂĆä¿ŃåęĆïijŽ

```
def print_actor():
    while True:

        try:
            msg = yield       # Get a message
            print('Got:', msg)
        except GeneratorExit:
            print('Actor terminating')

# Sample use
p = print_actor()
next(p)      # Advance to the yield (ready to receive)
p.send('Hello')
p.send('World')
p.close()
```

èőÍèőž

actoræÍąåijŘçŽĎé■ĚåŁŻåřśåIJÍäžŐåőČçŽĎçőĂå■ŢæĂğãĂĆ
åőđéŹĚäÿŁïijŇè£ŹéĞŇäżĚäżĚåŘłæIJL'äÿĂäÿłæăÿå£ČæŞ■ä¡IJ `send()` .
çŤŽèĞşïijŇåŕźäžŐåIJÍå§žäžŐactorçşżçż§äÿ■çŽĎâĂIJæűĹæĄŕâĂİçŽĎæşŻåŇŰæęĆå£ţåŘŕäżěåůšåď'Žçğ■æŰ
ä¿ŃåęĆïijŇä¡ăåŘŕäżěäżěåĚČçżĎå¡ćåijŘäijăéĂŠæăĞç■¿æűĹæĄŕïijŇèőľ'actoræL'ğèąŇäÿ■åŘŇçŽĎæŞ■ä¡IJïij

```
class TaggedActor(Actor):
    def run(self):
        while True:
            tag, *payload = self.recv()
            getattr(self,'do_'+tag)(*payload)

    # Methods correponding to different message tags
    def do_A(self, x):
        print('Running A', x)

    def do_B(self, x, y):
        print('Running B', x, y)

# Example
a = TaggedActor()
a.start()
a.send(('A', 1))       # Invokes do_A(1)
a.send(('B', 2, 3))    # Invokes do_B(2,3)
```

ä¡IJäÿžåŘęåď'ŰäÿĂäÿłä¿Ńå■ŘïijŇäÿŃéİćçŽĎactoråĚĄèőÿåIJÍäÿĂäÿłåůëä¡IJèĂĚäÿ■è£ŘèąŇäżżæĎŘçŽ
åźűäÿŤéĂŽè£ĞäÿĂäÿłçL'źæőŁçŽĎResultåŕźèśąè£ŤåŻđçżŞæđIJïijŽ

```
from threading import Event
class Result:
    def __init__(self):
        self._evt = Event()
        self._result = None

    def set_result(self, value):
        self._result = value

        self._evt.set()

    def result(self):
        self._evt.wait()
        return self._result

class Worker(Actor):
    def submit(self, func, *args, **kwargs):
        r = Result()
        self.send((func, args, kwargs, r))
        return r
```

(continues on next page)

(continued from previous page)

```
    def run(self):
        while True:
            func, args, kwargs, r = self.recv()
            r.set_result(func(*args, **kwargs))

# Example use
worker = Worker()
worker.start()
r = worker.submit(pow, 2, 3)
print(r.result())
```

æIJĂåŘŐïijŇâĂIJåŘŚéĂĄâĂİäÿĂäÿłäżżåŁąæűĹæĄŕçŽĎæęĆå£ţåŘŕäżěèćńæĽľåśŢåĹřåďŽè£ŻçĺŃçŤŽ
ä¿ŃåęĆïijŇäÿĂäÿłçśżactoråŕźèśąçŽĎ `send()` æŰźæşŢåŘŕäżěèćńçijŰçĺŃèőľåőČèČ¡åIJĺäÿĂäÿłåěŮæŐěå■Ů
æĹŰéĂŽè£Ğæ§ŘäžŻæűĹæĄŕäÿ■éŮťäżűïijĹæŕŤåęĆAMQPãĂĄZMQç■Ľ'ïijĽ'æİěåŘŚéĂĄãĂĆ

14.11 12.11 åőđçŐřæűĹæĄŕåŘŚåÿČ/èőćéŸĚæĺąåđŃ

éŮőéćŸ

ä¡äæIJĽäÿĂäÿłå§žäžŐçž£çĺŃéĂŽä£ąçŽĎçĺŃåžŘïijŇæČşèőľåőČäżňåőđçŐřåŘŚåÿČ/èőćéŸĚæĺąåijŘçŽĎ

èğčåĘşæŰźæąĹ

èęĄåőđçŐřåŘŚåÿČ/èőćéŸĚçŽĎæűĹæĄŕéĂŽä£ąæĺąåijŘïijŇ
ä¡äéĂŽåÿÿèęĄåijŢåĚěäÿĂäÿłå■ŢçŃňçŽĎâĂIJäžď'æ■ćæIJžâĂİæĹŰâĂIJç¡ŚåĚşâĂİåŕźèśąä¡IJäÿžæĽ'ĂæIJĽ'æ
äź§åŕśæŸŕèŕť'ïijŇäÿ■çŻť'æŐěåřĘæűĹæĄŕäżŐäÿĂäÿłäżżåŁąåŘŚéĂĄåĹřåŘęäÿĂäÿłïijŇèĂŇæŸŕåřĘåĚűåŘŚé
çĎűåŘŐçŤśäžď'æ■ćæIJžåřĘåőČåŘŚéĂĄçżŹäÿĂäÿłæĹŰåď'ŽäÿłèćńåĚşèĄŤäżżåŁąãĂĆäÿŃéİćæŸŕäÿĂäÿłéİđ

```
from collections import defaultdict

class Exchange:
    def __init__(self):
        self._subscribers = set()

    def attach(self, task):
        self._subscribers.add(task)

    def detach(self, task):
        self._subscribers.remove(task)

    def send(self, msg):
        for subscriber in self._subscribers:
            subscriber.send(msg)

# Dictionary of all created exchanges
_exchanges = defaultdict(Exchange)
```

(continues on next page)

(continued from previous page)

```
# Return the Exchange instance associated with a given name
def get_exchange(name):
    return _exchanges[name]
```

äÿĂäÿłäžď æ■ćæIJžåřśæŸŕäÿĂäÿłæŹőéĂŽåŕźèśąïijŇèť §èť čçžť æŁď äÿĂäÿłæť žèůČçŽĎèőćéŸĚèĂĚéŽĚ
æŕŘäÿłäžď æ■ćæIJžéĂŽè£ĞäÿĂäÿłåŘ■çğřåőŽä¡■ïijŇget_exchange()
éĂŽè£ĞçżŹåőŽäÿĂäÿłåŘ■çğřè£ŤåŻđçŻÿåžŤçŽĎ Exchange åőđä¿ŃãĂĆ

äÿŃéĺćæŸŕäÿĂäÿłçőĂå■Ţä¿Ńå■ŘïijŇæijŤçď žäžĘåęĆä¡Ţä¡£çŤĺäÿĂäÿłäžď æ■ćæIJžïijŽ

```
# Example of a task.  Any object with a send() method

class Task:
    ...
    def send(self, msg):
        ...

task_a = Task()
task_b = Task()

# Example of getting an exchange
exc = get_exchange('name')

# Examples of subscribing tasks to it
exc.attach(task_a)
exc.attach(task_b)

# Example of sending messages
exc.send('msg1')
exc.send('msg2')

# Example of unsubscribing
exc.detach(task_a)
exc.detach(task_b)
```

ář¡çőąåŕźäžŐè£ŹäÿłéŮőéćŸæIJL'å¿Ĺåď ŽçŽĎåŘŸçğ■ïijŇäÿ■è£ĞäÿĞåŘŸäÿ■çężåĚűåőŮãĂĆ
æűĹæĄŕäijŽèćńåŘŚéĂĄçżŹäÿĂäÿłäžď æ■ćæIJžïijŇçĎűåŘŐäžď æ■ćæIJžäijŽåřĘåőČäžňåŘŚéĂĄçżŹèćńçżŚ

èőĺèőž

éĂŽè£ĞéŸ§åĹŮåŘŚéĂĄæűĹæĄŕçŽĎäżżåŁąæĹŰçž£çĺŃçŽĎæĺąåijŘå¿ĹåőźæŸ ŞèćńåőđçŐřåźűäÿŤäź§
äÿ■è£ĞïijŇä¡£çŤĺåŘŚåÿČ/èőćéŸĚæĺąåijŘçŽĎåě¡åď ĎæŽť åŁäæŸŐæŸ¿ãĂĆ

éęŰåĚĹïijŇä¡£çŤĺäÿĂäÿłäžď æ■ćæIJžåŘŕäżěçőĂåŃŰåď ğéČĺåĹĘæűL'åŘŁåĹřçž£çĺŃéĂŽä£ąçŽĎåůěä¡I
æŮäéIJĂåŐżåĘŹéĂŽè£Ĝåď Žè£ŻçĺŃæĺąåİŮæİěæŞ■ä¡IJåď Žäÿłçž£çĺŃïijŇä¡åŘłéIJĂèęĄä¡£çŤĺè£Źäÿłäžď æ
æ§Řçğ■çĺŃåžęäÿŁïijŇè£Źäÿłåřśèů§æŮěå£ŮæĺąåİŮçŽĎåůěä¡IJåŐ§çŘĘçśżäijijãĂĆ
åőđéŹĚäÿŁïijŇåőČåŘŕäżěè¡żæİ¿çŽĎèğčèĂęçĺŃåžŘäÿ■åď ŽäÿłäżżåŁąãĂĆ

åĚűæňąïijŇäžď æ■ćæIJžåź£æŠ■æűĹæĄŕçżŹåď ŽäÿłèőćéŸĚèĂĚçŽĎèČ¡åŁŻåÿęæİěäžĘäÿĂäÿłåĚĺæŰřçŽ
ä¿ŃåęĆïijŇä¡ăåŘŕäżěä¡£çŤĺåď ŽäżżåŁąçşżçż§ãĂĄåź£æŠ■æĹŰæĽĞåĞžãĂĆ
ä¡ăè£ŸåŘŕäżěéĂŽè£ĞäżěæŹőéĂŽèőćéŸĚèĂĚèžńäż¡çżŚåőŽæİěæđĎåżžèřČèŕŢåŠŇèŕŁæŰ■åůěåĚůãĂĆ
ä¿ŃåęĆïijŇäÿŃéİćæŸŕäÿĂäÿłçőĂå■ŢçŽĎèŕŁæŰ■çśżïijŇåŘŕäżěæŸ¿çď žèćńåŘŚéĂĄçŽĎæűĹæĄŕïijŽ

```
class DisplayMessages:
    def __init__(self):
        self.count = 0
    def send(self, msg):
        self.count += 1
        print('msg[{}]: {!r}'.format(self.count, msg))

exc = get_exchange('name')
d = DisplayMessages()
exc.attach(d)
```

æIJĂåŘŐïijŇèŕěåőđçŐřçŽĎäÿĂäÿłéĞ■èęĄçĽźçĆźæŸŕåőČèČ¡åĚijåőźåď ŽäÿłâĂIJtask-likeâĂİåŕźèśąãĂĆ ä¿ŃåęĆïijŇæűĹæĄŕæŐěåŘŮèĂĚåŘŕäżěæŸŕactorïijĹ12.10åřŘèŁĆäżŃçż■ïijĽãĂĄå■ŘçĺŃ `send()` æŰźæşŢçŽĎäÿIJèě£ãĂĆ

åĚşäžŐäžď æ■ćæIJžçŽĎäÿĂäÿłåŘŕèČ¡éŮőéćŸæŸŕåŕźäžŐèőćéŸĚèĂĚçŽĎæ■čçąőçżŚåőŽåŠŇèğčçżŚãĂ
äÿžäžĘæ■čçąőçŽĎçőąçŘĘèţĎæžŘïijŇæŕŘäÿĂäÿłçżŚåőŽçŽĎèőćéŸĚèĂĚå£ĚéążæIJĂçżĹèęĄèğčçżŚãĂĆ
åIJĺäżčçăĄäÿ■éĂŽåÿÿäijŽæŸŕåČŘäÿŃéİćè£ŹæăůçŽĎæĺąåijŘïijŽ

```
exc = get_exchange('name')
exc.attach(some_task)
try:
    ...
finally:
    exc.detach(some_task)
```

æ§Řçğ■æĎŘäźĽäÿŁïijŇè£ŹäÿłåŠŇä¡£çŤĺæŰĞäżűãĂĄéŤĄåŠŇçśżäijijåŕźèśąå¿ĹåČŘãĂĆ
éĂŽåÿÿå¿Ĺåőźæ ŸŞäijŽå£ŸèőŕæIJĂåŘŐçŽĎ `detach()` æ■ěéłď ãĂĆ
äÿžäžĘçőĂåŃŰè£ŹäÿłïijŇä¡ăåŘŕäżěèĂČèŹŚä¡£çŤĺäÿŁäÿŃæŰĞçőąçŘĘåŹĺå■ŘèőőãĂĆ
ä¿ŃåęĆïijŇåIJĺäžď æ■ćæIJžåŕźèśąäÿŁåćđåŁääÿĂäÿł `subscribe()`
æŰźæşŢïijŇåęĆäÿŃïijŽ

```
from contextlib import contextmanager
from collections import defaultdict

class Exchange:
    def __init__(self):
        self._subscribers = set()

    def attach(self, task):
        self._subscribers.add(task)

    def detach(self, task):
        self._subscribers.remove(task)
```

(continues on next page)

(continued from previous page)

```
    @contextmanager
    def subscribe(self, *tasks):
        for task in tasks:
            self.attach(task)
        try:
            yield
        finally:
            for task in tasks:
                self.detach(task)

    def send(self, msg):
        for subscriber in self._subscribers:
            subscriber.send(msg)

# Dictionary of all created exchanges
_exchanges = defaultdict(Exchange)

# Return the Exchange instance associated with a given name
def get_exchange(name):
    return _exchanges[name]

# Example of using the subscribe() method
exc = get_exchange('name')
with exc.subscribe(task_a, task_b):
     ...
     exc.send('msg1')
     exc.send('msg2')
     ...

# task_a and task_b detached here
```

æIJĂåŘŐè£ŸåžŤèŕĕæşĺæĎŘçŽĎæŸŕåĚşäžŐäžď æ■ćæIJžçŽĎæĂİæČşæIJĽå¿Ĺåď Žçğ■çŽĎæĽľåśŢåőđ
ä¿ŃåęĆïijŇäžď æ■ćæIJžåŘŕäżĕåőđçŐřäÿĂæŢť äÿłæűĹæĄŕéĂŽéĄşéŻĘåŘĹæĹŰæŘŘä¿Żäžď æ■ćæIJžåŘ■çğ
äžď æ■ćæIJžè£ŸåŘŕäżĕèćńæĽľåśŢåĹŕåĹĘåÿČåijŘèőąçőŮçĺŃåžŘäÿ■ïijĹæŕŤåęĆïijŇåřĘæűĹæĄŕèůŕçŤśåĹŕä

14.12 12.12 ä¡£çŤĺçŤ§æĹŘåŹĺäżčæŻ£çž£çĺŃ

éŮőécŸ

ä¡ăæČşä¡£çŤĺçŤ§æĹŘåŹĺïijĹå■ŘçĺŃïijĽæŻ£äżčçşżçż§çž£çĺŃæİĕåőđçŐřåźůåŘŚãĂĆè£ŹäÿłæIJĽæŮűůåŘ

èğčåĘşæŰźæąĹ

èęĄä¡£çŤĺçŤ§æĹŘåŹĺåőđçŐřèĞłåůśçŽĎåźűåŘŚïijŇä¡ăéęŰåĚĹèęĄåŕźçŤ§æĹŘåŹĺåĞ¡æŢřåŠŇ
`yield` èŕ■åŘĕæIJĽæůśåĹżçŘĘèğčãĂĆ `yield` èŕ■åŘĕäijŽèőľäÿĂäÿłçŤ§æĹŘåŹĺæŃĆèţůåőČçŽĎæĽğèąŃïi
åřĘçŤ§æĹŘåŹĺå¡ŞåĄŽæ§Řçğ■âĂIJäżżåŁąâĂİåźűä¡£çŤĺäżżåŁąå■Ŕä¡IJåĹĞæ■ćæİĕæŻ£æ■ćåőČäżňçŽĎæĽğ

èęĄæijŤçď'žè£Źçğ■æĂİæČşïijŇèĂČèŹŚäÿŃéİćäÿď'äÿłä¡£çŤÍçőĂå■ŢçŽĎ `yield` èŕ■åŘěçŽĎçŤ§æĹŘåŹÍåĞ¡æŢřïijŽ

```
# Two simple generator functions
def countdown(n):
    while n > 0:
        print('T-minus', n)
        yield
        n -= 1
    print('Blastoff!')

def countup(n):
    x = 0
    while x < n:
        print('Counting up', x)
        yield
        x += 1
```

è£ŹäžŻåĞ¡æŢřåIJÍåĘĚéČÍä¡£çŤÍyieldèŕ■åŘěïijŇäÿŃéİćæŸŕäÿĂäÿłåőđçŐřäžĘçőĂå■ŢäżżåŁąèřČåžęåŹÍ

```
from collections import deque

class TaskScheduler:
    def __init__(self):
        self._task_queue = deque()

    def new_task(self, task):
        '''
        Admit a newly started task to the scheduler

        '''
        self._task_queue.append(task)

    def run(self):
        '''
        Run until there are no more tasks
        '''
        while self._task_queue:
            task = self._task_queue.popleft()
            try:
                # Run until the next yield statement
                next(task)
                self._task_queue.append(task)
            except StopIteration:
                # Generator is no longer executing
                pass

# Example use
sched = TaskScheduler()
sched.new_task(countdown(10))
sched.new_task(countdown(5))
```

(continues on next page)

(continued from previous page)

```
sched.new_task(countup(15))
sched.run()
```

`TaskScheduler` çśżåIJÍäÿĂäÿłå¿łçŐŕäÿ■è£ŘèąŇçŤ§æĹŘåŹÍéŻĘåŘĹâĂŤâĂŤæŕŔäÿłéČ¡è£ŘèąŇåĹřç
è£ŘèąŇè£ŹäÿłäÿŃå■ŘïijŇè¿ŞåĞžåęĆäÿŃïijŽ

```
T-minus 10
T-minus 5
Counting up 0
T-minus 9
T-minus 4
Counting up 1
T-minus 8
T-minus 3
Counting up 2
T-minus 7
T-minus 2
...
```

åĹřæ■d'äÿžæ■ćïijŇæĹŚäżňåőđéŹĚäÿŁåůšçżŘåőđçŐřäžĘäÿĂäÿłâĂIJæŞ■ä¡IJçşżçż§âĂİçŽĎæIJĂåřŘæäÿ
çŤ§æĹŘåŹÍåĞ¡æŢřåřśæŸŕèőď'äÿžïijŇèĂŇyieldèŕ■åŘĕæŸŕäżżåŁąæŇĆèţůçŽĎä£ąåŘůãĂĆ
èřČåžęåŹÍå¿łçŐŕæčĂæ§ĕäżżåŁąåĹŮèąÍçŻť'åĹřæşąæIJL'äżżåŁąèęĄæL'ğèąŇäÿžæ■ćãĂĆ

åőđéŹĚäÿŁïijŇä¡ăåŘŕèČ¡æČşèęĄä¡£çŤÍçŤ§æĹŘåŹÍæİĕåőđçŐřçőĂå■ŢçŽĎåźűåŘŚãĂĆ
éĆčäźĹïijŇåIJÍåőđçŐřactoræĹŰç¡ŚçżIJæIJ■åŁąåŹÍçŽĎæŮűåĂŹä¡ăåŘŕäžĕä¡£çŤÍçŤ§æĹŘåŹÍæİĕæŻ£äżčçž£ç

äÿŃéİćçŽĎäżčçăĄæijŤçď'žäžĘä¡£çŤÍçŤ§æĹŘåŹÍæİĕåőđçŐřäÿĂäÿłäÿ■ä¿İèţŰçž£çÍŃçŽĎactorïijŽ

```
from collections import deque

class ActorScheduler:
    def __init__(self):
        self._actors = { }          # Mapping of names to actors
        self._msg_queue = deque()   # Message queue

    def new_actor(self, name, actor):
        '''
        Admit a newly started actor to the scheduler and give it a␣
↪name
        '''
        self._msg_queue.append((actor,None))
        self._actors[name] = actor

    def send(self, name, msg):
        '''
        Send a message to a named actor
        '''
        actor = self._actors.get(name)
        if actor:
            self._msg_queue.append((actor,msg))
```

(continues on next page)

(continued from previous page)

```
    def run(self):
        '''
        Run as long as there are pending messages.
        '''
        while self._msg_queue:
            actor, msg = self._msg_queue.popleft()
            try:
                 actor.send(msg)
            except StopIteration:
                 pass

# Example use
if __name__ == '__main__':
    def printer():
        while True:
            msg = yield
            print('Got:', msg)

    def counter(sched):
        while True:
            # Receive the current count
            n = yield
            if n == 0:
                break
            # Send to the printer task
            sched.send('printer', n)
            # Send the next count to the counter task (recursive)

            sched.send('counter', n-1)

    sched = ActorScheduler()
    # Create the initial actors
    sched.new_actor('printer', printer())
    sched.new_actor('counter', counter(sched))

    # Send an initial message to the counter to initiate
    sched.send('counter', 10000)
    sched.run()
```

åőŇåĚÍåijĎæĞĆè£ŹæőţäżčçăĄéIJĂèęĄæŻťæůśåĚěçŽĎå■ęäźäïijŇä¡ĘæŸŕåĚşéŤőçĆźåIJÍäžŐæŤűéŻĘæ
æIJňèťÍäÿŁïijŇèřČåžęåŹÍåIJÍæIJĽéIJĂèęĄåŘŚéĂĄçŽĎæűĹæĄŕæŮűäijŽäÿĂçŻťè£ŘèąŇçİĂãĂĆ
èőąæŢřçŤ§æĹŘåŹÍäijŽçżŹèĞłåůśåŘŚéĂĄæűĹæĄŕåźűåIJÍäÿĂäÿłéĂŠå¡Šå¿łçŐŕäÿ■çżŞæİ§ãĂĆ

äÿŃéİćæŸŕäÿĂäÿłæŻťåŁäéńŸçžğçŽĎä¿Ńå■ŘïijŇæijŤçďžäžĘä¡£çŤÍçŤ§æĹŘåŹÍæİěåőđçŐŕäÿĂäÿłäźűå

```
from collections import deque
from select import select
```

(continues on next page)

(continued from previous page)

```
# This class represents a generic yield event in the scheduler
class YieldEvent:
    def handle_yield(self, sched, task):
        pass
    def handle_resume(self, sched, task):
        pass

# Task Scheduler
class Scheduler:
    def __init__(self):
        self._numtasks = 0        # Total num of tasks
        self._ready = deque()     # Tasks ready to run
        self._read_waiting = {}   # Tasks waiting to read
        self._write_waiting = {} # Tasks waiting to write

    # Poll for I/O events and restart waiting tasks
    def _iopoll(self):
        rset,wset,eset = select(self._read_waiting,
                                self._write_waiting,[])
        for r in rset:
            evt, task = self._read_waiting.pop(r)
            evt.handle_resume(self, task)
        for w in wset:
            evt, task = self._write_waiting.pop(w)
            evt.handle_resume(self, task)

    def new(self,task):
        '''
        Add a newly started task to the scheduler
        '''

        self._ready.append((task, None))
        self._numtasks += 1

    def add_ready(self, task, msg=None):
        '''
        Append an already started task to the ready queue.
        msg is what to send into the task when it resumes.
        '''
        self._ready.append((task, msg))

    # Add a task to the reading set
    def _read_wait(self, fileno, evt, task):
        self._read_waiting[fileno] = (evt, task)

    # Add a task to the write set
    def _write_wait(self, fileno, evt, task):
        self._write_waiting[fileno] = (evt, task)
```

(continues on next page)

(continued from previous page)

```
    def run(self):
        '''
        Run the task scheduler until there are no tasks
        '''
        while self._numtasks:
             if not self._ready:
                  self._iopoll()
             task, msg = self._ready.popleft()
             try:
                 # Run the coroutine to the next yield
                 r = task.send(msg)
                 if isinstance(r, YieldEvent):
                     r.handle_yield(self, task)
                 else:
                     raise RuntimeError('unrecognized yield event')
             except StopIteration:
                 self._numtasks -= 1

# Example implementation of coroutine-based socket I/O
class ReadSocket(YieldEvent):
    def __init__(self, sock, nbytes):
        self.sock = sock
        self.nbytes = nbytes
    def handle_yield(self, sched, task):
        sched._read_wait(self.sock.fileno(), self, task)
    def handle_resume(self, sched, task):
        data = self.sock.recv(self.nbytes)
        sched.add_ready(task, data)

class WriteSocket(YieldEvent):
    def __init__(self, sock, data):
        self.sock = sock
        self.data = data
    def handle_yield(self, sched, task):

        sched._write_wait(self.sock.fileno(), self, task)
    def handle_resume(self, sched, task):
        nsent = self.sock.send(self.data)
        sched.add_ready(task, nsent)

class AcceptSocket(YieldEvent):
    def __init__(self, sock):
        self.sock = sock
    def handle_yield(self, sched, task):
        sched._read_wait(self.sock.fileno(), self, task)
    def handle_resume(self, sched, task):
        r = self.sock.accept()
        sched.add_ready(task, r)
```

(continues on next page)

(continued from previous page)

```
# Wrapper around a socket object for use with yield
class Socket(object):
    def __init__(self, sock):
        self._sock = sock
    def recv(self, maxbytes):
        return ReadSocket(self._sock, maxbytes)
    def send(self, data):
        return WriteSocket(self._sock, data)
    def accept(self):
        return AcceptSocket(self._sock)
    def __getattr__(self, name):
        return getattr(self._sock, name)

if __name__ == '__main__':
    from socket import socket, AF_INET, SOCK_STREAM
    import time

    # Example of a function involving generators.  This should
    # be called using line = yield from readline(sock)
    def readline(sock):
        chars = []
        while True:
            c = yield sock.recv(1)
            if not c:
                break
            chars.append(c)
            if c == b'\n':
                break
        return b''.join(chars)

    # Echo server using generators
    class EchoServer:
        def __init__(self,addr,sched):
            self.sched = sched
            sched.new(self.server_loop(addr))

        def server_loop(self,addr):
            s = Socket(socket(AF_INET,SOCK_STREAM))

            s.bind(addr)
            s.listen(5)
            while True:
                c,a = yield s.accept()
                print('Got connection from ', a)
                self.sched.new(self.client_handler(Socket(c)))

        def client_handler(self,client):
            while True:
                line = yield from readline(client)
```

(continues on next page)

(continued from previous page)

```
                if not line:
                    break
                line = b'GOT:' + line
                while line:
                    nsent = yield client.send(line)
                    line = line[nsent:]
            client.close()
            print('Client closed')

    sched = Scheduler()
    EchoServer(('',16000),sched)
    sched.run()
```

è£Źæőţäžčçăąæ IJL'çĆźåď■æİĆãĂĆäÿ■è£ĞïijŇåőČåőđçŐřäžĘäÿĂäÿłåřŔåđŃçŽĎæŞ■ä¡IJçşżçż§ãĂĆ
æIJL'äÿĂäÿłåřśçżłçŽĎäżżåŁąéŸ§åĹŮïijŇåźűäÿŤè£ŸæIJL'åŽăI/OäijŚçIJăçŽĎäżżåŁąç■L'å¿ĚåŇžå§§ãĂĆ
è£ŸæIJL'å¿Ĺåď'ŽěřČåžęåŹÍèť§èť'čåIJÍåřśçżłéŸ§åĹŮåŠŇI/Oç■L'å¿ĚåŇžå§§äźŃéŮť'çğżåŁÍäżżåŁąãĂĆ

èőÍèőž

åIJÍæđĎåżżå§žäžŐçŤ§æĹŘåŹÍçŽĎåźűåŔŚæąĘæđűæŮűïijŇéĂŽåÿÿäijŽä¡£çŤÍæŻť'åÿÿèğĄçŽĎyieldå¡ćå

```
def some_generator():
    ...
    result = yield data
    ...
```

ä¡£çŤÍè£Źçğ■å¡ćåijŘçŽĎyieldèŕ■åŘěçŽĎåĞ¡æŢřéĂŽåÿÿèćńçğřäÿžâĂIJå■ŘçÍŃâĂİãĂĆ
éĂŽè£ĞèřČåžęåŹÍïijŇyieldèŕ■åŘěåIJÍäÿĂäÿłå¿łçŐŕäÿ■èćńåď'ĎçŘĘïijŇåęĆäÿŃïijŽ

```
f = some_generator()

# Initial result. Is None to start since nothing has been computed
result = None
while True:
    try:
        data = f.send(result)
        result = ... do some calculation ...
    except StopIteration:
        break
```

è£ŹéĞŇçŽĎéĂżè¿ŚçÍ■å¿őæIJL'çĆźåď'■æİĆãĂĆäÿ■è£ĞïijŇèćńäijăçżŹ
`send()` çŽĎåĂijåőŽäźL'äžĘåIJÍyieldèŕ■åŘěéĘŠæİěæŮűçŽĎè£ŤåŻđåĂijãĂĆ
åŻăæ■d'ïijŇåęĆæđIJäÿĂäÿłyieldåĞĘåď'ĞåIJÍåŕźäźŃåL'■yield-
æŢřæ■őçŽĎåŻđåžŤäÿ■è£ŤåŻđçżŞæđIJæŮűïijŇäijŽåIJÍäÿŃäÿĂæňą `send()`
æŞ■ä¡IJè£ŤåŻđãĂĆ åęĆæđIJäÿĂäÿłçŤ§æĹŘåŹÍåĞ¡æŢřåĹŽåijĂåğŃè£ŘèąŇïijŇåŘŚéĂĄäÿĂäÿłNoneåĂijäij

éŹď'äžĘåŘŚéĂĄåĂijåď'ŰïijŇè£ŸåŘŕäżěåIJÍäÿĂäÿłçŤ§æĹŘåŹÍäÿŁéİćæL'ğèąŇäÿĂäÿł
`close()` æŰźæşŢãĂĆ åőČäijŽåŕijèĞť'åIJÍæL'ğèąŇyieldèŕ■åŘěæŮűæŁŻåĞžäÿĂäÿł
`GeneratorExit` åijĆåÿÿïijŇäżŐèĂŇçżĹæ■ćæL'ğèąŇãĂĆ

åęĆæđIJè£ŻäÿĂæ■ěèő¿èőąïijŇäÿĂäÿłçŤ§æĹŘåŹĺåŘŕäżěæ■ŢèŐůè£ŹäÿłåijĆåÿÿåźűæĽğèąŇæÿĚçŘĘæŞ■ä¡I
åŘŇæăůè£ŸåŘŕäżěä¡£çŤĺçŤ§æĹŘåŹĺçŽĎ `throw()` æŰźæşŢåIJÍyield-
èŕ■åŘěæĽğèąŇæŮűçŤ§æĹŘäÿĂäÿłäżżæĎŘçŽĎæĽğèąŇæŇĞäżd'ãĂĆ
äÿĂäÿłäżżåŁąèřČåžęåŹĺåŘŕäĹ'çŤĺåőČæĺěåIJÍè£ŘèąŇçŽĎçŤ§æĹŘåŹĺäÿ■åd'ĎçŘĘéŤŹèŕŕãĂĆ

æIJĂåŘŐäÿĂäÿłä¿Ňå■Řäÿ■ä¡£çŤĺçŽĎ `yield from` èŕ■åŘěèćńçŤĺæĺěåőđçŐřå■ŘçĺŃïijŇåŘŕäżěèćńåĚ
æIJňèt'ĺäÿŁåřśæŸŕåřĘæŐğåĹűæĺČéĂŘæŸŐçŽĎäijäè¿ŞçżŹæŰřçŽĎåĞ¡æŢřãĂĆ
äÿ■åČŘæŹőéĂŽçŽĎçŤ§æĹŘåŹĺïijŇäÿĂäÿłä¡£çŤĺ `yield from`
èćńèřČçŤĺçŽĎåĞ¡æŢřåŘŕäżěè£ŤåŽđäÿĂäÿłä¡IJäÿž `yield from`
èŕ■åŘěçżŞæđIJçŽĎåĂijãĂĆ åĚşäžŐ `yield from` çŽĎæŻt'åd'Žä£ąæĄŕåŘŕäżěåIJÍ PEP 380 äÿ■æĽ¿åĹřãĂĆ

æIJĂåŘŐïijŇåęĆæđIJä¡£çŤĺçŤ§æĹŘåŹĺçijŰçĺŃïijŇèęĄæŘŘéĘŠä¡ăçŽĎæŸŕåőČè£ŸæŸŕæIJL'å¿Ĺåd'Žç
çL'źåĹńæŸŕïijŇä¡åå¿Ůäÿ■åĹřäżżä¡Ţçž£çĺŃåŘŕäżěæŘŘä¿ŻçŽĎåě¡åd'ĎãĂĆä¿ŃåęĆïijŇåęĆæđIJä¡ăæĽğèąŇ
åőČäijŽåřĘæŢt'äÿłäżżåŁąæŇĆètůç§ěéĄŞæŞ■ä¡IJåőŇæĹŘãĂĆäÿžäžĘèğčåĘşè£ŹäÿłéŮőécŸïijŇ
ä¡ååŘłèČ¡éĂĹ'æŃl'åřĘæŞ■ä¡IJåğŤæt'¿çżŹåŘęåd'ŰäÿĂäÿłåŘŕäżěçŃňçńŃè£ŘèąŇçŽĎçž£çĺŃæĹŰè£ŻçĺŃãĂ
åŘęåd'ŰäÿĂäÿłéŹŘåĹűæŸŕåd'ğéČĺåĹĘPythonåžŞåźűäÿ■èČ¡å¿Ĺåě¡çŽĎåĚijåőźå§žäžŐçŤ§æĹŘåŹĺçŽĎçž£çĺ
åęĆæđIJä¡ăéĂĹ'æŃl'è£ŹäÿłæŰźæąĹïijŇä¡ăäijŽåŘŚçŐřä¡ăéIJĂèęĄèĞłåůśæŤźåĘŹå¿Ĺåd'ŽæăĞåĞĘåžŞåĞ¡æ
ä¡IJäÿžæIJňèŁĆæŘŘåĹřçŽĎå■ŘçĺŃåŠŇçŻÿåĚşæŁĂæIJŕçŽĎäÿĂäÿłå§žçąĂèČŃæŹŕïijŇåŘŕäżěæ§ěçIJŃ
PEP 342 åŠŇ âĂIJå■ŘçĺŃåŠŇåźűåŘŚçŽĎäÿĂéŮĺæIJL'èűčèŕ¿çĺŃâĂİ

PEP 3156 åŘŇæăůæIJL'äÿĂäÿłåĚşäžŐä¡£çŤĺå■ŘçĺŃçŽĎåijĆæ■ěI/OæĺąåđŃãĂĆ
çL'źåĹńçŽĎïijŇä¡ăäÿ■åŘŕèČ¡èĞłåůśåŐżåőđçŐřäÿĂäÿłåžŢåśĆçŽĎå■ŘçĺŃèřČåžęåŹĺãĂĆ
äÿ■è£ĞïijŇåĚşäžŐå■ŘçĺŃçŽĎæĂİæČşæŸŕå¿Ĺåd'ŽæţĄèąŇåžŞçŽĎå§žçąĂïijŇ åŇĚæŃň
gevent, greenlet, Stackless Python äżěåŘŁåĚűäżŰçśżäijijåůěçĺŃãĂĆ

14.13 12.13 åd'Žäÿłçž£çĺŃéŸ§åĹŮè¡őèŕć

éŮőécŸ

ä¡ăæIJL'äÿĂäÿłçž£çĺŃéŸ§åĹŮéŻĘåŘĹïijŇæČşäÿžåĹřæĺěçŽĎåĚČçt'äè¡őèŕćåőČäżňïijŇ
åŕśèů§ä¡ăäÿžäÿĂäÿłåőćæĹůçńŕèŕůæśĆåŐżè¡őèŕćäÿĂäÿłç¡ŚçżIJè£đæŐěéŻĘåŘĹçŽĎæŰźåijŘäÿĂæăůãĂĆ

èğčåĘşæŰźæąĹ

åŕźäžŐè¡őèŕćéŮőécŸçŽĎäÿĂäÿłåÿÿèğĄèğčåĘşæŰźæąĹäÿ■æIJL'äÿłå¿ĹåřŚæIJL'äžžç§ěéĄŞçŽĎæŁĂåů
æIJňèt'ĺäÿŁèőšåĚűæĂİæČşåřśæŸŕïijŽåŕźäžŐæŕŘäÿłä¡ăæČşèęĄè¡őèŕćçŽĎéŸ§åĹŮïijŇä¡ăåĹŽåžžäÿĂåŕźè£đa
çŽűåŘŐä¡ăåIJĺåĚűäÿ■äÿĂäÿłåěŮæŐěå■ŮäÿŁéĺćçijŰåĘźäžčçăĄæĺěæăĞèŕĘå■ŸåIJĺçŽĎæŢřæ■őïijŇ
åŘęåd'ŰäÿĂäÿłåěŮæŐěå■ŮèćńäijăçżŹ `select()` æĹŰçśżäijijçŽĎäÿĂäÿłè¡őèŕćæŢřæ■őåĹřè¿¿çŽĎåĞ¡æŢř

```
import queue
import socket
import os

class PollableQueue(queue.Queue):
    def __init__(self):
        super().__init__()
```

(continues on next page)

(continued from previous page)

```
        # Create a pair of connected sockets
        if os.name == 'posix':
            self._putsocket, self._getsocket = socket.socketpair()
        else:
            # Compatibility on non-POSIX systems
            server = socket.socket(socket.AF_INET, socket.SOCK_
↪STREAM)
            server.bind(('127.0.0.1', 0))
            server.listen(1)
            self._putsocket = socket.socket(socket.AF_INET, socket.
↪SOCK_STREAM)
            self._putsocket.connect(server.getsockname())
            self._getsocket, _ = server.accept()
            server.close()

    def fileno(self):
        return self._getsocket.fileno()

    def put(self, item):
        super().put(item)
        self._putsocket.send(b'x')

    def get(self):
        self._getsocket.recv(1)
        return super().get()
```

åIJÍè£ŹäÿłäżčçăĄäÿ■ïijŇäÿĂäÿłæŰřçŽĎ `Queue` åőđä¿ŃçśżåđŃèćńåőŽäźL'ïijŇåžŢåśĆæŸŕäÿĂäÿłèćńè£
åIJÍUnixæIJžåŹÍäÿŁçŽĎ `socketpair()` åĞ¡æŢřèČ¡è¡żæİ¿çŽĎåĹŻåżžè£ŹæăůçŽĎåěŮæŐěå■ŮãĂĆ
åIJÍWindowsäÿŁéİćïijŇä¡åå£Ěéążä¡£çŤÍçśżäijijäżčçăĄæİěæÍąæŃ§åőČãĂĆ
çĎűåŘŐåőŽäźL'æŹőéĂŽçŽĎ `get()` åŠŇ `put()` æŰźæşŢåIJÍè£ŹäžŻåěŮæŐěå■ŮäÿŁéİćæİěæL'ğèąŇI/OæŞ
`put()` æŰźæşŢåĘ■åřĘæŢřæ■őæŤ¿åĚěéŸ§åĹŮåŘŐäijŽåĘŹäÿĂäÿłå■Ţå■ŮèŁĆåĹřæ§ŘäÿłåěŮæŐěå■Ůäÿ■å
èĂŇ `get()` æŰźæşŢåIJÍäżŐéŸ§åĹŮäÿ■çğżéŹď'äÿĂäÿłåĚČçť'äæŮűäijŽäżŐåŘęåď'ŰäÿĂäÿłåěŮæŐěå■Ůäÿ■

`fileno()` æŰźæşŢä¡£çŤÍäÿĂäÿłåĞ¡æŢřæŕŤåęĆ `select()` æİěèőľè£ŹäÿłéŸ§åĹŮåŘŕäżěèćńè¡őèŕćãĂĆ åőČäżĚäżĚåŘłæŸŕæŽť'éIJšäžĘåžŢåśĆèćń `get()` åĞ¡æŢřä¡£çŤÍåĹřçŽĎsocketçŽĎæŰĞäżűæŘŘè£řçňęèĂŇåůšãĂĆ

äÿŃéİćæŸŕäÿĂäÿłä¿Ńå■ŘïijŇåőŽäźL'äžĘäÿĂäÿłäÿžåĹřæİěçŽĎåĚČçť'äçŻŚæŐğåď'ŽäÿłéŸ§åĹŮçŽĎæŰ

```
import select
import threading

def consumer(queues):
    '''
    Consumer that reads data on multiple queues simultaneously
    '''
    while True:
        can_read, _, _ = select.select(queues,[],[])
        for r in can_read:
```

(continues on next page)

(continued from previous page)

```
            item = r.get()
            print('Got:', item)

q1 = PollableQueue()
q2 = PollableQueue()
q3 = PollableQueue()
t = threading.Thread(target=consumer, args=([q1,q2,q3],))
t.daemon = True
t.start()

# Feed data to the queues
q1.put(1)
q2.put(10)
q3.put('hello')
q2.put(15)
...
```

åęĆæđIJä¡ăèŕŢçİĂè£ŘèąŇåőČïijŇä¡ăäijŽåŔŚçŐřè£ŹäÿłæűĹèť źèĂĚäijŽæŐěåŘŮåĹřæĽĂæIJĽçŽĎèćń

èőĺèőž

åŕźäžŐè¡őèŕćéİđçśżæŰĞäżűåŕźèśąïijŇæŕŤåęĆéŸ§åĹŮéĂŽåÿÿéČ¡æŸŕæŕŤè¿ČæčŸæĽŃçŽĎéŮőećŸãĂ
ä¿ŃåęĆïijŇåęĆæđIJä¡ääÿ■ä¡£çŤĺäÿŁéİćçŽĎåěŮæŐěå■ŮæŁĂæIJŕïijŇ
ä¡ăåŤŕäÿĂçŽĎéĂĽæŃľåŕśæŸŕçijŰåĘŹäżčçăĄæİěå¿łçŐŕéĄ■åŐĘè£ŹäžŻéŸ§åĹŮåźűä¡£çŤĺäÿĂäÿłåőŽæŮűå

```
import time
def consumer(queues):
    while True:
        for q in queues:
            if not q.empty():
                item = q.get()
                print('Got:', item)

        # Sleep briefly to avoid 100% CPU
        time.sleep(0.01)
```

è£ŹæăůåĄŽåĚűåőđäÿ■åŘĹçŘĘïijŇè£ŸäijŽåijŢåĚěåĚűäżŰçŽĎæĂğèČ¡éŮőećŸãĂĆ
ä¿ŃåęĆïijŇåęĆæđIJæŰřçŽĎæŢŕæ■őèćńåŁäåĚěåĹřäÿĂäÿłéŸ§åĹŮäÿ■ïijŇèĞşåřŚèęĄèŁś10æŕńçğŠæĽ■èČ¡è
åęĆæđIJä¡ääźŃåĽ■çŽĎè¡őèŕćè£ŸèęĄåŐżè¡őèŕćåĚűäżŰåŕźèśąïijŇæŕŤåęĆç¡ŚçżIJåěŮæŐěå■ŮéĆčè£ŸäijŽæĽ
ä¿ŃåęĆïijŇåęĆæđIJä¡äæČşåŘŇæŮűè¡őèŕćåěŮæŐěå■ŮåŠŇéŸ§åĹŮïijŇä¡äåŘŕèČ¡èęĄåČŘäÿŃéİćè£Źæăůä¡£

```
import select

def event_loop(sockets, queues):
    while True:
        # polling with a timeout
        can_read, _, _ = select.select(sockets, [], [], 0.01)
        for r in can_read:
```

(continues on next page)

(continued from previous page)

```
            handle_read(r)
        for q in queues:
            if not q.empty():
                item = q.get()
                print('Got:', item)
```

è£ŹäÿłæŰźæąĹéĂŽè£ĞåřĘéŸ§åĹŮåŠŇåěŮæŐěå■Ůç■L'åŘŇåŕźå¿ĚæİěèğčåĘşäžĘåď'ğéČĺåĹĘçŽĎéŮőé
äÿĂäÿłå■ŢçŃňçŽĎ select() èřČçŤĺåŘŕèćńåŘŇæŮűçŤĺæİěè¡őèŕćãĂĆ
ä¡£çŤĺèűĚæŮűæĹŰåĚűäżŰå§žäžŐæŮűéŮť'çŽĎæIJžåĹűæİěæL'ğèąŇåŚĺæIJ§æĂğæčĂæ§ěåźűæšąæIJL'å£Ěèę
çŤŽèĞşïijŇåęĆæđIJæŢřæ■őèćńåŁååĚěåĹřäÿĂäÿłéŸ§åĹŮïijŇæűĹèť'źèĂĚåĞäźŐåŘŕäżěåőđæŮűçŽĎèćńéĂ
åř¡çőąäijŽæIJL'äÿĂçĆźçĆźåžŢåśĆçŽĎI/Oæ■§èĂŮïijŇä¡£çŤĺåőČéĂŽåÿÿäijŽèŐůå¿ŮæŻť'åě¡çŽĎåŞ■åžŤæŮ

14.14 12.14 åIJÍUnixçşżçż§äÿŁéİćåŘŕåŁĺåőĹæŁď'è£ŻçÍŃ

éŮőécŸ

ä¡äæČşçijŰåĘŹäÿĂäÿłä¡IJäÿžäÿĂäÿłåIJÍUnixæĹŰçśżUnixçşżçż§äÿŁéİćè£ŘèąŇçŽĎåőĹæŁď'è£ŻçÍŃè£

èğčåĘşæŰźæąĹ

åĹŻåżžäÿĂäÿłæ■čçąőçŽĎåőĹæŁď'è£ŻçÍŃéIJĂèęĄäÿĂäÿłçš¿çąőçŽĎçşżçż§èřČçŤĺåžŘåĹŮäżěåŘŁåŕźäż
äÿŃéİćçŽĎäžčçăĄåśŢçď'žäžĘæĂŐæăůåőŽäźĹ'äÿĂäÿłåőĹæŁď'è£ŻçÍŃïijŇåŘŕäżěåŘŕåŁĺåŘŐå¿Ĺåőźæ ŸŞçŽĎ

```
#!/usr/bin/env python3
# daemon.py

import os
import sys

import atexit
import signal

def daemonize(pidfile, *, stdin='/dev/null',
                          stdout='/dev/null',
                          stderr='/dev/null'):

    if os.path.exists(pidfile):
        raise RuntimeError('Already running')

    # First fork (detaches from parent)
    try:
        if os.fork() > 0:
            raise SystemExit(0)    # Parent exit
    except OSError as e:
        raise RuntimeError('fork #1 failed.')
```

(continues on next page)

(continued from previous page)

```
    os.chdir('/')
    os.umask(0)
    os.setsid()
    # Second fork (relinquish session leadership)
    try:
        if os.fork() > 0:
            raise SystemExit(0)
    except OSError as e:
        raise RuntimeError('fork #2 failed.')

    # Flush I/O buffers
    sys.stdout.flush()
    sys.stderr.flush()

    # Replace file descriptors for stdin, stdout, and stderr
    with open(stdin, 'rb', 0) as f:
        os.dup2(f.fileno(), sys.stdin.fileno())
    with open(stdout, 'ab', 0) as f:
        os.dup2(f.fileno(), sys.stdout.fileno())
    with open(stderr, 'ab', 0) as f:
        os.dup2(f.fileno(), sys.stderr.fileno())

    # Write the PID file
    with open(pidfile,'w') as f:
        print(os.getpid(),file=f)

    # Arrange to have the PID file removed on exit/signal
    atexit.register(lambda: os.remove(pidfile))

    # Signal handler for termination (required)
    def sigterm_handler(signo, frame):
        raise SystemExit(1)

    signal.signal(signal.SIGTERM, sigterm_handler)

def main():
    import time
    sys.stdout.write('Daemon started with pid {}\n'.format(os.
↪getpid()))
    while True:
        sys.stdout.write('Daemon Alive! {}\n'.format(time.ctime()))
        time.sleep(10)

if __name__ == '__main__':
    PIDFILE = '/tmp/daemon.pid'

    if len(sys.argv) != 2:
        print('Usage: {} [start|stop]'.format(sys.argv[0]),␣
↪file=sys.stderr)
```

(continues on next page)

(continued from previous page)

```
        raise SystemExit(1)

    if sys.argv[1] == 'start':
        try:
            daemonize(PIDFILE,
                      stdout='/tmp/daemon.log',
                      stderr='/tmp/dameon.log')
        except RuntimeError as e:
            print(e, file=sys.stderr)
            raise SystemExit(1)

        main()

    elif sys.argv[1] == 'stop':
        if os.path.exists(PIDFILE):
            with open(PIDFILE) as f:
                os.kill(int(f.read()), signal.SIGTERM)
        else:
            print('Not running', file=sys.stderr)
            raise SystemExit(1)

    else:
        print('Unknown command {!r}'.format(sys.argv[1]), file=sys.
↪stderr)
        raise SystemExit(1)
```

èęĄåŘŕåŁĺè£ŹäÿłåőĺæŁď'è£ŻçĺŃïijŇçŤĺæĹůéIJĂèęĄä¡£çŤĺåęĆäÿŃçŽĎåŚ¡äżď'ïijŽ

```
bash % daemon.py start
bash % cat /tmp/daemon.pid
2882
bash % tail -f /tmp/daemon.log
Daemon started with pid 2882
Daemon Alive! Fri Oct 12 13:45:37 2012
Daemon Alive! Fri Oct 12 13:45:47 2012
...
```

åőĺæŁď'è£ŻçĺŃåŘŕäżěåőŇåĚĺåIJĺåŘŐåŘŕè£ŘèąŇïijŇåŻäæ■ď'è£ŹäÿłåŚ¡äżď'äijŽçńŃå■ęè£ŤåŻđãĂĆäÿ■è£ĞïijŇä¡ääŘŕäżěåČŘäÿŁéİćéĆčæäůæ§ěçIJŃäÿŐåőČçŻÿåĚęçŽĎpidæŰĞäżűåŠŇæŮěå£ŮãĂĆèęĄåĄIJæ

```
bash % daemon.py stop
bash %
```

èőĺèőž

æIJňèŁĆåőŽäźL'äžĘäÿĂäÿłåĞ¡æŢř `daemonize()` ïijŇåIJĺçĺŃåžŘåŘŕåŁĺæŮűèćńèřČçŤĺä¡£å¿ŮçĺŃåžŘ `daemonize()` åĞ¡æŢřåŘłæŐěåŘŮåĚęéŤőå■ŮåŘĆæŢřïijŇè£ŹæäůçŽĎèŕĺåŘŕéĂL'åŘĆæŢřåIJĺèćńä¡£çŤĺæŮ åőČäijŽäijžåĹűçŤĺæĹůåČŘäÿŃéİćè£ŹæäůÄ¡£çŤĺåőČïijŽ

```
daemonize('daemon.pid',
          stdin='/dev/null,
          stdout='/tmp/daemon.log',
          stderr='/tmp/daemon.log')
```

èĂŇäÿ■æŸŕåČŔäÿŃéİćè£ŹæăůåŕńçşŁäÿ■æÿĚçŽĎèřČçŤĺïijŽ

```
# Illegal. Must use keyword arguments
daemonize('daemon.pid',
          '/dev/null', '/tmp/daemon.log','/tmp/daemon.log')
```

åĹŻåżžäÿĂäÿłåőĹæŁď'è£ŻçÍŃçŽĎæ■ěéłď'çIJŃäÿŁåŐżäÿ■æŸŕå¿ĹæŸŞæĞĆïijŃä¡ĘæŸŕåď'ğä¡ŞæĂİæČ
éęŮåĚĹïijŃäÿĂäÿłåőĹæŁď'è£ŻçÍŃå£Ěéążèę ĄżőçĹűè£ŻçÍŃäÿ■èĎśçężãĂĆ è£ŹæŸŕçŤś
os.fork() æŞ■ä¡IJæİěåőŃæĹŘçŽĎïijŃåźűçńŃå■şèćńçĹűè£ŻçÍŃçżĹæ■ćãĂĆ

åIJÍå■Řè£ŻçÍŃåŘŸæĹŘå■ď'åĎ£åŘŐïijŃěřČçŤÍ os.setsid()
åĹŻåżžäžĘäÿĂäÿłåĚĹæŮřçŽĎè£ŻçÍŃäijŽèŕİïijŃåźűèő¿ç¡őå■Řè£ŻçÍŃäÿžéęŮéćĘãĂĆ
åőČäijŽèő¿ç¡őè£Źäÿłå■Řè£ŻçÍŃäÿžæŮřçŽĎè£ŻçÍŃçżĎçŽĎéęŮéćĘïijŃåźűçąőä£İäÿ■äijŽåĘ■æIJL'æŐğåĹűç
åęĆæđIJè£ŹäžŽåŘňäÿŁåŐżåď'łé■ŤåźżïijŃåŻääÿžåőČéIJĂèęĄåřĘåőĹæŁď'è£ŻçÍŃåŘŃçżĹçńŕåĹĘçężåijĂåźű
èřČçŤÍ os.chdir() åŠŃ os.umask(0) æŤźåŘŸäžĘą¡ŞåL'■åůěä¡IJçŽőå¡Ţåźűé■G■ç¡őæŮĞäżűæİČéŹŘæ
ä£őæŤźçŻőå¡ŢéĂŽåÿÿæŸŕäÿłåě¡äÿżæĎŘïijŃåŻääÿžè£Źæăůåŕŕäżěä¡£å¿ŮåőČäÿ■åĘ■åůěä¡IJåIJÍèćńåŘŕåŁÍæ

åŘęåď'ŰäÿĂäÿłèřČçŤÍ os.fork() åIJÍè£ŹéĞŃæŻť'åŁăçěđçğŸçĆźãĂĆ
è£ŹäÿĂæ■ěä¡£å¿ŮåőĹæŁď'è£ŻçÍŃåď'śåŐżäžĘèŐůåŘŰæŮřçŽĎæŐğåĹűçżĹçńŕçŽĎèČ¡åŁŻåźűäÿŤèőľ'åőČæ
ïijĹæIJňèť'ĺäÿŁïijŃèřědaemonæŤ¿åijČäžĘåőČçŽĎäijŽèŕİéęŮéćĘä¡Őä¡■ïijŃåŻäæ■ď'åĘ■äź§æšąæIJL'æİČéŹŘ
ář¡çőąä¡åŘŕäżěå£¡çŢěè£ŹäÿĂæ■ěïijŃä¡ĘæŸŕæIJĂåě¡äÿ■èęĄè£ŹäźĹåĄŽãĂĆ

äÿĂæŮęåőĹæŁď'è£ŻçÍŃèćńæ■čçąőçŽĎåĹĘçężïijŃåőČäijŽéĞ■æŮřåĹİåğŃåŃŰæăĞåĞĘI/OæţĄæŃĞåŘ
è£ŹäÿĂéČĺåĹĘæIJL'çĆźéŽ¿æĞĆãĂĆèů§æăĞåĞĘI/OæţĄçŻÿåĚşçŽĎæŮĞäżűářźèšąçŽĎåijŢçŤĺåIJÍèğčéĞŁå
ïijĹsys.stdout, sys.__stdout__ç■L'ïijL'ãĂĆ äżĚäżĚçőĂå■ŢçŽĎåĚşéŮ■
sys.stdout åźűéĞ■æŮřæŃĞåőŽåőČæŸŕèąŃäÿ■éĂŽçŽĎïijŃ
åŻääÿżæšąåŁđæşŢç§ěéĄŞåőČæŸŕåŘęåĚĺéČĺéČ¡æŸŕçŤĺçŽĎæŸŕ sys.stdout ãĂĆ
è£ŹéĞŃïijŃæĹŚäżňæL'ŞåijĂäžĘäÿĂäÿłå■ŢçŃňçŽĎæŮĞäżűářźèšąïijŃåźűèřČçŤÍ os.
dup2() ïijŃ çŤĺåőČæĺěäżčæŻ£èćń sys.stdout ä¡£çŤĺçŽĎæŮĞäżűæŘŘè£řçňęãĂĆ
è£ŹæăůïijŃsys.stdout ä¡£çŤĺçŽĎåŐ§åğŃæŮĞäżűäijŽèćńåĚşéŮ■åźűçŤśæŮřçŽĎæĺěæŻ£æ■ćãĂĆ
è£ŸèęĄåijžèřČçŽĎæŸŕäżżä¡ŢçŤĺäžŐæŮĞäżűçijŮçĂąæĹŮæŮĞæIJňåď'ĎçŘĘçŽĎæăĞåĞĘI/OæţĄè£ŸäijŽä

åőĹæŁď'è£ŻçÍŃçŽĎäÿĂäÿłéĂŽåÿÿåőđèůţæŸŕåIJÍäÿĂäÿłæŮĞäżűäÿ■åĘŹåĚèè£ŻçÍŃIDïijŃåŕäżěèćńåĚ
daemonize() åĞ¡æŢřçŽĎæIJĂåŘŐéČĺåĹĘåĘźäžĘè£ŹäÿłæŮĞäżűïijŃä¡ĘæŸŕåIJÍçÍŃåžŘçżĹæ■ćæŮűåĹăć
atexit.register() åĞ¡æŢřæşĺåĘŃäžĘäÿĂäÿłåĞ¡æŢřåIJÍPythonèğčéĞŁåŹÍçżĹæ■ćæŮűæL'ğèąÑãĂĆ
äÿĂäÿłåŕźäžŐSIGTERMçŽĎä£ąåŘůåď'ĎçŘĘåŹÍçŽĎåőŽäźL'åŘŃæăůéIJĂèęĄèćńäijŸéŻĚçŽĎåĚşéŮ■ãĂĆ
ä£ąåŘůåď'ĎçŘĘåŹÍçőĂå■ŢçŽĎæŁŻåĞžäžĘ SystemExit() åijĆåÿÿãĂĆ
æĹŰèőÿè£ŹäÿĂæ■ěçIJŃäÿŁåŐżæšąå£ĚèęĄïijŃä¡ĘæŸŕæšąæIJL'åőČïijŃ
çżĹæ■ćä£ąåŘůäijŽä¡£å¿Ůäÿ■æL'ğèąŃ atexit.register()
æşĺåĘŃçŽĎæÿĚçŘĘæŞ■ä¡IJçŽĎæŮůåĂŹåŕśæİĂæŐL'äžĘèğčéĞŁåŹÍãĂĆ
äÿĂäÿłæİĂæŐL'è£ŻçÍŃçŽĎä¿Ńå■Řäžčçą ąåŘŕäżěåIJÍçÍŃåžŘæIJĂåŘŐçŽĎ stop
åŚ¡äżď'çŽĎæŞ■ä¡IJäÿ■çIJŃåĹřãĂĆ

æŻť'åď'ŽåĚşäžŐçijŮåĘŹåőĹæŁď'è£ŻçÍŃçŽĎä£ąæĄŕåŘŕäżěæ§ěçIJŃãĂŁUNIX
çŐŕåćČéńŸçžğçijŮçÍŃãĂŃ, çňňäžŃçL'Ĺ by W. Richard
Stevens and Stephen A. Rago (Addison-Wesley, 2005)ãĂĆ

åř¡çőąåőČæŸŕåĚşæşÍäÿŐCèŕ■èÍĂçijŰçÍŃïijŇä¡ĘæŸŕæĽĂæIJĽçŽĎåĘĚåőźéČ¡éĂĆçŤÍäžŐPythonïijŇ
åŻääÿžæĽĂæIJĽéIJĂèęĄçŽĎPOSIXåĞ¡æŢřéČ¡åŘŕäżěåIJÍæăĞåĞĘåžŞäÿ■æĽ¿åĹřãĂĆ

15 çňňå■ĄäÿĽçńäïijŽèĎŽæIJňçijŰçÍŃäÿŐçşżçż§çőąçŘĘ

èőÿåď Žäžžä¡£çŤÍPythonä¡IJäÿžäÿĂäÿłshellèĎŽæIJňçŽĎæŻ£äżčïijŇçŤÍæİěåőđçŐřåÿÿçŤÍçşżçż§äżżåŁą

Contents:

15.1 13.1 éĂŽè£ĞéĞ■åőŽåŘŚ/çőąéĄŞ/æŰĞäżűæŐěåŘŮè¿ŞåĚě

éŮőécŸ

ä¡ăåÿŇæIJŻä¡äçŽĎèĎŽæIJňæŐěåŘŮäżżä¡ŢçŤÍæĹůèőď äÿžæIJĂçőĂå■ŢçŽĎè¿ŞåĚěæŰźåijŘãĂĆåŇĚæ
éĞ■åőŽåŘŚæŰĞäżűåĹřèŕěèĎŽæIJňïijŇæĹŰåIJÍåŚ¡äżď èąŇäÿ■äijăéĂŠäÿĂäÿłæŰĞäżűåŘ■æĹŰæŰĞäżűåŘ■

èğčåĘşæŰźæąĹ

PythonåĘĚç¡őçŽĎ `fileinput` æÍąåİŮèőľ è£ŹäÿłåŘŸå¿ŮçőĂå■ŢãĂĆåęĆæđIJä¡ăæIJĽäÿĂäÿłäÿŃéİćè

```
#!/usr/bin/env python3
import fileinput

with fileinput.input() as f_input:
    for line in f_input:
        print(line, end='')
```

éĆčäźĹä¡ăåřśèČ¡äżěåĽ■éİćæŘŘåĹřçŽĎæĽĂæIJĽæŰźåijŘæİěäÿžæ■ď èĎŽæIJňæŘŘä¿Żè¿ŞåĚěãĂĆåĄ
`filein.py` åźűåřĘåĚűåŘŸäÿžåŘŕæĽğèąŇæŰĞäżűïijŇ éĆčäźĹä¡ăåŘŕäżěåČŘäÿŃéİćè£ŹæăůěřČçŤÍåőČïijŇ

```
$ ls | ./filein.py          # Prints a directory listing to stdout.
$ ./filein.py /etc/passwd   # Reads /etc/passwd to stdout.
$ ./filein.py < /etc/passwd # Reads /etc/passwd to stdout.
```

èőÍèőž

`fileinput.input()` åĹŻåżžåźűè£ŤåŻđäÿĂäÿł `FileInput` çśżçŽĎåőđä¿ŃãĂĆ
èŕěåőđä¿ŃéŹď äžĘæŃěæIJĽäÿĂäžŻæIJĽçŤÍçŽĎåÿőåŁľæŰźæşŢåď ŰïijŇåőČè£ŸåŘŕèćńå¡ŞåĄŽäÿĂäÿłäÿŁä
åŻäæ■ď ïijŇæŢť åŘĹèţůæİěïijŇåęĆæđIJæĹŚäżňèęĄåĘŹäÿĂäÿłæĽŞå■řåď Žäÿłæ ŰĞäżűè¿ŞåĞžçŽĎèĎŽæIJň

```
>>> import fileinput
>>> with fileinput.input('/etc/passwd') as f:
>>>     for line in f:
...         print(f.filename(), f.lineno(), line, end='')
```

(continues on next page)

(continued from previous page)

```
...
/etc/passwd 1 ##
/etc/passwd 2 # User Database
/etc/passwd 3 #

<other output omitted>
```

éĂŽè£ĞåřĘåőČä¡IJäÿžäÿĂäÿłäÿŁäÿŃæŰĞçőąçŘĘåŹÍä¡£çŤÍïijŇåŘŕäżěçąőä£ÍåőČäÿ■åĘ■ä¡£çŤÍæŮűæŰ
èĂŇäÿŤæĹŚäżňåIJÍäźŃåŘŐè£ŸæijŤçď žäžĘ `FileInput` çŽĎäÿĂäžŻæIJL'çŤÍçŽĎåÿőåŁľæŰźæşŢæÍěèŐů

15.2 13.2 çżĹæ■ćçÍŃåžŘåźűçżŹåĞžéŤŹèŕŕä£ąæĄŕ

éŮőéćŸ

ä¡äæČşåŘŚæäĞåĞĘéŤŹèŕŕæL'Şå■řäÿĂæÍąæűĹæĄŕåźűè£ŤåŻđæ§ŘäÿłéİđéŻűçŁűæĂĄçăĄæÍěçżĹæ■ćçÍ

èğčåĘşæŰźæąĹ

ä¡äæIJL'äÿĂäÿłçÍŃåžŘåČŘäÿŃéÍćè£ŹæäůçżĹæ■ćïijŇæŁŻåĞžäÿĂäÿł `SystemExit`
åijĆåÿÿïijŇä¡£çŤÍéŤŹèŕŕæűĹæĄŕä¡IJäÿžåŘĆæŢřăĂĆä¿ŃåęĆïijŽ

```
raise SystemExit('It failed!')
```

åőČäijŽåřĘæűĹæĄŕåIJÍ `sys.stderr` äÿ■æL'Şå■řïijŇçĎűåŘŐçÍŃåžŘäżěçŁűæĂĄçăĄ1éĂĂåĞžăĂĆ

èőÍèőž

æIJňèŁĆèŹ¡çĎűå¿Ĺç§■åřŘïijŇä¡ĘæŸŕåőČèČ¡èğčåĘşåIJÍåĘŹèĎŽæIJňæŮűçŽĎäÿĂäÿłåÿÿèğĄéŮőéćŸã
äź§åřśæŸŕèŕť'ïijŇå¡Şä¡äæČşèęĄçżĹæ■ćæ§Řäÿłçí ŃåžŘæŮűïijŇä¡ääŘŕèČ¡äijŽåČŘäÿŃéÍćè£ŹæäůåĘŹïijŽ

```
import sys
sys.stderr.write('It failed!\n')
raise SystemExit(1)
```

åęĆæđIJä¡äçŻť'æŐěåřĘæűĹæĄŕä¡IJäÿžåŘĆæŢřäijäçżŹ `SystemExit()`
ïijŇéĆčäźĹä¡ääŘŕäżěçIJĄçŢěåĚűäżŮæ■ěéłď'ïijŇ æŕŤåęĆim-
portèŕ■åŘěæĹŮåřĘéŤŹèŕŕæűĹæĄŕåĘŹåĚě `sys.stderr`

15.3 13.3 èğčæđŘåŚ¡äżď'èąŇéĂĽéąź

éŮőéćŸ

ä¡äçŽĎçÍŃåžŘåęĆä¡ŢèČ¡åď'§èğčæđŘåŚ¡äżď'èąŇéĂĽéąźïijĹä¡■äžŐsys.argväÿ■ïijL'

èğčåĘşæŰźæąĹ

argparse æĺąåİŮåŔŕèćńçŤĺæİěèğčæđŘåŠ¡äžď'èąŇéĂĽ'éąźãĂĆäÿŃéİćäÿĂäÿłçőĂå■Ţä¿Ńå■ŘæijŤçď'ž

```
# search.py
'''
Hypothetical command-line tool for searching a collection of
files for one or more text patterns.
'''
import argparse
parser = argparse.ArgumentParser(description='Search some files')

parser.add_argument(dest='filenames',metavar='filename', nargs='*')

parser.add_argument('-p', '--pat',metavar='pattern', required=True,
                    dest='patterns', action='append',
                    help='text pattern to search for')

parser.add_argument('-v', dest='verbose', action='store_true',
                    help='verbose mode')

parser.add_argument('-o', dest='outfile', action='store',
                    help='output file')

parser.add_argument('--speed', dest='speed', action='store',
                    choices={'slow','fast'}, default='slow',
                    help='search speed')

args = parser.parse_args()

# Output the collected arguments
print(args.filenames)
print(args.patterns)
print(args.verbose)
print(args.outfile)
print(args.speed)
```

èŕěçĺŃåžŘåőŽäźĽ'äžĘäÿĂäÿłąęĆäÿŃä¡£çŤĺçŽĎåŠ¡äžď'èąŇèğčæđŘåŹĺïijŽ

```
bash % python3 search.py -h
usage: search.py [-h] [-p pattern] [-v] [-o OUTFILE] [--speed {slow,
↪fast}]
                 [filename [filename ...]]

Search some files

positional arguments:
  filename

optional arguments:
```

(continues on next page)

(continued from previous page)

```
  -h, --help            show this help message and exit
  -p pattern, --pat pattern
                        text pattern to search for
  -v                    verbose mode
  -o OUTFILE            output file
  --speed {slow,fast}   search speed
```

äÿŃéİćçŽĎéČĺåĹĘæijŤçď žäžĘçÍŃåžŘäÿ■çŽĎæŢřæ■ŏéČĺåĹĘãĂĆäżŤçżĘèğĆåŕ§print()èŕ■åŘěçŽĎæĽ'Ş

```
bash % python3 search.py foo.txt bar.txt
usage: search.py [-h] -p pattern [-v] [-o OUTFILE] [--speed {fast,
↪slow}]
                 [filename [filename ...]]
search.py: error: the following arguments are required: -p/--pat

bash % python3 search.py -v -p spam --pat=eggs foo.txt bar.txt
filenames = ['foo.txt', 'bar.txt']
patterns  = ['spam', 'eggs']
verbose   = True
outfile   = None
speed     = slow

bash % python3 search.py -v -p spam --pat=eggs foo.txt bar.txt -o␣
↪results
filenames = ['foo.txt', 'bar.txt']
patterns  = ['spam', 'eggs']
verbose   = True
outfile   = results
speed     = slow

bash % python3 search.py -v -p spam --pat=eggs foo.txt bar.txt -o␣
↪results \
             --speed=fast
filenames = ['foo.txt', 'bar.txt']
patterns  = ['spam', 'eggs']
verbose   = True
outfile   = results
speed     = fast
```

åŕźäžŐéĂĽéąźåĂijçŽĎè£ŻäÿĂæ■ěåď ĎçŘĘçŤśçÍŃåžŘæĺěåĘşåőŽïijŇçŤĺä¡äèĞłåůśçŽĎéĂżè¿ŚæĺěæŻ£ `print()` åĞ¡æŢřãĂĆ

èőĺèőž

`argparse` æĺąåĺŮæŸŕæăĞåĞĘåžŞäÿ■æIJĂåď ğçŽĎæĺąåĺŮäźŃäÿĂïijŇæŃěæIJĽ'åď ğéĞŘçŽĎéĚ■ç¡ŏé æIJňèŁĆåŘłæŸŕæijŤçď žäžĘåĚűäÿ■æIJĂå§žçąĂçŽĎäÿĂäžŻçĽ'źæĂğïijŇåÿőåŁl'ä¡ăåĚěéŮĺãĂĆ

äÿžäžĘèğčæđŘåŚ¡äżď èąŇéĂĽéąźïijŇä¡äéęŰåĚĹèęĄåĹŻåżžäÿĂäÿł `ArgumentParser` åőđä¿ŃïijŇ åźűä¡£çŤÍ `add_argument()`

æŰźæşŢåčřæŸŐä¡ăæČşèęĄæŤřæŇĄçŽĎéĂĽ'éąźãĂĆ åIJÍæŕŘäÿł add_argument() èřČçŤÍäÿ■ïijŇdest åŘĆæŢřæŇĞåőŽèğčæđŘçżŞæđIJèćńæŇĞæť¿çżŹåśđæĂğçŽĎåŘ■å■ŮãĂĆ metavar åŘĆæŢřèćńçŤÍæİěçŤ§æĹŘåÿőåŁľä£ąæĄŕãĂĆaction åŘĆæŢřæŇĞåőŽèů§åśđæĂğåŕźåžŤçŽĎåď'ĎçŘĘéĂżè¿ŚïijŇ éĂŽåÿÿçŽĎåĂijäÿž store ,èćńçŤÍæİěå■ŸåĆÍæ§ŘäÿłåĂijæĹŰåřĘåď'ŽäÿłåŘĆæŢřåĂijæŤűéŻĘåĹřäÿĂäÿłåĹŮèąÍäÿ■ãĂĆ äÿŃéİćçŽĎåŘĆæŢřæŤűéŻĘæĽ'ĂæIJĽ'åĽľä¡ŹçŽĎåŚ¡äżď'èąŇåŘĆæŢřåĹřäÿĂäÿłåĹŮèąÍäÿ■ãĂĆåIJÍæIJňä¿Ń

```
parser.add_argument(dest='filenames',metavar='filename', nargs='*')
```

äÿŃéİćçŽĎåŘĆæŢřæăźæ■őåŘĆæŢřæŸŕåŘęå■ŸåIJÍæİěèő¿ç¡őäÿĂäÿł Boolean æăĞå£ŮïijŽ

```
parser.add_argument('-v', dest='verbose', action='store_true',
                    help='verbose mode')
```

äÿŃéİćçŽĎåŘĆæŢřæŐěåŘŮäÿĂäÿłå■ŢçŃňåĂijåźűåřĘåĚűå■ŸåĆÍäÿžäÿĂäÿłå■ŮçňęäÿšïijŽ

```
parser.add_argument('-o', dest='outfile', action='store',
                    help='output file')
```

äÿŃéİćçŽĎåŘĆæŢřèŕť'æŸŐåĚĄèőÿæ§ŘäÿłåŘĆæŢřéĞ■åď'■åĞžçŐřåď'ŽæňąïijŇåźűåřĘåőČäżňè£¡åŁäåĹ required æăĞå£ŮèąÍçď'žèřěåŘĆæŢřèĞşåřŚèęĄæIJĽ'äÿĂäÿłãĂĆ-p åŠŇ --pat èąÍçď'žäÿď'äÿłåŘĆæŢřåŘ■å¡ćåijŘéČ¡åŘŕä¡£çŤÍãĂĆ

```
parser.add_argument('-p', '--pat',metavar='pattern', required=True,
                    dest='patterns', action='append',
                    help='text pattern to search for')
```

æIJĂåŘŐïijŇäÿŃéİćçŽĎåŘĆæŢřèŕť'æŸŐæŐěåŘŮäÿĂäÿłåĂijïijŇä¡ĘæŸŕäijŽåřĘåĚűåŠŇåŘŕèČ¡çŽĎéĂ

```
parser.add_argument('--speed', dest='speed', action='store',
                    choices={'slow','fast'}, default='slow',
                    help='search speed')
```

äÿĂæŮęåŘĆæŢřéĂĽ'éąźèćńæŇĞåőŽïijŇä¡ăářśåŘŕäżěæĽ'ğèąŇ parser.parse() æŰźæşŢäžĘãĂĆ åőČäijŽåď'ĎçŘĘ sys.argv çŽĎåĂijåźűè£ŤåŻđäÿĂäÿłçżŞæđIJåőđä¿ŃãĂĆ æŕŘäÿłåŘĆæŢřåĂijäijŽèćńèő¿ç¡őæĹŘèŕěåőđä¿Ńäÿ■ add_argument() æŰźæşŢçŽĎ dest åŘĆæŢřæŇĞåőŽçŽĎåśđæĂğåĂijãĂĆ

è£Ÿå¿Ĺåď'Žçğ■åĚűäžŰæŰźæşŢèğčæđŘåŚ¡äżď'èąŇéĂĽ'éąźãĂĆ ä¿ŃåęĆïijŇä¡ăåŘŕèČ¡äijŽæĽ'ŃåŁĺçŽĎåď'ĎçŘĘ sys.argv æĹŰèĂĚä¡£çŤÍ getopt æĺąåİŮãĂĆ ä¡ĘæŸŕïijŇåęĆæđIJä¡äéĞĞçŤÍæIJňèŁĆçŽĎæŰźåijŘïijŇåřĘäijŽåĞŘåřŚå¿Ĺåď'ŽåĘŮä¡Źäžčçăąï argparse æĺąåİŮåůšçżŘåÿőä¡ăåď'ĎçŘĘäžĘãĂĆ ä¡ăåŘŕèČ¡è£ŸäijŽçćřåĹřä¡£çŤÍ optparse åžŞèğčæđŘéĂĽ'éąźçŽĎäžčçăąãĂĆ åř¡çőą optparse åŠŇ argparse å¿ĹåČŘïijŇä¡ĘæŸŕåŘŐèĂĚæŻť'åĚĹè£ŻïijŇåŻăæ■ď'åIJÍæŰŕçŽĎçÍŇåžŘäÿ■ä¡ăåžŤèŕěä¡£çŤÍåőČãĂĆ

15.4 13.4 è£ŘèąŇæŮűåijźåĞžåŕĘçăĄè¿ŞåĚěæŔŘçď ž

éŮőéćŸ

ä¡ăåĘŹäžĘäÿłèĎŽæIJňïijŇè£ŘèąŇæŮűéIJĂèęĄäÿĂäÿłåŕĘçăĄãĂĆæ■ď'èĎŽæIJňæŸŕäžď'äžŠåijŔçŽĎïij
èĂŇæŸŕéIJĂèęĄåijźåĞžäÿĂäÿłåŕĘçăĄè¿ŞåĚěæŔŘçď'žïijŇèőľçŤĺæĹůèĞłåůśè¿ŞåĚěãĂĆ

èğčåĘşæŰźæąĹ

è£ŹæŮűåĂŹPythonçŽĎ `getpass` æĺąåİŮæ■čæŸŕä¡ăæL'ĂéIJĂèęĄçŽĎãĂĆä¡ăåŔŕäžěèőľä¡ăå¿Ĺè¡żæİ¿
åźűäÿŤäÿ■äijŽåIJĺçŤĺæĹůçżĹçńŕåŻđæŸ¿åŕĘçăĄãĂĆäÿŃéİćæŸŕåĚůä¡Şäžčçăąïij Ž

```
import getpass

user = getpass.getuser()
passwd = getpass.getpass()

if svc_login(user, passwd):     # You must write svc_login()
   print('Yay!')
else:
   print('Boo!')
```

åIJĺæ■ď'äžčçăĄäÿ■ïijŇ`svc_login()` æŸŕä¡ăèęĄåőđçŐřçŽĎåď'ĎçŘĘåŕĘçăĄçŽĎåĞ¡æŢřïijŇåĚůä¡Şç

èőĺèőž

æşĺæĎŔåIJĺåL'■éİćäžčçăĄäÿ■ `getpass.getuser()`
äÿ■äijŽåijźåĞžçŤĺæĹůåŘ■çŽĎè¿ŞåĚěæŔŘçď'žãĂĆ åőČäijŽæăźæ■őèřěçŤĺæĹůçŽĎshel-
lçŐŕåćČæĹŰèĂĚäijŽä¿İæ■őæIJňåIJřçşżçż§çŽĎåŕĘçăĄåžŞïijĹæŤŕæŇĄ *pwd*
æĺąåİŮçŽĎåźşåŔřïijL'æİěä¡£çŤĺå¡ŞåL'■çŤĺæĹůçŽĎçŹżå¡ŢåŘ■ïijŇ

åęĆæđIJä¡ăæČşæŸ¿çď'žçŽĎåijźåĞžçŤĺæĹůåŘ■è¿ŞåĚěæŔŘçď'žïijŇä¡£çŤĺåĘĚç¡őçŽĎ
`input` åĞ¡æŢřïijŽ

```
user = input('Enter your username: ')
```

è£ŸæIJL'äÿĂçĆźå¿ĹéĞ■èęĄïijŇæIJL'äžŻçşżçż§åŔŕèČ¡äÿ■æŤŕæŇĄ `getpass()`
æŰźæşŢéŽŘèŮŔè¿ŞåĚěåŕĘçăĄãĂĆ è£Źçğ■æČĚåĘţäÿŃïijŇPythonäijŽæŔŘåL'■è■ęåŚŁä¡ăè£ŹäžŽéŮőéćŸïij

15.5 13.5 èŐůåŔŰçżĹçńŕçŽĎåď'ğåřŔ

éŮőéćŸ

ä¡ăéIJĂèęĄç§ěéĄŞå¡ŞåL'■çżĹçńŕçŽĎåď'ğåřŔäžěä¿£æ■čçąőçŽĎæăijåijŔåŇŰè¿ŞåĞžãĂĆ

èğčåĘşæŰźæąĹ

ä¡£çŤÍ `os.get_terminal_size()` åĞ¡æŢřæİěåĄŽåĹřè£ŹäÿĂçĆźãĂĆ

äżčçăĄçď žä¿ŃïijŽ

```
>>> import os
>>> sz = os.get_terminal_size()
>>> sz
os.terminal_size(columns=80, lines=24)
>>> sz.columns
80
>>> sz.lines
24
>>>
```

èőÍèőž

æIJL'åď łåď ŽæŰźåijŔæİěå¿Ůç§ěçżĹçńřåď ğåřŔäžĘïijŇäżŐèŕżåŔŰçŐŕåćČåŔŸéĞŔåĹřæL'ğèąŇåžŢåśĆç
`ioctl()` åĞ¡æŢřç■L'ç■L'ãĂĆ äÿ■è£ĞïijŇäÿžäżĂäźĹèęĄåŐżçăŤçl'űè£ŹäžŽåď■æİĆçŽĎåŁđæşŢèĂŇäÿ■æ

15.6 13.6 æL'ğèąŇåď ŰéČĺåŚ¡äżď åźűèŐůåŔŰåőČçŽĎè¿ŞåĞž

éŮőéćŸ

ä¡ăæČşæL'ğèąŇäÿĂäÿłåď ŰéČĺåŚ¡äżď åźűäžěPythonå■ŮçňęäÿšçŽĎå¡ćåijŔèŐůåŔŰæL'ğèąŇçżŞæđIJãĂ

èğčåĘşæŰźæąĹ

ä¡£çŤÍ `subprocess.check_output()` åĞ¡æŢřãĂĆä¿ŃåęĆïijŽ

```
import subprocess
out_bytes = subprocess.check_output(['netstat','-a'])
```

è£ŹæőţäžčçăĄæL'ğèąŇäÿĂäÿłæŇĞåőŽçŽĎåŚ¡äżď åźűåřĘæL'ğèąŇçżŞæđIJäžěäÿĂäÿłå■ŮèŁĆå■Ůçňęäÿ
åęĆæđIJä¡äéIJĂèęĄæŰĞæIJňå¡ćåijŔè£ŤåŻđïijŇåŁääÿĂäÿłèğčçăĄæ■ěéłď å■şåŔŕãĂĆä¿ŃåęĆïijŽ

```
out_text = out_bytes.decode('utf-8')
```

åęĆæđIJèćńæL'ğèąŇçŽĎåŚ¡äżď äžěéİđéŻűçăĄè£ŤåŻđïijŇåřśäijŽæŁŻåĞžåijĆåÿÿãĂĆ
äÿŃéİćçŽĎä¿Ňå■Ŕæ■ŢèŐůåĹřéŤŹèŕŕåźűèŐůåŔŰè£ŤåŻđçăĄïijŽ

```
try:
    out_bytes = subprocess.check_output(['cmd','arg1','arg2'])
except subprocess.CalledProcessError as e:
    out_bytes = e.output        # Output generated before error
    code      = e.returncode    # Return code
```

éżŸèőď æČĚåĘţäÿŃïijŇcheck_output() äżĚäżĚè£ŤåŻđè¿ŞåĚěåĹřæăĞåĞĘè¿ŞåĞžçŽĎåĂijãĂĆ åęĆæđIJä¡ăéIJĂèęĄåŘŇæŮűæŤűéŻĘæăĞåĞĘè¿ŞåĞžåŠŇéŤŹèŕŕè¿ŞåĞžïijŇä¡£çŤÍ stderr åŘĆæŢřïijŽ

```
out_bytes = subprocess.check_output(['cmd','arg1','arg2'],
                                    stderr=subprocess.STDOUT)
```

åęĆæđIJä¡ăéIJĂèęĄçŤÍäÿĂäÿłèűĚæŮűæIJžåĹűæİěæĽğèąŇåŚ¡äżď ïijŇä¡£çŤÍ timeout åŘĆæŢřïijŽ

```
try:
    out_bytes = subprocess.check_output(['cmd','arg1','arg2'],␣
↪timeout=5)
except subprocess.TimeoutExpired as e:
    ...
```

éĂŽåÿÿæİěèőšïijŇåŚ¡äżď çŽĎæĽğèąŇäÿ■éIJĂèęĄä¡£çŤÍåĹřåžŢåśĆshellçŐŕåćČïijĹæŕŤåęĆshãĂĄbash äÿĂäÿłå■ŮçňęäÿšåĹŮèąĺäijŽèćńäijăéĂŠçżŹäÿĂäÿłä¡Őçžğçşżçż§åŚ¡äżď ïijŇæŕŤåęĆ os.execve() ãĂĆ åęĆæđIJä¡ăæČşèőľ åŚ¡äżď ècńäÿĂäÿłshellæĽğèąŇïijŇäijăéĂŠäÿĂäÿłå■ŮçňęäÿšåŘĆæŢřïij shell=True. æIJĽæŮűåĂŹä¡ăæČşèęĄPythonåŐżæĽğèąŇäÿĂäÿłåď■æİĆçŽĎshellåŚ¡äżď çŽĎæŮűåĂŹè

```
out_bytes = subprocess.check_output('grep python | wc > out',␣
↪shell=True)
```

éIJĂèęĄæşĺæĎŘçŽĎæŸŕåIJĺshelläÿ■æĽğèąŇåŚ¡äżď äijŽå■ŸåIJĺäÿĂåőŽçŽĎåőĽåĚĺéčŐéŹľïijŇçĽźåĹ è£ŹæŮűåĂŹåŘŕäżěä¡£çŤÍ shlex.quote() åĞ¡æŢřæİěåřĘåŘĆæŢřæ■čçąőçŽĎçŤĺåŘŇåijŢçŤĺåijŢèţůæİěã

èőĺèőž

ä¡£çŤÍ check_output() åĞ¡æŢřæŸŕæĽğèąŇåď ŰéČĺåŚ¡äżď åźűèŐůåŘŰåĚűè£ŤåŻđåĂijçŽĎæIJĂçõ ä¡ĘæŸŕïijŇåęĆæđIJä¡ăéIJĂèęĄåŕźå■Řè£ŻçĺŃåĄŽæŻť åď■æİĆçŽĎäžď äžŠïijŇæŕŤåęĆçżŹåőČåŘŚéĂĄè¿Ş è£ŹæŮűåĂŹåŘŕçŻť æŐěä¡£çŤÍ subprocess.Popen çśżãĂĆä¿ŃåęĆïijŽ

```
import subprocess

# Some text to send
text = b'''
hello world
this is a test
goodbye
'''

# Launch a command with pipes
p = subprocess.Popen(['wc'],
          stdout = subprocess.PIPE,
          stdin = subprocess.PIPE)

# Send the data and get the output
stdout, stderr = p.communicate(text)
```

(continues on next page)

(continued from previous page)

```
# To interpret as text, decode
out = stdout.decode('utf-8')
err = stderr.decode('utf-8')
```

subprocess æĺąåİŮåŕźäžŐä¿İèţŰTTYçŽĎåď ŰéČĺåŚ¡äżď äÿ■åŘĺéĂĆçŤĺãĂĆ
ä¿ŃåęĆïijŇä¡ääÿ■èČ¡ä¡£çŤĺåőČæİěèĞłåŁĺåŇŰäÿĂäÿłçŤĺæĹůè¿ŞåĚěåŕĘçăĄçŽĎäżżåŁąïijĹæŕŤåęĆäÿĂäÿłss
è£ŹæŮűåĂŹïijŇä¡ăéIJĂèęĄä¡£çŤĺåĹřçňňäÿĽæŰźæĺąåİŮäžĘïijŇæŕŤåęĆå§žäžŐèŚŮåŘ■çŽĎ
expect åőűæŮŘçŽĎåůěåĚůïijĹpexpectæĹŰçśżäijijçŽĎïijĽ

15.7 13.7 åď■åĹűæĹŰèĂĚçğżåŁĺæŰĞäżűåŠŇçŻőå¡Ţ

éŮőéćŸ

ä¡ăæČşèęĄåď■åĹűæĹŰçğżåŁĺæŰĞäżűåŠŇçŻőå¡ŢïijŇä¡ĘæŸŕåŘĺäÿ■æČşèřČçŤĺshellåŚ¡äżď ãĂĆ

èğčåĘşæŰźæąĹ

shutil æĺąåİŮæIJĽå¿Ĺåď Žä¿£æ■ůçŽĎåĞ¡æŢřåŘŕäžěåď■åĹűæŰĞäżűåŠŇçŻőå¡ŢãĂĆä¡£çŤĺèţůæİěé

```
import shutil

# Copy src to dst. (cp src dst)
shutil.copy(src, dst)

# Copy files, but preserve metadata (cp -p src dst)
shutil.copy2(src, dst)

# Copy directory tree (cp -R src dst)
shutil.copytree(src, dst)

# Move src to dst (mv src dst)
shutil.move(src, dst)
```

è£ŹäžŻåĞ¡æŢřçŽĎåŘĆæŢřéČ¡æŸŕå■Ůçňęäÿšå¡ćåijŘçŽĎæŰĞäżűæĹŰçŻőå¡ŢåŘ■ãĂĆ
åžŢåśĆèŕ■äźĽæĺąæŃ§äžĘçśżäijijçŽĎUnixåŚ¡äżď ïijŇåęĆäÿŁéİćçŽĎæşĺéĞŁéČĺåĹĘãĂĆ

éżŸèőď æČĚåĘţäÿŃïijŇåŕźäžŐçňęåŘůéŞ¿æŐěèĂŇåůšè£ŹäžŻåŚ¡äżď åď ĎçŘĘçŽĎæŸŕåőČæŇĞåŘŚçŽ
ä¿ŃåęĆïijŇåęĆæđIJæžŘæŰĞäżűæŸŕäÿĂäÿłçňęåŘůéŞ¿æŐěïijŇéĆčäźĹçŻőæăĞæŰĞäżűåřĘäijŽæŸŕçňęåŘůé
åęĆæđIJä¡ääŘłæČşåď■åĹűçňęåŘůéŞ¿æŐěæIJňèžńïijŇéĆčäźĹéIJĂèęĄæŇĞåőŽåĚşéŤőå■ŮåŘĆæŢř
follow_symlinks ,åęĆäÿŃïijŽ

åęĆæđIJä¡ăæČşä£ĺçŢŹèćńåď■åĹűçŻőå¡Ţäÿ■çŽĎçňęåŘůéŞ¿æŐěïijŇåČŘè£ŹæăůåĄŽïijŽ

```
shutil.copytree(src, dst, symlinks=True)
```

copytree() åŘŕäžěèőľä¡ššIJĺåď■åĹűè£ĞçÍŃäÿ■éĂĽæŃľæĂğçŽĎå£¡çŢěæ§ŘäžŻæŰĞäžűæĹŰçŻőå
ä¡ššŘŕäžěæŘŘä¿ŻäÿĂäÿłå£¡çŢěåĞ¡æŢřïjŇæŐěåŘŮäÿĂäÿłçŻőå¡ŢåŘ■åŠŇæŰĞäžűåŘ■åĹŮèąĺä¡IJäÿžè¿ŞåĚ

```
def ignore_pyc_files(dirname, filenames):
    return [name in filenames if name.endswith('.pyc')]

shutil.copytree(src, dst, ignore=ignore_pyc_files)
```

çŤśäžŐå£¡çŢěæ§Řçğ■æĺąåijŘçŽĎæŰĞäžűåŘ■æŸŕå¿Ĺåÿÿèğ ĄçŽĎïijŇåŻäæ■ď'äÿĂäÿłä¿£æ■ůçŽĎåĞ¡æ
ignore_patterns() åůšçżŘåŇĚåŘńåIJĺéĞŇéĺćäžĘãĂĆä¿ŃåęĆïijŽ

```
shutil.copytree(src, dst, ignore=shutil.ignore_patterns('*~', '*.pyc
↪'))
```

èőĺèőž

ä¡£çŤĺ shutil åď■åĹűæŰĞäžűåŠŇçŻőå¡Ţäź§å£ŠçőĂå■ŢäžĘçĆźåŘğãĂĆ
äÿ■è£ĞïijŇåŕźäžŐæŰĞäžűåĚĆæŢřæ■őä£ąæĄŕïijŇcopy2() è£ŹæăůçŽĎåĞ¡æŢřåŘłèČ¡ăř¡èĞłåůśæIJĂåď'ğè
èő£éŮőæŮűéŮť'ãĂĄåĹŻåżżæŮűéŮť'åŠŇæĺČéŹŘè£ŹäžŻå§žæIJňä£ąæĄŕäijŽèćńä£ĺçŢŹïijŇ
ä¡ĘæŸŕåŕźäžŐæĽ'ĂæIJĽ'èĂĚãĂĄACLsãĂĄèţĎæžŘforkåŠŇåĚűäžŰæŻť'æůśåśĆæňąçŽĎæŰĞäžűåĚČä£ąæA
è£ŹäÿłèŸå¿Ůä¿ĺèţŰäžŐåžŢåśĆæŞ■ä¡IJçşżçż§çśżåđŃåŠŇçŤĺæĽůæĽ'ĂæŃěæIJĽçŽĎèő£éŮőæĺČéŹŘãĂĆ
ä¡äéĂŽåÿÿäÿ■äijŽåŐżä¡£çŤĺ shutil.copytree() åĞ¡æŢřæĺěæĽ'ğèąŇçşżçż§åď'Ğäż¡ãĂĆ
å¡Şåď'ĎçŘĘæŰĞäžűåŘ■çŽĎæŮűåĂŹïijŇæIJĂåě¡ä¡£çŤĺ os.path
äÿ■çŽĎåĞ¡æŢřæĺěçąőä£ĺæIJĂåď'ğçŽĎåŘŕçğżæď■æĂğïijĹçĽźåĹńæŸŕåŘŇæŮűèęĄéĂĆçŤĺäžŐUnixåŠŇWi
ä¿ŃåęĆïijŽ

```
>>> filename = '/Users/guido/programs/spam.py'
>>> import os.path
>>> os.path.basename(filename)
'spam.py'
>>> os.path.dirname(filename)
'/Users/guido/programs'
>>> os.path.split(filename)
('/Users/guido/programs', 'spam.py')
>>> os.path.join('/new/dir', os.path.basename(filename))
'/new/dir/spam.py'
>>> os.path.expanduser('~/guido/programs/spam.py')
'/Users/guido/programs/spam.py'
>>>
```

ä¡£çŤĺ copytree() åď■åĹűæŰĞäžűåď'źçŽĎäÿĂäÿłæčŸæĽ'ŃçŽĎéŮőéćŸæŸŕåŕźäžŐéŤŹèŕŕçŽĎåď'Ď
ä¿ŃåęĆïijŇåIJĺåď■åĹűè£ĞçÍŃäÿ■ïijŇåĞ¡æŢřåŘŕèČ¡äijŽçćřåĹřæ■§åĺŘçŽĎçňęåŘůéŞ¿æŐěïijŇåŻääÿżæĺČéŹ
äÿżäžĘèğčåĘşè£ŹäÿłéŮőéćŸïijŇæĽ'ĂæIJĽçćřåĹřçŽĎéŮőéćŸäijŽèćńæŤűéŻęåĹřäÿĂäÿłåĹŮèąĺäÿ■åźűæĽ'Ş
äÿŃéĺćæŸŕäÿĂäÿłä¿Ňå■ŘïijŽ

```
try:
    shutil.copytree(src, dst)
except shutil.Error as e:
```

(continues on next page)

(continued from previous page)

```
    for src, dst, msg in e.args[0]:
        # src is source name
        # dst is destination name
        # msg is error message from exception
        print(dst, src, msg)
```

åęĆæđIJä¡ăæŘŘä¿ŻåĚşéŤőå■ŮåŘĆæŢř ignore_dangling_symlinks=True ïijŇè£ŹæŮűåĂŹ copytree() äijŽå£¡çŢĕæŐĽæŮăæŢĹçňęåŘůéŞ¿æŐĕãĂĆ

æIJňèŁĆæijŤçď žçŽĎè£ŹäžŻåĞ¡æŢřéČ¡æŸŕæIJĂåÿÿèğĄçŽĎãĂĆäÿ■è£ĞïijŇshutil è£ŸæIJĽæŻť'åď'ŽçŽĎåŠŇåď'■åĹűæŢřæ■őçŻÿåĚşçŽĎæŞ■ä¡IJãĂĆ åőČçŽĎæŰĞæącå¿ĹåĂijå¿ŮäÿĂçIJŃïijŇåŘĆèĂČ Python documentation

15.8 13.8 åĹŻåżżåŠŇèğčåŐŃå¡Šæącæ̆ŰĞäżű

éŮőécŸ

ä¡ăéIJĂèęĄåĹŻåżżæĹŰèğčåŐŃåÿÿèğĄæăijåijŘçŽĎå¡Šæącæ̆ŰĞäżűïijĹæŕŤåęĆ.tar, .tgzæĹŰ.zipïijĽ

èğčåĘşæŰźæąĹ

shutil æĺąåİŮæŃĕæIJĽäÿď'äÿłåĞ¡æŢřâĂŤâĂŤ make_archive() åŠŇ unpack_archive() åŘŕæť'¿äÿŁçŤĺåIJžãĂĆ ä¿ŃåęĆïijŽ

```
>>> import shutil
>>> shutil.unpack_archive('Python-3.3.0.tgz')

>>> shutil.make_archive('py33','zip','Python-3.3.0')
'/Users/beazley/Downloads/py33.zip'
>>>
```

make_archive() çŽĎçňňäžŃäÿłåŘĆæŢřæŸŕæIJ§æIJŻçŽĎè¿ŞåĞžæăijåijŘãĂĆ åŘŕäżěä¡£çŤĺ get_archive_formats() èŐůåŘŰæĽĂæIJĽæŤŕæŃçŽĎå¡Šæącæăijåij̇ŘåĹŮèąĺãĂĆä¿Ń

```
>>> shutil.get_archive_formats()
[('bztar', "bzip2'ed tar-file"), ('gztar', "gzip'ed tar-file"),
 ('tar', 'uncompressed tar file'), ('zip', 'ZIP file')]
>>>
```

èőĺèőž

Pythonè£ŸæIJĽåĚűäżŰçŽĎæĺąåİŮåŘŕçŤĺæĺěåď'ĎçŘĘåď'Žçğ■å¡Šæącæăijåij̇ŘïijĹæŕŤåęĆtarfile, zipfile, gzip, bz2ïijĽçŽĎåžŢåśĆçżĘèŁĆãĂĆ äÿ■è£ĞïijŇåęĆæđIJä¡ääżĚäżĚåŘłæŸŕèęĄåĹŻåżżæĹŰæŘŘåŘŰ åŘŕäżěçŻť'æŐěä¡£çŤĺ shutil äÿ■çŽĎè£ŹäžŻéńŸåśĆåĞ¡æŢřãĂĆ

è£ŹäžŻåĞ¡æŢřè£ŸæIJL’å¿Ĺåď’ŽåĚűäžŰéĂL’éąźïijŇçŤĺäžŐæŮěå£ŮæL’Şå■řãĂĄéĆĎæčĂãĂĄæŰĞäžűå
åŘĆèĂČ shutilæŰĞæąč

15.9 13.9 éĂŽè£ĞæŰĞäžűåŘ■æ§ěæL’¿æŰĞäžű

éŮőéćŸ

ä¡ăéIJĂèęĄåĘŹäÿĂäÿłæűL’åŘŁåĹřæŰĞäžűæ§ěæL’¿æŞ■ä¡IJçŽĎèĎŽæIJňïijŇæŕŤåęĆåŕźæŮěå£Ůå¡Šæą
ä¡ääÿ■æČşåIJĺPythonèĎŽæIJňäÿ■èřČçŤĺshellïijŇæĹŰèĂĚä¡äèęĄåőđçŐřäÿĂäžŻshelläÿ■èČ¡åĄŽçŽĎåŁ§èČ

èğčåĘşæŰźæąĹ

æ§ěæL’¿æŰĞäžűïijŇåŘŕä¡£çŤĺ `os.walk()` åĞ¡æŢřïijŇäijääÿĂäÿłéąűçžğçŻőå¡ŢåŘ■çżŹåőČãĂĆ
äÿŇéĺćæŸŕäÿĂäÿłä¿Ňå■ŘïijŇæ§ěæL’¿çL’źåőŽçŽĎæŰĞäžűåŘ■åźűç■ŤåžŤæL’ĂæIJL’çňęåŘĹæĺąäžűçŽĎæŰĞ

```
#!/usr/bin/env python3.3
import os

def findfile(start, name):
    for relpath, dirs, files in os.walk(start):
        if name in files:
            full_path = os.path.join(start, relpath, name)
            print(os.path.normpath(os.path.abspath(full_path)))

if __name__ == '__main__':
    findfile(sys.argv[1], sys.argv[2])
```

ä£Ĺå■ŸèĎŽæIJňäÿžæŰĞäžűfindfile.pyïijŇçĎűåŘŐåIJĺåŚ¡äžď’èąŇäÿ■æL’ğèąŇåőČãĂĆ
æŇĞåőŽåĹĺåğŃæ§ěæL’¿çŻőå¡ŢäžěåŘŁåŘ■å■Ůä¡IJäÿžä¡■ç¡őåŘĆæŢřïijŇåęĆäÿŃïijŽ

èőĺèőž

`os.walk()` æŰźæşŢäÿžæĹŚäžňéĄ■åŐEçŻőå¡ŢæăŚïijŇ
æŕŘæňąè£ŻåĚěäÿĂäÿłçŻőå¡ŢïijŇåőČäijŽè£ŤåŻđäÿĂäÿłäÿL’åĚČçżĎïijŇåŇĚåŘńçŻÿåŕźäžŐæ§ěæL’¿çŻőå¡Ţ
äžěåŘŁéĆčäÿłçŻőå¡ŢäÿŃéĺćçŽĎæŰĞäžűåŘ■åĹŮèąĺãĂĆ

åŕźäžŐæŕŘäÿłåĚČçżĎïijŇåŘłéIJĂæčĂæţŃäÿĂäÿŃçŻőæăĞæŰĞäžűåŘ■æŸŕåŘęåIJĺæŰĞäžűåĹŮèąĺäÿ■
`os.path.join()` åŘĹåźűèůŕå¿ĎãĂĆ äÿžäžĘéĄ£åĚ■åěĞæĂłçŽĎèůŕå¿ĎåŘ■æŕŤåęĆ `./` `./foo//bar` ïijŇä¡£çŤĺäžĘåŘęåď’Ůäÿď’äÿłåĞ¡æŢřæĺěä£őæ■čçżŞæđIJãĂĆ çňňäÿĂäÿłæŸŕ `os.path.abspath()` ,åőČæŐěåŘŮäÿĂäÿłèůŕå¿ĎïijŇåŘŕèČ¡æŸŕçŻÿåŕźèůŕå¿ĎïijŇæIJĂåŘŐè£ŤåŻđçżĺåŕ
çňňäžŇäÿłæŸŕ `os.path.normpath()` ïijŇçŤĺæĺěè£ŤåŻđæ■čåÿÿèůŕå¿ĎïijŇåŘŕäžěèğčåĘşåŘŇæŰIJæĺĘã

åř¡çőąè£ŹäÿłèĎŽæIJňçŻÿåŕźäžŐUNIXåźşåŘŕäÿŁéĺćçŽĎå¿Ĺåď’Žæ§ěæL’¿æĺěěőšèęĄçőĂå■Ţå¿Ĺåď’Žïij
åźűäÿŤïijŇè£ŸèČ¡å¿Ĺè¡żæĺ¿çŽĎåŁäåĚěåĚűäžŰçŽĎåŁ§èČ¡ãĂĆ
æĹŚäžňåĘ■æijŤçď’žäÿĂäÿłä¿Ňå■ŘïijŇäÿŃéĺćçŽĎåĞ¡æŢřæL’Şå■řæL’ĂæIJL’æIJĂè£Śèćńä£őæŤźè£ĞçŽĎæŰ

```
#!/usr/bin/env python3.3

import os
import time

def modified_within(top, seconds):
    now = time.time()
    for path, dirs, files in os.walk(top):
        for name in files:
            fullpath = os.path.join(path, name)
            if os.path.exists(fullpath):
                mtime = os.path.getmtime(fullpath)
                if mtime > (now - seconds):
                    print(fullpath)

if __name__ == '__main__':
    import sys
    if len(sys.argv) != 3:
        print('Usage: {} dir seconds'.format(sys.argv[0]))
        raise SystemExit(1)

    modified_within(sys.argv[1], float(sys.argv[2]))
```

åIJÍæ■d'åĞ¡æŢřçŽĎå§žçąĂäźŃäÿŁïijŇä¡£çŤÍos,os.path,globç■L'çśżäijijæÍąåİŮïijŇä¡ääřśèČ¡åőđçŐřæŻt
åŘřåŘĆèĂČ5.11åřŘèŁĆåŠŇ5.13åřŘèŁĆç■L'çŻÿåĚşçńäèŁĆãĂĆ

15.10 13.10 èŕżåŘŰéĚ■ç¡őæŰĞäżű

éŮőéćŸ

æĂŐæäůèŕżåŘŰæŹőéĂŽ.iniæäijåijŘçŽĎéĚ■ç¡őæŰĞäżűïij§

èğčåĘşæŰźæąĹ

configparser æÍąåİŮèČ¡èćńçŤÍæİěèŕżåŘŰéĚ■ç¡őæŰĞäżűãĂĆä¿ŃåęĆïijŇåĄĞèő¿ä¡äæIJL'åęĆäÿŃç

```
; config.ini
; Sample configuration file

[installation]
library=%(prefix)s/lib
include=%(prefix)s/include
bin=%(prefix)s/bin
prefix=/usr/local

# Setting related to debug configuration
[debug]
```

(continues on next page)

(continued from previous page)

```
log_errors=true
show_warnings=False

[server]
port: 8080
nworkers: 32
pid-file=/tmp/spam.pid
root=/www/root
signature:
    =================================
    Brought to you by the Python Cookbook
    =================================
```

äÿŃéĮćæŸŕäÿĂäÿłèŕżåŔŰåŠŃæŔŘåŔŰåĚűäÿ■åĂijçŽĎä¿Ńå■ŘïijŽ

```
>>> from configparser import ConfigParser
>>> cfg = ConfigParser()
>>> cfg.read('config.ini')
['config.ini']
>>> cfg.sections()
['installation', 'debug', 'server']
>>> cfg.get('installation','library')
'/usr/local/lib'
>>> cfg.getboolean('debug','log_errors')

True
>>> cfg.getint('server','port')
8080
>>> cfg.getint('server','nworkers')
32
>>> print(cfg.get('server','signature'))

\=================================
Brought to you by the Python Cookbook
\=================================
>>>
```

åęĆæđIJæIJL'éIJĂèęĄïijŃä¡ăè£ŸèČ¡ä£őæŤźéĚ■ç¡őåźűä¡£çŤÍ `cfg.write()` æŰźæşŢåřĘåĚűåĘŹåŻđåĹřæŰĞäżűäÿ■ãĂĆä¿ŃåęĆïijŽ

```
>>> cfg.set('server','port','9000')
>>> cfg.set('debug','log_errors','False')
>>> import sys
>>> cfg.write(sys.stdout)
```

```
[installation]
library = %(prefix)s/lib
include = %(prefix)s/include
bin = %(prefix)s/bin
```

(continues on next page)

(continued from previous page)

```
prefix = /usr/local

[debug]
log_errors = False
show_warnings = False

[server]
port = 9000
nworkers = 32
pid-file = /tmp/spam.pid
root = /www/root
signature =
          ==================================
          Brought to you by the Python Cookbook
          ==================================
>>>
```

èőÍèőž

éĚ■ç¡őæŰĞäžűä¡IJäÿžäÿĂçğ■åŘŕèŕżæĂğå¿Ĺåě¡çŽĎæăijåijŘïijŇéİđåÿÿéĂĆçŤÍäžŐå■ŸåĆÍçÍŃåžŘäÿ■çŽ
åIJÍæŕŘäÿłéĚ■ç¡őæŰĞäžűäÿ■ïijŇéĚ■ç¡őæŢřæ■őäijŽèćńåĹĘçżĎïijĹæŕŤåęĆä¿Ńå■Řäÿ■çŽĎâĂIJinstallationâ
âĂIJdebugâĂİ åŠŇ âĂIJserverâĂİïijL'ãĂĆ æŕŘäÿłåĹĘçżĎåIJÍåĚűäÿ■æŇĞåőŽåŕźåžŤçŽĎåŘĎäÿłåŘŸéĞŘåĂ

åŕźäžŐåŘŕåőđçŐřåŘŇæăůåŁ§èČ¡çŽĎéĚ■ç¡őæŰĞäžűåŠŇPythonæžŘæŰĞäžűæŸŕæIJL'å¿Ĺåď'ğçŽĎäÿ■
éęŰåĚĹïijŇéĚ■ç¡őæŰĞäžűçŽĎèŕ■æşŢèęĄæŻť'èĞłçŤśäžŻïijŇäÿŃéİćçŽĎètŃåĂijèŕ■åŘěæŸŕç■L'æŢĹçŽĎïij

```
prefix=/usr/local
prefix: /usr/local
```

éĚ■ç¡őæŰĞäžűäÿ■çŽĎåŘ■å■ŮæŸŕäÿ■åŇžåĹĘåď'ğåřŘåĘŹçŽĎãĂĆä¿ŃåęĆïijŽ

```
>>> cfg.get('installation','PREFIX')
'/usr/local'
>>> cfg.get('installation','prefix')
'/usr/local'
>>>
```

åIJÍèğčæđŘåĂijçŽĎæŮűåĂŹïijŇgetboolean() æŰźæşŢæ§ěæL'¿äżä¡ŢåŘŕèąŇçŽĎåĂijãĂĆä¿ŃåęĆ

```
log_errors = true
log_errors = TRUE
log_errors = Yes
log_errors = 1
```

æĹŰèőÿéĚ■ç¡őæŰĞäžűåŠŇPythonäžčçăĄæIJĂåď'ğçŽĎäÿ■åŘŇåIJÍäžŐïijŇåőČåźűäÿ■æŸŕäżŐäÿŁèĂŇ
æŰĞäžűæŸŕåőL'èčĚäÿĂäÿłæŢť'ä¡Ş èćńèŕżåŘŰçŽĎãĂĆåęĆæđIJçćřåĹřäžĘåŘŸéĞŘæŻ£æ■ćïijŇåőČåőđéŹĚä
ä¿ŃåęĆïijŇåIJÍäÿŃéİćè£ŹäÿłéĚ■ç¡őäÿ■ïijŇprefix åŘŸéĞŘåIJÍä¡£çŤÍåőČçŽĎåŘŸéĞŘäźŃåL'■æĹŰäźŃåŘ

```
[installation]
library=%(prefix)s/lib
include=%(prefix)s/include
bin=%(prefix)s/bin
prefix=/usr/local
```

`ConfigParser` æIJL'äÿłåőźæŸŞèćńå£¡èğĘçŽĎçL'źæĂğæŸŕåőČèČ¡äÿĂæňąèŕżåŘŰåď'ŽäÿłéĚ■ç¡őæŰ
ä¿ŃåęĆïijŇåĄĞèő¿äÿĂäÿłçŤĺæĹůåČŘäÿŃéİćè£ŹæăůæđĎéĂăäžĘäżŰäżňçŽĎéĚ■ç¡őæŰĞäżűïijŽ

```
; ~/.config.ini
[installation]
prefix=/Users/beazley/test

[debug]
log_errors=False
```

èŕżåŘŰè£ŹäÿłæŰĞäżűïijŇåőČåřśèČ¡èů§äźŃåL'■çŽĎéĚ■ç¡őåŘĹåźűèţůæİěãĂĆåęĆïijŽ

```
>>> # Previously read configuration
>>> cfg.get('installation', 'prefix')
'/usr/local'

>>> # Merge in user-specific configuration
>>> import os
>>> cfg.read(os.path.expanduser('~/.config.ini'))
['/Users/beazley/.config.ini']

>>> cfg.get('installation', 'prefix')
'/Users/beazley/test'
>>> cfg.get('installation', 'library')
'/Users/beazley/test/lib'
>>> cfg.getboolean('debug', 'log_errors')
False
>>>
```

äżŤçżĘèğĆåŕ§äÿŃ `prefix` åŘŸéĞŘæŸŕæĂŐæăůèęĘçŻŰåĚűäżŰçŻÿåĚşåŘŸéĞŘçŽĎïijŇæŕŤåęĆ
`library` çŽĎèő¿åőŽåĂijãĂĆ äžğçŤ§è£Źçğ■çżŞæđIJçŽĎåŐ§åŻäæŸŕåŘŸéĞŘçŽĎæŤźåĘźéĞĞåŘŰçŽĎæ
ä¡ääŘŕäżěåČŘäÿŃéİćè£ŹæăůåĄŽèŕŢéłŇïijŽ

```
>>> cfg.get('installation','library')
'/Users/beazley/test/lib'
>>> cfg.set('installation','prefix','/tmp/dir')
>>> cfg.get('installation','library')
'/tmp/dir/lib'
>>>
```

æIJĂåŘŐè£ŸæIJL'å¿ĹéĞ■èęĄäÿĂçĆźèęĄæşĺæĎŘçŽĎæŸŕPythonåźűäÿ■èČ¡æŤŕæŇĄ.iniæŰĞäżűåIJĺåĺ
çąőä£İä¡ääůšçżŘåŘĆéŸĚäžĘconfigparseræŰĞæąčäÿ■çŽĎèŕ■æşŢèŕęæČĚäżěåŘŁæŤŕæŇĄçL'źæĂğãĂĆ

15.11 13.11 çżŹçőĂå■ŢèĎŽæIJňåćđåŁăæŮěå£ŮåŁ§èČ¡

éŮőéćŸ

ä¡ăåÿŇæIJŻåIJÍèĎŽæIJňåŠŇçÍŃåžŘäÿ■åřĘèŕŁæŰ■ä£ąæĄŕåĘŹåĚěæŮěå£ŮæŰĞäżűãĂĆ

èğčåĘşæŰźæąĹ

æĽŞå■řæŮěå£ŮæIJĂçőĂå■ŢæŰźåijŘæŸŕä¡£çŤÍ `logging` æÍąåİŮãĂĆä¿ŃåęĆïijŽ

```
import logging

def main():
    # Configure the logging system
    logging.basicConfig(
        filename='app.log',
        level=logging.ERROR
    )

    # Variables (to make the calls that follow work)
    hostname = 'www.python.org'
    item = 'spam'
    filename = 'data.csv'
    mode = 'r'

    # Example logging calls (insert into your program)
    logging.critical('Host %s unknown', hostname)
    logging.error("Couldn't find %r", item)
    logging.warning('Feature is deprecated')
    logging.info('Opening file %r, mode=%r', filename, mode)
    logging.debug('Got here')

if __name__ == '__main__':
    main()
```

äÿŁéİćäžŤäÿłæŮěå£ŮèřČçŤÍïijĹcritical(), error(), warning(), info(), debug()ïijĽäžěéŹ■åžŘæŰźåijŘèąÍçď žäÿ■åŘŇçŽĎäÿěéĞ■çžğåĹńãĂĆ `basicConfig()` çŽĎ `level` åŘĆæŢřæŸŕäÿĂäÿłè£ĞæżďåŹÍãĂĆ æĽĂæIJĽçžğåĹńä¡ŐäžŐæ■ďçžğåĹńçŽĎæŮěå£ŮæŰĹæĄŕéČ¡äijŽèćńå£¡çŢěæŐĽãĂĆ æŕŘäÿłloggingæŞ■ä¡IJçŽĎåŘĆæŢřæŸŕäÿĂäÿłæŰĹæĄŕå■ŮçňęäÿšïijŇåŘŐéİćåĘ■èů§äÿĂäÿłæĹŰåď ŽäÿłåŘĆ æđĎéĂăæIJĂçżĹçŽĎæŮěå£ŮæŰĹæĄŕçŽĎæŮűåĂŹæĹŚäžňä¡£çŤÍäžĘ%æŞ■ä¡IJçňęæİěæăijåijŘåŇŰæŰĹæĄ

è£ŘèąŇè£ŹäÿłçÍŃåžŘåŘŐïijŇåIJÍæŰĞäżű `app.log` äÿ■çŽĎåĘĚåőźåžŤèŕěæŸŕäÿŃéİćè£ŹæăůïijŽ

```
CRITICAL:root:Host www.python.org unknown
ERROR:root:Could not find 'spam'
```

åęĆæđIJä¡ăæČşæŤźåŘŸè¿ŞåĞžç■ĽçžğïijŇä¡ăåŘŕäžěä£őæŤź `basicConfig()` èřČçŤÍäÿ■çŽĎåŘĆæŢřãĂĆä¿ŃåęĆïijŽ

```
logging.basicConfig(
     filename='app.log',
     level=logging.WARNING,
     format='%(levelname)s:%(asctime)s:%(message)s')
```

æIJĂåŘŐè¿ŞåĞžåŘŸæĹŘåęĆäÿŃïijŽ

```
CRITICAL:2012-11-20 12:27:13,595:Host www.python.org unknown
ERROR:2012-11-20 12:27:13,595:Could not find 'spam'
WARNING:2012-11-20 12:27:13,595:Feature is deprecated
```

äÿŁéİćçŽĎæŮěå£ŮéĚ■ç¡őéČ¡æŸŕçąňçijŰçăĄåĹřçÍŃåžŘäÿ■çŽĎãĂĆåęĆæđIJä¡ăæČşä¡£çŤÍéĚ■ç¡őæŰČ åŘŕäżěåČŘäÿŃéİćè£Źæăůä£őæŤź `basicConfig()` èřČçŤÍïijŽ

```
import logging
import logging.config

def main():
    # Configure the logging system
    logging.config.fileConfig('logconfig.ini')
    ...
```

åĹŻåżžäÿĂäÿłäÿŃéİćè£Źæăůç ŽĎæŰĞäżűïijŇåŘ■å■ŮåŘń `logconfig.ini` ïijŽ

```
[loggers]
keys=root

[handlers]
keys=defaultHandler

[formatters]
keys=defaultFormatter

[logger_root]
level=INFO
handlers=defaultHandler
qualname=root

[handler_defaultHandler]
class=FileHandler
formatter=defaultFormatter
args=('app.log', 'a')

[formatter_defaultFormatter]
format=%(levelname)s:%(name)s:%(message)s
```

åęĆæđIJä¡ăæČşä£őæŤźéĚ■ç¡őïijŇåŘŕäżěçŻť æŐěçijŰè¿ŚæŰĞäżűlogconfig.iniå■şåŘŕãĂĆ

èőÍèőž

åř¡çőąåŕźäžŐ `logging` æÍąåİŮèĂŇåůšæIJL'å¿Ĺåď'ŽæŻť'éńŸçžğçŽĎéĚ■ç¡őéĂL'éąźïijŇäÿ■è£ĞèŹéĞŇçŽĎæŰźæąĹåŕźäžŐçőĂå■ŢçŽĎçÍŃåžŘåŠŇèĎŽæIJňåůšçżŘèűşåď'§äžĘãĂĆåŘłæČşåIJÍèřČçŤÍæŮěå£ŮæŞ■ä¡IJåL'■åĚĹæL'ğèąŇäÿŃbasicConfig()åĞ¡æŢřæŰźæşŢïijŇä¡äçŽĎçÍŃåžŘåŕśè

åęĆæđIJä¡äæČşèęĄä¡äçŽĎæŮěå£ŮæűĹæĄŕåĘŹåĹřæăĞåĞĘéŤŹèŕŕäÿ■ïijŇèĂŇäÿ■æŸŕæŮěå£ŮæŰĞäż `basicConfig()` æŮűäÿ■äijäæŰĞäżűåŘ■åŘĆæŢřå■şåŘŕãĂĆä¿ŃåęĆïijŽ

```
logging.basicConfig(level=logging.INFO)
```

`basicConfig()` åIJÍçÍŃåžŘäÿ■åŘłèČ¡èćńæL'ğèąŇäÿĂæňąãĂĆåęĆæđIJä¡äçÍ■åŘŐæČşæŤźåŘŸæŮěå åŕśéIJĂèęĄåĚĹèŐůåŘŰ `root logger` ïijŇçĎűåŘŐçŻť'æŐëä£őæŤźåőČãĂĆä¿ŃåęĆïijŽ

```
logging.getLogger().level = logging.DEBUG
```

éIJĂèęĄåijžèřČçŽĎæŸŕæIJňèŁĆåŘłæŸŕæijŤçď'žäžĘ `logging` æÍąåİŮçŽĎäÿĂäžŻå§žæIJňçŤÍæşŢãĂĆ åőČåŘŕäżěåĄŽæŻť'åď'ŽæŻť'éńŸçžğçŽĎåőŽåĹűãĂĆ åĚşäžŐæŮěå£ŮåőŽåĹűåŇŰäÿĂäÿłå¿Ĺåě¡çŽĎèţĎæžŘæŸŕ Logging Cookbook

15.12 13.12 çżŹåĞ¡æŢřåžŞåćđåŁäæŮěå£ŮåŁ§èČ¡

éŮőécŸ

ä¡äæČşçżŹæ§ŘäÿłåĞ¡æŢřåžŞåćđåŁäæŮěå£ŮåŁ§èČ¡ïijŇä¡ĘæŸŕåŘĹäÿ■èČ¡å¡śåŞ■åĹřéĆčäžŻäÿ■ä¡£çŤÍ

èğčåĘşæŰźæąĹ

åŕźäžŐæČşèęĄæL'ğèąŇæŮěå£ŮæŞ■ä¡IJçŽĎåĞ¡æŢřåžŞèĂŇåůšïijŇä¡äåžŤèřěåĹŻåżżäÿĂäÿłäÿŞåśđçŽĎ `logger` åŕźèśąïijŇåźűäÿŤåČŘäÿŃéİćè£ŹæăůåĹİåğŃåŇŰéĚ■ç¡őïijŽ

```
# somelib.py

import logging
log = logging.getLogger(__name__)
log.addHandler(logging.NullHandler())

# Example function (for testing)
def func():
    log.critical('A Critical Error!')
    log.debug('A debug message')
```

ä¡£çŤÍè£ŹäÿłéĚ■ç¡őïijŇéżŸèőď'æČĚåĘţäÿŃäÿ■äijŽæL'Şå■řæŮěå£ŮãĂĆä¿ŃåęĆïijŽ

```
>>> import somelib
>>> somelib.func()
>>>
```

äÿ■è£ĞïijŇåęĆæđIJéĚ■ç¡őè£ĞæŮěå£Ůçşżçż§ïijŇéĆčäźĹæŮěå£ŮæűĹæĄŕæĽŞå■řářśåijĂåğŃçŤ§æŢĹï

```
>>> import logging
>>> logging.basicConfig()
>>> somelib.func()
CRITICAL:somelib:A Critical Error!
>>>
```

èőĺèőž

éĂŽåÿÿæİěèőšïijŇä¡ääÿ■åžŤèŕěåIJĺåĞ¡æŢřåžŞäżčçăĄäÿ■èĞłåůśéĚ■ç¡őæŮěå£Ůçşżçż§ïijŇæĹŰèĂĚæŸ

èřČçŤĺ `getLogger(__name__)` åĹŻåżžäÿĂäÿłåŠŇèřČçŤĺæĺąåİŮåŘŇåŘ■çŽĎlog-
geræĺąåİŮãĂĆ çŤśäžŐæĺąåİŮéČ¡æŸŕåŤŕäÿĂçŽĎïijŇåŻäæ■ď åĹŻåżžçŽĎloggeräź§åřĘæŸŕåŤŕäÿĂçŽĎãĂĆ

`log.addHandler(logging.NullHandler())` æŞ■ä¡IJåřĘäÿĂäÿłçľžåď ĎçŘĘåŹĺçżŚåőŽåĹřåĹ
äÿĂäÿłçľžåď ĎçŘĘåŹĺéżŸèőď äijŽå£¡çŢěřČçŤĺæĽĂæIJĽçŽĎæŮěå£ŮæűĹæĄŕãĂĆ
åŻäæ■ď ïijŇåęĆæđIJä¡£çŤĺèŕěåĞ¡æŢřåžŞçŽĎæŮűåĂŹè£ŸæšąæIJĽéĚ■ç¡őæŮěå£ŮïijŇéĆčäźĹåřĘäÿ■äijŽæ

è£ŸæIJĽäÿĂçĆźåřśæŸŕåŕźäžŐåŘĎäÿłåĞ¡æŢřåžŞçŽĎæŮěå£ŮéĚ■ç¡őåŘŕäżěæŸŕçŻÿäžŠçŃňçńŃçŽĎïij
ä¿ŃåęĆïijŇåŕźäžŐåęĆäÿŃçŽĎäżčçăĄïijŽ

```
>>> import logging
>>> logging.basicConfig(level=logging.ERROR)

>>> import somelib
>>> somelib.func()
CRITICAL:somelib:A Critical Error!

>>> # Change the logging level for 'somelib' only
>>> logging.getLogger('somelib').level=logging.DEBUG
>>> somelib.func()
CRITICAL:somelib:A Critical Error!
DEBUG:somelib:A debug message
>>>
```

åIJĺè£ŹéĞŃïijŇæäźæŮěå£ŮèćńéĚ■ç¡őæĹŘäżĚäżĚè¿ŞåĞžERRORæĹŰæŻť éńŸçžğåĹńçŽĎæűĹæĄŕãĂ
äÿ■è£Ğ ïijŇ`somelib` çŽĎæŮěå£ŮçžğåĹńèćńå■ŢçŃňéĚ■ç¡őæĹŘåŘŕäżěè¿ŞåĞždebugçžğåĹńçŽĎæűĹæĄŕïi
åČŘè£ŹæäůæŻť æŤźå■ŢçŃňæĺąåİŮçŽĎæŮěå£ŮéĚ■ç¡őåŕźäžŐèřČèŕŢæİěèőšæŸŕå¿ĹæŰźä¿£çŽĎïijŇ
åŻääÿžä¡äæŮäéIJĂåŐżæŻť æŤźäżżä¡ŢçŽĎåĚĺåśĂæŮěå£ŮéĚ■ç¡ôâŤâĂŤåŘłéIJĂèęĄä£őæŤźä¡äæČşèęĄæŻ

Logging HOWTO èŕęçżĘäżŃçż■äžĘåęĆä¡ŢéĚ■ç¡őæŮěå£ŮæĺąåİŮåŠŇåĚűäżŰæIJĽçŤĺæŁĂåůğïijŇåŘŕä

15.13 13.13 åőđçŐŕäÿĂäÿłèőąæŮűåŹĺ

éŮőécŸ

ä¡äæČşèőŕå¡ŢçĺŃåžŘæĽğèąŇåď Žäÿłäżżå£ąæĽĂèŁśèť źçŽĎæŮűéŮť

èğčåĘşæŰźæąĹ

`time` æĺąåİŮåŇĚåŘńå¿Ĺåď ŽåĞ¡æŢřæİěæĽğèąŇèů§æŮűéŮť æIJĽåĚşçŽĎåĞ¡æŢřãĂĆ åř¡çőąåęĆæ■ď ïijŇéĂŽåÿÿæĹŚäżňäijŽåIJĺæ■ď å§žçąĂäźŃäÿŁæđĎéĂääÿĂäÿłæŻť éńŸçžğçŽĎæŐěåŘčæİěæİ

```
import time

class Timer:
    def __init__(self, func=time.perf_counter):
        self.elapsed = 0.0
        self._func = func
        self._start = None

    def start(self):
        if self._start is not None:
            raise RuntimeError('Already started')
        self._start = self._func()

    def stop(self):
        if self._start is None:
            raise RuntimeError('Not started')
        end = self._func()
        self.elapsed += end - self._start
        self._start = None

    def reset(self):
        self.elapsed = 0.0

    @property
    def running(self):
        return self._start is not None

    def __enter__(self):
        self.start()
        return self

    def __exit__(self, *args):
        self.stop()
```

è£ŹäÿłçśżåőŽäźĽäžĘäÿĂäÿłåŘŕäżěèćńçŤĺæĹůæäźæ■őéIJĂèęĄåŘŕåŁĺãĂĄåĄIJæ■ćåŠŇéĞ■ç¡őçŽĎèőąą åőČäijŽåIJĺ `elapsed` åśđæĂğäÿ■èőřå¡ŢæŢť äÿłæűĹèĂŮæŮűéŮť ãĂĆ äÿŃéĺćæŸŕäÿĂäÿłä¿Ńå■ŘæİěæijŤçď žæĂŐæăůä¡£çŤĺåőČïijŽ

```
def countdown(n):
    while n > 0:
        n -= 1

# Use 1: Explicit start/stop
t = Timer()
t.start()
```

(continues on next page)

(continued from previous page)

```
countdown(1000000)
t.stop()
print(t.elapsed)

# Use 2: As a context manager
with t:
    countdown(1000000)

print(t.elapsed)

with Timer() as t2:
    countdown(1000000)
print(t2.elapsed)
```

èőÍèőž

æIJňèŁĆæŔŘä¿ŻäžĘäÿĂäÿłçőÅå■ŢèĂŇåőđçŤÍçŽĎçśżæÌěåőđçŐřæŮűéŮť'èőřå¡ŢäżěåŔŁèĂŮæŮűèőąç
åŘŇæŮűäź§æŸŕåŕźä¡£çŤÍwithèŕ■åŔěäżěåŔŁäÿŁäÿŃæŰĞçőąçŘĘåŹÍå■ŔèőőçŽĎäÿĂäÿłå¿Ĺåě¡çŽĎæijŤçď'ž

åIJÍèőąæŮűäÿ■èęĄèĂČèŹŚäÿĂäÿłážŢåśĆçŽĎæŮűéŮť'åĞ¡æŢřéŮőécŸãĂĆäÿĂèĹňæÌěèŕť'ïijŇ
ä¡£çŤÍ `time.time()` æĹŰ `time.clock()` èőąçőŮçŽĎæŮűéŮť'çš¿åžęåŻäæŞ■ä¡IJçšżçż§çŽĎäÿ■åŘŇäijŽ
èĂŇä¡£çŤÍ `time.perf_counter()` åĞ¡æŢřåŕźäżěçąőä£Ìä¡£çŤÍçšżçż§äÿŁéÌćæIJĂçš¿çąőçŽĎèőąæŮűåŹ

äÿŁè£řäżčçăĄäÿ■çŤś `Timer` çśżèőřå¡ŢçŽĎæŮűéŮť'æŸŕéŠ§èąĺæŮűéŮť'ïijŇåźűåŇĚåŘńäžĘæĽ'ĂæIJĽ'ä
åęĆæđIJä¡ääŔłæČşèőąçőŮèřěè£ŻçÍŃæĽ'ĂèŁśèť'źçŽĎCPUæŮűéŮť'ïijŇåžŤèřěä¡£çŤÍ `time.process_time()` æÌěäżčæŻ£ïijŽ

```
t = Timer(time.process_time)
with t:
    countdown(1000000)
print(t.elapsed)
```

`time.perf_counter()` åŠŇ `time.process_time()` éČ¡äijŽè£ŤåŻđåřŔæŢřå¡ćåijŔçŽĎçğŠæŢřæŮűéŮť'ãĂĆ ääđéŹĚçŽĎæŮűéŮť'åĂijæšąæIJĽ'äżżä¡ŢæĎŔäźĽ'ïij

æŻť'åď'ŽåĚşäžŐèőąæŮűåŠŇæĂğèČ¡åĹĘæđŘçŽĎä¿Ńå■ŘèřůåŔĆèĂČ14.13åřŔèŁĆãĂĆ

15.14 13.14 éŹŘåĹűåĘĚå■ŸåŠŇCPUçŽĎä¡£çŤÍéĞŔ

éŮőécŸ

ä¡äæČşåŕźåIJÍUnixçšżçż§äÿŁéÌćè£ŘèąŇçŽĎçÍŃåžŔèő¿ç¡őåĘĚå■ŸæĹŰCPUçŽĎä¡£çŤÍéŹŘåĹűãĂĆ

èğčåĘşæŰźæąĹ

`resource` æÍąåÌŮèČ¡åŘŇæŮűæĽ'ğèąŇè£ŹäÿďäÿłäżżåŁąãĂĆä¿ŃåęĆïijŇèęĄéŹŘåĹűCPUæŮűéŮť'ïij

```
import signal
import resource
import os

def time_exceeded(signo, frame):
    print("Time's up!")
    raise SystemExit(1)

def set_max_runtime(seconds):
    # Install the signal handler and set a resource limit
    soft, hard = resource.getrlimit(resource.RLIMIT_CPU)
    resource.setrlimit(resource.RLIMIT_CPU, (seconds, hard))
    signal.signal(signal.SIGXCPU, time_exceeded)

if __name__ == '__main__':
    set_max_runtime(15)
    while True:
        pass
```

çÍŃåžŘè£ŘèąŇæŮűïijŇSIGXCPU ä£ąåŘůåIJÍæŮűéŮť'è£ĞæIJ§æŮűèćńçŤ§æĹŘïijŇçĎűåŘŐæĽ'ğèąŇæÿ

èęĄéŹŘåĹűåĘĚå■Ÿä¡£çŤÍïijŇèő¿ç¡őåŘŕä¡£çŤÍçŽĎæĂżåĘĚå■ŸåĂijå■şåŘŕïijŇåęĆäÿŃïijŽ

```
import resource

def limit_memory(maxsize):
    soft, hard = resource.getrlimit(resource.RLIMIT_AS)
    resource.setrlimit(resource.RLIMIT_AS, (maxsize, hard))
```

åČŘè£Źæåůèő¿ç¡őäžĘåĘĚå■ŸéŹŘåĹűåŘŐïijŇçÍŃåžŘè£ŘèąŇåĹřæšąæIJĽåď'Žä¡ŹåĘĚå■ŸæŮűäijŽæŁŽ MemoryError åijĆåÿÿãĂĆ

èőÍèőž

åIJÍæIJňèŁĆä¿Ńå■Řäÿ■ïijŇsetrlimit() åĞ¡æŢřèćńçŤÍæİěèő¿ç¡őçĽźåőŽèţĎæžŘäÿŁéİćçŽĎè¡řéŹŘ è¡řéŹŘåĹűæŸŕäÿĂäÿłåĂijïijŇå¡ŞèűĚè£ĞèŁŹäÿłåĂijçŽĎæŮűåĂŹæŞ■ä¡IJçşżçż§éĂŽåÿÿäijŽåŘŚéĂĄäÿĂäÿł çąňéŹŘåĹűæŸŕçŤÍæİěæŇĞåőŽè¡řéŹŘåĹűèČ¡èő¿åőŽçŽĎæIJĂåď'ğåĂijãĂĆéĂŽåÿÿæİěèőšïijŇè£ŹäÿłçŤśçşż åř¡çőąçąňéŹŘåĹűåŘŕäžěæŤźåřŘäÿĂçĆźïijŇä¡ĘæŸŕæIJĂåě¡äÿ■èęĄä¡£çŤÍçŤÍæĹůè£ŻçÍŃåŐżä£őæŤźãĂĆ

setrlimit() åĞ¡æŢřè£ŸèČ¡èćńçŤÍæİěèő¿ç¡őå■Řè£ŻçÍŃæŢřéĞŘãĂĄæĽŞåijĂæŰĞäžűæŢřäžěåŘŁç æŻť'åď'ŽèŕęæČĚèŕůåŘĆèĂČ resource æÍąåIŮçŽĎæŰĞæąčãĂĆ

éIJĂèęĄæşÍæĎŘçŽĎæŸŕæIJňèŁĆåĘĚåőźåŘłèČ¡éĂĆçŤÍäžŐUnixçşżçż§ïijŇåźűäÿŤäÿ■ä£İèŕĄæĽ'ĂæIJĹ æŕŤåęĆæĹŚäžňåIJÍæţŃèŕŢçŽĎæŮűåĂŹïijŇåőČèČ¡åIJÍLinuxäÿŁéİćæ■čåÿÿè£ŘèąŇïijŇä¡ĘæŸŕåIJÍOS XäÿŁå■ť'äÿ■èČ¡ãĂĆ

15.15 13.15 åŘŕåŁĺäÿĂäÿłWEBæţŔèğĹåŹĺ

éŮőéćŸ

ä¡ăæČşéĂŽè£ĞèĎŽæIJňåŘŕåŁĺæţŔèğĹåŹĺåźűæL'ŞåijĂæŇĞåőŽçŽĎURLç¡Śéąţ

èğčåĘşæŰźæąĹ

webbrowser æĺąåİŮèČ¡èćńçŤĺæİěåŘŕåŁĺäÿĂäÿłæţŔèğĹåŹĺïijŇåźűäÿŤäÿŐåźşåŘŕæŮăåĚşãĂĆä¿ŃåęĆ

```
>>> import webbrowser
>>> webbrowser.open('http://www.python.org')
True
>>>
```

åőČäijŽä¡£çŤĺéżŸèőd'æţŔèğĹåŹĺæL'ŞåijĂæŇĞåőŽç¡ŚéąţãĂĆåęĆæđIJä¡ăè£ŸæČşåŕźç¡ŚéąţæL'ŞåijĂæŮ

```
>>> # Open the page in a new browser window
>>> webbrowser.open_new('http://www.python.org')
True
>>>

>>> # Open the page in a new browser tab
>>> webbrowser.open_new_tab('http://www.python.org')
True
>>>
```

è£ŹæăůåřśåŘŕäžěæL'ŞåijĂäÿĂäÿłæŰřçŽĎæţŔèğĹåŹĺçłŮåŘčæĹŰèĂĚæăĞç■¿ïijŇåŘłèęĄæţŔèğĹåŹĺæŤ

åęĆæđIJä¡ăæČşæŇĞåőŽæţŔèğĹåŹĺçśżåđŃïijŇåŘŕäžěä¡£çŤĺ webbrowser.get() åĞ¡æŢřæİěæŇĞåőŽæ§Řäÿłçĺ'źåőŽæţŔèğĹåŹĺãĂĆä¿ŃåęĆïijŽ

```
>>> c = webbrowser.get('firefox')
>>> c.open('http://www.python.org')
True
>>> c.open_new_tab('http://docs.python.org')
True
>>>
```

åŕźäžŐæŤŕæŇĄçŽĎæţŔèğĹåŹĺåŘ■çğřåĹŮèąĺåŘŕæ§ěéŸĚ'PythonæŰĞæąč <http://docs.python.org/3/library/webbrowser.html>'_

èőĺèőž

åIJĺèĎŽæIJňäÿ■æL'ŞåijĂæţŔèğĹåŹĺæIJL'æŮűåĂŹäijŽå¿ĹæIJL'çŤĺãĂĆä¿ŃåęĆïijŇæ§ŘäÿłèĎŽæIJňæL'
ä¡ăæČşå£ńéĂ§æL'ŞåijĂäÿĂäÿłæţŔèğĹåŹĺæİěçąőä£ĺåőČåůšçżŔæ■čåÿÿè£ŘèąŇäžĘãĂĆ
æĹŰèĂĚæŸŕæ§Řäÿłçĺ Ňåž Ŕäžě HTMLç¡ŚéąţæăijåijŔè¿ŞåĞžæŢřæ■őïijŇä¡ăæČşæL'ŞåijĂæţŔèğĹåŹĺæ§ěçIJŇ
äÿ■çőąæŸŕäÿŁéĺćåŞłçğ■æČĚåĘţïijŇä¡£çŤĺ webbrowser æĺąåİŮéČ¡æŸŕäÿĂäÿłçőĂå■ŢåőđçŤĺçŽĎèğčåĘşa

16 çňňå■ĄåŻŻçńäïijŽæţŃèŕŢãĂĄèřČèŕŢåŠŇåijĆåÿÿ

èŕŢéłŇè£ŸæŸŕå¿ĹæčŠçŽĎïijŇä¡ĘæŸŕèřČèŕŢïij§åřśæšąéĆčäźĹæIJL'èűčäžĘãĂĆäžŃåðđæŸŕïijŇåIJÍPyt

Contents:

16.1 14.1 æţŃèŕŢstdoutè¿ŞåĞž

éŮőécŸ

ä¡äçŽĎçÍŃåžŘäÿ■æIJL'äÿłæŰźæşŢäijŽè¿ŞåĞžåĹřæăĞåĞĘè¿ŞåĞžäÿ■ïijĹsys.stdoutïijL'ãĂĆäź§åřśæŸŕè
ä¡äæČşåĘŹäÿłæţŃèŕŢæĺěèŕĄæŸŐåőČïijŇçżŹåőŽäÿĂäÿłè¿ŞåĚěïijŇçŻÿåžŤçŽĎè¿ŞåĞžèČ¡æ■čåÿÿæŸ¿çd'žå

èğčåĘşæŰźæąĹ

ä¡£çŤÍ `unittest.mock` æĺąåİŮäÿ■çŽĎ `patch()` åĞ¡æŢřïijŇ
ä¡£çŤÍètůæĺěéİđåÿÿçőĂå■ŢïijŇåŘŕäżěäÿžå■ŢäÿłæţŃèŕŢæĺąæŃ§ `sys.stdout`
çĎűåŘŐåŻđæżŽïijŇ åźűäÿŤäÿ■äžğçŤ§åd'ğéĞŘçŽĎäÿt'æŮűåŘŸéĞŘæĹŰåIJĺæţŃèŕŢçŤĺä¿ŃçŻt'æŐěæŽt'éIJ

ä¡IJäÿžäÿĂäÿłä¿Ńå■ŘïijŇæĹŚäżňåIJÍ `mymodule` æĺąåİŮäÿ■åőŽäźL'åęĆäÿŃäÿĂäÿłåĞ¡æŢřïijŽ

```
# mymodule.py

def urlprint(protocol, host, domain):
    url = '{}://{}.{}'.format(protocol, host, domain)
    print(url)
```

éżŸèőd'æČĚåĘţäÿŃåĘĚç¡őçŽĎ `print` åĞ¡æŢřäijŽåřĘè¿ŞåĞžåŘŚéĂĄåĹř `sys.stdout` ãĂĆ äÿžäžĘæţŃèŕŢè¿ŞåĞžçIJ§çŽĎåIJĺéĆčéĞŃïijŇä¡ääŘŕäżěä¡£çŤĺäÿĂäÿłæŻ£èżńåŕźèśąæĺěæĺąæŃ
ä¡£çŤÍ `unittest.mock` æĺąåİŮçŽĎ `patch()` æŰźæşŢåŘŕäżěå¿ĹæŰźä¿£çŽĎåIJĺæţŃèŕŢè£ŘèąŃçŽĎäÿŁ
åźűäÿŤå¡ŞæţŃèŕŢåőŃæĹŘæŮűåĂŹèĞłåŁĺè£ŤåŻđåőČäżňçŽĎåŐ§æIJL'çŁűæĂĄãĂĆäÿŃéĺćæŸŕåŕź
`mymodule` æĺąåİŮçŽĎæţŃèŕŢäżčçăĄïijŽ

```
from io import StringIO
from unittest import TestCase
from unittest.mock import patch
import mymodule

class TestURLPrint(TestCase):
    def test_url_gets_to_stdout(self):
        protocol = 'http'
        host = 'www'
        domain = 'example.com'
        expected_url = '{}://{}.{}\n'.format(protocol, host, domain)

        with patch('sys.stdout', new=StringIO()) as fake_out:
```

(continues on next page)

(continued from previous page)

```
            mymodule.urlprint(protocol, host, domain)
            self.assertEqual(fake_out.getvalue(), expected_url)
```

èőÍèőž

urlprint() åĞ¡æŢřæŐěåŘŮäÿL'äÿłåŘĆæŢřïijŇæţŃèŕŢæŰźæşŢåijĂåğŃäijŽåĚĹèő¿ç¡őæŕŘäÿĂäÿłåł
expected_url åŘŸéĞŘèćńèő¿ç¡őæĹŘåŇĚåŘńæIJ§æIJŻçŽĎè¿ŞåĞžçŽĎå■ŮçňęäÿšãĂĆ

unittest.mock.patch() åĞ¡æŢřèćńçŤĺä¡IJäÿĂäÿłäÿŁäÿŃæŰĞçőąçŘĘåŹĺïijŇä¡£çŤĺ
StringIO åŕźèśąæİěäžčæŻ£ sys.stdout . fake_out
åŘŸéĞŘæŸŕåIJĺèŕěè£ŻçĺŃäÿ■èćńåĹŻåżžçŽĎæĺąæŃ§åŕźèśąãĂĆ åIJĺwith-
èŕ■åŘěäÿ■ä¡£çŤĺåőČåŘŕäżěæL'ğèąŇåŘĎçğ■æčĂæ§ěãĂĆå¡Şwithèŕ■åŘěçżŞæİ§æŮűïijŇpatch
äijŽåřĘæL'ĂæIJL'äÿIJèě£æĄćåď■åĹřæţŃèŕŢåijĂåğŃåL'■çŽĎçŁűæĂĄãĂĆ
æIJL'äÿĂçĆźéIJĂèęĄæşĺæĎŘçŽĎæŸŕæ§ŘäżŽåŕźPythonçŽĎCæL'l'åśŢåŘŕèČ¡äijŽå£¡çŢěæŐL'
sys.stdout çŽĎéĚ■ç¡őèĂŇçŻť'æŐěåĘŹåĚěåĹřæăĞåĞĘè¿ŞåĞžäÿ■ãĂĆ
éŹŘäžŐçŕĞåźĚïijŇæIJňèŁĆäÿ■äijŽæűL'åŘŁåĹřè£ŹæŰźéĺćçŽĎèőšèğčïijŇåőČéĂĆçŤĺäžŐçžŕPythonäżčçăĄ
åęĆæđIJä¡ăçIJ§çŽĎéIJĂèęĄåIJĺĆæL'l'åśŢäÿ■æ■ŢèŐůI/OïijŇä¡ăåŕźäżěåĚĹæL'ŞåijĂäÿĂäÿłäÿť'æŮűæŰĞäżű
æŻť'åď'ŽåĚşäžŐæ■ŢèŐůäżěå■Ůçňęäÿšå¡ćåijŘæ■ŢèŐůI/OåŠŇ StringIO
åŕźèśąèŕůåŘĆéŸĚ5.6åřŘèŁĆãĂĆ

16.2 14.2 åIJĺå■ŢåĚČæţŃèŕŢäÿ■çżŹåŕźèśąæL'ŞèąěäÿĄ

éŮőécŸ

ä¡ăåĘźçŽĎå■ŢåĚČæţŃèŕŢäÿ■éIJĂèęĄçżŹæŇĞåőŽçŽĎåŕźèśąæL'ŞèąěäÿĄïijŇ
çŤĺæĬěæŰ■èĺĂåőČäżňåIJĺæţŃèŕŢäÿ■çŽĎæIJ§æIJŻèąŇäÿžïijĹæŕŤåęĆïijŇæŰ■èĺĂèćńèřČçŤĺæŮűçŽĎåŘĆæŢ

èğčåĘşæŰźæąĹ

unittest.mock.patch() åĞ¡æŢřåŘŕèćńçŤĺæĬěèğčåĘşè£ŹäÿłéŮőécŸăĂĆ
patch() è£ŸåŘŕèćńçŤĺä¡IJäÿĂäÿłèčĚéěřåŹĺăĂĄäÿŁäÿŃæŰĞçőąçŘĘåŹĺæĹŮå■ŢçŃňä¡£çŤĺïijŇåř¡çőąåźűä
ä¿ŃåęĆïijŇäÿŃéĬćæŸŕäÿĂäÿłåřĘåőČå¡ŞåĄŽèčĚéěřåŹĺä¡£çŤĺçŽĎä¿Ňå■ŘïijŽ

```
from unittest.mock import patch
import example

@patch('example.func')
def test1(x, mock_func):
    example.func(x)       # Uses patched example.func
    mock_func.assert_called_with(x)
```

åőČè£ŸåŘŕäżěèćńå¡ŞåĄŽäÿĂäÿłäÿŁäÿŃæŰĞçőąçŘĘåŹĺïijŽ

```
with patch('example.func') as mock_func:
    example.func(x)       # Uses patched example.func
    mock_func.assert_called_with(x)
```

æIJĂåŘŐïijŇä¡ăè£ŸåŘŕäżěæL'ŃåŁĺçŽĎä¡£çŤĺåőČæL'Ş èąěäÿĄïijŽ

```
p = patch('example.func')
mock_func = p.start()
example.func(x)
mock_func.assert_called_with(x)
p.stop()
```

åęĆæđIJåŘŕèČ¡çŽĎèŕĺïijŇä¡ăèČ¡åď§åŘäåŁăèčĚéěřåŹĺåŠŇäÿŁäÿŃæŰĞçőąçŘĘåŹĺæĺěçżŹåď Žäÿłåŕźèś

```
@patch('example.func1')
@patch('example.func2')
@patch('example.func3')
def test1(mock1, mock2, mock3):
    ...

def test2():
    with patch('example.patch1') as mock1, \
         patch('example.patch2') as mock2, \
         patch('example.patch3') as mock3:
    ...
```

èőĺèőž

patch() æŐěåŘŮäÿĂäÿłåůšå■ŸåIJĺåŕźèśąçŽĎåĚĺèůŕå¿ĎåŘ■ïijŇåřĘåĚűæŻ£æ■ćäÿžäÿĂäÿłæŰřçŽĎåĂ
åŐ§æĺěçŽĎåĂijäijŽåIJĺèčĚéěřåŹĺåĞ¡æŢřæĹŰäÿŁäÿŃæŰĞçőąçŘĘåŹĺåőŇæĹŘåŘŐèĞłåŁĺæĄćåď■åŻđæĺěã
éżŸèőď æČĚåĘţäÿŃïijŇæL'ĂæIJL'åĂijäijŽèćń MagicMock åőđä¿ŃæŻ£äżčãĂĆä¿ŃåęĆïijŽ

```
>>> x = 42
>>> with patch('__main__.x'):
...     print(x)
...
<MagicMock name='x' id='4314230032'>
>>> x
42
>>>
```

äÿ■è£ĞïijŇä¡ăåŘŕäżěéĂŽè£ĞçżŹ patch() æŘŘä¿ŻçňňäžŇäÿłåŘĆæŢřæĺěåřĘåĂijæŻ£æ■ćæĹŘäżżä¡Ţ

```
>>> x
42
>>> with patch('__main__.x', 'patched_value'):
...     print(x)
...
```

(continues on next page)

(continued from previous page)

```
patched_value
>>> x
42
>>>
```

èćńçŤÍæİěä¡IJäÿžæŻ£æ■ćåĂijçŽĎ `MagicMock` åðđä¿ŃèČ¡åď§æĺąæŃ§åŕŕèřČçŤÍåŕźèśąåŠŇåðđä¿ŃãĂäżŰäżňèðřå¡ŢåŕźèśąçŽĎä¡£çŤÍä£ąæĄŕåźűåĚĄèðÿä¡ăæĽğèąŇæŰ■èÍĂæčĂæ§ěïijŇä¿ŃåęĆïijŽ

```
>>> from unittest.mock import MagicMock
>>> m = MagicMock(return_value = 10)
>>> m(1, 2, debug=True)
10
>>> m.assert_called_with(1, 2, debug=True)
>>> m.assert_called_with(1, 2)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File ".../unittest/mock.py", line 726, in assert_called_with
    raise AssertionError(msg)
AssertionError: Expected call: mock(1, 2)
Actual call: mock(1, 2, debug=True)
>>>

>>> m.upper.return_value = 'HELLO'
>>> m.upper('hello')
'HELLO'
>>> assert m.upper.called

>>> m.split.return_value = ['hello', 'world']
>>> m.split('hello world')
['hello', 'world']
>>> m.split.assert_called_with('hello world')
>>>

>>> m['blah']
<MagicMock name='mock.__getitem__()' id='4314412048'>
>>> m.__getitem__.called
True
>>> m.__getitem__.assert_called_with('blah')
>>>
```

äÿĂèĹňæİěèðšïijŇè£ŹäžŻæŞ■ä¡IJäijŽåIJÍäÿĂäÿłå■ŢåĚČæţŃèŕŢäÿ■åðŇæĹŘãĂĆä¿ŃåęĆïijŇåĄĞèð¿ä¡ä

```
# example.py
from urllib.request import urlopen
import csv

def dowprices():
    u = urlopen('http://finance.yahoo.com/d/quotes.csv?s=@^DJI&f=sl1
↪')
```

(continues on next page)

(continued from previous page)

```
    lines = (line.decode('utf-8') for line in u)
    rows = (row for row in csv.reader(lines) if len(row) == 2)
    prices = { name:float(price) for name, price in rows }
    return prices
```

æ■čåÿÿæĮěèőšïijŇè£ŹäÿłåĞ¡æŢřäijŽä¡£çŤÍ urlopen() äžŐWebäÿŁéĮćèŐůåŘŰæŢřæ■őåźűèğčæđŘåőČãĂĆ åIJÍå■ŢåĚČæţŃèŕŢäÿ■ïijŇä¡ššŘŕäžěçżŹåőČäÿĂäÿłécĎåĚĹåőŽ

```
import unittest
from unittest.mock import patch
import io
import example

sample_data = io.BytesIO(b'''\
"IBM",91.1\r
"AA",13.25\r
"MSFT",27.72\r
\r
''')

class Tests(unittest.TestCase):
    @patch('example.urlopen', return_value=sample_data)
    def test_dowprices(self, mock_urlopen):
        p = example.dowprices()
        self.assertTrue(mock_urlopen.called)
        self.assertEqual(p,
                         {'IBM': 91.1,
                          'AA': 13.25,
                          'MSFT' : 27.72})

if __name__ == '__main__':
    unittest.main()
```

æIJňä¿Ńäÿ■ïijŇä¡■äžŐ example æĺąåĮŮäÿ■çŽĎ urlopen() åĞ¡æŢřèćńäÿĂäÿłæĺąæŃ§åŕźèśąæŻ£äžčïijŇ èŕěåŕźèśąäijŽè£ŤåŻđäÿĂäÿłåŇĚåŘńæţŃèŕŢæŢřæ■őçŽĎ ByteIO().

è£ŸæIJL'äÿĂçĆźïijŇåIJĺæL'Şèąěäÿ ĄæŮűæĹŚäžňä¡£çŤĺäžĘ example.urlopen æĮěäžčæŻ£ urllib.request.urlopen ãĂĆ å¡Şä¡ššĹŻåżžèąěäÿ ĄçŽĎæŮűåĂŹïijŇä¡ššĚéążä¡£çŤĺäőČäžňåIJĺæţŃèŕŢäžčçăĄäÿ■çŽĎåŘ■çğřãĂĆ çŤśäžŐæţŃèŕŢäžčçăĄä¡£çŤĺäžĘ from urllib.request import urlopen ,éĆčäźĹ dowprices() åĞ¡æŢř äÿ■ä¡£çŤĺçŽĎ urlopen() åĞ¡æŢřåőđéŹĚäÿŁåŕśä¡■äžŐ example æĺąåĮŮäžĘãĂĆ

æIJňèŁĆåőđéŹĚäÿŁåŘłæŸŕåŕź unittest.mock æĺąåĮŮçŽĎäÿĂæňąæţĚåŕĮè¿Ďæ■ćãĂĆ æŻť'åď'ŽæŻť'éńŸçžğçŽĎçL'źæĂğïijŇèŕůåŘĆèĂČ åőŸæŰźæŰĞæąč

16.3 14.3 ålJÍå■ŢåĚČæţŃèŕŢäÿ■æţŃèŕŢåijĆåÿÿæČĚåĘţ

éŮőéćŸ

ä¡ăæČşåĘŹäÿłæţŃèŕŢçŤÍä¿ŃæÌěåĞĘçąőçŽĎåĹď æŰ■æ§ŘäÿłåijĆåÿÿæŸŕåŘęèćńæŁŻåĞžãĂĆ

èğčåĘşæŰźæąĹ

åŕźäžŐåijĆåÿÿçŽĎæţŃèŕŢåŘŕä¡£çŤÍ `assertRaises()` æŰźæşŢãĂĆ ä¿ŃåęĆïijŇåęĆæđIJä¡ăæČşæţŃèŕŢæ§ŘäÿłåĞ¡æŢřæŁŻåĞžäžĘ `ValueError` åijĆåÿÿïijŇåČŘäÿŃéÌćè£ŹæăůåĘŹïijŽ

```
import unittest

# A simple function to illustrate
def parse_int(s):
    return int(s)

class TestConversion(unittest.TestCase):
    def test_bad_int(self):
        self.assertRaises(ValueError, parse_int, 'N/A')
```

åęĆæđIJä¡ăæČşæţŃèŕŢåijĆåÿÿçŽĎåĚůä¡ŞåĂijïijŇéIJĂèęAçŤÍåĹřåŘęåď ŰäÿĂçğ■æŰźæşŢïijŽ

```
import errno

class TestIO(unittest.TestCase):
    def test_file_not_found(self):
        try:
            f = open('/file/not/found')
        except IOError as e:
            self.assertEqual(e.errno, errno.ENOENT)

        else:
            self.fail('IOError not raised')
```

èőÍèőž

`assertRaises()` æŰźæşŢäÿžæţŃèŕŢåijĆåÿÿå■ŸåIJÍæĂğæŘŘä¿ŻäžĘäÿĂäÿłçőĂä¿£æŰźæşŢãĂĆ äÿĂäÿłåÿÿèğAçŽĎéŹůéŸśæŸŕæĹ'ŃåŁÍåŐżè£ŻèąŇåijĆåÿÿæčĂæţŃãĂĆæŕŤåęĆïijŽ

```
class TestConversion(unittest.TestCase):
    def test_bad_int(self):
        try:
            r = parse_int('N/A')
        except ValueError as e:
            self.assertEqual(type(e), ValueError)
```

è£Źçğ■æŰźæşŢçŽĎéŮőéćŸåIJÍäžŐåőČå¿ĹåőźæŸ§éĄŮæijŘåĚűäżŰæČĚåĘţïijŇæŕŤåęĆæšąæIJĽäżżä¡T
éĆčäźĹä¡äè£Ÿå¿ŮéIJĂèęĄåćđåŁäåŘęåď'ŰçŽĎæčĂæţŃè£ĞçÍŃïijŇåęĆäÿŃéİćè£ŹæäůïijŽ

```
class TestConversion(unittest.TestCase):
    def test_bad_int(self):
        try:
            r = parse_int('N/A')
        except ValueError as e:
            self.assertEqual(type(e), ValueError)
        else:
            self.fail('ValueError not raised')
```

`assertRaises()` æŰźæşŢäijŽåď'ĎçŘĘæĽ'ĂæIJĽçżĘèŁĆïijŇåŻäæ■ď'ä¡ääžŤèŕĕä¡£çŤÍåőČãĂĆ

`assertRaises()` çŽĎäÿĂäÿłçijžçĆźæŸŕäőČæţŃäÿ■äžĘäijĆäÿÿäĚůä¡ŞçŽĎåĂijæŸŕäď'ŽäřŚãĂĆ äÿžäžĘæţŃèŕŢäijĆäÿÿäĂijïijŇåŘŕäżĕä¡£çŤÍ `assertRaisesRegex()` æŰźæşŢïijŇ åőČäŘŕäŘŇæŮűæţŃèŕŢäijĆäÿÿçŽĎå■ŸåIJÍäżĕäŘŁéĂŽè£Ğæ■čäĹŹäijŘäŇźéĚ■äijĆäÿÿçŽĎå■ŮçňęäÿšèąÍçd

```
class TestConversion(unittest.TestCase):
    def test_bad_int(self):
        self.assertRaisesRegex(ValueError, 'invalid literal .*',
                               parse_int, 'N/A')
```

`assertRaises()` åŠŇ `assertRaisesRegex()` è£ŸæIJĽäÿĂäÿłåőźæŸŞå£¡çŢĕçŽĎåIJřæŰźäřśæŸŕäőČäżňè£ŸèČ¡èćňå¡ŞåĄŽäÿŁäÿŃæŰĞçőąçŘĘåŹĺä¡£çŤÍ

```
class TestConversion(unittest.TestCase):
    def test_bad_int(self):
        with self.assertRaisesRegex(ValueError, 'invalid literal .*
↪'):
            r = parse_int('N/A')
```

ä¡Ęä¡äçŽĎæţŃèŕŢæűĽ'åŘŁåĹŕåď'ŽäÿłæĽ'ğèąŃæ■ĕéłď'çŽĎæŮűåĂŹè£Źçğ■æŰźæşŢäŕśå¿ĹæIJĽçŤÍäžĘ

16.4 14.4 äřĘæţŃèŕŢè¿ŞåĞžçŤÍæŮĕå£Ůèőřå¡ŢåĹřæŰĞäżűäÿ■

éŮőéćŸ

ä¡ääÿŃæIJŻäřĘå■ŢåĚĈæţŃèŕŢçŽĎè¿ŞåĞžåĘŹåĹŕåĹŕæ§ŘäÿłæŰĞäżűäÿ■åŐżïijŇèĂŃäÿ■æŸŕæĽ'Şå■řä

èğčåĘşæŰźæąĹ

è£ŘèąŃå■ŢåĚĈæţŃèŕŢäÿĂäÿłåÿÿèğĄæŁĂæIJŕäřśæŸŕåIJÍæţŃèŕŢæŰĞäżűäžŢéČÍåŁääĚĕäÿŃéİćè£Źæőţ

```
import unittest

class MyTest(unittest.TestCase):
    pass
```

(continues on next page)

(continued from previous page)

```
if __name__ == '__main__':
    unittest.main()
```

è£ŹæăůçŽĎèŕÍæţŃèŕŢæŰĞäżűåřśæŸŕåŘŕæĽğèąŇçŽĎïijŇåźűäÿŤäijŽåřĘè£ŘèąŇæţŃèŕŢçŽĎçżŞæđIJæ
åęĆæđIJä¡ăæČşéĞ■åőŽåŘŚè¿ŞåĞžïijŇåřśéIJĂèęĄåČŘäÿŃéÍćè£ŹæăůäŁőæŤź `main()`
åĞ¡æŢřïijŽ

```
import sys

def main(out=sys.stderr, verbosity=2):
    loader = unittest.TestLoader()
    suite = loader.loadTestsFromModule(sys.modules[__name__])
    unittest.TextTestRunner(out,verbosity=verbosity).run(suite)

if __name__ == '__main__':
    with open('testing.out', 'w') as f:
        main(f)
```

èőÍèőž

æIJňèŁĆæĎ§åĚť'èűčçŽĎéČÍåĹĘåźűäÿ■æŸŕåřĘæţŃèŕŢçżŞæđIJéĞ■åőŽåŘŚåĹřäÿĂäÿłæŰĞäżűäÿ■ïijŇ
èĂŇæŸŕéĂŽè£ĞÈ£ŹæăůåĄŽåŘŚä¡ăåśŢçď'žäžĘ `unittest`
æÍąåİŮäÿ■äÿĂäžŻåĂijå¿ŮåĚşæşÍçŽĎåĘĚéČÍåůěä¡IJåŐ§çŘĘãĂĆ

`unittest` æÍąåİŮéęŰåĚĹäijŽçżĎèčĚäÿĂäÿłæţŃèŕŢåěŮäżűãĂĆ
è£ŹäÿłæţŃèŕŢåěŮäżűåŇĚåŘńäžĘä¡ăåőŽäźĽçŽĎåŘĎçğ■æŰźæşŢãĂĆäÿĂæŮęåěŮäżűçżĎèčĚåőŇæĹŘïijŇå

è£Źäÿď'æ■ěæŸŕåĹĘåijĂçŽĎïijŇ`unittest.TestLoader`
åőđä¿ŃèćńçŤÍæİěçżĎèčĚæţŃèŕŢåěŮäżűãĂĆ `loadTestsFromModule()`
æŸŕåőČåőŽäźĽçŽĎæŰźæşŢäźŃäÿĂïijŇçŤÍæİěæŤűéŻĘæţŃèŕŢçŤÍä¿ŃãĂĆ åőČäijŽäÿž
`TestCase` çśżæĽńæŘŕæ§ŘäÿłæĺąåİŮåźűåřĘåĚűäÿ■çŽĎæţŃèŕŢæŰźæşŢæŘŕåŘŰåĞžæİěãĂĆ
åęĆæđIJä¡ăæČşè£ŻèąŇçżĘçšŠåžęçŽĎæŐğåĹűïijŇ åŘŕäżěä¡£çŤÍ
`loadTestsFromTestCase()` æŰźæşŢæİěäżŐæ§ŘäÿłçżğæĽ£TestCaseçŽĎçśżäÿ■æŘŕåŘŰæţŃèŕŢæŰźæ
`TextTestRunner` çśżæŸŕäÿĂäÿłæţŃèŕŢè£ŘèąŇçśżçŽĎä¿Ńå■ŘïijŇ
è£ŹäÿłçśżçŽĎäÿżèęĄçŤÍéĂŤæŸŕæĽğèąŇæ§ŘäÿłæţŃèŕŢåěŮäżűäÿ■åŇĚåŘńçŽĎæţŃèŕŢæŰźæşŢãĂĆ
è£Źäÿłçśżèů§æĽğèąŇ`unittest.main()` åĞ¡æŢřæĽĂä¡£çŤÍçŽĎæţŃèŕŢè£ŘèąŇåŹÍæŸŕäÿĂæăůçŽĎãĂĆ
äÿ■è£ĞïijŇæĹŚäżňåIJÍè£ŹéĞŇåŕźåőČè£ŻèąŇäžĘäÿĂäžŻåĹŮåžŢåśĆéĚ■ç¡őïijŇåŇĚæŃňè¿ŞåĞžæŰĞäżűåŠ
åř¡çőąæIJňèŁĆä¿Ńå■Řäżčçą̊Ąå¿ŁåřŚïijŇä¡ĘæŸŕèČ¡æŇĞåŕijä¡ăåęĆä¡Ţåŕź
`unittest` æąĘæđűè£ŻèąŇæŻť'è£ŻäÿĂæ■ěçŽĎèĞłåőŽäźĽãĂĆ
èęĄæČşèĞłåőŽäźĽæţŃèŕŢåěŮäżűçŽĎèčĚéĚ■æŰźåijŘïijŇä¡ăåŘŕäżěåŕź `TestLoader`
çśżæĽğèąŇæŻť'åď'ŽçŽĎæŞ■ä¡IJãĂĆ äÿžäžĘèĞłåőŽäźĽæţŃèŕŢè£ŘèąŇïijŇä¡ăåŘŕäżěæđĎéĂääÿĂäÿłèĞłåůš
`TextTestRunner` çŽĎåŁ§èČ¡ãĂĆ èĂŇè£ŹäžŻåůšçżŘèűĚåĞžäžĘæIJňèŁĆçŽĎèŇČåŻť'ãĂĆ`unittest`
æÍąåİŮçŽĎæŰĞæąčåŕźåžŢåśĆåőđçŐřåŐ§çŘĘæIJĽæŻť'æůśåĚěçŽĎèőšèğčïijŇåŘŕäżěåŐżçIJŃçIJŃãĂĆ

16.5 14.5 å£¡çŢĕæĹŰæIJ§æIJŻæţŃèŕŢåďśèťĕ

éŮőéćŸ

ä¡ăæČşåIJĺå■ŢåĚČæţŃèŕŢäÿ■å£¡çŢĕæĹŰæăĞèőŕæ§ŘäžŻæţŃèŕŢäijŻæŇL'çĚğéćĎæIJ§è£ŘèąŇåď'śèť'ĕã

èğčåĘşæŰźæąĹ

unittest æĺąåİŮæIJL'èčĚéĕřåŹĺåŘŕçŤĺæİĕæŐğåĹűåŕźæŇĞåőŽæţŃèŕŢæŰźæşŢçŽĎåď'ĎçŘĘïijŇä¿Ń

```
import unittest
import os
import platform

class Tests(unittest.TestCase):
    def test_0(self):
        self.assertTrue(True)

    @unittest.skip('skipped test')
    def test_1(self):
        self.fail('should have failed!')

    @unittest.skipIf(os.name=='posix', 'Not supported on Unix')
    def test_2(self):
        import winreg

    @unittest.skipUnless(platform.system() == 'Darwin', 'Mac␣
↪specific test')
    def test_3(self):
        self.assertTrue(True)

    @unittest.expectedFailure
    def test_4(self):
        self.assertEqual(2+2, 5)

if __name__ == '__main__':
    unittest.main()
```

åęĆæđIJä¡ăåIJĺMacäÿŁè£ŘèąŇè£ŹæőţäžčçăĄïijŇä¡ăäijŽå¿ŮåĹřåęĆäÿŃè¿ŞåĞžïijŽ

```
bash % python3 testsample.py -v
test_0 (__main__.Tests) ... ok
test_1 (__main__.Tests) ... skipped 'skipped test'
test_2 (__main__.Tests) ... skipped 'Not supported on Unix'
test_3 (__main__.Tests) ... ok
test_4 (__main__.Tests) ... expected failure

----------------------------------------------------------------------
↪--
```

(continues on next page)

(continued from previous page)

```
Ran 5 tests in 0.002s

OK (skipped=2, expected failures=1)
```

èőÍèőž

skip() èčĚéěřåŹÍèČ¡èćńçŤÍæİěå£¡çŢěæ§Řäÿłä¡ääÿ■æČşè£ŘèąŇçŽĎæţŃèřŢãĂĆ skipIf() åŠŇ skipUnless() åŕźäžŐä¡ääŘłæČşåIJÍæ§Řäÿłç ĽźåőŽåźşåŘřæĹŰPythonçĽĹæIJňæĹŰåĚŭ ä¡£çŤÍ @expected çŽĎåď śèť ěèčĚéěřåŹÍæİěæäĞèőřéĆčäžŻçąőåőŽäijŽåď śèť ěçŽĎæţŃèřŢïijŇåźűäÿŤåŕźè£

å£¡çŢěæŰźæşŢçŽĎèčĚéěřåŹÍè£ŸåŘŕäżěèćńçŤÍæİěèčĚéěřæŢť äÿłæţŃèřŢçśżïijŇæřŤåęĆïijŽ

```
@unittest.skipUnless(platform.system() == 'Darwin', 'Mac specific␣
↪tests')
class DarwinTests(unittest.TestCase):
    pass
```

16.6 14.6 åď ĎçŘĘåď Žäÿłåij Ćåÿÿ

éŮőéćŸ

ä¡äæIJĽäÿĂäÿłäżčçăĄçĽĞæőţåŘŕèČ¡äijŽæŁŻåĞžåď Žäÿłäÿ■åŘŇçŽĎåijĆåÿÿïijŇæĂŐæăůæĽ■èČ¡äÿ■

èğčåĘşæŰźæąĹ

åęĆæđIJä¡ääŘŕäżěçŤÍå■ŢäÿłäżčçăĄåİŮåď ĎçŘĘäÿ■åŘŇçŽĎåijĆåÿÿïijŇåŘŕäżěåřĘåőČäžňæŤ¿åĚěäÿĂä

```
try:
    client_obj.get_url(url)
except (URLError, ValueError, SocketTimeout):
    client_obj.remove_url(url)
```

åIJÍè£Źäÿłä¿Ňå■Řäÿ■ïijŇåĚČçěŰäÿ■äżżä¡ŢäÿĂäÿłåijĆåÿÿåŘŚçŤ§æŮűéČ¡äijŽæĽğèąŇ remove_url() æŰźæşŢãĂĆ åęĆæđIJä¡äæČşåŕźåĚűäÿ■æ§Řäÿłåij Ćåÿÿè£ŻèąŇäÿ■åŘŇçŽĎåď ĎçŘĘïijŇåŘ except èŕ■åŘěäÿ■ïijŽ

```
try:
    client_obj.get_url(url)
except (URLError, ValueError):
    client_obj.remove_url(url)
except SocketTimeout:
    client_obj.handle_url_timeout(url)
```

å¿Łåď ŽçŽĎåijĆåÿÿäijŽæIJĽåśĆçžğåĚşçşżïijŇåŕźäžŐè£Źçğ■æČĚåĘţïijŇä¡äåŘŕèČ¡ä¡£çŤÍåőČäžňçŽĎä

```
try:
    f = open(filename)
except (FileNotFoundError, PermissionError):
    pass
```

åŘŕäžěèćńéĞ■åĘŹäÿžïijŽ

```
try:
    f = open(filename)
except OSError:
    pass
```

OSError æŸŕ FileNotFoundError åŠŇ PermissionError åijĆåÿÿçŽĎå§žçśżãĂĆ

èőĺèőž

ař¡çőąåď ĎçŘĘåď Žäÿłåij Ćåÿÿæ IJňèžňåźűæšąäżĂäźĹçĽźæőŁçŽĎïijŇäÿ■è£Ğä¡ššŘŕäžěä¡£çŤĺ as åĚşéŤőå■ŮæĺěèŐůå¿Ůèćńæ Ł ŻåĞžåijĆåÿÿçŽĎåij ŢçŤĺïijŽ

```
try:
    f = open(filename)
except OSError as e:
    if e.errno == errno.ENOENT:
        logger.error('File not found')
    elif e.errno == errno.EACCES:
        logger.error('Permission denied')
    else:
        logger.error('Unexpected error: %d', e.errno)
```

è£Źäÿłä¿Ňå■Řäÿ■ïijŇ e åŘŸéĞŘæŇĞåŘŚäÿĂäÿłèćńæŁŻåĞžçŽĎ OSError åijĆåÿÿåðđä¿ŇãĂĆ è£ŹäÿłåIJĺä¡äæČşæŻť è£ŻäÿĂæ■ěåĹĘæđŘè£ŹäÿłåijĆåÿÿçŽĎæŮűåĂŹäijŽå¿ĹæIJĽçŤĺï

åŘŇæŮűè£ŸèęĄæşĺæĎŘçŽĎæŮűåĂŹ except èŕ■åŘěæŸŕéąžåžŘæčĂæ§ěçŽĎïijŇçňňäÿĂäÿłåŇźéĚ■ç ä¡ššŘŕäžěå¿Ĺåőźæ ŸŞçŽĎæđĎéĂåď Žäÿł except åŘŇæŮűåŇźéĚ■çŽĎæČĚå¡ćïijŇæŕŤåęĆïijŽ

```
>>> f = open('missing')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
FileNotFoundError: [Errno 2] No such file or directory: 'missing'
>>> try:
...     f = open('missing')
... except OSError:
...     print('It failed')
... except FileNotFoundError:
...     print('File not found')
...
It failed
>>>
```

è£ŹéĞŇçŽĎ FileNotFoundError èŕ■åŘěåźűæšąæIJĽæĽğèąŇçŽĎåŐ§åŻäæŸŕ OSError æŻťäÿĂèĹňïijŇåőČåŘŕåŇźéĚ■ FileNotFoundError åijĆåÿÿïijŇ äžŐæŸŕåřśæŸŕçňňäÿĂäÿłåŇźéĚ■çŽĎãĂĆ åIJĺèřČèŕŢçŽĎæŮűåĂŹïijŇåęĆæđIJä¡ăåŕźæ§ŘäÿłçĽźåőŽåijĆåÿÿ ä¡ăåŘŕäžěéĂŽè£ĞæšěçIJŃèŕěåijĆåÿÿçŽĎ __mro__ åśđæĂğæİěå£ńéĂ§æţŘèğĹãĂĆæŕŤåęĆïijŽ

```
>>> FileNotFoundError.__mro__
(<class 'FileNotFoundError'>, <class 'OSError'>, <class 'Exception'>
↪,
 <class 'BaseException'>, <class 'object'>)
>>>
```

äÿŁéİćåĹŮèąĺäÿ■äżżä¡ŢäÿĂäÿłçŻťåĹř BaseException çŽĎçśżéČ¡èČ¡èćńçŤĺäžŐ except èŕ■åŘěãĂĆ

16.7 14.7 æ■ŢèŐůæĽĂæIJĽåijĆåÿÿ

éŮőéćŸ

æĂŐæăůæ■ŢèŐůäżčçăĄäÿ■çŽĎæĽĂæIJĽåijĆåÿÿïij§

èğčåĘşæŰźæąĹ

æČşèęĄæ■ŢèŐůæĽĂæIJĽçŽĎåijĆåÿÿïijŇåŘŕäżěçŻťæŐěæ■ŢèŐů Exception å■şåŘŕïijŽ

```
try:
   ...
except Exception as e:
   ...
   log('Reason:', e)        # Important!
```

è£ŹäÿłåřĘäijŽæ■ŢèŐůéŹďäžĘ SystemExit ãĂĄ KeyboardInterrupt åŠŇ GeneratorExit äźŃåďŰçŽĎæĽĂæIJĽåijĆåÿÿãĂĆ åęĆæđIJä¡äè£ŸæČşæ■ŢèŐůè£ŹäÿĽäÿłåijĆåÿÿïijŇåřĘ Exception æŤźæĹŘ BaseException å■şåŘŕãĂĆ

èőĺèőž

æ■ŢèŐůæĽĂæIJĽåijĆåÿÿéĂŽåÿÿæŸŕçŤśäžŐçĺŃåžŘåŚŸåIJĺæ§ŘäžŻåď■æİĆæŞ■ä¡IJäÿ■åźűäÿ■èČ¡èőŕ åęĆæđIJä¡ääÿ■æŸŕå¿ĹçżĘå£ČçŽĎäžžïijŇè£ŹäźŕçijŰåĘŹäÿ■æŸŞèřČèŕŢäžčçăĄçŽĎäÿĂäÿłçőĂå■ŢæŰź

æ■čåŽäåęĆæ■ď'ïijŇåęĆæđIJä¡äéĂĽ'æŃl'æ■ŢèŐůæĽĂæIJĽåijĆåÿÿïijŇéĆčäźĹåIJĺæ§ŘäÿłåIJřæŰźïijĹæ åęĆæđIJä¡äæšąæIJĽè£ŹæăůåĄŽïijŇæIJĽæŮűåĂŹä¡äçIJŃåĹřåijĆåÿÿæĽŞå■řæŮűåŘŕèČ¡æŚÿäÿ■çİĂåď'èĎ

```
def parse_int(s):
    try:
```

(continues on next page)

(continued from previous page)

```
        n = int(v)
    except Exception:
        print("Couldn't parse")
```

èŕŢçİĂè£ŘèąŇè£ŹäÿłåĞ¡æŢřïijŇçżŞæđIJåęĆäÿŃïijŽ

```
>>> parse_int('n/a')
Couldn't parse
>>> parse_int('42')
Couldn't parse
>>>
```

è£ŹæŮűåĂŹä¡ăåřśäijŽæŇăåď'ť'æČşïijŽâĂIJè£ŹåŠŃåŻđäžŃåŢŁïij§âĂİ
åĄĞåęĆä¡ăåČŔäÿŃéİćè£ŹæăůéĞ■åĘŹè£ŹäÿłåĞ¡æŢřïijŽ

```
def parse_int(s):
    try:
        n = int(v)
    except Exception as e:
        print("Couldn't parse")
        print('Reason:', e)
```

è£ŹæŮűåĂŹä¡ăèČ¡èŐůåŔŰåęĆäÿŃè¿ŞåĞžïijŇæŇĞæŸŐäžĘæIJL'äÿłçijŰçÍŃéŤŹèŕřïijŽ

```
>>> parse_int('42')
Couldn't parse
Reason: global name 'v' is not defined
>>>
```

å¿ŁæŸŐæŸ¿ïijŇä¡ăåžŤèŕěåř¡åŔŕèČ¡åřĘåijĆåÿÿåď'ĎçŘĘåŹÍåőŽäźL'çŽĎçš¿åĞĘäÿĂäžŻãĂĆ
äÿ■è£ĞïijŇèęĄæŸŕä¡ăå£Ěéążæ■ŢèŐůæL'ĂæIJL'åijĆåÿÿïijŇçąőä£ÍæL'Şå■řæ■čçąőçŽĎèŕŁæŰ■ä£ąæĄŕæŁŰ

16.8 14.8 åŁŻåżžèĞłåőŽäźL'åijĆåÿÿ

éŮőécŸ

åIJÍä¡ăæđĎåżžçŽĎåžŤçŤÍçÍŃåžŔäÿ■ïijŇä¡ăæČşåřĘåžŢåśĆåijĆåÿÿåŇĚèčĚæŁŘèĞłåőŽäźL'çŽĎåijĆåÿÿ

èğčåĘşæŰźæąŁ

åŁŻåżžæŰřçŽĎåijĆåÿÿå¿ŁçőĂå■ŢâĂŤâĂŤåőŽäźL'æŰřçŽĎçśżïijŇèőľ'åőČçżğæL'£èĞł
`Exception` ïijŁæŁŰèĂĚæŸŕäżżä¡ŢäÿĂäÿłåůšå■ŸåIJÍçŽĎåijĆåÿÿçśżåđŃïijL'ãĂĆ
ä¿ŃåęĆïijŇåęĆæđIJä¡ăçijŰåĘŹç¡ŚçżIJçŻÿåĚşçŽĎçÍŃåžŔïijŇä¡ăåŔŕèČ¡äijŽåőŽäźL'äÿĂäžŻçśżäijijåęĆäÿŃçŽ

```
class NetworkError(Exception):
    pass
```

(continues on next page)

(continued from previous page)

```
class HostnameError(NetworkError):
    pass

class TimeoutError(NetworkError):
    pass

class ProtocolError(NetworkError):
    pass
```

çĎűåŘŐçŤĺæĹůářśåŔŕäżěåČŔéĂŽåÿÿéĆčæăůä¡£çŤĺè£ŹäžŻåijĆåÿÿäžĘïijŇä¿ŃåęĆïijŽ

```
try:
    msg = s.recv()
except TimeoutError as e:
    ...
except ProtocolError as e:
    ...
```

èőĺèőž

èĞłåőŽäźĽåijĆåÿÿçśżåžŤèŕěæĂżæŸŕçżğæĽ£èĞłåĘĚç¡őçŽĎ Exception çśżïijŇ æĹŰèĂĚæŸŕçżğæĽ£èĞłéĆčäžŻæIJňèžńåřśæŸŕäżŐ Exception çżğæĽ£èĂŇæĺěçŽĎçśżãĂĆ åř¡çőąæĽĂæIJĽçśżåŘŇæŮűäź§çżğæĽ£èĞł BaseException ïijŇä¡Ęä¡ääÿ■åžŤèŕěä¡£çŤĺè£Źäÿłåŧ§žçśżæĺěåőŽäźĽæŰřçŽĎåijĆåÿÿãĂĆ BaseException æŸŕäÿžçşżçż§éĂĂåĞžåijĆåÿÿèĂŇä£ĺçŢŹçŽĎïijŇæŕŤåęĆ KeyboardInterrupt æĹŰ SystemExit äżěåŘŁåĚűäżŰéĆčäžŻäijŽçżŹåžŤçŤĺåŔŚéĂĄä£ąåŘůèĂŇéĂĂåĞžçŽĎåijĆåÿÿãĂĆ åŽäæ■d'ïijŇæ■ŢèŐůè£ŹäžŻåijĆåÿÿæIJňèžńæšąäżĂäźĽæĎŔäźĽ'ãĂĆ è£ŹæăůçŽĎèŕĺïijŇåĄĞåęĆä¡äçżğæĽ'£ BaseException åŔŕèČ¡äijŽåŕijèĞť'ä¡äçŽĎèĞłåőŽäźĽåijĆåÿÿäÿ■ä

åIJĺçĺŃåžŔäÿ■åijŢåĚěèĞłåőŽäźĽåijĆåÿÿåŔŕäżěä¡£å¿Ůä¡äçŽĎäżčçăĄæŻť'åĚůåŔŕèŕżæĂğïijŇèČ¡æÿĚæŽ è£ŸæIJĽäÿĂçğ■èő¿èőąæŸŕåřĘèĞłåőŽäźĽåijĆåÿÿéĂŽè£ĞçżğæĽ£çżĎåŘĹèţůæĺěãĂĆåIJĺåď■æĺĆåžŤçŤĺçĺŃ ä¡£çŤĺå§žçśżæĺěåĹĘçżĎåŘĎçğ■åijĆåÿÿçśżäź§æŸŕå¿ĹæIJĽçŤĺçŽĎãĂĆåőČåŔŕäżěèőľçŤĺæĹůæ■ŢèŐůäÿĂä

```
try:
    s.send(msg)
except ProtocolError:
    ...
```

ä¡äè£ŸèČ¡æ■ŢèŐůæŻť'åď'ğèŇČåŻť'çŽĎåijĆåÿÿïijŇåřśåČŔäÿŃéĺćè£ŹæăůïijŽ

```
try:
    s.send(msg)
except NetworkError:
    ...
```

åęĆæđIJä¡äæČşåőŽäźĽçŽĎæŰřåijĆåÿÿéĞ■åĘŹäžĘ `__init__()` æŰźæşŢïijŇ çąőä£ĺä¡ää¡£çŤĺæĽ'ĂæIJĽåŔĆæŢřèřČçŤĺ `Exception.__init__()` ïijŇä¿ŃåęĆïijŽ

```
class CustomError(Exception):
    def __init__(self, message, status):
        super().__init__(message, status)
        self.message = message
        self.status = status
```

çIJŃäÿŁåŐżæIJL'çĆźåĕĞæĂłïijŃäÿ■è£ĞExceptionçŽĎéżŸèőď'èąŃäÿžæŸŕæŐĕåŘŮæL'ĂæIJL'äijăéĂŠç
`.args` åśđæĂğäÿ■. å¿Ĺåď'ŽåĚűäżŰåĞ¡æŢřåžŞåŠŃéČĺåĹĘPythonåžŞéżŸèőď'æL'ĂæIJL'åijĆåÿÿéČ¡å£Ĕéążą
`.args` åśđæĂğïijŃ åŽăæ■ď'åęĆæđIJä¡ăå£¡çŢĕäžĘè£ŹäÿĂæ■ĕïijŃä¡ääijŽåŘŚçŐřæIJL'äžŻæŮűåĂŹä¡ääőŽäź
äÿžäžĘæijŤçď'ž `.args` çŽĎä¡£çŤĺïijŃèĂČèŹŚäÿŃäÿŃéĺćè£Źäÿłä¡£çŤĺåĘĚç¡őçŽĎ *RuntimeError*' åijĆåÿÿçŽĎäžď'äžŠäijŽèŕĺïijŃ æşĺæĎŘçIJŃraiseèŕ■åŘĕäÿ■ä¡£çŤĺçŽĎåŘĆæŢřäÿłæŢřæŸŕæĂŐæă

```
>>> try:
...     raise RuntimeError('It failed')
... except RuntimeError as e:
...     print(e.args)
...
('It failed',)
>>> try:
...     raise RuntimeError('It failed', 42, 'spam')
... except RuntimeError as e:

...     print(e.args)
...
('It failed', 42, 'spam')
>>>
```

åĚşäžŐåĹżåżżèĞłåőŽäźL'åijĆåÿÿçŽĎæŻť'åď'Žä£ąæĄŕïijŃèŕůåŘĆèĂČ'PythonåőŸæŰźæŰĞæąč <https://docs.python.org/3/tutorial/errors.html>'_

16.9 14.9 æ■ŢèŐůåijĆåÿÿåŘŐæŁŻåĞžåŘęåď'ŰçŽĎåijĆåÿÿ

éŮőéćŸ

ä¡äæČşæ■ŢèŐůäÿĂäÿłåijĆåÿÿåŘŐæŁŻåĞžåŘęåď'ŰäÿĂäÿłäÿ■åŘŃçŽĎåijĆåÿÿïijŃåŘŃæŮűè£Ÿå¿ŮåIJ

èğčåĘşæŰźæąĹ

äÿžäžĘéŞ¿æŐĕåijĆåÿÿïijŃä¡£çŤĺ `raise from` èŕ■åŘĕæĺĕäžčæŻ£çőĂå■ŢçŽĎ `raise` èŕ■åŘĕãĂĆ åőČäijŽèőľä¡ääŘŃæŮűä£ĺçŢŹäÿď'äÿłåijĆåÿÿçŽĎä£ąæĄŕãĂĆä¿ŃåęĆïijŽ

```
>>> def example():
...     try:
...             int('N/A')
...     except ValueError as e:
...             raise RuntimeError('A parsing error occurred') from␣
↪e
```

(continues on next page)

(continued from previous page)

```
...
>>> example()
Traceback (most recent call last):
  File "<stdin>", line 3, in example
ValueError: invalid literal for int() with base 10: 'N/A'
```

äÿŁéĺćçŽĎåijĆåÿÿæŸŕäÿŃéĺćçŽĎåijĆåÿÿäžğçŤ§çŽĎçŻťæŐěåŐ§åŻăïijŽ

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "<stdin>", line 5, in example
RuntimeError: A parsing error occurred
>>>
```

åIJĺåŻđæžŕäÿ■åŔŕäžěçIJŃåĹřïijŇäÿďäÿłåijĆåÿÿéČ¡èćńæ■ŢèŐůãĂĆ èęĄæČşæ■ŢèŐůè£ŹæăůçŽĎåijĆåÿÿïijŇä¡ăåŔŕäžěä¡£çŤĺäÿĂäÿłçőĂå■ŢçŽĎ `except` èŕ■åŘěãĂĆ äÿ■è£ĞïijŇä¡ăè£ŸåŔŕäžěéĂŽè£Ğæ§ěçIJŃåijĆåÿÿåŕźèśąçŽĎ `__cause__` åśđæĂğæĺěèů§èÿłåijĆåÿÿéŞ¿ãĂĆä¿ŃåęĆïijŽ

```
try:
    example()
except RuntimeError as e:
    print("It didn't work:", e)

    if e.__cause__:
        print('Cause:', e.__cause__)
```

å¡ŞåIJĺ `except` åĺŮäÿ■åŔĹæIJL'åŘęåď'ŰçŽĎåijĆåÿÿèćńæŁŻåĞžæŮűäijŽåŕijèĞť'äÿĂäÿłéŽŘèŮŔçŽĎå

```
>>> def example2():
...     try:
...             int('N/A')
...     except ValueError as e:
...             print("Couldn't parse:", err)
...
>>>
>>> example2()
Traceback (most recent call last):
  File "<stdin>", line 3, in example2
ValueError: invalid literal for int() with base 10: 'N/A'
```

åIJĺåď'ĎçŘĘäÿŁè£řåijĆåÿÿçŽĎæŮűåĂŹïijŇåŘęåď'ŰäÿĂäÿłåijĆåÿÿåŔŚçŤ§äžĘïijŽ

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "<stdin>", line 5, in example2
NameError: global name 'err' is not defined
>>>
```

è£Źäÿłä¿Ńå■Řäÿ■ïijŇä¡ăåŘŇæŮűèŐůå¿ŮäžĘäÿď'äÿłåijĆåÿÿçŽĎä£ąæĄŕïijŇä¡ĘæŸŕåŕźåijĆåÿÿçŽĎèğč

è£ŹæŮűåĂŹïijŇNameError åijĆåÿÿèćńä¡IJäÿžçÍŃåžŔæIJĂçżĹåijĆåÿÿèćńæŁŻåĞžïijŇèĂŇäÿ■æŸŕä¡■äžŐ

åęĆæđIJïijŇä¡ăæČşå£¡çŢěæŐĽåijĆåÿÿéŞ¿ïijŇåŔŕä¡£çŤÍ raise from None:

```
>>> def example3():
...     try:
...             int('N/A')
...     except ValueError:
...             raise RuntimeError('A parsing error occurred') from␣
↪None
...
>>>
example3()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "<stdin>", line 5, in example3
RuntimeError: A parsing error occurred
>>>
```

èőÍèőž

åIJÍèő¿èőąäżčçăĄæŮűïijŇåIJÍåŔęåď ŰäÿĂäÿł except äżčçăĄåİŮäÿ■ä¡£çŤÍ raise èŕ■åŔěçŽĎæŮűåĂŹä¡ăèęĄçĽ ŹåĹńåřŔå£ČäžĘãĂĆ åď ğåď ŽæŢřæČĔåĘţäÿŃïijŇè£Źçğ■ raise èŕ■åŔěéČ¡åžŤèŕěèćńæŤźæĹŘ raise from èŕ■åŔěãĂĆäź§åřśæŸŕèŕť ä¡ăåžŤèŕěä¡£çŤÍäÿŃéİćè£Źçğ

```
try:
   ...
except SomeException as e:
   raise DifferentException() from e
```

è£ŹæăůåĄŽçŽĎåŐ§åŻăæŸŕä¡ăåžŤèŕěæŸ¿çď žçŽĎåřĘåŐ§åŻăéŞ¿æŐěèţůæİěãĂĆ äź§åřśæŸŕèŕť ïijŇDifferentException æŸŕçŻť æŐěäżŐ SomeException èą■çŤ§èĂŇæİěãĂĆ è£Źçğ■åĚşçşżåŔŕäżěäżŐåŻđæžŕçżŞæđIJäÿ■çIJŃåĞžæİěãĂĆ

åęĆæđIJä¡ăåČŔäÿŃéİćè£ŹæăůåĘŹäżčçăĄïijŇä¡ăäż■çĎűäijŽå¿ŮåĹřäÿĂäÿłéŞ¿æŐěåijĆåÿÿïijŇ äÿ■è£Ğè£ŹäÿłåźűæšąæľJĽå¿ĹæÿĚæŹřçŽĎèŕť æŸŐè£ŹäÿłåijĆåÿÿéŞ¿åĹřåžŢæŸŕåĘĚéČĺåijĆåÿÿè£ŸæŸŕæ§

```
try:
   ...
except SomeException:
   raise DifferentException()
```

å¡Şä¡ăä¡£çŤÍ raise from èŕ■åŔěçŽĎèŕĺïijŇåřśå¿ĹæÿĚæěŽçŽĎèąĺæŸŐæŁŻåĞžçŽĎæŸŕçňňäžŇäÿłå

æIJĂåŘŐäÿĂäÿłä¿Ńå■Řäÿ■éŽŘèŮŔåijĆåÿÿéŞ¿ä£ąæĄŕãĂĆ åř¡çőąéŽŘèŮŔåijĆåÿÿéŞ¿ä£ąæĄŕäÿ■åĹľ äžŐåŻđæžŕïijŇåŘŇæŮűåőČäź§äÿćåď śäžĘå¿Ĺåď ŽæIJĽçŤÍçŽĎèřČ äÿ■è£ĞäÿĞäžŃçŽĘåźşç■ĽïijŇæIJĽæŮűåĂŹåŔłä£İçŢŹéĂĆå¡ŞçŽĎä£ąæĄŕäź§æŸŕå¿ĹæIJĽçŤÍçŽĎãĂĆ

16.10 14.10 éĞ■æŰřæŁŻåĞžèćńæ■ŢèŐůçŽĎåijĆåÿÿ

éŮőéćŸ

ä¡ăåIJÍäÿĂäÿł `except` åİŮäÿ■æ■ŢèŐůäžĘäÿĂäÿłåijĆåÿÿïijŇçŐřåIJÍæČşéĞ■æŰřæŁŻåĞžåőČãĂĆ

èğčåĘşæŰźæąĹ

çőĂå■ŢçŽĎä¡£çŤÍäÿĂäÿłå■ŢçŃňçŽĎ `rasie` èŕ■åŘěå■şåŘŕïijŇä¿ŃåęĆïijŽ

```
>>> def example():
...     try:
...             int('N/A')
...     except ValueError:
...             print("Didn't work")
...             raise
...

>>> example()
Didn't work
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "<stdin>", line 3, in example
ValueError: invalid literal for int() with base 10: 'N/A'
>>>
```

èőÍèőž

è£ŹäÿłéŮőéćŸéĂŽåÿÿæŸŕå¡Şä¡ăéIJĂèęĄåIJÍæ■ŢèŐůåijĆåÿÿåŘŐæĽğèąŇæ§ŘäÿłæŞ■ä¡IJïijĹæŕŤåęĆèć
äÿĂäÿłå¿ĹåÿÿèğĄçŽĎçŤÍæşŢæŸŕåIJÍæ■ŢèŐůæĽĂæIJĽåijĆåÿÿçŽĎåď ĎçŘĘåŹÍäÿ■ïijŽ

```
try:
    ...
except Exception as e:
    # Process exception information in some way
    ...

    # Propagate the exception
    raise
```

16.11 14.11 è¿ŞåĞžè■ęåŚŁä£ąæĄŕ

éŮőéćŸ

ä¡ăåÿŇæIJŻèĞłåůśçŽĎçÍŃåžŘèČ¡çŤ§æĹŘè■ęåŚŁä£ąæĄŕïijĹæŕŤåęĆåž§åijČçĽźæĂğæĹŰä¡£çŤÍéŮőéć

èğčåĘşæŰźæąĹ

èęĄè¿ŞåĞžäÿĂäÿłè■ęåŚŁæűĹæĄŕïijŇåŔŕä¡£çŤÍ `warning.warn()` åĞ¡æŢřãĂĆä¿ŃåęĆïijŽ

```
import warnings

def func(x, y, logfile=None, debug=False):
    if logfile is not None:
         warnings.warn('logfile argument deprecated',␣
↪DeprecationWarning)
    ...
```

`warn()` çŽĎåŘĆæŢřæŸŕäÿĂäÿłè■ęåŚŁæűĹæĄŕåŠŇäÿĂäÿłè■ęåŚŁçśżïijŇè■ęåŚŁçśżæIJĹ'åęĆäÿŃåĞăć DeprecationWarning, SyntaxWarning, RuntimeWarning, ResourceWarning, æĹŰ FutureWarning.

åŕźè■ęåŚŁçŽĎåď'ĎçŘĘåŘŰåĘşäžŐä¡ääęĆä¡Ţè£ŘèąŇèğčéĞŁåŹĺäżěåŘŁäÿĂäžŻåĚűäżŰéĚ■ç¡őãĂĆ ä¿ŃåęĆïijŇåęĆæđIJä¡ää¡£çŤÍ `-W all` éĂĹ'éąźåŐżè£ŘèąŇPythonïijŇä¡ääijŽå¿ŮåĹřåęĆäÿŃçŽĎè¿ŞåĞžïijŽ

```
bash % python3 -W all example.py
example.py:5: DeprecationWarning: logfile argument is deprecated
  warnings.warn('logfile argument is deprecated',␣
↪DeprecationWarning)
```

éĂŽåÿÿæÌěèőšïijŇè■ęåŚŁäijŽè¿ŞåĞžåĹřæăĞåĞĘéŤŹèŕŕäÿŁãĂĆåęĆæđIJä¡äæČşèőšè■ęåŚŁè¡ňæ■ćäÿžå `-W error` éĂĹ'éąźïijŽ

```
bash % python3 -W error example.py
Traceback (most recent call last):
  File "example.py", line 10, in <module>
    func(2, 3, logfile='log.txt')
  File "example.py", line 5, in func
    warnings.warn('logfile argument is deprecated',␣
↪DeprecationWarning)
DeprecationWarning: logfile argument is deprecated
bash %
```

èőÍèőž

åIJÍä¡äçżť'æŁď'è¡ŕäżűïijŇæŔŘçď'žçŤÍæĹůæ§ŘäžŻä£ąæĄŕïijŇä¡ĘæŸŕåŔĹäÿ■éIJĂèęĄåřĘåĚűäÿŁå■Ğäÿ ä¿ŃåęĆïijŇåĄĞèő¿ä¡ääĞĘåď'Ğä£őæŤźæ§ŘäÿłåĞ¡æŢřåžŞæĹŰæąĘæđűçŽĎåŁ§èČ¡ïijŇä¡ääŘŕäżěåĚĹäÿžä¡äč ä¡äè£ŸåŘŕäżěè■ęåŚŁçŤÍæĹůäÿĂäžŻåŕźäżčçăĄæIJĹ'éŮőécŸçŽĎä¡£çŤÍæŰźåijŘãĂĆ

ä¡IJäÿžåŘęåď'ŰäÿĂäÿłåĘĚç¡őåĞ¡æŢřåžŞçŽĎè■ęåŚŁä¡£çŤÍä¿Ńå■ŘïijŇäÿŃéÌćæijŤçď'žäžĘäÿĂäÿłæšąæ

```
>>> import warnings
>>> warnings.simplefilter('always')
>>> f = open('/etc/passwd')
```

(continues on next page)

(continued from previous page)

```
>>> del f
__main__:1: ResourceWarning: unclosed file <_io.TextIOWrapper name=
↪'/etc/passwd'
 mode='r' encoding='UTF-8'>
>>>
```

éżŸèőď æČĚåĘţäÿŃïijŇåźűäÿ■æŸŕæĽ ĂæIJĽè■ęåŚŁæűĹæĄŕéČ¡äijŽåĞžçŐřãĂĆ-W éĂĽéąźèČ¡æŐğåĹűè■ęåŚŁæűĹæĄŕçŽĎè¿ŞåĞžãĂĆ -W all äijŽè¿ŞåĞžæĽ ĂæIJĽè■ęåŚŁæűĹæĄŕïijŇ-W ignore å£¡çŢěæŐĽ æĽ ĂæIJĽè■ęåŚŁïijŇ-W error åřĘè■ęåŚŁè¡ňæ■ćæĹŘåijĆåÿÿãĂĆ åŘęåď ŰäÿĂçğ■éĂĽ æŃľïijŇä¡ăè£ŸåŘŕäżěä¡£çŢĺ warnings.simplefilter() åĞ¡æŢřæŐğåĹűè¿ŞåĞžãĂĆ always åŘĆæŢřäijŽèőľ æĽ ĂæIJĽè■ęåŚŁæűĹæĄŕåĞžçŐřïijŇ`ignore å£¡çŢěěřČæĽ ĂæIJĽçŽĎè■ęåŚŁïijŇerror åřĘè■ęåŚŁè¡ňæ■ćæĹŘåijĆåÿÿãĂĆ

åŕźäžŐçőĂå■ŢçŽĎçŢ§æĹŘè■ęåŚŁæűĹæĄŕçŽĎæČĚåĘţè£ŹäžŻåůšçżŘèűşåď §äžĘãĂĆ warnings æĺąåİŮåŕźè£ĞæżďåŠŇè■ęåŚŁæűĹæĄŕåď ĎçŘĘæŘŘä¿ŻäžĘåď ğéĞŘçŽĎæŻť énŸçžğçŽĎéĚ■ç¡ æŻť åď Žä£ąæĄŕèřůåŘĆèĂČ PythonæŰĞæąč

16.12 14.12 èřČèŕŢå§žæIJňçŽĎçÍŃåžŘåť ľæžČéŤŹèŕŕ

éŮőécŸ

ä¡äçŽĎçÍŃåžŘåť ľæžČåŘŐèřěæĂŐæăůåŐżèřČèŕŢåőČïij§

èğčåĘşæŰźæąĹ

åęĆæđIJä¡äçŽĎçÍŃåžŘåŻääÿžæ§Řäÿłåij Ćåÿÿè ĂŇåť ľæžČïijŇè£ŘèąŇ python3 -i someprogram.py åŘŕæĽ ğèąŇçőĂå■ŢçŽĎèřČèŕŢãĂĆ -i éĂĽéąźåŘŕèőľçÍŃåžŘçżŞæİ§åŘŐæĽ ŞåijĂäÿĂäÿłäžď ažŠåijŘshellãĂĆ çĎűåŘŐä¡ăářśèČ¡æ§ěçIJŃçŐřăćČïijŇä¿ŃåęĆïijŇåĄĞèő¿ä¡ăæIJĽäÿŃéİćçŽĎäžčçăĄïijŽ

```
# sample.py

def func(n):
    return n + 10

func('Hello')
```

è£ŘèąŇ python3 -i sample.py äijŽæIJĽçśżäijijåęĆäÿŃçŽĎè¿ŞåĞžïijŽ

```
bash % python3 -i sample.py
Traceback (most recent call last):
  File "sample.py", line 6, in <module>
    func('Hello')
  File "sample.py", line 4, in func
    return n + 10
```

(continues on next page)

(continued from previous page)

```
TypeError: Can't convert 'int' object to str implicitly
>>> func(10)
20
>>>
```

åęĆæđIJä¡ăçIJŃäÿ■åĹřäÿŁéİćè£ŹæăůçŽĎïijŇåŘŕäžěåIJÍçÍŃåžŘåť'ľæžČåŘŐæĽŞåijĂPythonçŽĎèřČèŕŢ

```
>>> import pdb
>>> pdb.pm()
> sample.py(4)func()
-> return n + 10
(Pdb) w
  sample.py(6)<module>()
-> func('Hello')
> sample.py(4)func()
-> return n + 10
(Pdb) print n
'Hello'
(Pdb) q
>>>
```

åęĆæđIJä¡ăçŽĎäžčçăĄæĽĂåIJÍçŽĎçŐŕåćČå¿ĹéŽ¿èŐůåŘŰäžď'äžŠshellïijĹæŕŤåęĆåIJÍæ§Řäÿłæ IJ■åŁą
éĂŽåÿÿåŘŕäžěæ■ŢèŐůåijĆåÿÿåŘŐèĞłåůśæĽŞå■řèů§èÿłä£ąæĄŕãĂĆä¿ŃåęĆïijŽ

```
import traceback
import sys

try:
    func(arg)
except:
    print('**** AN ERROR OCCURRED ****')
    traceback.print_exc(file=sys.stderr)
```

èęĄæŸŕä¡ăçŽĎçÍŃåžŘæšąæIJĽ'åť'ľæžČïijŇèĂŇåŘłæŸŕäžğçŤ§äžĘäÿĂäžŻä¡ăçIJŃäÿ■æĞĆçŽĎçżŞæđI
ä¡ăåIJÍæĎ§åĚť'èűčçŽĎåIJřæŰźæŘŠåĚěäÿĂäÿŃ `print()` èŕ■åŘěäź§æŸŕäÿłäÿ■éŤŹçŽĎéĂĽ'æŃľ'ãĂĆ
äÿ■è£ĞïijŇèęĄæŸŕä¡ăæĽ'ŞçőŮè£ŹæăůåĄŽïijŇæIJĽ'äÿĂäžŻåřŘæŁĂåůğåŘŕäžěåÿőåŁľ'ä¡ăãĂĆ
éęŰåĚĹïijŇ`traceback.print_stack()` åĞ¡æŢŕäijŽä¡ăçÍŃåžŘè£ŘèąŇåĹřéĆčäÿłçĆźçŽĎæŮűåĂŹåĹŻ

```
>>> def sample(n):
...     if n > 0:
...             sample(n-1)
...     else:
...             traceback.print_stack(file=sys.stderr)
...
>>> sample(5)
  File "<stdin>", line 1, in <module>
  File "<stdin>", line 3, in sample
  File "<stdin>", line 3, in sample
  File "<stdin>", line 3, in sample
```

(continues on next page)

(continued from previous page)

```
  File "<stdin>", line 3, in sample
  File "<stdin>", line 3, in sample
  File "<stdin>", line 5, in sample
>>>
```

åŘęåď ŰïijŇä¡ăè£ŸåŘŕäżěåČŘäÿŃéİćè£ŹæăůäĭĭçŤÍ pdb.set_trace() åIJÍäżżä¡ŢåIJřæŰźæĽŃåŁÍçŽĎåŘŕåŁÍèřČèŕŢåŹÍïijŽ

```
import pdb

def func(arg):
    ...
    pdb.set_trace()
    ...
```

å¡ŞçÍŃåžŘæŕŤè¿Čåď ğèĂŇä¡ăæČşèřČèŕŢæŐğåĹűæţĄçÍŃäżěåŘŁåĞ¡æŢřåŘĆæŢřçŽĎæŮűåĂŹè£Źäÿłåĭĭ ä¿ŃåęĆïijŇäÿĂæŮęèřČèŕŢåŹÍåijĂåğŃè£ŘèąŇïijŇä¡ăåŕśèČ¡åď§ä¡£çŤÍ print æİěèğĆæţŃåŘŸéĞŘåĂijæĹŰæŢšåĞżæ§ŘäÿłåŚ¡äżď æŕŤåęĆ w æİěèŐůåŘŰè£¡èÿłä£ąæĄŕãĂĆ

èőÍèőž

äÿ■èęĄåřEèřČèŕŢåijĎçŽĎè£ĞäžŐåď■æİĆåŇŰãĂĆäÿĂäžŻçőĂå■ŢçŽĎéŤŹèŕŕåŘłéIJĂèęĄèğĆåŕ§çÍŃåž åőđéŹĚçŽĎéŤŹèŕŕäÿĂèĹňæŸŕåăĘæăĹçŽĎæIJĂåŘŐäÿĂèąÑãĂĆ ä¡ăåIJÍåijĂåŘŚçŽĎæŮűåĂŹïijŇäź§åŘŕäżěåIJÍä¡ăéIJĂèęĄèřČèŕŢçŽĎåIJřæŰźæŘŠåĚěäÿĂäÿŃ print() åĞ¡æŢřæİěèŕŁæŰ■ä£ąæĄŕïijĹåŘłéIJĂèęĄæIJĂåŘŐåŘŚåÿČçŽĎæŮűåĂŹåĹăéŹď è£ŹäžŻæĽŞå■ř

èřČèŕŢåŹÍçŽĎäÿĂäÿłåÿÿèğĄçŤÍæşŢæŸŕèğĆæţŃæ§ŘäÿłåůšçżŘåť ľæžČçŽĎåĞ¡æŢřäÿ■çŽĎåŘŸéĞŘãĂ ç§ěéĄŞæĂŐæăůåIJÍåĞ¡æŢřåť ľæžČåŘŐè£ŻåĚěèřČèŕŢåŹÍæŸŕäÿĂäÿłå¿ĹæIJĽçŤÍçŽĎæŁĂèČ¡ãĂĆ

å¡Ş ä¡ăæČşèğčåĽŰäÿĂäÿłéİđåÿÿåď■æİĆçŽĎçÍŃåžŘïijŇåžŢåśĆçŽĎæŐğåĹűéĂżè¿Śä¡ăäÿ■æŸŕå¿ĹæÿĚ æŘŠåĚě pdb.set_trace() è£ŹæăůçŽĎèŕ■åŘěåŕśå¿ĹæIJĽçŤÍäžĘãĂĆ

åőđéŹĚäÿŁïijŇçÍŃåžŘäijŽäÿĂçŻť è£ŘèąŇåĹŕçćŕåĹř set_trace() èŕ■åŘěä¡■ç¡őïijŇçĎűåŘŐçńŃél ňè£ŻåĚěèřČèŕŢåŹÍãĂĆ çĎűåŘŐä¡ăåŕśåŘŕäżěåĄŽæŻť åď ŽçŽĎäžŃäžĘãĂĆ

åęĆæđIJä¡ăä¡£çŤÍIDEæİěåĄŽPythonåijĂåŘŚïijŇéĂŽåÿÿIDEéČ¡äijŽæŘŘä¿ŻèĞłåůšçŽĎèřČèŕŢåŹÍæİěa æŻť åď Žè£ŹæŰźéİćçŽĎä£ąæĄŕåŘŕäżěåŘĆèĂČä¡ăä¡£çŤÍçŽĎIDEæĽŃåĘŇãĂĆ

16.13 14.13 çżŹä¡ăçŽĎçÍŃåžŘåĄŽæĂğèČ¡æţŃèŕŢ

éŮőécŸ

ä¡ăæČşæţŃèŕŢä¡ăçŽĎçÍŃåžŘè£ŘèąŇæĽĂèŁśèť źçŽĎæŮűéŮť åźůåĄŽæĂğèČ¡æţŃèŕŢãĂĆ

èğčåĘşæŰźæąĹ

åęĆæđIJä¡ăåŘłæŸŕçőĂå■ŢçŽĎæČşæţŃèŕŢäÿŃä¡ăçŽĎçĺŃåžŘæŢť'ä¡ŞèŁśèť'źçŽĎæŮűéŮť'ïijŇéĂŽåÿÿä¡£çŤĺUnixæŮűéŮť'åĞ¡æŢřåřśèąŇäžĘïijŇæŕŤåęĆïijŽ

```
bash % time python3 someprogram.py
real 0m13.937s
user 0m12.162s
sys  0m0.098s
bash %
```

åęĆæđIJä¡ăè£ŸéIJĂèęĄäÿĂäÿłçĺŃåžŘåŘĎäÿłçżĘèŁĆçŽĎèŕęçżĘæŁěåŚŁïijŇåŘŕäżěä¡£çŤĺ `cProfile` æĺąåĬŮïijŽ

```
bash % python3 -m cProfile someprogram.py
         859647 function calls in 16.016 CPU seconds

   Ordered by: standard name

   ncalls  tottime  percall  cumtime  percall␣
↪filename:lineno(function)
   263169    0.080    0.000    0.080    0.000 someprogram.
↪py:16(frange)
      513    0.001    0.000    0.002    0.000 someprogram.
↪py:30(generate_mandel)
   262656    0.194    0.000   15.295    0.000 someprogram.py:32(
↪<genexpr>)
        1    0.036    0.036   16.077   16.077 someprogram.py:4(
↪<module>)
   262144   15.021    0.000   15.021    0.000 someprogram.py:4(in_
↪mandelbrot)
        1    0.000    0.000    0.000    0.000 os.py:746(urandom)
        1    0.000    0.000    0.000    0.000 png.py:1056(_readable)
        1    0.000    0.000    0.000    0.000 png.py:1073(Reader)
        1    0.227    0.227    0.438    0.438 png.py:163(<module>)
      512    0.010    0.000    0.010    0.000 png.py:200(group)
    ...
bash %
```

äÿ■è£ĞéĂŽåÿÿæČĚåĘţæŸŕäżŃäžŐè£Źäÿď'äÿłæđĄçńŕäźŃéŮť'ãĂĆæŕŤåęĆä¡ăåůšçżŘç§ěéĄŞäżčçăĄè£Ì
åŕźäžŐè£ŹäžŻåĞ¡æŢřçŽĎæĂğèČ¡æţŃèŕŢïijŇåŘŕäżěä¡£çŤĺäÿĂäÿłçőĂå■ŢçŽĎèčĚéěřåŹĺïijŽ

```python
# timethis.py

import time
from functools import wraps

def timethis(func):
    @wraps(func)
    def wrapper(*args, **kwargs):
```

(continues on next page)

(continued from previous page)

```
        start = time.perf_counter()
        r = func(*args, **kwargs)
        end = time.perf_counter()
        print('{}.{} : {}'.format(func.__module__, func.__name__,␣
↪end - start))
        return r
    return wrapper
```

èęĄä¡£çŤÍè£ŹäÿłèčĚéěřåŹÍïijŇåŘłéIJĂèęĄåřĘåĚűæŤ¿ç¡őåIJÍä¡ăèęĄè£ŻèąŇæĂğèČ¡æţŃèŕŢçŽĎåĞ¡æŢ

```
>>> @timethis
... def countdown(n):
...     while n > 0:
...             n -= 1
...
>>> countdown(10000000)
__main__.countdown : 0.803001880645752
>>>
```

èęĄæţŃèŕŢæ§ŘäÿłäżčçăĄåİŮè£ŘèąŇæŮűéŮť'ïijŇä¡ăåŘŕäżěåőŽäźL'äÿĂäÿłäÿŁäÿŃæŰĞçőąçŘĘåŹÍïijŇ

```
from contextlib import contextmanager

@contextmanager
def timeblock(label):
    start = time.perf_counter()
    try:
        yield
    finally:
        end = time.perf_counter()
        print('{} : {}'.format(label, end - start))
```

äÿŃéİćæŸŕä¡£çŤÍè£ŹäÿłäÿŁäÿŃæŰĞçőąçŘĘåŹÍçŽĎä¿Ńå■ŘïijŽ

```
>>> with timeblock('counting'):
...     n = 10000000
...     while n > 0:
...             n -= 1
...
counting : 1.5551159381866455
>>>
```

åŕźäžŐæţŃèŕŢå¿ŁåřŘçŽĎäżčçăĄçL'Ğæőţè£ŘèąŇæĂğèČ¡ïijŇä¡£çŤÍ timeit æÍąåİŮäijŽå¿ŁæŰźä¿£ïijŇä¿ŃåęĆïijŽ

```
>>> from timeit import timeit
>>> timeit('math.sqrt(2)', 'import math')
0.1432319980012835
>>> timeit('sqrt(2)', 'from math import sqrt')
```

(continues on next page)

(continued from previous page)

```
0.10836604500218527
>>>
```

timeit äijŽæL'ğèąŇçňňäÿĂäÿłåŔĆæŢřäÿ■èŕ■åŔě100äÿĞæňąåźűèőąçőŮè£ŘèąŇæŮűéŮť'ãĂĆçňňäžŇäÿłåŔĆæŢřæŸŕè£ŘèąŇæţŃèŕŢäźŃåL'■éĚ■ç¡őçŐŕåćČãĂĆåęĆæđIJä¡äæČşæŤźåŔŸå¿łçŐŕæL'ğèąŇæňåŔŕäžěåČŔäÿŃéİćè£Źæăůèő¿ç¡ő number åŔĆæŢřçŽĎåĂijïijŽ

```
>>> timeit('math.sqrt(2)', 'import math', number=10000000)
1.434852126003534
>>> timeit('sqrt(2)', 'from math import sqrt', number=10000000)
1.0270336690009572
>>>
```

èőĺèőž

å¡ŞæL'ğèąŇæĂğèČ¡æţŃèŕŢçŽĎæŮűåĂŹïijŇéIJĂèęĄæşĺæĎŔçŽĎæŸŕä¡äèŐůåŔŰçŽĎçżŞæđIJéČ¡æŸŕè time.perf_counter() åĞ¡æŢřäijŽåIJĺçżŹåőŽåźşåŘřäÿŁèŐůåŔŰæIJĂénŸçš¿ážęçŽĎèőąæŮűåĂijãĂĆäÿ■è£ĞïijŇåőČäż■çĎűè£ŸæŸŕå§žäžŐæŮűéŠ§æŮűéŮť'ïijŇå¿Ĺåď'ŽåŻäçť'ääijŽå¡šåŞ■åĹřåőČçŽĎçš¿çąőážęïęĆæđIJä¡äåŕźäžŐæL'ğèąŇæŮűéŮť'æŻť'æĎ§åĚť'èűčïijŇä¡£çŤĺ time.process_time() æİěäżčæŻ£åőČãĂĆä¿ŃåęĆïijŽ

```
from functools import wraps
def timethis(func):
    @wraps(func)
    def wrapper(*args, **kwargs):
        start = time.process_time()
        r = func(*args, **kwargs)
        end = time.process_time()
        print('{}.{} : {}'.format(func.__module__, func.__name__,␣
↪end - start))
        return r
    return wrapper
```

æIJĂåŘŐïijŇåęĆæđIJä¡äæČşè£ŻèąŇæŻť'æůśåĚěçŽĎæĂğèČ¡åĹĘæđŘïijŇéĆčäźĹä¡äéIJĂèęĄèŕęçżĘéŸİ time ãĂĄtimeit åŠŇåĚűäżŰçŻÿåĚşæĺąåİŮçŽĎæŰĞæąčãĂĆè£ŹæăůäąäåŔŕäžěçŘĘèğčåŠŇåźşåŘřçŻÿåĚşçŽĎäůőåijĆäżěåŘŁäÿĂäžŻåĚűäżŰéŹůéŸśãĂĆè£ŸåŔŕäżěåŘĆèĂČ13.13åřŘèŁĆäÿ■çŻÿåĚşçŽĎäÿĂäÿłåĹŻäżžèőąæŮűåŹĺçśżçŽĎä¿Ńå■ŘãĂĆ

16.14 14.14 åŁäéĂ§çĺŃåžŔè£ŘèąŇ

éŮőécŸ

ä¡äçŽĎçĺŃåžŔè£ŘèąŇåď'łæĚćïijŇä¡äæČşåIJĺäÿ■ä¡£çŤĺåď'■æİĆæŁĂæIJŕæŕŤåęĆCæL'ľåśŢæĹŰJITçijŰ

èğčåĘşæŰźæąĹ

åĚşäžŐçÍŃåžŘäijŸåŇŰçŽĎçňňäÿĂäÿłåĞĘåĹŹæŸŕâĂIJäÿ■èęĄäijŸåŇŰâĂİïijŇçňňäžŇäÿłåĞĘåĹŹæŸŕ
åęĆæđIJä¡äçŽĎçÍŃåžŘè£ŘèąŇçijŞæĚćïijŇéęŰåĚĹä¡åå¿Ůä¡£çŤÍ14.13åřŘèŁĆçŽĎæŁĂæIJŕåĚĹåŕźåőČè£Żè

éĂŽåÿÿæİěèőšä¡ääijŽåŘŚçŐřä¡åå¿ŮçÍŃåžŘåIJÍåřŚæŢřåĞääÿłçČ■çĆźåIJřæŰźèŁśèť źäžĘåď ğéĞŘæŮűé
æŕŤåęĆåĘĚå■ŸçŽĎæŢřæ■őåď ĎçŘĘå¿łçŐŕãĂĆäÿĂæŮęä¡åőŽä¡■åĹřè£ŹäžŻçĆźïijŇä¡åřśåŘŕäžěä¡£çŤÍäÿŇ

ä¡£çŤÍåĞ¡æŢř

å¿Ĺåď ŽçÍŃåžŘåŚŸåĹŽåijĂåğŃäijŽä¡£çŤÍPythonèŕ■èÍĂåĘŹäÿĂäžŻçőĂå■ŢèĎŽæIJňãĂĆ
å¡ŞçijŰåĘŹèĎŽæIJňçŽĎæŮűåĂŹïijŇéĂŽåÿÿäźăæČŕäžĘåĘŹæŕńæŮăçżŞæđĎçŽĎäžčçăĄïijŇæŕŤåęĆïijŽ

```
# somescript.py

import sys
import csv

with open(sys.argv[1]) as f:
     for row in csv.reader(f):

         # Some kind of processing
         pass
```

å¿ĹåřŚæIJL'äžžç§ěéĄŞïijŇåČŘè£ŹæăůåőŽäźL'åIJÍåĚÍåśĂèŇČåŻť çŽĎäžčçăĄè£ŘèąŇèţůæİěèęĄæŕŤåőŽ
è£Źçğ■éĂ§åžęåůőåijĆæŸŕçŤśäžŐåśĂéČÍåŘŸéĞŘåŠŇåĚÍåśĂåŘŸéĞŘçŽĎåőđçŐřæŰźåijŘïijÍä¡£çŤÍåśĂéČÍ
åŻäæ■ď ïijŇåęĆæđIJä¡äæČşèőľ çÍŃåžŘè£ŘèąŇæŻť å£ńäžŻïijŇåŘłéIJĂèęĄåřĘèĎŽæIJňèŕ■åŘěæŤ¿åĚěåĞ¡æŢ

```
# somescript.py
import sys
import csv

def main(filename):
    with open(filename) as f:
         for row in csv.reader(f):
             # Some kind of processing
             pass

main(sys.argv[1])
```

éĂ§åžęçŽĎåůőåijĆåŘŰåĘşäžŐåőđéŹĚè£ŘèąŇçŽĎçÍŃåžŘïijŇäÿ■è£Ğæăźæ■őçżŘéłŇïijŇä¡£çŤÍåĞ¡æŢï
30%çŽĎæĂğèČ¡æŘŘå■ĞæŸŕå¿Ĺåÿÿèğ ĄçŽĎãĂĆ

åř¡åŘŕèČ¡åŐżæŐL'åśđæĂğèő£éŰő

æŕŘäÿĂæňąä¡£çŤÍçĆź(.)æŞ■ä¡IJçňęæİěèő£éŰőåśđæĂğçŽĎæŮűåĂŹäijŽåÿęæİěéćÍåď ŰçŽĎåijĂéŤĂãĂĆ
åőČäijŽèğęåŘŚçL'źåőŽçŽĎæŰźæşŢïijŇæŕŤåęĆ `__getattribute__()` åŠŇ
`__getattr__()` ïijŇè£ŹäžŽæŰźæşŢäijŽè£ŻèąŇå■ŮåĚÿæŞ■ä¡IJæŞ■ä¡IJãĂĆ

éĂŽåÿÿä¡ååŘŕäžěä¡£çŤÍ `from module import name`
è£ŹæăůçŽĎåŕijåĚěå¡ćåijŘïijŇäžěåŘŁä¡£çŤÍçżŚåőŽçŽĎæŰźæşŢãĂĆ
åĄĞèő¿ä¡äæIJL'åęĆäÿŇçŽĎäžčçăĄçL'ĞæőţïijŽ

```
import math

def compute_roots(nums):
    result = []
    for n in nums:
        result.append(math.sqrt(n))
    return result

# Test
nums = range(1000000)
for n in range(100):
    r = compute_roots(nums)
```

åIJÍæĹŚäžňæIJžåŹÍäÿŁéİćæţŃèŕŢçŽĎæŮűåĂŹïijŇè£ŹäÿłçÍŃåžŘèŁśèť źäžĘåď ğæęĆ40çğŠãĂĆçŐřåIJÍa compute_roots() åĞ¡æŢřåęĆäÿŃïijŽ

```
from math import sqrt

def compute_roots(nums):

    result = []
    result_append = result.append
    for n in nums:
        result_append(sqrt(n))
    return result
```

ä£őæŤźåŘŐçŽĎçL'ĹæIJňè£ŘèąŇæŮűéŮť åď ğæęĆæŸŕ29çğŠãĂĆåŤŕäÿĂäÿ■åŘŇäźŃåď ĎåřśæŸŕæűĹéZ çŤÍ sqrt() äžčæŻ£äžĘ math.sqrt() ãĂĆ The result. append() æŰźæşŢèćńèţŃçżŹäÿĂäÿłåśĂéČÍåŘŸéĞŘ result_append ïijŇçĎűåŘŐåIJÍåĘĚéČÍå¿łçŐŕäÿ■ä¡£çŤÍåőČãĂĆ

äÿ■è£ĞïijŇè£ŹäžŻæŤźåŘŸåŘłæIJL'åIJÍåď ğéĞŘéĞ■åď ■äžčçăĄäÿ■æL'■æIJL'æĎŘäźL'ïijŇæŕŤåęĆå¿łçĆ åŻäæ■ď ïijŇè£ŹäžŻäijŸåŇŰäź§åŘłæŸŕåIJÍæ§ŘäžŻçL'źåőŽåIJŕæŰźæL'■åžŤèŕěèćńä¡£çŤÍãĂĆ

çŘĘèğčåśĂéČÍåŘŸéĞŘ

äźŃåL'■æŘŘè£ĞïijŇåśĂéČÍåŘŸéĞŘäijŽæŕŤåĚÍåśĂåŘŸéĞŘè£ŘèąŇéĂ§åžęå£ńãĂĆ åŕźäžŐéćŚçźĄèő£éŮőçŽĎåŘ■çğřïijŇéĂŽè£ĞåřĘè£ŹäžŻåŘ■çğřåŘŸæĹŘåśĂéČÍåŘŸéĞŘåŘŕäžěåŁäéĂ§çÍŃä ä¿ŃåęĆïijŇçIJŃäÿŃäźŃåL'■åŕźäžŐ compute_roots() åĞ¡æŢřè£ŻèąŇä£őæŤźåŘŐçŽĎçL'ĹæIJňïijŽ

```
import math

def compute_roots(nums):
    sqrt = math.sqrt
    result = []
    result_append = result.append
    for n in nums:
        result_append(sqrt(n))
    return result
```

åIJÍè£ŹäÿłçL'ĹæIJňäÿ■ïijŇsqrt äžŐ match æÍąåİŮèćńæŃ£åĞžåźűæŤ¿åĚěäžĘäÿĂäÿłåśĂéČÍåŘŸéĞŘ

åęĆæđIJä¡ăè£ŘèąŇè£ŹäÿłäżčçăĄïijŇåď ğæęĆèŁśèť ź25çğŠïijĹåŕźäžŐäźŃåĽ■29çğŠåŘĹæŸŕäÿĂäÿłæŤźè£Żż
è£Źäÿłéćİåď ŰçŽĎåŁăéĂ§åŐ§åŻăæŸŕåŻääÿžåŕźäžŐåśĂéČĺåŘŸéĞŘ sqrt
çŽĎæ§ěæĽ¿èęĄå£ńäžŐåĚĺåśĂåŘŸéĞŘ sqrt

åŕźäžŐçśżäÿ■çŽĎåśđæĂğèő£éŮőäź§åŘŇæăůéĂĆçŤĺäžŐè£ŹäÿłåŐ§çŘĘãĂĆ
éĂŽåÿÿæİěèőšïijŇæ§ěæĽ¿æ§ŘäÿłåĂijæŕŤåęĆ self.name
äijŽæŕŤèő£éŮőäÿĂäÿłåśĂéČĺåŘŸéĞŘèęĄæĚćäÿĂäžŽãĂĆ åIJĺåĘĚéČĺå¿łçŐŕäÿ■ïijŇåŘŕäžěåřĘæ§ŘäÿłéIJĂè

```
# Slower
class SomeClass:
    ...
    def method(self):
        for x in s:
            op(self.value)

# Faster
class SomeClass:

    ...
    def method(self):
        value = self.value
        for x in s:
            op(value)
```

éĄ£åĚ■äÿ■å£ĚèęĄçŽĎæŁ¡èśą

äżżä¡ŢæŮűåĂŹå¡Şä¡ää¡£çŤĺéćİåď ŰçŽĎåď ĎçŘĘåśĆïijĹæŕŤåęĆèčĚéěřåŹĺãĂĄåśđæĂğèő£éŮőãĂĄæŘ
æŕŤåęĆçIJŃäÿŃåęĆäÿŃçŽĎè£ŹäÿłçśżïijŽ

```
class A:
    def __init__(self, x, y):
        self.x = x
        self.y = y
    @property
    def y(self):
        return self._y
    @y.setter
    def y(self, value):
        self._y = value
```

çŐřåIJĺè£ŻèąŇäÿĂäÿłçőĂå■ŢæţŃèŕŢïijŽ

```
>>> from timeit import timeit
>>> a = A(1,2)
>>> timeit('a.x', 'from __main__ import a')
0.07817923510447145
>>> timeit('a.y', 'from __main__ import a')
0.35766440676525235
>>>
```

åŘŕäżěçIJŃåĹřïijŇèő£éŮőåśđæĂğyçŻÿæŕŤåśđæĂğxèĂŇèĺĂæĚćçŽĎäÿ■æ■ćäÿĂçĆźçĆźïijŇåď ğæęĆæ
åęĆæđIJä¡ăåIJĺæĎŘæĂğèČ¡çŽĎèŕĺïijŇéĆčäźĹåřśéIJĂèęĄéĞ■æŰřåőąèğĘäÿŃåŕźäžŐyçŽĎåśđæĂğèő£éŮőåŽ

åęĆæđIJæšąæIJĽå£ĚèęĄïijŇåřśä¡£çŤÍçőĂå■ŢåśđæĂğåŘğãĂĆ
åęĆæđIJäżĚäżĚæŸŕåŻääÿżåĚűäżŰçijŰçÍŃèŕ■èÍĂéIJĂèęĄä¡£çŤÍgetter/setteråĞ¡æŢřåřśåŐżä£őæŤźäżčçäĄéč

ä¡£çŤÍåĘĚç¡őçŽĎåőźåŹÍ

åĘĚç¡őçŽĎæŢřæ■őçśżåđŃæŕŤåęĆå■ŮçňęäÿšãĂĄåĚČçżĎãĂĄåĹŮèąÍãĂĄéŻĘåŘĹåŠŇå■ŮåĚÿéČ¡æŸŕ
åęĆæđIJä¡äæČşèĞłåůśåőđçŐřæŰřçŽĎæŢřæ■őçżŞæđĎïijĹæŕŤåęĆéŞ¿æŐĕåĹŮèąÍãĂĄåźşèąąæăŚç■Ľ'ïijĽ'ïijŇ
éĆčäźĹèęĄæČşåIJÍæĂğèČ¡äÿŁè¿¿åĹřåĘĚç¡őçŽĎéĂ§åžęåĞääźŐäÿ■åŘŕèČ¡ïijŇåŽäæ■đ'ïijŇè£ŸæŸŕäźŰäźŰ

éĄ£åĚ■åĹŻåżżäÿ■å£ĚèęĄçŽĎæŢřæ■őçżŞæđĎæĹŰåđ'■åĹű

æIJĽæŮűåĂŹçÍŃåžŘåŚŸæČşæŸ¿æŚĘäÿŃïijŇæđĎéĂääÿĂäżŻåźűæšąæIJĽå£ĚèęĄçŽĎæŢřæ■őçżŞæđĎ

```
values = [x for x in sequence]
squares = [x*x for x in values]
```

äź§èőÿè£ŹéĞŇçŽĎæČşæşŢæŸŕéęŰåĚĹåřĘäÿĂäżŻåĂijæŤűéŻĘåĹřäÿĂäÿłåĹŮèąÍäÿ■ïijŇçĎűåŘŐä¡£çŤ
äÿ■è£ĞïijŇçňňäÿĂäÿłåĹŮèąÍåőŇåĚÍæšąæIJĽå£ĚèęĄïijŇåŘŕäżěçőĂå■ŢçŽĎåČŘäÿŃéÍćè£ŹæäůåĘŹïijŽ

```
squares = [x*x for x in sequence]
```

äÿŐæ■đ'çŻÿåĚşïijŇè£ŸèęĄæşÍæĎŘäÿŃéĆčäžŻåŕźPythonçŽĎåĚśäžńæŢřæ■őæIJżåĹűè£ĞäžŐåĄŘæĽ'ğ
æIJĽäżŻäżżåźűæšąæIJĽå¿Ĺåě¡çŽĎçŘĘèğčæĹŰä£ąäżżPythonçŽĎåĘĚå■ŸæÍąåđŃïijŇæżěçŤÍ
`copy.deepcopy()` äźŃçśżçŽĎåĞ¡æŢřäĆ éĂŽåÿÿåIJÍè£ŹäžŻäżčçäĄäÿ■æŸŕåŘŕäżěåŐżæŐĽ'åđ'■åĹűæŞ

èőÍèőž

åIJÍäijŸåŇŰäźŃåĽ'■ïijŇæIJĽå£ĚèęĄåĚĹçäŤçĺ'űäÿŃä¡£çŤÍçŽĎçőŮæşŢäĆ
éĂĽæŃĺ'äÿĂäÿłåđ'■æÍĆåžęäÿž O(n log n) çŽĎçőŮæşŢèęĄæŕŤä¡äåŐżèřĆæŢť'äÿĂäÿłåđ'■æÍĆåžęäÿž
O(n**2) çŽĎçőŮæşŢæĽ'ĂåÿęæİěçŽĎæĂğèČ¡æŘŘå■ĞèęĄåđ'ğå¿Ůåđ'ŽäĆ

åęĆæđIJä¡äèğĽ'å¿Ůä¡äè£ŸæŸŕå¿Ůè£ŻèąŃäijŸåŇŰïijŇéĆčäźĹèŕůäżŐæŢť'ä¡ŞèĂČèŹŚäĆ
ä¡IJäÿżäÿĂèĹňåĞĘåĹŹïijŇäÿ■èęĄåŕźçÍŃåžŘçŽĎæŕŘäÿĂäÿłéČÍåĹĘéČ¡åŐżäijŸåŇŰ,åŻääÿżè£ŹäžŻä£őæŤź
ä¡äåžŤèŕěäÿŞæşÍäžŐäijŸåŇŰäžğçŤ§æĂğèČ¡çŞűéĆĹçŽĎåIJŕæŰźïijŇæŕŤåęĆåĘĚéČÍå¿łçŐŕäĆ

ä¡äè£ŸèęĄæşÍæĎŘå¿őåřŘäijŸåŇŰçŽĎçżŞæđIJäĆä¿ŃåęĆèĂČèŹŚäÿŃéÍćåĹŻåżżäÿĂäÿłå■ŮåĚÿçŽĎä

```
a = {
    'name' : 'AAPL',
    'shares' : 100,
    'price' : 534.22
}

b = dict(name='AAPL', shares=100, price=534.22)
```

åŘŐéÍćäÿĂçğ■åĘŹæşŢæŻť'çőĂæť'ĄäÿĂäżŻïijĹä¡ääÿ■éIJĂèęĄåIJÍåĚşéŤőå■ŮäÿŁè¿ŞåĚěåijŢåŘůïijĽ'äĂ
äÿ■è£ĞïijŇåęĆæđIJä¡äåřĘè£ŹäÿđäÿłäžčçäĄçĽ'Ğæőţè£ŻèąŃæĂğèČ¡æţŃèŕŢåŕźæŕŤæŮűïijŇäijŽåŘŚçŐŕä¡£ç
`dict()` çŽĎæŰźåijŘäijŽæĚćäžĘ3åĂ■ãĂĆ çIJŃåĹřè£ŹäÿłïijŇä¡äæŸŕäÿ■æŸŕæIJĽ'åĘšåŁÍæŁŁæĽ'ĂæIJĽ'ä¡
`dict()` çŽĎäžčçäĄéČ¡æŻ£æ■ćæĹŘçňňäÿĂçğ■ãĂĆ äÿ■åđ'§ïijŇèĄłæŸŐçŽĎçÍŃåžŘåŚŸåŘłäijŽåĚşæşÍäžŰ

åęĆæđIJä¡äçŽĎäijŸåŇŰèęĄæśĆæŕŤè¿ČéńŸïijŇæIJňèŁĆçŽĎè£ŹäžŻçŐĂå■ŢæŁĂæIJŕæżąèűşäÿ■äžĘïij
ä¿ŃåęĆïijŇPyPyåůěçÍŃæŸŕPythonèğčéĞŁåŹÍçŽĎåŘęåđ'ŰäÿĂçğ■åőđçŐŕïijŇåőČäijŽåĹĘæđŘä¡äçŽĎçÍŃåžł

åőČæIJĽæŮűåĂŹèČ¡æđĄåď ğçŽĎæŘŘå■ĞæĂğèČ¡ïijŇéĂŽåÿÿåŘŕäžěæŐěè£ŚCäžčçăĄçŽĎéĂ§åžęăĂĆ
äÿ■è£ĞåŘŕæČIJçŽĎæŸŕïijŇåĹŕåĘŹè£ŹæIJňäźęä¡■ç¡őïijŇPyPyè£Ÿäÿ■èČ¡åőŇåĚĺæŤŕæŇĄPython3.
åŻäæ■ď ïijŇè£ŹäÿłæŸŕä¡ääřĘæİěéIJĂèęĄåŐżçăŤçľ űçŽĎãĂĆä¡äè£ŸåŘŕäžěèĂČèŹŚäÿŃNumbaåůěçÍŃïijŇ
NumbaæŸŕäÿĂäÿłåIJÍä¡ää¡£çŤÍèčĚéěřåŹÍæİěéĂĽæŃľPythonåĞ¡æŢřè£ŻèąŇäijŸåŇŰæŮűçŽĎåŁÍæĂĄçijŰ
è£ŹäžŻåĞ¡æŢřäijŽä¡£çŤÍLLVMèćńçijŰèŕŚæĹŘæIJňåIJřæIJžåŹÍçăĄãĂĆåőČåŘŇæäůåŘŕäžěæđĄåď ğçŽĎæŘ
ä¡ĘæŸŕïijŇèů§PyPyäÿĂæäůïijŇåőČåŕźäžŐPython 3çŽĎæŤŕæŇĄçŐřåIJÍè£ŸåĄIJçŢŹåIJÍåőđéłŇéŸűæőţãĂĆ

æIJĂåŘŐæĹŚåijŢçŤÍJohn Ousterhoutèŕť è£ĞçŽĎèŕÍä¡IJäÿžçżŞåř¿ïijŽâĂIJæIJĂåě¡çŽĎæĂğèČ¡äijŸåŇŰ
çŻť åĹŕä¡äçIJ§çŽĎéIJĂèęĄäijŸåŇŰçŽĎæŮűåĂŹåĘ■åŐżèĂČèŹŚåőČãĂĆçąőä£Íä¡äçÍŃåžŘæ■čçąőçŽĎè£Řè

17 çňňå■ĄäžŤçńäïijŽCèŕ■èÍĂæĽľåśŢ

æIJňçńäçÍĂçIJijäžŐäżŐPythonèő£éŮőCäžčçăĄçŽĎéŮőéćŸãĂĆèőÿåď ŽPythonåĘĚç¡őåžŞæŸŕçŤÍCåĘŹ
èő£éŮőCæŸŕèőľ PythonçŽĎåŕźçŐřæIJĽåžŞè£ŻèąŇäžď äžŠäÿĂäÿłéĞ■èęĄçŽĎçżĎæĹŘéČÍåĹĘãĂĆ
è£Źäź§æŸŕäÿĂäÿłå¡Şä¡äéÍćäÿť äżŐPython 2 åĹř Python 3æĽľåśŢäžčçăĄçŽĎéŮőéćŸãĂĆ
èŹ¡çĎűPythonæŘŘä¿ŻäžĘäÿĂäÿłåź£æşŻçŽĎçijŰçÍŃAPIïijŇåőđéŹĚäÿŁæIJĽå¿Ĺåď ŽæŰźæşŢæÍěåď ĎçŘĘ
çŻÿæŕŤèŕŢåŻ¿çżŹåĞžåŕźäžŐæŕŘäÿĂäÿłåŘŕèČ¡çŽĎåůěåĚůæĹŰæŁĂæIJŕçŽĎèŕęçżĘåŘĆèĂČïijŇ
æĹŚäźĹéĞĞçŤÍçŽĎæŸŕæŸŕéŻĘäÿ■åIJÍäÿĂäÿłåřŘçĽĞæőţçŽĎC++äżčçăĄïijŇäżěåŘŁäÿĂäžŻæIJĽäżčèąÍæ
è£ŹäÿłçżőæăĞæŸŕæŘŘä¿ŻäÿĂçşżåĹŮçŽĎçijŰçÍŃæÍąæÍ£ïijŇæIJĽçżŘéłŇçŽĎçÍŃåžŘåŚŸåŘŕäžěæĽľåśŢè

è£ŹéĞŇæŸŕæĹŚäžňåřĘåIJÍåď ğéČÍåĹĘçğŸçś■äÿ■åůěä¡IJçŽĎäžčçăĄïijŽ

```
/* sample.c */_method
#include <math.h>

/* Compute the greatest common divisor */
int gcd(int x, int y) {
    int g = y;
    while (x > 0) {
        g = x;
        x = y % x;
        y = g;
    }
    return g;
}

/* Test if (x0,y0) is in the Mandelbrot set or not */
int in_mandel(double x0, double y0, int n) {
    double x=0,y=0,xtemp;
    while (n > 0) {
        xtemp = x*x - y*y + x0;
        y = 2*x*y + y0;
        x = xtemp;
        n -= 1;
        if (x*x + y*y > 4) return 0;
    }
    return 1;
}
```

(continues on next page)

(continued from previous page)

```
/* Divide two numbers */
int divide(int a, int b, int *remainder) {
    int quot = a / b;
    *remainder = a % b;
    return quot;
}

/* Average values in an array */
double avg(double *a, int n) {
    int i;
    double total = 0.0;
    for (i = 0; i < n; i++) {
        total += a[i];
    }
    return total / n;
}

/* A C data structure */
typedef struct Point {
    double x,y;
} Point;

/* Function involving a C data structure */
double distance(Point *p1, Point *p2) {
    return hypot(p1->x - p2->x, p1->y - p2->y);
}
```

è£ŹæőţäżčçăĄåŇĚåŘńäžĘåď Žçğ■äÿ■åŘŇçŽĎCèŕ■èÍĂçijŰçÍŃçĽźæĂğãĂĆ éęŰåĚĹïijŇè£ŹéĞŇæIJĽå¿Ĺåď ŽåĞ¡æŢřæŕŤåęĆ `gcd()` åŠŇ `is_mandel()` ãĂĆ `divide()` åĞ¡æŢřæŸŕäÿĂäÿłè£ŤåŻđåď ŽäÿłåĂijçŽĎCåĞ¡æŢřä¿Ńå■ŘïijŇåĚűäÿ■æIJĽäÿĂäÿłæŸŕéĂŽè£Č `avg()` åĞ¡æŢřéĂŽè£ĞäÿĂäÿłCæŢřçżĎæĽğèąŇæŢřæ■őèĄŽéŻĘæŞ■ä¡IJãĂĆ`Point` åŠŇ `distance()` åĞ¡æŢřæűĽåŘŁåĹřäžĘCçżŞæđĎä¡ŞãĂĆ

åŕźäžŐæŐěäÿŃæĺěçŽĎæĽĂæIJĽåřŘèŁĆïijŇåĚĹåĄĞåőŽäÿŁéĺćçŽĎäżčçăĄåůšçżŘèćńåĘŹåĚěäžĘäÿĂçĎűåŘŐåőČäžňçŽĎåőŽäźĽèćńåĘŹåĚěäÿĂäÿłåŘ■åŘńâĂIJsample.hâĂĺçŽĎåď ťæŰĞäżűäÿ■ïijŇ åźűäÿŤèćńçijŰèŕŚäÿžäÿĂäÿłåžŞåŘńâĂIJlibsampleâĂĺïijŇèČ¡èćńéŞ¿æŐěåĹřåĚűäżŰCèŕ■èÍĂäżčçăĄäÿ■ãĂĆ çijŰèŕŚåŠŇéŞ¿æŐěçŽĎçżĘèŁĆä¿Ìæ■őçşżçż§çŽĎäÿ■åŘŇèĂŇäÿ■åŘŇïijŇä¡ĘæŸŕè£Źäÿłäÿ■æŸŕæĹŚäżňåĚ åęĆæđIJä¡äèęĄåď ĎçŘĘCäżčçăĄïijŇæĹŚäżňåĄĞåőŽè£ŹäżŽå§žçąĂçŽĎäÿIJèě£ä¡äéČ¡æŐŇæŘąäžĘãĂĆ

Contents:

17.1 15.1 ä¡£çŤÍctypesèő£éŮőCäžčçăĄ

éŮőéćŸ

ä¡ăæIJL'äÿĂäžŻCåĞ¡æŢřåůšçżŘèćńçijŰèřŚåĹřåĚśäžńåžŞæĹŰDLLäÿ■ãĂĆä¡ăåÿŇæIJŻåŘřäžěä¡£çŤÍçží
èĂŇäÿ■çŤÍçijŰåĘŹéćÍåď'ŰçŽĎCäžčçăĄæĹŰä¡£çŤÍçňňäÿL'æŰźæL'l'åśŢåůěåĚůãĂĆ

èğčåĘşæŰźæąĹ

åŕźäžŐéIJĂèęĄèřČçŤÍCäžčçăĄçŽĎäÿĂäžŻåřŘçŽĎéŮőéćŸïijŇéĂŽåÿÿä¡£çŤÍPythonæăĞåĞĘåžŞäÿ■çŽ
`ctypes` æÍąåÍŮåřśèűşåď'§äžĘãĂĆ èęĄä¡£çŤÍ `ctypes` ïijŇä¡ăéęŰåĚĹèęĄçąőä£Íä¡ăèęĄèő£éŮőçŽĎCäžčçăĄ
ïijĹåŘŇæăůçŽĎæđűæđĎãĂĄå■Ůåď'ğåřŘãĂĄçijŰèřŚåŹĺç■L'ïijL'çŽĎæ§ŘäÿłåĚśäžńåžŞäÿ■äžĘãĂĆ
äÿžäžĘè£ŻèąŇæIJňèŁĆçŽĎæijŤçď'žïijŇåĄĞèő¿ä¡ăæIJL'äÿĂäÿłåĚśäžńåžŞåŘ■å■ŮåŘń
`libsample.so` ïijŇéĞŇéÍćçŽĎåĘĚåőźåřśæŸŕ15çńäážŃçż■éČÍåĹĘéĆčæăůãĂĆ
åŘęåď'Ůè£ŸåĄĞèő¿è£ŹäÿŁ `libsample.so` æŰĞäžűèćńæŤ¿ç¡őåĹřä¡■äžŐ `sample.py` æŰĞäžűçŻÿåŘŇçŽĎçŻőå¡Ţäÿ■äžĘãĂĆ

èęĄèő£éŮőè£ŹäÿłåĞ¡æŢřåžŞïijŇä¡ăèęĄåĚĹæđĎåżžäÿĂäÿłåŇĚèčĚåőČçŽĎPythonæÍąåÍŮïijŇåęĆäÿŇè

```
# sample.py
import ctypes
import os

# Try to locate the .so file in the same directory as this file
_file = 'libsample.so'
_path = os.path.join(*(os.path.split(__file__)[:-1] + (_file,)))
_mod = ctypes.cdll.LoadLibrary(_path)

# int gcd(int, int)
gcd = _mod.gcd
gcd.argtypes = (ctypes.c_int, ctypes.c_int)
gcd.restype = ctypes.c_int

# int in_mandel(double, double, int)
in_mandel = _mod.in_mandel
in_mandel.argtypes = (ctypes.c_double, ctypes.c_double, ctypes.c_
↪int)
in_mandel.restype = ctypes.c_int

# int divide(int, int, int *)
_divide = _mod.divide
_divide.argtypes = (ctypes.c_int, ctypes.c_int, ctypes.
↪POINTER(ctypes.c_int))
_divide.restype = ctypes.c_int

def divide(x, y):
    rem = ctypes.c_int()
    quot = _divide(x, y, rem)
```

(continues on next page)

(continued from previous page)

```
    return quot,rem.value

# void avg(double *, int n)
# Define a special type for the 'double *' argument
class DoubleArrayType:
    def from_param(self, param):
        typename = type(param).__name__
        if hasattr(self, 'from_' + typename):
            return getattr(self, 'from_' + typename)(param)
        elif isinstance(param, ctypes.Array):
            return param
        else:
            raise TypeError("Can't convert %s" % typename)

    # Cast from array.array objects
    def from_array(self, param):
        if param.typecode != 'd':
            raise TypeError('must be an array of doubles')
        ptr, _ = param.buffer_info()
        return ctypes.cast(ptr, ctypes.POINTER(ctypes.c_double))

    # Cast from lists/tuples
    def from_list(self, param):
        val = ((ctypes.c_double)*len(param))(*param)
        return val

    from_tuple = from_list

    # Cast from a numpy array
    def from_ndarray(self, param):
        return param.ctypes.data_as(ctypes.POINTER(ctypes.c_double))

DoubleArray = DoubleArrayType()
_avg = _mod.avg
_avg.argtypes = (DoubleArray, ctypes.c_int)
_avg.restype = ctypes.c_double

def avg(values):
    return _avg(values, len(values))

# struct Point { }
class Point(ctypes.Structure):
    _fields_ = [('x', ctypes.c_double),
                ('y', ctypes.c_double)]

# double distance(Point *, Point *)
distance = _mod.distance
distance.argtypes = (ctypes.POINTER(Point), ctypes.POINTER(Point))
```

(continues on next page)

(continued from previous page)

```
distance.restype = ctypes.c_double
```

åęĆæđIJäÿĂåĹĞæ■čåÿÿïijŇä¡ăåřśåŘŕäžěåŁăè¡¡åźűä¡£çŤĺéĞŇéĺćåőŽäźĽçŽĎCåĞ¡æŢřäžĘãĂĆä¿ŃåęĆï

```
>>> import sample
>>> sample.gcd(35,42)
7
>>> sample.in_mandel(0,0,500)
1
>>> sample.in_mandel(2.0,1.0,500)
0
>>> sample.divide(42,8)
(5, 2)
>>> sample.avg([1,2,3])
2.0
>>> p1 = sample.Point(1,2)
>>> p2 = sample.Point(4,5)
>>> sample.distance(p1,p2)
4.242640687119285
>>>
```

èőĺèőž

æIJňåřŘèŁĆæIJĽå¿Ĺåď ŽåĂijå¿ŮæĹŚäžňèŕęçżĘèőĺèőžçŽĎåIJřæŰźãĂĆ éęŰåĚĹæŸŕåŕźäžŐCåŠŇPythonäžčçăĄäÿĂèţůæĽŞåŇĚçŽĎéŮőéćŸïijŇåęĆæđIJä¡ăåIJĺä¡£çŤĺ `ctypes` æĺěèő£éŰőçijŰèřŚåŘŐçŽĎCäžčçăĄïijŇ éĆčäźĹéIJĂèęĄçąőä£Ìè£ŹäÿłåĚśäžňåžŞæŤ¿åIJĺ `sample.py` æĺąåĺŮåŘŇäÿĂäÿłåIJřæŰźãĂĆ äÿĂçğ■åŕŕèČ¡æŸŕåřĘçŤ§æĹŘçŽĎ . `so` æŰĞäżűæŤ¿ç¡őåIJĺèęĄä¡£çŤĺőČçŽĎPythonäžčçăĄåŘŇäÿĂäÿłçŻőå¡ŢäÿŇãĂĆ æĹŚäžňåIJĺ `recipeâĂĭsample.py` äÿ■ä¡£çŤĺ `__file__` åŘŸéĞŕæĺěæ§ěçIJŃőČèćňőĽèčĚçŽĎä¡■ç¡ŏïijŇ çĎűåŘŐæđĎéĂääÿĂäÿłæŇĞåŘŚåŘŇäÿĂäÿłçŻőå¡ŢäÿŇç `libsample.so` æŰĞäżűçŽĎèůŕå¿ĎãĂĆ

åęĆæđIJCåĞ¡æŢřäžŞèćňőĽèčĚåĹřåĚűäżŰåIJřæŰźïijŇéĆčäźĹä¡ăåřśèęĄä£őæŤźçŻÿäžŤçŽĎèůŕå¿ĎãĂĆ åęĆæđIJCåĞ¡æŢřäžŞåIJĺä¡ăæIJżäŹĺäÿŁèćňőĽèčĚäÿžäÿĂäÿłæăĞåĞĘäžŞäžĘïijŇ éĆčäźĹåŕŕäżěä¡£çŤĺ `ctypes.util.find_library()` åĞ¡æŢřæĺěæ§ěæĽ¿ïijŽ

```
>>> from ctypes.util import find_library
>>> find_library('m')
'/usr/lib/libm.dylib'
>>> find_library('pthread')
'/usr/lib/libpthread.dylib'
>>> find_library('sample')
'/usr/local/lib/libsample.so'
>>>
```

äÿĂæŮęä¡äç§ěéĄŞäžĘCåĞ¡æŢřäžŞçŽĎä¡■ç¡ŏïijŇéĆčäźĹåřśåŘŕäżěåČŘäÿŇéĺćè£Źæăůä¡£çŤĺ `ctypes.cdll.LoadLibrary()` æĺěåŁăè¡¡åőČïijŇ åĚűäÿ■ `_path` æŸŕæăĞåĞĘäžŞçŽĎåĚĺèůŕå¿ĎïijŽ

```
_mod = ctypes.cdll.LoadLibrary(_path)
```

åĞ¡æŢřåžŞèćńåŁäè¡¡åŘŐïijŇä¡äéIJĂèęĄçijŰåĘźåĞääÿłèŕ■åŘěæİěæŘŘåŘŰçĽźåőŽçŽĎçňęåŘůåźűæŇĞ
åřśåČŘäÿŇéİćè£ŹäÿłäžčçăĄçĽĞæőţäÿĂæăůïijŽ

```
# int in_mandel(double, double, int)
in_mandel = _mod.in_mandel
in_mandel.argtypes = (ctypes.c_double, ctypes.c_double, ctypes.c_
↪int)
in_mandel.restype = ctypes.c_int
```

åIJÍè£ŹæőţäžčçăĄäÿ■ïijŇ`.argtypes` åśđæĂğæŸŕäÿĂäÿłåĚČçżĎïijŇåŇĚåŘńäžĘæ§ŘäÿłåĞ¡æŢřçŽĎè
èĂŇ `.restype` åřśæŸŕçŻÿäžŤçŽĎè£ŤåŻđçśżåđŇãĂĆ `ctypes`
åőŽäźĽäžĘåď ğéĞŘçŽĎçśżåđŇåŕźèśąïijĹæŕŤåęĆc_double, c_int, c_short, c_floatç■L'ïijL'ïijŇ
äžčèąĹäžĘåŕźåžŤçŽĎCæŢřæ■őçśżåđŇãĂĆåęĆæđIJä¡äæČşèől'PythonèČ¡åď§äijäéĂŠæ■čçąőçŽĎåŘĆæŢřçśżå
éĆčäźĹè£ŹäžŻçśżåđŇç■¿åŘ■çŽĎçżŚåőŽæŸŕå¿ĹéĞ■èęĄçŽĎäÿĂæ■ěãĂĆåęĆæđIJä¡äæšąæIJL'è£ŹäźĹåĄŽïi
è£ŸåŘŕèČ¡äijŽåŕijèĞť æŢť äÿłèğčéĞŁåŹĹè£ŽçĹŇæŇĆæŐL'ãĂĆ
ä¡£çŤĹctypesæIJL'äÿĂäÿłéžżçČęçĆźçŽĎåIJřæŰźæŸŕåŐ§çŤ§çŽĎCäžčçăĄä¡£çŤĹçŽĎæIJŕèŕ■åŘŕèČ¡èů§Pytho
`divide()` åĞ¡æŢřæŸŕäÿĂäÿłå¿Ĺåě¡çŽĎä¿Ňå■ŘïijŇåőČéĂŽè£ĞäÿĂäÿłåŘĆæŢřéŹď äžěåŘęäÿĂäÿłåŘĆæŢ
åř¡çőąè£ŹæŸŕäÿĂäÿłå¿ĹåÿÿèğĄçŽĎCæŁĂæIJŕïijŇä¡ĘæŸŕåIJĹPythonäÿ■å■ť äÿ■ç§ěéĄŞæĂŐæăůæÿĚæŹřçŽ
ä¿ŇåęĆïijŇä¡ääÿ■èČ¡åČŘäÿŇéİćè£ŹæăůçőĂå■ŢçŽĎåĄŽïijŽ

```
>>> divide = _mod.divide
>>> divide.argtypes = (ctypes.c_int, ctypes.c_int, ctypes.
↪POINTER(ctypes.c_int))
>>> x = 0
>>> divide(10, 3, x)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ctypes.ArgumentError: argument 3: <class 'TypeError'>: expected LP_
↪c_int
instance instead of int
>>>
```

åřśçőŮè£ŹäÿłèČ¡æ■čçąőçŽĎåůëä¡IJïijŇåőČäijŽè£ÍåŘ■PythonåŕźäžŐæŢť æŢřçŽĎäÿ■åŘŕæŻť æŤźåŐ§å
åŕźäžŐæűĽ'åŘŁåĹřæŇĞéŠĹçŽĎåŘĆæŢřïijŇä¡äéĂŽåÿÿéIJĂèęĄåĚĹæđĎåżžäÿĂäÿłçŻÿäžŤçŽĎctypesåŕźèśą

```
>>> x = ctypes.c_int()
>>> divide(10, 3, x)
3
>>> x.value
1
>>>
```

åIJÍè£ŹéĞŇïijŇäÿĂäÿł `ctypes.c_int` åőđä¿ŃèćńåĹŻåżżåźűä¡IJäÿžäÿĂäÿłæŇĞéŠĹèćńäijäè£ŻåŐżã
èů§æŹőéĂŽPythonæŢť å¡ćäÿ■åŘŇçŽĎæŸŕïijŇäÿĂäÿł `c_int`
åŕźèśąæŸŕåŘŕäżěèćńä£őæŤźçŽĎãĂĆ `.value` åśđæĂğåŘŕèćńçŤĹæĺěèŐůåŘŰæĹŰæŻť æŤźè£ŹäÿłåĂijãĂĆ

åŕźäžŐéĆčäžŻäÿ■åČŘPythonçŽĎCèřČçŤĹïijŇéĂŽåÿÿåŘŕäżěåĘŹäÿĂäÿłåřŘçŽĎåŇĚèčĚåĞ¡æŢřãĂĆ
è£ŹéĞŇïijŇæĹŚäžňèől' `divide()` åĞ¡æŢřéĂŽè£ĞåĚČçżĎæĺěè£ŤåŻđäÿď äÿłçżŞæđIJïijŽ

```
# int divide(int, int, int *)
_divide = _mod.divide
_divide.argtypes = (ctypes.c_int, ctypes.c_int, ctypes.
↪POINTER(ctypes.c_int))
_divide.restype = ctypes.c_int

def divide(x, y):
    rem = ctypes.c_int()
    quot = _divide(x,y,rem)
    return quot, rem.value
```

avg() åĞ¡æŢřåŔĹæŸŕäÿĂäÿłæŰřçŽĎæŇŚæĹŸãĂĆCäżčçăĄæIJ§æIJŻæŐěåŔŮåĹřäÿĂäÿłæŇĞéŚĹåŠ
ä¡ĘæŸŕïijŇåIJĺPythonäÿ■ïijŇæĹŚäżňå£ĚéążèĂČèŹŚè£ŹäÿłéŮőéćŸïijŽæŢřçżĎæŸŕåŢěïij§åőČæŸŕäÿĂäÿłåĹ
è£ŸæŸŕ array æĺąåĺŮäÿ■çŽĎäÿĂäÿłæŢřçżĎïij§è£ŸæŸŕäÿĂäÿł numpy
æŢřçżĎïij§è£ŸæŸŕèŕť æĽĂæIJĽéČ¡æŸŕïij§ åőđéŹĚäÿŁïijŇäÿĂäÿłPythonâĂIJæŢřçżĎâĂİæIJĽåď Žçğ■å¡ćå

DoubleArrayType æijŤçď žäžĘæĂŐæăůåď ĎçŘĘè£Źçğ■æČĚåĘţãĂĆ
åIJĺè£Źäÿłçśżäÿ■åőŽäźĽäžĘäÿĂäÿłå■ŢäÿłæŰźæşŢ from_param() ãĂĆ
è£ŹäÿłæŰźæşŢçŽĎèğŠèĽšæŸŕæŐěåŔŮäÿĂäÿłå■ŢäÿłåŔĆæŢřçĎůåŘŐåřĘåĚůåŘŚäÿŃè¡ňæ■ćäÿžäÿĂäÿłåŘĺ
ïijĹæIJňä¿Ńäÿ■æŸŕäÿĂäÿł ctypes.c_double çŽĎæŇĞéŚĹïijĽãĂĆ
åIJĺ from_param() äÿ■ïijŇä¡ăåŔŕäžěåĄŽäżżä¡Ţä¡äæČşåĄŽçŽĎäžŃãĂĆ
åŘĆæŢřçŽĎçśżåđŃåŘ■èćňæŘŘåŔŮåĞžæİěåźűèćňçŤĺäžŐåĹĘåŘŚåĹřäÿĂäÿłæŻť åĚůä¡ŞçŽĎæŰźæşŢäÿ■åŐ
ä¿ŃåęĆïijŇåęĆæđIJäÿĂäÿłåĹŮèąĺèćňäijäéĂŠè£ĞæİěïijŇéČčäźĹ typename åŕśæŸŕ list
ïijŇ çĎůåŘŐ from_list æŰźæşŢèćňèřČçŤĺãĂĆ

åŕźäžŐåĹŮèąĺåŠŇåĚČçżĎïijŇfrom_list æŰźæşŢåřĘåĚűè¡ňæ■ćäÿžäÿĂäÿł ctypes
çŽĎæŢřçżĎåŕźèśąãĂĆ è£ŹäÿłçIJŃäÿŁåŐżæIJĽçĆźåěĞæĂłïijŇäÿŃéİćæĹŚäżňä¡£çŤĺäÿĂäÿłäžď äžŠåijŔä¿Ń
ctypes æŢřçżĎïijŽ

```
>>> nums = [1, 2, 3]
>>> a = (ctypes.c_double * len(nums))(*nums)
>>> a
<__main__.c_double_Array_3 object at 0x10069cd40>
>>> a[0]
1.0
>>> a[1]
2.0
>>> a[2]
3.0
>>>
```

åŕźäžŐæŢřçżĎåŕźèśąïijŇfrom_array() æŘŘåŔŮåžŢåśĆçŽĎåĘĚå■ŸæŇĞéŚĹáźůåřĘåĚűè¡ňæ■ćäÿž
ctypes æŇĞéŚĹåŕźèśąãĂĆä¿ŃåęĆïijŽ

```
>>> import array
>>> a = array.array('d',[1,2,3])
>>> a
array('d', [1.0, 2.0, 3.0])
>>> ptr_ = a.buffer_info()
>>> ptr
```

(continues on next page)

(continued from previous page)

```
4298687200
>>> ctypes.cast(ptr, ctypes.POINTER(ctypes.c_double))
<__main__.LP_c_double object at 0x10069cd40>
>>>
```

from_ndarray() æijŤçď žäžĘåŕźäžŐ numpy æŢřçżĎçŽĎè¡ňæ■ćæŞ■ä¡IJãĂĆ éĂŽè£ĞåőŽäźĽ DoubleArrayType çśżåźűåIJÍ avg() çśżåđŃç■¿åŘ■äÿ■ä¡£çŤÍåőČïijŇ éĆčäźĹè£ŹäÿłåĞ¡æŢřåŕśèČ¡æŐěåŘŮåď Žäÿłäÿ■åŘŇçŽĎçśżæŢřçżĎè¿ŞåĚěäžĘïijŽ

```
>>> import sample
>>> sample.avg([1,2,3])
2.0
>>> sample.avg((1,2,3))
2.0
>>> import array
>>> sample.avg(array.array('d',[1,2,3]))
2.0
>>> import numpy
>>> sample.avg(numpy.array([1.0,2.0,3.0]))
2.0
>>>
```

æIJňèŁĆæIJĂåŘŐäÿĂéČÍåĹĘåŘŚä¡ăæijŤçď žäžĘæĂŐæăůåď ĎçŘĘäÿĂäÿłçőĂå■ŢçŽĎCçżŞæđĎãĂĆ åŕźäžŐçżŞæđĎä¡ŞïijŇä¡ăåŘłéIJÄèęĄåČŘäÿŃéİćè£ŹæăůçőĂå■ŢçŽĎåőŽäźĽäÿĂäÿłçśżïijŇåŇĚåŘńçŻÿåžŤç

```
class Point(ctypes.Structure):
    _fields_ = [('x', ctypes.c_double),
                ('y', ctypes.c_double)]
```

äÿĂæŮęçśżèćńåőŽäźĽåŘŐïijŇä¡ăåŕśåŘŕäžěåIJÍçśżåđŃç■¿åŘ■äÿ■æĹŰèĂĚæŸŕéIJÄèęĄåőđä¿ŃåŇŰçżŞ

```
>>> p1 = sample.Point(1,2)
>>> p2 = sample.Point(4,5)
>>> p1.x
1.0
>>> p1.y
2.0
>>> sample.distance(p1,p2)
4.242640687119285
>>>
```

æIJĂåŘŐäÿĂäžŻåŕŘçŽĎæŘŘçď žïijŽåęĆæđIJä¡ăæČşåIJÍPythonäÿ■èő£éŮőäÿĂäžŻåŕŘçŽĎCåĞ¡æŢřïijŇ ctypes æŸŕäÿĂäÿłå¿ĹæIJĹçŤÍçŽĎåĞ¡æŢřåžŞãĂĆ åř¡çőąåęĆæ■d'ïijŇåęĆæđIJä¡ăæČşèęĄåŐżèő£éŮőäÿĂä Swig (15.9èŁĆäijŽèőšåĹř) æĹŰ CythonïijĹ15.10èŁĆïijĹ'ãĂĆ

åŕźäžŐåď ğåđŃåžŞçŽĎèő£éŮőæIJĹäÿłäÿżèęĄéŮőéćŸïijŇçŤśäžŐctypesåźűäÿ■æŸŕåőŇåĚÍèĞłåŁÍåŇŰï éĆčäźĹä¡ăåŕśå£Ěéążè£śèť źåď ğéĞŕæŮűéŮť æĚçijŰåĘźæĹ'ĂæIJĹ'çŽĎçśżåđŃç■¿åŘ■ïijŇåŕśåČŕä¿Ňå■Řäÿ åęĆæđIJåĞ¡æŢřåžŞåď §åď■æĚĆïijŇä¡ăè£Ÿå¿ŮåŐżçijŰåĘźå¿Ĺåď ŽåŕŘçŽĎåŇĚèčĚåĞ¡æŢřåŠŇæŤŕæŇĄçśż åŘęåď ŰïijŇéŹď éĚđä¡ăåůšçżŘåőŇåĚĺçš¿éĂŽäžĘæĹ'ĂæIJĹ'ăžŢåśĆçŽĎCæŐěåŘčçżĘèŁĆïijŇåŇĚæŃňåĘĚå■ éĂŽåÿÿäÿĂäÿłå¿ĹåŕŘçŽĎäžčçăĄçijžéŹůãĂĄèő£éŮőèűŁçŢŇæĹŰåĚűäžŰçśżäijijéŤŹèŕŕåŕśèČ¡èőľ PythonçÍŇ

ä¡IJäÿž `ctypes` çŽĎäÿĂäÿłæŻ£äżčïijŇä¡ăè£ŸåŕŕäżěèĂČèŹŚäÿŃCFFIãĂĆCFFIæŘŘä¿ŻäžĘå¿Ĺåď Žç
ä¡ĘæŸŕä¡£çŤĺCèŕ■æşŢåźűæŤŕæŇĄæŻť åď ŽéńŸçžğçŽĎCäžčçăĄçśżåđŃãĂĆ
åĹřåĘŹè£ŹæIJňäźęäÿžæ■ćïijŇCFFIè£ŸæŸŕäÿĂäÿłçŻÿåŕźè¿ČæŰřçŽĎåůěçĺŃïijŇ
ä¡ĘæŸŕåőČçŽĎæţĄèąŇåžęæ■čåIJĺå£ńéĂ§äÿŁå■ĞäĂĆ çŤŽèĞşè£ŸæIJL'åIJĺèőĺèőžåIJĺPythonåřĘæĺěçŽĎçL'

17.2 15.2 çőĂå■ŢçŽĎCæL'ľåśŢæĺąåİŮ

éŮőécŸ

ä¡ăæČşäÿ■ä¿ĺéĺăåĚűäżŰåůěåĚůïijŇçŻť æŐěä¡£çŤĺPythonçŽĎæL'ľåśŢAPIæĺěçijŰåĘŹäÿĂäžŻçőĂå■Ţç

èğčåĘşæŰźæąĹ

åŕźäžŐçőĂå■ŢçŽĎCäžčçăĄïijŇæđĎåżžäÿĂäÿłèĞłåőŽäźL'æL'ľåśŢæĺąåİŮæŸŕå¿ĹåőźæŸŞçŽĎãĂĆ
ä¡IJäÿžçňňäÿĂæ■ěïijŇä¡äéIJĂèęĄçąőä£Ìä¡äçŽĎCäžčçăĄæIJL'äÿĂäÿłæ■čçąőçŽĎåď ť æŰĞäźűãĂĆä¿ŃåęĆïij

```
/* sample.h */

#include <math.h>

extern int gcd(int, int);
extern int in_mandel(double x0, double y0, int n);
extern int divide(int a, int b, int *remainder);
extern double avg(double *a, int n);

typedef struct Point {
    double x,y;
} Point;

extern double distance(Point *p1, Point *p2);
```

éĂŽåÿÿæĺěèőšïijŇè£Źäÿłåď ť æŰĞäźűèęĄåŕźåžŤäÿĂäÿłåůšçżŔèćńå■ŢçŃňçijŰèŕŚè£ĞçŽĎåžŞãĂĆ
æIJL'äžĘè£ŹäžŻïijŇäÿŃéĺćæĹŚäžňæijŤçď žäÿŃçijŰåĘŹæL'ľåśŢåĞ¡æŢřçŽĎäÿĂäÿłçőĂå■Ţä¿Ńå■ŘïijŽ

```
#include "Python.h"
#include "sample.h"

/* int gcd(int, int) */
static PyObject *py_gcd(PyObject *self, PyObject *args) {
  int x, y, result;

  if (!PyArg_ParseTuple(args,"ii", &x, &y)) {
    return NULL;
  }
  result = gcd(x,y);
  return Py_BuildValue("i", result);
}
```

(continues on next page)

(continued from previous page)

```
/* int in_mandel(double, double, int) */
static PyObject *py_in_mandel(PyObject *self, PyObject *args) {
  double x0, y0;
  int n;
  int result;

  if (!PyArg_ParseTuple(args, "ddi", &x0, &y0, &n)) {
    return NULL;
  }
  result = in_mandel(x0,y0,n);
  return Py_BuildValue("i", result);
}

/* int divide(int, int, int *) */
static PyObject *py_divide(PyObject *self, PyObject *args) {
  int a, b, quotient, remainder;
  if (!PyArg_ParseTuple(args, "ii", &a, &b)) {
    return NULL;
  }
  quotient = divide(a,b, &remainder);
  return Py_BuildValue("(ii)", quotient, remainder);
}

/* Module method table */
static PyMethodDef SampleMethods[] = {
  {"gcd",  py_gcd, METH_VARARGS, "Greatest common divisor"},
  {"in_mandel", py_in_mandel, METH_VARARGS, "Mandelbrot test"},
  {"divide", py_divide, METH_VARARGS, "Integer division"},
  { NULL, NULL, 0, NULL}
};

/* Module structure */
static struct PyModuleDef samplemodule = {
  PyModuleDef_HEAD_INIT,

  "sample",           /* name of module */
  "A sample module",  /* Doc string (may be NULL) */
  -1,                 /* Size of per-interpreter state or -1 */
  SampleMethods       /* Method table */
};

/* Module initialization function */
PyMODINIT_FUNC
PyInit_sample(void) {
  return PyModule_Create(&samplemodule);
}
```

èęĄçżŚåőŽè£ŹäÿłæL'ľåśŢæÍąåİŮïijŇåČŔäÿŃéİćè£ŹæăůåĹŻåżžäÿĂäÿł setup.py æŰĞäżűïijŽ

```
# setup.py
from distutils.core import setup, Extension

setup(name='sample',
      ext_modules=[
        Extension('sample',
                  ['pysample.c'],
                  include_dirs = ['/some/dir'],
                  define_macros = [('FOO','1')],
                  undef_macros = ['BAR'],
                  library_dirs = ['/usr/local/lib'],
                  libraries = ['sample']
                  )
        ]
)
```

äÿžäžĘæđĎåžžæIJĂçżĹçŽĎåĞ¡æŢřåžŞïijŇåŔłéIJĂçőĂå■ŢçŽĎä¡£çŤÍ python3 buildlib.py build_ext --inplace åŚ¡äžď'å■şåŔŕïijŽ

```
bash % python3 setup.py build_ext --inplace
running build_ext
building 'sample' extension
gcc -fno-strict-aliasing -DNDEBUG -g -fwrapv -O3 -Wall -Wstrict-
 ↪prototypes
 -I/usr/local/include/python3.3m -c pysample.c
 -o build/temp.macosx-10.6-x86_64-3.3/pysample.o
gcc -bundle -undefined dynamic_lookup
build/temp.macosx-10.6-x86_64-3.3/pysample.o \
 -L/usr/local/lib -lsample -o sample.so
bash %
```

åęĆäÿŁæL'Ăçď'žïijŇåőČäijŽåĹŻåżžäÿĂäÿłåŘ■å■ŮåŔń sample.so çŽĎåĚśäžńåžŞãĂĆå¡Şèćńçij ŰèŕŚåŘŐïijŇä¡ăåřśèČ¡åřĘåőČä¡IJäÿžäÿĂäÿłæĺąåİŮåŕijåĚěè£ŻæİěäžĘïijŽ

```
>>> import sample
>>> sample.gcd(35, 42)
7
>>> sample.in_mandel(0, 0, 500)
1
>>> sample.in_mandel(2.0, 1.0, 500)

0
>>> sample.divide(42, 8)
(5, 2)
>>>
```

åęĆæđIJä¡äæŸŕåIJĺWindowsæIJžåŹĺäÿŁéĺćåřĺèŕŢè£ŹäžŻæ■ěéłď'ïijŇåŔŕèČ¡äijŽéĄĞåĹřåŘĎçğ■çŐŕåćČ PythonçŽĎäžŇè£ŻåĹűåĹĘåŘŚéĂŽåÿÿä¡£çŤĺäžĘMicrosoft Visual StudioæİěæđĎåżžãĂĆ äÿžäžĘèől'è£ŹäžŻæL'l'åśŢèČ¡æ■čåÿÿåůěä¡IJïijŇä¡äéIJĂèęĄä¡£çŤĺåŘŇæäůæĹŰåĚijåőźçŽĎåůěåĚůæİěçijŰè åŘĆèĂČçŻÿäžŤçŽĎ PythonæŰĞæąč

èőÍèőž

åIJÍåřÌèřŢäżżä¡ŢæĽŃåĘźæĽľåśŢäźŃåĽ■ïijŇæIJĂåě¡èČ¡åĚÍåŘĆèĂČäÿŃPythonæŰĞæąčäÿ■çŽĎ
æĽľåśŢåŠŇåţŇåĚěPythonèğčéĞŁåŹÍ . PythonçŽĎCæĽľåśŢAPIå¿Ĺåď ğïijŇåIJÍè£ŹéĞŇæŢť äÿłåŐżèőšè£řåć
äÿ■è£ĞåŕźäžŐæIJĂæäÿå£ČçŽĎéČÍåĹĘè£ŸæŸŕåŘŕäżěèőÍèőžäÿŃçŽĎãĂĆ

éęŰåĚĹïijŇåIJÍæĽľåśŢæĺąåİŮäÿ■ïijŇä¡ššĘźçŽĎåĞ¡æŢřéČ¡æŸŕåČŘäÿŃéİćè£ŹæšůçŽĎäÿĂäÿłæŹőéĂ

```
static PyObject *py_func(PyObject *self, PyObject *args) {
  ...
}
```

PyObject æŸŕäÿĂäÿłèČ¡èąĺçď žäżżä¡ŢPythonåŕźèśąçŽĎCæŢřæ■őçśżåđŃãĂĆ
åIJÍäÿĂäÿłéńŸçžğåśĆéİćïijŇäÿĂäÿłæĽľåśŢåĞ¡æŢřåřśæŸŕäÿĂäÿłæŐěåŘŮäÿĂäÿłPythonåŕźèśą
ïijĹåIJÍ PyObject *argsäÿ■ïijĽåĚČçżĎåźűè£ŤåŻđäÿĂäÿłæŰřPythonåŕźèśąçŽĎCåĞ¡æŢřãĂĆ
åĞ¡æŢřçŽĎ self åŘĆæŢřåŕźäžŐçőĂå■ŢçŽĎæĽľåśŢåĞ¡æŢřæšąæIJĽèćńä¡£çŤĺåĹřïijŇ
äÿ■è£ĞåęĆæđIJä¡äæČşåőŽäźĽæŰřçŽĎçśżæĹŰèĂĚæŸŕĆäÿ■çŽĎåŕźèśąçśżåđŃçŽĎèŕÍåřśèČ¡æť¿äÿŁçŤĺåIJž
éĆčäźĹ self åřśèČ¡åijŢçŤĺéĆčäÿłåőđä¿ŃäžĘãĂĆ

PyArg_ParseTuple() åĞ¡æŢřèćńçŤĺæİěåřĘPythonäÿ■çŽĎåĂijè¡ňæ■ćæĹŘCäÿ■åŕźåžŤèąĺçď žãĂĆ
åőČæŐěåŘŮäÿĂäÿłæŇĞåőŽè¿ŞåĚěæĭijåijŘçŽĎæĭijåijŘåŇŰå■Ůçňęäÿšä¡IJäÿžè¿ŞåĚěïijŇæŕŤåęĆâĂIJiâĂİ
åŘŇæĭůè£ŸæIJĽå■ŸæŤ¿è¡ňæ■ćåŘŐçżŞæđIJçŽĎCåŘŸéĞŘçŽĎåIJřåİĂãĂĆ
åęĆæđIJè¿ŞåĚěçŽĎåĂijäÿ■åŇźéĚ■è£ŹäÿłæĭijåijŘåŇŰå■ŮçňęäÿšïijŇåřśäijŽæŁŻåĞžäÿĂäÿłåijĆåÿÿåźűè£ŤĂ
éĂŽè£ĞæčĂæ§ěåźűè£ŤåŻđNULLïijŇäÿĂäÿłåŘĹéĂĆçŽĎåijĆåÿÿäijŽåIJÍèřČçŤĺäžčçăĄäÿ■èćńæŁŻåĞžãĂĆ

Py_BuildValue() åĞ¡æŢřèćńçŤĺæİěæĭźæ■őCæŢřæ■őçśżåđŃåĹŻåżżPythonåŕźèśąãĂĆ
åőČåŘŇæĭůæŐěåŘŮäÿĂäÿłæĭijåijŘåŇŰå■ŮçňęäÿšæİěæŇĞåőŽæIJ§æIJŻçśżåđŃãĂĆ
åIJÍæĽľåśŢåĞ¡æŢřäÿ■ïijŇåőČèćńçŤĺæİěè£ŤåŻđçżŞæđIJçżŹPythonãĂĆ
Py_BuildValue() çŽĎäÿĂäÿłçĽźæĂğæŸŕåőČèČ¡æđĎåżżæŻť åŁååď■æİĆçŽĎåŕźèśąçśżåđŃïijŇæŕŤåęĆ
åIJÍ py_divide() äžčçăĄäÿ■ïijŇäÿĂäÿłä¿Ňå■ŘæijŤçď žäžĘæĂŐæĭůè£ŤåŻđäÿĂäÿłåĚČçżĎãĂĆäÿ■è£Ğïï

```
return Py_BuildValue("i", 34);       // Return an integer
return Py_BuildValue("d", 3.4);      // Return a double
return Py_BuildValue("s", "Hello"); // Null-terminated UTF-8 string
return Py_BuildValue("(ii)", 3, 4); // Tuple (3, 4)
```

åIJÍæĽľåśŢæĺąåİŮåžŢéČÍïijŇä¡ääijŽåŘŚçŐŕäÿĂäÿłåĞ¡æŢřèąĺïijŇæŕŤåęĆæIJňèŁĆäÿ■çŽĎ
SampleMethods èąĺãĂĆ è£ŹäÿłèąĺåŘŕäżěåĹŮåĞžCåĞ¡æŢřãĂĄPythonäÿ■ä¡£çŤĺçŽĎåŘ■å■ŮãĂĄæŰĞæa
æĽĂæIJĽæĺąåİŮéČ¡éIJĂèęĄæŇĞåőŽè£ŹäÿłèąĺïijŇåŻääÿžåőČåIJÍæĺąåİŮåĹĺåğŃåŇŰæŮűèęĄèćńä¡£çŤĺåĹř

æIJĂåŘŐçŽĎåĞ¡æŢř PyInit_sample() æŸŕæĺąåİŮåĹĺåğŃåŇŰåĞ¡æŢřïijŇä¡ĘèŕěæĺąåİŮçňňäÿĂæŕ
è£ŹäÿłåĞ¡æŢřçŽĎäÿżèęĄåůěä¡IJæŸŕåIJÍèğčéĞŁåŹÍäÿ■æşĺåĘŇæĺąåİŮåŕźèśąãĂĆ

æIJĂåŘŐäÿĂäÿłèęĄçĆźéIJĂèęĄæŘŘåĞžæİěïijŇä¡£çŤĺCåĞ¡æŢřæİěæĽľåśŢPythonèęĄèĂČèŹŚçŽĎäžł
ïijĹåőđéŹĚäÿŁïijŇC APIåŇĚåŘńäžĘèŭĚè£Ğ500äÿłåĞ¡æŢřïijĽãĂĆä¡ššŽŤèŕěåřĘæIJňèŁĆå¡ŞåĄŽæŸŕäÿĂäÿ
æŽť åď ŽéńŸçžğåĘĚåőźïijŇåŘŕäżěçIJŃçIJŃ PyArg_ParseTuple() åŠŇ
Py_BuildValue() åĞ¡æŢřçŽĎæŰĞæąčïijŇ çĎůåŘŐè£ŽäÿĂæ■ěæĽľåśŢåijĂãĂĆ

17.3 15.3 çijŰåĘŹæĽľåśŢåĞ¡æŢřæŞ■ä¡IJæŢřçżĎ

éŮőéćŸ

ä¡ăæČşçijŰåĘŹäÿĂäÿłCæĽľåśŢåĞ¡æŢřæĺěæŞ■ä¡IJæŢřçżĎïijŇåŘŕèČ¡æŸŕèćńarrayæÍąåĺŮæĹŰçśżäijijN
äÿ■è£ĞïijŇä¡ăæČşèőľä¡ăçŽĎåĞ¡æŢřæŻť'åŁăéĂŽçŤÍïijŇèĂŇäÿ■æŸŕéŠĹåŕźæ§ŘäÿłçĽźåőŽçŽĎåžŞæĽĂçŤ

èğčåĘşæŰźæąĹ

äÿžäžĘèČ¡èőľæŐěåŘŮåŠŇåď'ĎçŘĘæŢřçżĎåĚůæIJĽåŘŕçğżæď'■æĂğïijŇä¡ăéIJĂèęĄä¡£çŤÍåĹř
Buffer Protocol . äÿŃéĺćæŸŕäÿĂäÿłæĽŃåĘŹçŽĎCæĽľåśŢåĞ¡æŢřä¿Ńå■ŘïijŇ
çŤÍæĺěæŐěåŘŮæŢřçżĎæŢřæ■őåźűèřČçŤÍæIJňçńäåijĂçŕĞéČÍåĹĘçŽĎ `avg(double *buf, int len)` åĞ¡æŢřïijŽ

```
/* Call double avg(double *, int) */
static PyObject *py_avg(PyObject *self, PyObject *args) {
  PyObject *bufobj;
  Py_buffer view;
  double result;
  /* Get the passed Python object */
  if (!PyArg_ParseTuple(args, "O", &bufobj)) {
    return NULL;
  }

  /* Attempt to extract buffer information from it */

  if (PyObject_GetBuffer(bufobj, &view,
      PyBUF_ANY_CONTIGUOUS | PyBUF_FORMAT) == -1) {
    return NULL;
  }

  if (view.ndim != 1) {
    PyErr_SetString(PyExc_TypeError, "Expected a 1-dimensional array
↪");
    PyBuffer_Release(&view);
    return NULL;
  }

  /* Check the type of items in the array */
  if (strcmp(view.format,"d") != 0) {
    PyErr_SetString(PyExc_TypeError, "Expected an array of doubles
↪");
    PyBuffer_Release(&view);
    return NULL;
  }

  /* Pass the raw buffer and size to the C function */
  result = avg(view.buf, view.shape[0]);
```

(continues on next page)

(continued from previous page)

```
  /* Indicate we're done working with the buffer */
  PyBuffer_Release(&view);
  return Py_BuildValue("d", result);
}
```

äÿŃéİćæĹŚäżňæijŤçď žäÿŃè£ŹäÿłæĽľ ľåśŢåĞ¡æŢřæŸŕåęĆä¡Ţåůëä¡IJçŽĎïijŽ

```
>>> import array
>>> avg(array.array('d',[1,2,3]))
2.0
>>> import numpy
>>> avg(numpy.array([1.0,2.0,3.0]))
2.0
>>> avg([1,2,3])
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'list' does not support the buffer interface
>>> avg(b'Hello')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: Expected an array of doubles
>>> a = numpy.array([[1.,2.,3.],[4.,5.,6.]])
>>> avg(a[:,2])
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: ndarray is not contiguous
>>> sample.avg(a)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: Expected a 1-dimensional array
>>> sample.avg(a[0])

2.0
>>>
```

èőĺèőž

åřĘäÿĂäÿłæŢřçżĎåŕźèśąäijäçżŹCåĞ¡æŢřåŔŕèČ¡æŸŕäÿĂäÿłæĽľ ľåśŢåĞ¡æŢřåĄŽçŽĎæIJĂåÿÿèğĄçŽĎäż
å¿Ĺåď ŽPythonåžŤçŤĺçĺŃåžŔïijŇäżŐåŻ¿åČŔåď ĎçŘĘåĹřçğŚå■ęèőąçőŮïijŇéČ¡æŸŕå§žäžŐénŸæĂğèČ¡çŽĎ
éĂŽè£ĞçijŰåĘŹèČ¡æŐěåŔŮåźűæŞ■ä¡IJæŢřçżĎçŽĎäżčçăĄïijŇä¡ăåŔŕäżěçijŰåĘŹå¿Ĺåě¡çŽĎåĚijåőźè£ŹäžŽ
èĂŇäÿ■æŸŕåŔłèČ¡åĚijåőźä¡äèĞłåůśçŽĎäżčçăĄãĂĆ

äżčçăĄçŽĎåĚşéŤőçĆźåIJĺäžŐ `PyBuffer_GetBuffer()` åĞ¡æŢřãĂĆ
çżŹåőŽäÿĂäÿłäżżæĎŔçŽĎPythonåŕźèśąïijŇåőČäijŽèŕŢçĺĂåŐżèŐůåŔŰåžŢåśĆåĘĚå■Ÿä£ąæĄŕïijŇåőČçőĂå
1. äijäçżŹ `PyBuffer_GetBuffer()` çŽĎçĽźæőŁæăĞå£ŮçżŹåĞžäžĘæĽĂéIJĂçŽĎåĘĚå■ŸçijŞåĘśçśżåč
ä¿ŃåęĆïijŇ`PyBUF_ANY_CONTIGUOUS` èąĺçď žæŸŕäÿĂäÿłè£đçż■çŽĎåĘĚå■ŸåŇžå§§ãĂĆ

åŕźäžŐæŢřçżĎãĂĄå■ŮèŁĆå■Ůçňęäÿšå ŠŇåĚűäżŰçśżäijijåŕźèśąèĂŇèĺĂïijŇäÿĂäÿł

Py_buffer çżŞæđĎä¡ŞåŇĚåŘńäžĘæĽĂæIJĽåžŢåŚĆåĘĚå■ŸçŽĎä£ąæĄŕãĂĆ
åőČåŇĚåŘńäÿĂäÿłæŇĞåŘŚåĘĚå■ŸåIJřåĬĂãĂĄåď ğåřŘãĂĄåĚČçť ăåď ğåřŘãĂĄæăijåijŘåŠŇåĚűäžŰçżĘèŁ

```
typedef struct bufferinfo {
    void *buf;              /* Pointer to buffer memory */
    PyObject *obj;          /* Python object that is the owner */
    Py_ssize_t len;         /* Total size in bytes */
    Py_ssize_t itemsize;    /* Size in bytes of a single item */
    int readonly;           /* Read-only access flag */
    int ndim;               /* Number of dimensions */
    char *format;           /* struct code of a single item */
    Py_ssize_t *shape;      /* Array containing dimensions */
    Py_ssize_t *strides;    /* Array containing strides */
    Py_ssize_t *suboffsets; /* Array containing suboffsets */
} Py_buffer;
```

æIJňèŁĆäÿ■ïijŇæĹŚäžňåŘłåĚşæşĺæŐěåŘŮäÿĂäÿłåŘŇçš¿åžęæţőçĆźæŢřæŢřçżĎä¡IJäÿžåŘĆæŢřãĂĆ
èęĄæčĂæ§ěåĚČçť ăæŸŕåŘęæŸŕäÿĂäÿłåŘŇçš¿åžęæţőçĆźæŢřïijŇåŘłéIJĂéłŇèřĄ
format åśđæĂğæŸŕäÿ■æŸŕå■ŮçňęäÿšâĂİdâĂİ. è£ŹäÿłäŹ§æŸŕ struct
æĺąåİŮçŤĺæİěçijŰçăĄäžŇè£ŻåĹűæŢřæ■őçŽĎãĂĆ éĂŽåÿÿæİěèőšïijŇformat
åŘŕäžěæŸŕäżżä¡ŢåĚijåőź struct æĺąåİŮçŽĎæăijåijŘåŇŰå■Ůçňęäÿšïij Ň
åźűäÿŤåęĆæđIJæŢřçżĎåŇĚåŘńäžĘCçżŞæđĎçŽĎèŕİåőČåŘŕäžěåŇĚåŘńåď ŽäÿłåĂijãĂĆ
äÿĂæŮęæĹŚäžňåůšçżŘçąőåőŽäžĘåžŢåśĆçŽĎçijŞå■ŸåŇžä£ąæĄŕïijŇéĆčåŘłéIJĂèęĄçőĂå■ŢçŽĎåřĘåőČäij
åőđéŹĚäÿŁïijŇæĹŚäžňäÿ■å£ĚæŇĚå£ČæŸŕæĂŐæăůçŽĎæŢřçżĎçśżåđŇæĹŰèĂĚåőČæŸŕèćńäžĂäźĹåžŞåĹŽ
è£Źäź§æŸŕäÿžäżĂäźĹè£ŹäÿłåĞ¡æŢřèČ¡åĚijåőź array æĺąåİŮäź§èČ¡åĚijåőź numpy
æĺąåİŮäÿ■çŽĎæŢřçżĎäžĘãĂĆ

åIJĺè£ŤåŻđæIJĂçżĹçżŞæđIJäźŇåĽ■ïijŇåžŢåśĆçŽĎçijŞåĘšåŇžèğĘåŻ¿å£Ĕéążä¡£çŤĺ
PyBuffer_Release() éĞŁæŤ¿æŐĽãĂĆ äźŇæĽĂäżěèęĄè£ŹäÿĂæ■ěæŸŕäÿžäžĘèČ¡æ■čçąőçŽĎçőąçŘ

åŘŇæăůïijŇæIJňèŁĆäź§äžĚäżĚåŘłæŸŕæijŤçď žäžĘæŐěåŘŮæŢřçżĎçŽĎäÿĂäÿłåřŘçŽĎäżčçăĄçĽĞæő
åęĆæđIJä¡äçIJ§çŽĎèęĄåď ĎçŘĘæŢřçżĎïijŇä¡ääŘŕèČ¡äijŽçćřåĹřåď Žçżť æŢřæ■õãĂĄåď ğæŢřæ■õãĂĄäÿ■å
éĆčäźĹåřšå¿ŮåŐżå■ęæŽť éńŸçžğçŽĎäÿIJèě£äžĘãĂĆä¡äéIJĂèęĄåŘĆèĂČåőŸæŰźæŰĞæąčæĬěèŐůåŘŰæŽť

åęĆæđIJä¡äéIJĂèęĄçijŰåĘŹæűĽåŘŁåĹřæŢřçżĎåď ĎçŘĘçŽĎåď ŽäÿłæĽľåśŢïijŇéĆčäźĹéĂŽè£ĞCytho

17.4 15.4 åIJĺCæĽľåśŢæĺąåİŮäÿ■æŞ■ä¡IJéŽŘå¡ćæŇĞéŠĹ

éŮőécŸ

ä¡äæIJĽäÿĂäÿłæĽľåśŢæĺąåİŮéIJĂèęĄåď ĎçŘĘĆçżŞæđĎä¡Ş äÿ■çŽĎæŇĞéŠĹïijŇ
ä¡ĘæŸŕä¡äåŘĹäÿ■æČşæŽť éIJšçżŞæđĎä¡Ş äÿ■äżżä¡ŢåĘĚéČĺçżĘèŁĆçżŹPythonãĂĆ

èğčåĘşæŰźæąĹ

éŽŘå¡ćçżŞæđĎä¡ŞåŘŕäžěå¿Ĺåőźæ ŸŞçŽĎéĂŽè£ĞåřĘåőČäžňåŇĚèčĚåIJĺèČűåŻŁåŕźèśąäÿ■æĬěåď ĎçŘĘ
èĂČèŹŚæĹŚäžňä¿Ňå■Řäžčçăąäÿ■çŽĎäÿŇåĹŮCäžčçăĄçĽĞæőţïijŽ

```
typedef struct Point {
    double x,y;
} Point;

extern double distance(Point *p1, Point *p2);
```

äÿŃéĬćæŸŕäÿĂäÿłä¡£çŤĺèČűåŻŁåŃĚèčĚPointçżŞæđĎä¡ŞåŠŃ `distance()` åĞ¡æŢřçŽĎæĽľåśŢäżčçăĄåőđä¿ŃïijŽ

```
/* Destructor function for points */
static void del_Point(PyObject *obj) {
  free(PyCapsule_GetPointer(obj,"Point"));
}

/* Utility functions */
static Point *PyPoint_AsPoint(PyObject *obj) {
  return (Point *) PyCapsule_GetPointer(obj, "Point");
}

static PyObject *PyPoint_FromPoint(Point *p, int must_free) {
  return PyCapsule_New(p, "Point", must_free ? del_Point : NULL);
}

/* Create a new Point object */
static PyObject *py_Point(PyObject *self, PyObject *args) {

  Point *p;
  double x,y;
  if (!PyArg_ParseTuple(args,"dd",&x,&y)) {
    return NULL;
  }
  p = (Point *) malloc(sizeof(Point));
  p->x = x;
  p->y = y;
  return PyPoint_FromPoint(p, 1);
}

static PyObject *py_distance(PyObject *self, PyObject *args) {
  Point *p1, *p2;
  PyObject *py_p1, *py_p2;
  double result;

  if (!PyArg_ParseTuple(args,"OO",&py_p1, &py_p2)) {
    return NULL;
  }
  if (!(p1 = PyPoint_AsPoint(py_p1))) {
    return NULL;
  }
  if (!(p2 = PyPoint_AsPoint(py_p2))) {
    return NULL;
```

(continues on next page)

(continued from previous page)

```
  }
  result = distance(p1,p2);
  return Py_BuildValue("d", result);
}
```

åIJÍPythonäÿ■åŘŕäżěåČŘäÿŃéİćè£Źæăůæİěä¡£çŤÍè£ŹäžŻåĞ¡æŢřïjŽ

```
>>> import sample
>>> p1 = sample.Point(2,3)
>>> p2 = sample.Point(4,5)
>>> p1
<capsule object "Point" at 0x1004ea330>
>>> p2
<capsule object "Point" at 0x1005d1db0>
>>> sample.distance(p1,p2)
2.8284271247461903
>>>
```

èőÍèőž

èČűåŻŁåŠŃCæŃĞéŠĹçśżäijijãĂĆåIJÍåĘĚéČÍïijŃåőČäżňèŐůåŘŰäÿĂäÿłéĂŽçŤÍæŃĞéŠĹåŠŃäÿĂäÿłåŘ
PyCapsule_New() åĞ¡æŢřå¿ĹåőźæŸŞçŽĎèćńåĹŻåżžãĂĆ
åŘęåď ŰïijŃäÿĂäÿłåŘŕéĂĽçŽĎæđŘæđĎåĞ¡æŢřèČ¡èćńçżŚåőŽåĹřèČűåŻŁäÿŁïijŃçŤÍæİěåIJÍèČűåŻŁåŕźèśąè

èęĄæŘŘåŘŰèČűåŻŁäÿ■çŽĎæŃĞéŠĹïijŃåŘŕä¡£çŤÍ PyCapsule_GetPointer()
åĞ¡æŢřåźűæŃĞåőŽåŘ■çğřãĂĆ åęĆæđIJæŘŘä¿ŻçŽĎåŘ■çğřåŠŃèČűåŻŁäÿ■åŃźéĚ■æĹŰåĚűäżŰéŤŹèŕŕåĞž

æIJňèŁĆäÿ■ïijŃäÿĂåŕźåůěåĚůåĞ¡æŢřâĂŤâĂŤ PyPoint_FromPoint()
åŠŃ PyPoint_AsPoint() èćńçŤÍæİěåĹŻåżžåŠŃäżŐèČűåŻŁåŕźèśąäÿ■æŘŘåŘŰ-
Pointåőđä¿ŃãĂĆ åIJÍäżżä¡ŢæĽľåśŢåĞ¡æŢřäÿ■ïijŃæĹŚäżňäijŽä¡£çŤÍè£ŹäžŻåĞ¡æŢřèĂŃäÿ■æŸŕçŻť æŐěä¡
è£Źçğ■èő¿èőąä¡£å¿ŮæĹŚäżňåŘŕäżěå¿ĹåőźæŸŞçŽĎåžŤåŕźåřĘæİěåŕźPointåžŢäÿŃçŽĎåŃĚèčĚçŽĎæŻť æŤźä¿ŃåęĆïijŃåęĆæđIJä¡ääĘşåőŽä¡£çŤÍåŘęåď ŰäÿĂäÿłèČűåŻŁäžĘïijŃéĆčäźĹåŘłéIJĂèęĄæŻť æŤźè£Źäÿďäÿłå

åŕźäžŐèČűåŻŁåŕźèśąäÿĂäÿłéŽ¿çĆźåIJÍäžŐåđČåIJ¿åŻđæŤűåŠŃåĘĚå■ŸçőąçŘĘãĂĆ
PyPoint_FromPoint() åĞ¡æŢřæŐěåŘŮäÿĂäÿł must_free
åŘĆæŢřïijŃ çŤÍæİěæŃĞåőŽå¡ŞèČűåŻŁèćńéŤĂæŕĄæŮűåžŢåśĆPoint *
çżŞæđĎä¡ŞæŸŕåŘęåžŤèŕěèćńåŻđæŤűãĂĆ åIJÍæ§ŘäżŻCäżčçăĄäÿ■ïijŃå¡ŠåśđéŮőéćŸéĂŽåÿÿå¿ĹéŽ¿èćńåď Ì
çÍŃåžŘåŚŸåŘŕäżěä¡£çŤÍ extra åŘĆæŢřæİěæŐğåĹűïijŃèĂŃäÿ■æŸŕå■ŢæŰźéÍćçŽĎåĘşåőŽåđČåIJ¿åŻđæŤ
èęĄæşÍæĎŘçŽĎæŸŕåŠŃçŐřæIJL'èČűåŻŁæIJL'åĚşçŽĎæđŘæđĎåŹÍèČ¡ä¡£çŤÍ
PyCapsule_SetDestructor() åĞ¡æŢřæİěæŻť æŤźãĂĆ

åŕźäžŐæűL'åŘŁåĹřçżŞæđĎä¡ŞçŽĎCäżčçăĄèĂŃèÍĂïijŃä¡£çŤÍèČűåŻŁæŸŕäÿĂäÿłæŕŤè¿ČåŘĹçŘĘçŽĎè
ä¿ŃåęĆïijŃæIJL'æŮűåĂŹä¡ääźűäÿ■åĚşå£ČæŻť éIJšçżŞæđĎä¡ŞçŽĎåĘĚéČÍä£ąæĄŕæĹŰèĂĚåřĘåĚűè¡ňæ■ćæ
éĂŽè£Ğä¡£çŤÍèČűåŻŁïijŃä¡ääŘŕäżěåIJÍåőČäÿŁéÍćæŤ¿äÿĂäÿłè¡żéĞŘçžğçŽĎåŃĚèčĚåŹÍïijŃçĎűåŘŐåřĘåőČ

17.5 15.5 äżŐæĽľåśŢæÍąåİŮäÿ■åőŽäźĽåŠŇåŕijåĞžCçŽĎAPI

éŮőéćŸ

ä¡ăæIJĽäÿĂäÿłCæĽľåśŢæÍąåİŮïijŇåIJÍåĘĚéČÍåőŽäźĽäžĘå¿ĹåďŽæIJĽçŤÍçŽĎåĞ¡æŢřïijŇä¡ăæČşåřĘå
APIä¿ŻåĚűäżŮåIJřæŰźä¡£çŤÍãĂĆ ä¡ăæČşåIJÍåĚűäżŮæĽľåśŢæÍąåİŮäÿ■ä¡£çŤÍè£ŹäžŻåĞ¡æŢřïijŇä¡ĘæŸŕäÿ
åźűäÿŤéĂŽè£ĞCçijŰèŕŚåŹÍ/éŞ¿æŐěåŹÍæİěåĄŽçIJŃäÿŁåŐżçĽźåĹńåď■æİĆïijĹæĹŰèĂĚäÿ■åŘŕèČ¡åĄŽåĹř

èğčåĘşæŰźæąĹ

æIJňèŁĆäÿżèęĄéŮőéćŸæŸŕåęĆä¡Ţåď ĎçŘĘ15.4åřŘèŁĆäÿ■æŘŘåĹřçŽĎPointåŕźèśąãĂĆäżŤçżĘåŻđäÿ

```
/* Destructor function for points */
static void del_Point(PyObject *obj) {

  free(PyCapsule_GetPointer(obj,"Point"));
}

/* Utility functions */
static Point *PyPoint_AsPoint(PyObject *obj) {
  return (Point *) PyCapsule_GetPointer(obj, "Point");
}

static PyObject *PyPoint_FromPoint(Point *p, int must_free) {
  return PyCapsule_New(p, "Point", must_free ? del_Point : NULL);
}
```

çŐřåIJÍçŽĎéŮőéćŸæŸŕæĂŐæăůåřĘ `PyPoint_AsPoint()` åŠŇ `Point_FromPoint()` åĞ¡æŢřä¡IJäÿžAPIåŕijåĞžïijŇ è£ŹæăůåĚűäżŮæĽľåśŢæÍąåİŮèČ¡ä¡£çŤÍåźűéŞ¿æŐěåőČäżňïijŇæŕŤåęĆåęĆæđIJä¡ăæIJĽåĚűäżŮæĽľåśŢäź§

èęĄèğčåĘşè£ŹäÿłéŮőéćŸïijŇéęŮåĚÍèęĄäÿž `sample` æĽľåśŢåĘŹäÿłæŰřçŽĎåď ť æŰĞäżűåŘ■åŘń `pysample.h` ïijŇåęĆäÿŃïijŽ

```
/* pysample.h */
#include "Python.h"
#include "sample.h"
#ifdef __cplusplus
extern "C" {
#endif

/* Public API Table */
typedef struct {
  Point *(*aspoint)(PyObject *);
  PyObject *(*frompoint)(Point *, int);
} _PointAPIMethods;

#ifndef PYSAMPLE_MODULE
/* Method table in external module */
```

(continues on next page)

(continued from previous page)

```
static _PointAPIMethods *_point_api = 0;

/* Import the API table from sample */
static int import_sample(void) {
  _point_api = (_PointAPIMethods *) PyCapsule_Import("sample._point_
↪api",0);
  return (_point_api != NULL) ? 1 : 0;
}

/* Macros to implement the programming interface */
#define PyPoint_AsPoint(obj) (_point_api->aspoint)(obj)
#define PyPoint_FromPoint(obj) (_point_api->frompoint)(obj)
#endif

#ifdef __cplusplus
}
#endif
```

è£ŹéĞŇæIJĂéĞ■èęĄçŽĎéČĺåĹĘæŸŕåĞ¡æŢřæŇĞéŠĹèąĺ `_PointAPIMethods` . åőČäijŽåIJĺåŕijåĞžæĺąåİŮæŮűèćńåĹİåğŃåŇŰïijŇçĎűåŘŐåŕijåĚěæĺąåİŮæŮűèćńæ§ěæL'¿åĹřãĂĆä£őæŤźåŐ§åğŃçŽĎæL'l'åśŢæĺąåİŮæİěåąńåĚĚèąĺæăijåźűåřĘåőČåČŔäÿŃéİćè£ŹæăůåŕijåĞžïijŽ

```
/* pysample.c */

#include "Python.h"
#define PYSAMPLE_MODULE
#include "pysample.h"

...
/* Destructor function for points */
static void del_Point(PyObject *obj) {
  printf("Deleting point\n");
  free(PyCapsule_GetPointer(obj,"Point"));
}

/* Utility functions */
static Point *PyPoint_AsPoint(PyObject *obj) {
  return (Point *) PyCapsule_GetPointer(obj, "Point");
}

static PyObject *PyPoint_FromPoint(Point *p, int free) {
  return PyCapsule_New(p, "Point", free ? del_Point : NULL);
}

static _PointAPIMethods _point_api = {
  PyPoint_AsPoint,
  PyPoint_FromPoint
};
...
```

(continues on next page)

(continued from previous page)

```
/* Module initialization function */
PyMODINIT_FUNC
PyInit_sample(void) {
  PyObject *m;
  PyObject *py_point_api;

  m = PyModule_Create(&samplemodule);
  if (m == NULL)
    return NULL;

  /* Add the Point C API functions */
  py_point_api = PyCapsule_New((void *) &_point_api, "sample._point_
↪api", NULL);
  if (py_point_api) {
    PyModule_AddObject(m, "_point_api", py_point_api);
  }
  return m;
}
```

æIJĂåŘŐïijŇäÿŃéİćæŸŕäÿĂäÿłæŰřçŽĎæĽľåśŢæĺąåİŮä¿Ńå■ŘïijŇçŤĺæİěåŁăè¡¡åźűä¡£çŤĺè£ŹäžŻAPIå

```
/* ptexample.c */

/* Include the header associated with the other module */
#include "pysample.h"

/* An extension function that uses the exported API */
static PyObject *print_point(PyObject *self, PyObject *args) {
  PyObject *obj;
  Point *p;
  if (!PyArg_ParseTuple(args,"O", &obj)) {
    return NULL;
  }

  /* Note: This is defined in a different module */
  p = PyPoint_AsPoint(obj);
  if (!p) {
    return NULL;
  }
  printf("%f %f\n", p->x, p->y);
  return Py_BuildValue("");
}

static PyMethodDef PtExampleMethods[] = {
  {"print_point", print_point, METH_VARARGS, "output a point"},
  { NULL, NULL, 0, NULL}
};
```

(continues on next page)

(continued from previous page)

```
static struct PyModuleDef ptexamplemodule = {
  PyModuleDef_HEAD_INIT,
  "ptexample",           /* name of module */
  "A module that imports an API",  /* Doc string (may be NULL) */
  -1,                 /* Size of per-interpreter state or -1 */
  PtExampleMethods       /* Method table */
};

/* Module initialization function */
PyMODINIT_FUNC
PyInit_ptexample(void) {
  PyObject *m;

  m = PyModule_Create(&ptexamplemodule);
  if (m == NULL)
    return NULL;

  /* Import sample, loading its API functions */
  if (!import_sample()) {
    return NULL;
  }

  return m;
}
```

çijŰèŕŚè£ŹäÿłæŰřæÍąåİŮæŮűïijŇä¡ăçŤŽèĞşäÿ■éIJĂèęĄåŐżèĂČèŹŚæĂŐæăůåřĘåĞ¡æŢřåžŞæĹŰäżčçä ä¿ŃåęĆïijŇä¡ăåŔŕäżěåČŔäÿŃéİćè£ŹæăůåĹŻåżžäÿĂäÿłçőĂå■ŢçŽĎ `setup.py` æŰĞäżűïijŽ

```
# setup.py
from distutils.core import setup, Extension

setup(name='ptexample',
      ext_modules=[
        Extension('ptexample',
                  ['ptexample.c'],
                  include_dirs = [],  # May need pysample.h␣
↪directory
                  )
        ]
)
```

åęĆæđIJäÿĂåĹĞæ■čåÿÿïijŇä¡ääijŽåŔŚçŐřä¡ăçŽĎæŰřæL'l'åśŢåĞ¡æŢřèČ¡åŠŇåőŽäźL'åIJÍåĚűäżŰæÍąåİ APIåĞ¡æŢřäÿĂèţůè£ŘèąŇçŽĎå¿Ĺåě¡ãĂĆ

```
>>> import sample
>>> p1 = sample.Point(2,3)
>>> p1
<capsule object "Point *" at 0x1004ea330>
>>> import ptexample
```

(continues on next page)

(continued from previous page)

```
>>> ptexample.print_point(p1)
2.000000 3.000000
>>>
```

èőÍèőž

æIJňèŁĆå§žäžŐäÿĂäÿłåĽ■æŘŘåŕśæŸŕŕijŇèČűåŻŁåŕźèśąèČ¡èŐůåŘŰäżżä¡Ţä¡äæČşèęĄçŽĎåŕźèśąçŽĎ
è£Źæăůç ŽĎèŕÍŕijŇåőŽäźĽæĺąåÍŮäijŽåąńåĚĚäÿĂäÿłåĞ¡æŢŕæŇĞéŠĹçŽĎçżŞæđĎä¡ŞŕijŇåĹŽåżžäÿĂäÿłæŇĞ
ä¿ŃåęĆ `sample._point_api`.

åĚűäżŰæĺąåÍŮèČ¡åď§åIJÍåŕijåĚěæŮűèŐůåŘŰåĹřè£ŹäÿłåśđæĂğåźűæŘŘåŘŰåžŢåśĆçŽĎæŇĞéŠĹãĂĆ
äžŃåőđäÿŁŕijŇPythonæŘŘä¿ŻäžĘ `PyCapsule_Import()`
åůěåĚůåĞ¡æŢŕŕijŇäÿžäžĘåőŇæĹŘæĽĂæIJĽçŽĎæ■ěéłďãĂĆ
ä¡ääŘłéIJĂæŘŘä¿ŻåśđæĂğçŽĎåŘ■å■Ůå■şåŘŕŕijĹæŕŤåęĆsample._point_apiŕijĽŕijŇçĎűåŘŐäżŰåŕśäijŽäÿĂä

åIJÍåŕĘèćńåŕijåĞžåĞ¡æŢŕåŘŸäÿžåĚűäżŰæĺąåÍŮäÿ■æŹőéĂŽåĞ¡æŢŕæŮűŕijŇæIJĽäÿĂäžŻCçijŰçÍŃéŹů
åIJÍ `pysample.h` æŰĞäżűäÿ■ŕijŇäÿĂäÿł `_point_api`
æŇĞéŠĹèćńçŤÍæÍěæŇĞåŘŚåIJÍåŕijåĞžæĺąåÍŮäÿ■èćńåĹÍåğŃåŇŰçŽĎæŰźæşŢèąĺãĂĆ
äÿĂäÿłçŻÿåĚşçŽĎåĞ¡æŢŕ `import_sample()` èćńçŤÍæÍěæŇĞåŘŚèČűåŻŁåŕijåĚěåźűåĹÍåğŃåŇŰè£Źäÿłæ
è£ŹäÿłåĞ¡æŢŕå£Ěéążåĺäżżä¡ŢåĞ¡æŢŕèćńä¡£çŤÍäźŃåĽ■èćńèřČçŤÍãĂĆéĂŽåÿÿæÍěèőšŕijŇáőČäijŽåIJÍæĺąåÍ
æIJĂåŘŐŕijŇCçŽĎéćĎďĎçŘĘáőŘèćńáőŽäźĽŕijŇèćńçŤÍæÍěéĂŽè£ĞæŰźæşŢèąĺåŐżåĹĘåŘŚè£ŹäżŻAPIåĞ
çŤÍæĹůåŘłéIJĂèęĄä¡£çŤÍè£ŹäżŽåŐ§åğŃåĞ¡æŢŕåŘ■çğŕå■şåŘŕŕijŇäÿ■éIJĂèęĄéĂŽè£ĞáőŘåŐżäżĘèğčåĚűä

æIJĂåŘŐŕijŇè£ŸæIJĽäÿĂäÿłéĞ■èęĄçŽĎåŐ§åŻäèőľä¡ääŐżä¡£çŤÍè£ŹäÿłæŁĂæIJŕæÍěéŞ¿æŐěæĺąåÍŮâ
åęĆæđIJä¡ääÿ■æČşä¡£çŤÍæIJňæIJżçŽĎæŁĂæIJŕŕijŇéĆčä¡ääŕśå£Ěéążä¡£çŤÍåĚśäžňåžŞçŽĎéńŸçžğçĽźæĂğå
ä¿ŃåęĆŕijŇåŕĘäÿĂäÿłæŹőéĂŽçŽĎAPIåĞ¡æŢŕæŤ¿åĚěäÿĂäÿłåĚśäžňåžŞåźűçąőä£ÍæĽĂæIJĽæĽľäśŢæĺąåÍŮ
è£Źçğ■æŰźæşŢçąőäőđåŘŕèąŇŕijŇä¡ĘæŸŕäőČçŻÿäŕźçźĄçŘŘŕijŇçĽźäĹńæŸŕäIJÍäď ğäđŃçşżçż§äÿ■ãĂĆ
æIJňèŁĆæijŤçď žäžĘäęĆä¡ŢéĂŽè£ĞPythonçŽĎæŹőéĂŽåŕijåĚěæIJžäĹůåŠŇäżĚäżĚäĞääÿłèČűåŻŁèřČçŤÍæĺ
äŕźäžŐæĺąåÍŮçŽĎçijŰèŕŚŕijŇä¡ääŘłéIJĂèęĄäőŽäźĽäď ť æŰĞäżűŕijŇèĂŇäÿ■éIJĂèęĄèĂČèŹŚäĞ¡æŢŕäžŞçŽ

æŻť äď ŽäĚşäžŐäĹľçŤÍC APIæÍěæđĎéĂäæĽľäśŢæĺąåÍŮçŽĎä£ąæĄŕäŘŕäżěäŘĆèĂČ
PythonçŽĎæŰĞæąč

17.6 15.6 äżŐCèŕ■èÍĂäÿ■èřČçŤÍPythonäżčçăĄ

éŮőéćŸ

ä¡äæČşäIJÍCäÿ■äőĽäĚÍçŽĎæĽğèąŇæ§ŘäÿłPythonèřČçŤÍäźűè£ŤäŻđçżŞæđIJçżŹCãĂĆ
ä¿ŃåęĆŕijŇä¡äæČşäIJÍCèŕ■èÍĂäÿ■ä¡£çŤÍæ§ŘäÿłPythonäĞ¡æŢŕä¡IJäÿžäÿĂäÿłäŻđèřČãĂĆ

èğčäĘşæŰźæąĹ

äIJÍCèŕ■èÍĂäÿ■èřČçŤÍPythonéÍđäÿÿçőĂä■ŢŕijŇäÿ■è£ĞèŐ¿èőąäĹřäÿĂäžŻäřŘçł■éŮÍãĂĆ
äÿŃéÍćçŽĎCäżčçăĄäŚŁèřĽä¡äæĂŐæăůäőĽäĚÍçŽĎèřČçŤÍŕijŽ

```
#include <Python.h>

/* Execute func(x,y) in the Python interpreter.  The
   arguments and return result of the function must
   be Python floats */

double call_func(PyObject *func, double x, double y) {
  PyObject *args;
  PyObject *kwargs;
  PyObject *result = 0;
  double retval;

  /* Make sure we own the GIL */
  PyGILState_STATE state = PyGILState_Ensure();

  /* Verify that func is a proper callable */
  if (!PyCallable_Check(func)) {
    fprintf(stderr,"call_func: expected a callable\n");
    goto fail;
  }
  /* Build arguments */
  args = Py_BuildValue("(dd)", x, y);
  kwargs = NULL;

  /* Call the function */
  result = PyObject_Call(func, args, kwargs);
  Py_DECREF(args);
  Py_XDECREF(kwargs);

  /* Check for Python exceptions (if any) */
  if (PyErr_Occurred()) {
    PyErr_Print();
    goto fail;
  }

  /* Verify the result is a float object */
  if (!PyFloat_Check(result)) {
    fprintf(stderr,"call_func: callable didn't return a float\n");
    goto fail;
  }

  /* Create the return value */
  retval = PyFloat_AsDouble(result);
  Py_DECREF(result);

  /* Restore previous GIL state and return */
  PyGILState_Release(state);
  return retval;

fail:
```

(continues on next page)

(continued from previous page)

```
  Py_XDECREF(result);
  PyGILState_Release(state);
  abort();   // Change to something more appropriate
}
```

èęĄä¡£çŤĺè£ŹäÿłåĞ¡æŢřïijŇä¡ăéIJĂèęĄèŐůåŘŰäijăéĂŠè£ĞæİěçŽĎæ§Řäÿłåůšå■ŸåIJÍPythonèřČçŤÍçŽ
æIJL'å¿Ĺåď'Žçğ■æŰźæşŢåŘŕäżěèőľä¡ăè£ŹæăůåĄŽïijŇ æŕŤåęĆåřĘäÿĂäÿłåŘŕèřČçŤÍåŕźèśąäijăçżŹäÿĂäÿłæL'

äÿŃéİćæŸŕäÿĂäÿłçőĂå■Ţä¿Ňå■ŘçŤÍæİěæŐľéěřäżŐäÿĂäÿłåţŇåĚěçŽĎPythonèğčéĞŁåŹÍäÿ■èřČçŤÍäÿ

```
#include <Python.h>

/* Definition of call_func() same as above */
...

/* Load a symbol from a module */
PyObject *import_name(const char *modname, const char *symbol) {
  PyObject *u_name, *module;
  u_name = PyUnicode_FromString(modname);
  module = PyImport_Import(u_name);
  Py_DECREF(u_name);
  return PyObject_GetAttrString(module, symbol);
}

/* Simple embedding example */
int main() {
  PyObject *pow_func;
  double x;

  Py_Initialize();
  /* Get a reference to the math.pow function */
  pow_func = import_name("math","pow");

  /* Call it using our call_func() code */
  for (x = 0.0; x < 10.0; x += 0.1) {
    printf("%0.2f %0.2f\n", x, call_func(pow_func,x,2.0));
  }
  /* Done */
  Py_DECREF(pow_func);
  Py_Finalize();
  return 0;
}
```

èęĄæđĎåżżä¿Ňå■ŘäżčçăĄïijŇä¡ăéIJĂèęĄçijŰèŕŚCåźűåřĘåőČéŞ¿æŐěåĹřPythonèğčéĞŁåŹÍãĂĆ
äÿŃéİćçŽĎMakefileåŘŕäżěæŢŹä¡ăæĂŐæăůåĄŽïijĹäÿ■è£ĞåIJÍä¡ăæIJżåŹÍäÿŁéİćéIJĂèęĄäÿĂäżŽéĚ■ç¡őïijL'

```
all::
        cc -g embed.c -I/usr/local/include/python3.3m \
          -L/usr/local/lib/python3.3/config-3.3m -lpython3.3m
```

çijŰèŕŚåźűè£ŘèąŇäijŽäžğçŤ§çśżäijijäÿŃéİćçŽĎè¿ŞåĞžïijŽ

```
0.00 0.00
0.10 0.01
0.20 0.04
0.30 0.09
0.40 0.16
...
```

äÿŃéİćæŸŕäÿĂäÿłçÍ■å¿őäÿ■åŘŇçŽĎä¿Ńå■ŘïijŇåśŢçď žäžĘäÿĂäÿłæĽľåśŢåĞ¡æŢřïijŇ
åőČæŐěåŘŮäÿĂäÿłåŕŕèřČçŤĺåŕźèśąåŠŇåĚűäżŰåŘĆæŢřïijŇåźűåřĘåőČäžňäijäéĂŠçżŹ
`call_func()` æİěåĄŽæţŃèŕŢïijŽ

```
/* Extension function for testing the C-Python callback */
PyObject *py_call_func(PyObject *self, PyObject *args) {
  PyObject *func;

  double x, y, result;
  if (!PyArg_ParseTuple(args,"Odd", &func,&x,&y)) {
    return NULL;
  }
  result = call_func(func, x, y);
  return Py_BuildValue("d", result);
}
```

ä¡£çŤĺè£ŹäÿłæĽľåśŢåĞ¡æŢřïijŇä¡ăèęĄåČŕäÿŃéİćè£ŹæăůæţŃèŕŢőČïijŽ

```
>>> import sample
>>> def add(x,y):
...     return x+y
...
>>> sample.call_func(add,3,4)
7.0
>>>
```

èőĺèőž

åęĆæđIJä¡ăåIJĺCèŕ■èĺĂäÿ■èřČçŤĺPythonïijŇèęĄèőŕä¡ŕæIJĂéĞ■èęĄçŽĎæŸŕCèŕ■èĺĂäijŽæŸŕäÿżä¡ŞãĂ
äź§åŕśæŸŕèŕť ïijŇCèŕ■èĺĂèť §èť čæđĎéĂäåŕĆæŢřãĂĄèřČçŤĺPythonåĞ¡æŢřãĂĄæčĂæ§ěåijĆåÿÿãĂĄæčĂæ

ä¡IJäÿžçňňäÿĂæ■ěïijŇä¡ăå£ĚéążåĚĺæIJĽäÿĂäÿłèąĺçď žä¡ăåřĘèęĄèřČçŤĺçŽĎPythonåŕŕèřČçŤĺåŕźèśąãĂ
è£ŹåŕäżěæŸŕäÿĂäÿłåĞ¡æŢřãĂĄçśżãĂĄæŰźæşŢãĂĄåĘĚç¡őæŰźæşŢæĹŰåĚűäżŰäżżæĎŕåőđçŐřäžĘ
`__call__()` æŞ■ä¡IJçŽĎäÿIJèě£ãĂĆ äÿžäžĘçąőä£ĺæŸŕåŕŕèřČçŤĺçŽĎïijŇåŕäżěåČŕäÿŃéİćçŽĎäžčçăĄè
`PyCallable_Check()` åĄŽæčĂæ§ěïijŽ

```
double call_func(PyObject *func, double x, double y) {
  ...
  /* Verify that func is a proper callable */
  if (!PyCallable_Check(func)) {
```

(continues on next page)

(continued from previous page)

```
    fprintf(stderr,"call_func: expected a callable\n");
    goto fail;
  }
  ...
```

åIJÍCäžčçăĄéĞŇåď ĎçŘĘéŤŹèŕŕä¡ăéIJĂèęĄæăijåď ŰçŽĎåřŘå£ČãĂĆäÿĂèĹňæĺěèőšïijŇä¡ăäÿ■èČ¡äżĚä
éŤŹèŕŕåžŤèŕěä¡£çŤÍCäžčçăĄæŰźåijŘæĺěèćńåď ĎçŘĘãĂĆåIJÍè£ŹéĞŇïijŇæĹŚäžňæL'ŞçőŮåřĘåŕźéŤŹèŕŕçŽĎ
abort() çŽĎéŤŹèŕŕåď ĎçŘĘåŹÍãĂĆ åőČäijŽçżŞæĺ§æŐL'æŢť äÿłçÍŃåžŘïijŇåIJÍçIJ§åđçŐŕåćČäÿŃéĺćä¡ăä
ä¡ăèęĄèőŕä¡ŘçŽĎæŸŕåIJÍè£ŹéĞŇCæŸŕäÿżèğŠïijŇåŻăæ■ď åźűæšąæIJL'èů§æŁŻåĞžåijĆåÿÿçŻÿåŕźåžŤçŽĎæ
éŤŹèŕŕåď ĎçŘĘæŸŕä¡ăåIJÍçijŰçÍŃæŮűå£ĚéążèęĄèĂČèŹŚçŽĎäžŃæČĚãĂĆ

èřČçŤÍäÿĂäÿłåĞ¡æŢřçŻÿåŕźæĺěèőšå¿ĹçőĂå■ŢâĂŤâĂŤåŘłéIJĂèęĄä¡£çŤÍ
PyObject_Call() ïijŇ äijăäÿĂäÿłåŘŕèřČçŤÍåŕźèśąçżŹåőČãĂĄäÿĂäÿłåŘĆæŢřåĚČçżĎåŠŇäÿĂäÿłåŘŕéĂ
èęĄæđĎåżžåŘĆæŢřåĚČçżĎæĹŰå■ŮåĚÿïijŇä¡ăåŘŕäżěä¡£çŤÍ Py_BuildValue()
,åęĆäÿŃïijŽ

```
double call_func(PyObject *func, double x, double y) {
  PyObject *args;
  PyObject *kwargs;

  ...
  /* Build arguments */
  args = Py_BuildValue("(dd)", x, y);
  kwargs = NULL;

  /* Call the function */
  result = PyObject_Call(func, args, kwargs);
  Py_DECREF(args);
  Py_XDECREF(kwargs);
  ...
```

åęĆæđIJæšąæIJL'åĚşéŤőå■ŮåŘĆæŢřïijŇä¡ăåŘŕäżěäijăéĂŠNULLãĂĆå¡Şä¡ăèęĄèřČçŤÍåĞ¡æŢřæŮűïijŇ
éIJĂèęĄçąőä£Íä¡£çŤÍäžĘ Py_DECREF() æĹŰèĂĚ Py_XDECREF() æÿĚçŘĘåŘĆæŢřãĂĆ
çňňäžŇäÿłåĞ¡æŢřçŻÿåŕźåőL'åĚÍçĆźïijŇåŻääÿžåőČåĚĄèőÿäijăéĂŠNULLæŇĞéŠĹïijĹçŻť'æŐěå£¡çŢěåőČïijĺ
è£Źäź§æŸŕäÿžäżĂäźĹæĹŚäżňä¡£çŤÍåőČæĺěæÿĚçŘĘåŘŕéĂL'çŽĎåĚşéŤőå■ŮåŘĆæŢřãĂĆ

èřČçŤÍäÿĞPythonåĞ¡æŢřäźŃåŘŐïijŇä¡ăå£ĚéążæčĂæ§ěæŸŕåŘęæIJL'åijĆåÿÿåŘŚçŤ§ãĂĆ
PyErr_Occurred() åĞ¡æŢřåŘŕèćńçŤÍæĺěåĄŽè£ŹäżűäžŇãĂĆ
åŕźåŕźäžŐåijĆåÿÿçŽĎåď ĎçŘĘåŕśæIJL'çĆźéżżçČęäžĘïijŇçŤśäžŐæŸŕçŤÍCèŕ■èÍĂåĘźçŽĎïijŇä¡ăæšąæIJL'åĆ
åŻăæ■ď'ïijŇä¡ăå£ĔéążèęĄèő¿ç¡őäÿĂäÿłåijĆåÿÿçŁűæĂĄçĂąïijŇæL'Ş å■řåijĆåÿÿä£ąæĄŕæĹŰåĚűäżŰçŻÿåžŤ
åIJÍè£ŹéĞŇïijŇæĹŚäżňéĂL'æŃl'äžĘçőĂå■ŢçŽĎ abort()
æĺěåď ĎçŘĘãĂĆåŘęåď ŰïijŇäijăçż§CçÍŃåžŘåŚŸåŘŕèČ¡äijŽçŻť'æŐěěől'çÍŃåžŘåěŤæžČãĂĆ

```
...
/* Check for Python exceptions (if any) */
if (PyErr_Occurred()) {
  PyErr_Print();
  goto fail;
}
```

(continues on next page)

(continued from previous page)

```
...
fail:
  PyGILState_Release(state);
  abort();
```

äżŐèřČçŤÍPythonåĞ¡æŢřçŽĎè£ŤåŻđåĂijäÿ■æŘŘåŘŰä£ąæĄŕéĂŽåÿÿèęĄè£ŻèąŇçśżåđŃæčĂæ§ěåŠŇæ
èęĄè£ŹæŭůåĄŽçŽĎèŕÍïijŇä¡ăå£Ěéążäÿ¡£çŤÍPythonåŕźèśąåśĆäÿ■çŽĎåĞ¡æŢřãĂĆ
åIJÍè£ŹéĞŇæĹŚäżňä¡£çŤÍäžĘ `PyFloat_Check()` åŠŇ `PyFloat_AsDouble()`
æÍěæčĂæ§ěåŠŇæŘŘåŘŰPythonæţőçĆźæŢřãĂĆ

æIJĂåŘŐäÿĂäÿłéŮőéćŸæŸŕåŕźäžŐPythonåĚÍåśĂéŤĄçŽĎçőąçŘĘãĂĆ
åIJÍCèŕ■èÍĂäÿ■èő£éŮőPythonçŽĎæŮűůåĂŹïijŇä¡ăéIJĂèęĄçąőä£ÍGILèćńæ■čçąőçŽĎèŐůåŘŰůåŠŇéĞŁæŤ¿äž
äÿ■çĎűçŽĎèŕÍïijŇåŘŕèČ¡äijŽåŕijèĞť èğčéĞŁåŹÍè£ŤåŻđéŤŹèŕŕæŢřæ■őæĹŰèĂĚçŻť æŐěåěŤæžČãĂĆ
èřČçŤÍ `PyGILState_Ensure()` åŠŇ `PyGILState_Release()`
åŘŕäżěçąőä£ÍäÿĂåĹĞéČ¡èČ¡æ■čåÿÿãĂĆ

```
double call_func(PyObject *func, double x, double y) {
  ...
  double retval;

  /* Make sure we own the GIL */
  PyGILState_STATE state = PyGILState_Ensure();
  ...
  /* Code that uses Python C API functions */
  ...
  /* Restore previous GIL state and return */
  PyGILState_Release(state);
  return retval;

fail:
  PyGILState_Release(state);
  abort();
}
```

äÿĂæŮęè£ŤåŻđïijŇPyGILState_Ensure() åŘŕäżěçąőä£ÍèřČçŤÍçž£çÍŃçŃňå■ăPythonèğčéĞŁåŹÍã
åŕśçőŮCäżčçăĄè£ŘèąŇäžŐåŘęåď ŰäÿĂäÿłèğčéĞŁåŹÍäÿ■ç§ěéĄŞçŽĎçž£çÍŃäź§æšąäžŃãĂĆ
è£ŹæŮůåĂŹïijŇCäżčçăĄåŘŕäżěèĞłçŤśçŽĎä¡£çŤÍäżżä¡ŢåőČæČşèęĄçŽĎPython C-API
åĞ¡æŢřãĂĆ èřČçŤÍæĹŘåŁ§åŘŐïijŇPyGILState_Release()èćńçŤÍæÍěèőšèğčéĞŁåŹÍæĄćåď ■åĹřåŐ§åğŃçŁűa

èęĄæşÍæĎŘçŽĎæŸŕæŕŘäÿĂäÿł `PyGILState_Ensure()`
èřČçŤÍå£Ěéążèů§çÍĂäÿĂäÿłåŃźéĚ■çŽĎ `PyGILState_Release()`
èřČçŤÍâĂŤâĂŤå■şä¿£æIJL'éŤŹèŕŕåŘŚçŤ§ãĂĆ åIJÍè£ŹéĞŃïijŇæĹŚäżňä¡£çŤÍäÿĂäÿł `goto`
èŕ■åŘěçIJŃäÿŁåŐżæŸŕäÿłåŘŕæĂŢçŽĎèő¿èőąïijŇ ä¡ĘæŸŕåőđéŹĚäÿŁæĹŚäżňä¡£çŤÍåőČæÍěèőšæŐğåĹűæÍČè
åIJÍ `fail:` æăĞç■¿åŘŐéÍćçŽĎäżčçăĄåŠŇPythonçŽĎ `fianl:`
åÍŮçŽĎçŤÍéĂŤæŸŕäÿĂæăůçŽĎãĂĆ

åęĆæđIJä¡ăä¡£çŤÍæL'ĂæIJL'è£ŹäžŻçžęåőŽæÍěçijŰåĘŹCäżčçăĄïijŇåŇĚæŃňåŕźGILçŽĎçőąçŘĘãĂĄåij
ä¡ăäijŽåŘŚçŐŕäżŐCèŕ■èÍĂäÿ■èřČçŤÍPythonèğčéĞŁåŹÍæŸŕåŘŕéÍăçŽĎâĂŤâĂŤåŕśçőŮåĘ■åď ■æÍĆçŽĎçÍŃåž

17.7 15.7 äżŐCæĽľåśŢäÿ■éĞŁæŤ¿åĚĺåśĂéŤĄ

éŮőéćŸ

ä¡ăæČşèőľCæĽľåśŢäżčçăĄåŠŇPythonèğčéĞŁåŹĺäÿ■çŽĎåĚűäżŰè£ŻçĺŃäÿĂèţůæ■čçąőçŽĎæĽğèąŇïij
éĆčäźĹä¡ăåřśéIJĂèęĄåŐżéĞŁæŤ¿åźűéĞ■æŰřèŐůåŘŮåĚĺåśĂèğčéĞŁåŹĺéŤĄïijĹGILïijĽãĂĆ

èğčåĘşæŰźæąĹ

åIJĺCæĽľåśŢäżčçăĄäÿ■ïijŇGILåŘŕäżěéĂŽè£ĞåIJĺäżčçăĄäÿ■æŘŠåĚěäÿŃéĺćè£ŹæăůçŽĎåőŘæĺěéĞŁæ

```
#include "Python.h"
...

PyObject *pyfunc(PyObject *self, PyObject *args) {
   ...
   Py_BEGIN_ALLOW_THREADS
   // Threaded C code.  Must not use Python API functions
   ...
   Py_END_ALLOW_THREADS
   ...
   return result;
}
```

èőĺèőž

åŘłæIJĽå¡Şä¡äçąőä£ĺæšąæIJĽPython C APIåĞ¡æŢřåIJĺCäÿ■æĽğèąŇçŽĎæŮűåĂŹä¡ăæĽ■èČ¡åőĽåĚĺçŽ
GILéIJĂèęĄèćńéĞŁæŤ¿çŽĎåÿÿèğĄçŽĎåIJžæŹŕæŸŕåIJĺèőąçőŮåŕĘéŻĘåđŃäżčçăĄäÿ■éIJĂèęĄåIJĺCæŢřçżĎ
æĹŰèĂĚæŸŕèęĄæĽğèąŇéŸżåąđçŽĎI/OæŞ■ä¡IJæŮűïijĹærŤåęĆåIJĺäÿĂäÿłæŰĞäżűæŘŘè£řçňęäÿŁèŕżåŘŰa

å¡ŞGILèćńéĞŁæŤ¿åŘŐïijŇåĚűäżŰPythonçž£çĺŃæĽ■èćńåĚĄèőÿåIJĺèğčéĞŁåŹĺäÿ■æĽğèąŇãĂĆ
`Py_END_ALLOW_THREADS` åőŘäijŽéŸżåąđæĽğèąŇçŻťåĹřèřČçŤĺçž£çĺŃéĞ■æŰřèŐůåŘŰäžĘGILãĂĆ

17.8 15.8 CåŠŇPythonäÿ■çŽĎçž£çĺŃæůůçŤĺ

éŮőéćŸ

ä¡ăæIJĽäÿĂäÿłçĺŃåžŘéIJĂèęĄæůůåŘĹä¡£çŤĺCãĂĄPythonåŠŇçž£çĺŃïijŇ
æIJĽäžŻçž£çĺŃæŸŕåIJĺCäÿ■åĹŻåżžçŽĎïijŇèűĚåĞžäžĘPythonèğčéĞŁåŹĺçŽĎæŐğåĹűèŇČåŻťãĂĆ
åźűäÿŤäÿĂäžŻçž£çĺŃè£Ÿä¡£çŤĺäžĘPython C APIäÿ■çŽĎåĞ¡æŢřãĂĆ

èğčåĘşæŰźæąĹ

åęĆæđIJä¡ăæČşåřĘCãĂĄPythonåŠŇçž£çĺŃæůůåŘĹåIJĺäÿĂèţůïijŇä¡ăéIJĂèęĄçąőçŽĎåĹĺåğŇ
èęĄæČşè£ŹæăůåĄŽïijŇåŘŕäżěåřĘäÿŃåĹŮäżčçăĄæŤ¿åĹřä¡ăçŽĎCäżčçăĄäÿ■åźűçąőä£ĺåőČåIJĺäżżä¡Ţçž£çĺŃ

```
#include <Python.h>
  ...
  if (!PyEval_ThreadsInitialized()) {
    PyEval_InitThreads();
  }
  ...
```

åŕźäžŐäżżä¡ŢèřČçŤÍPythonåŕźèśąæĹŰPython C APIçŽĎCäžčçăĄïijŇçąőä£Ìä¡äéęŰåĚĹåůšçżŕæ■čçąőåĺ
è£ŹåŘŕäžěçŤÍ `PyGILState_Ensure()` åŠŇ `PyGILState_Release()` æÌěåĄŽåĹřïijŇåęĆäÿŇæĽĂçď žïijŽ

```
...
/* Make sure we own the GIL */
PyGILState_STATE state = PyGILState_Ensure();

/* Use functions in the interpreter */
...
/* Restore previous GIL state and return */
PyGILState_Release(state);
...
```

æŕŘæňąèřČçŤÍ `PyGILState_Ensure()` éČ¡èęAçŻÿåžŤçŽĎèřČçŤÍ `PyGILState_Release()`.

èőÍèőž

åIJÍæűĽåŘŁåĹřCåŠŇPythonçŽĎéńŸçžğçÍŃåžŘäÿ■ïijŇå¿Ĺåď ŽäžŃæČĚäÿĂèţůåĄŽæŸŕå¿Ĺåÿÿèğ ĄçŽĎ
åŘŕèČ¡æŸŕåŕźCãĂĄPythonãĂĄCçž£çÍŃãĂĄPythonçž£çÍŃçŽĎæůůåŘĹä¡£çŤÍãĂĆ
åŘłèęĄä¡äçąőä£ÌèğčéĞŁåŹÍèćńæ■čçąőçŽĎåĹÌåğŃåŇŰïijŇåźűäÿŤæűĽåŘŁåĹřèğčéĞŁåŹÍçŽĎCäžčçăĄæĽğ

èęĄæşÍæĎŘçŽĎæŸŕèřČçŤÍ `PyGILState_Ensure()` åźűäÿ■äijŽçńŃåĹżæŁćå■äæĹŰäÿ■æŰ■èğčéĞŁåŹÍãĂĆ åęĆæđIJæIJĽåĚűäžŰäžčçăĄæ■čåIJÍæĽğèąŃïijŇè£Ź
åIJÍåĘĚéČÍïijŇèğčéĞŁåŹÍäijŽæĽğèąŃåŚÍæIJ§æĂğçŽĎçž£çÍŃåĹĞæ■ćïijŇåŽäæ■ď åęĆæđIJåĚűäžŰçž£çÍŃåĺ
èřČçŤÍèĂĚæIJĂçżĹè£ŸæŸŕåŕźžèè£ŘèąŃçŽĎïijĹåř¡çőąåŘŕèČ¡èęĄåĚĹç■L'äÿĂäijŽïijL'ãĂĆ

17.9 15.9 çŤÍSWIGåŇĚèčĚCäžčçăĄ

éŮőécŸ

ä¡äæČşèőĺ'ä¡äåĘŹçŽĎCäžčçăĄä¡IJäÿžäÿĂäÿłCæĽĺ'åśŢæÍąåÌŮæÌěèő£éŮőïijŇæČşéĂŽè£Ğä¡£çŤÍ SwigåŇĚèčĚçŤ§æĹŘåŹÍ æÌěåőŇæĹŘãĂĆ

èğčåĘşæŰźæąĹ

SwigéĂŽè£Ğèğčæđŕ Cåď ť'æŰĞäžűåźűèĞłåŁÍåĹŻåžžæĽĺ'åśŢäžčçăĄæÌěæŞ■ä¡IJãĂĆ
èęĄä¡£çŤÍåőČïijŇä¡äåĚĹèęĄæIJĽäÿĂäÿłCåď ť'æŰĞäžűãĂĆä¿ŃåęĆïijŇæĹŚäžňçď žä¿ŃçŽĎåď ť'æŰĞäžűåę

```
/* sample.h */

#include <math.h>
extern int gcd(int, int);
extern int in_mandel(double x0, double y0, int n);
extern int divide(int a, int b, int *remainder);
extern double avg(double *a, int n);

typedef struct Point {
    double x,y;
} Point;

extern double distance(Point *p1, Point *p2);
```

äÿĂæŮęä¡ăæIJL'äžĘè£Źäÿłåď'ťæŰĞäżűïijŇäÿŃäÿĂæ■ěářśæŸŕçijŰåĘŹäÿĂäÿłSwigâĂİæŐěåŘčâĂİæŰ
æŇL'çĚğçžęåőŽïijŇè£ŹäžŻæŰĞäżűäžěâĂİ.iâĂİåŘŐçijĂåźűäÿŤçśżäijijäÿŃéİćè£ŹæăůïijŽ

```
// sample.i - Swig interface
%module sample
%{
#include "sample.h"
%}

/* Customizations */
%extend Point {
    /* Constructor for Point objects */
    Point(double x, double y) {
        Point *p = (Point *) malloc(sizeof(Point));
        p->x = x;
        p->y = y;
        return p;
   };
};

/* Map int *remainder as an output argument */
%include typemaps.i
%apply int *OUTPUT { int * remainder };

/* Map the argument pattern (double *a, int n) to arrays */
%typemap(in) (double *a, int n)(Py_buffer view) {
  view.obj = NULL;
  if (PyObject_GetBuffer($input, &view, PyBUF_ANY_CONTIGUOUS |␣
↪PyBUF_FORMAT) == -1) {
    SWIG_fail;
  }
  if (strcmp(view.format,"d") != 0) {
    PyErr_SetString(PyExc_TypeError, "Expected an array of doubles
↪");
    SWIG_fail;
  }
```

(continues on next page)

(continued from previous page)

```
  $1 = (double *) view.buf;
  $2 = view.len / sizeof(double);
}

%typemap(freearg) (double *a, int n) {
  if (view$argnum.obj) {
    PyBuffer_Release(&view$argnum);
  }
}

/* C declarations to be included in the extension module */

extern int gcd(int, int);
extern int in_mandel(double x0, double y0, int n);
extern int divide(int a, int b, int *remainder);
extern double avg(double *a, int n);

typedef struct Point {
    double x,y;
} Point;

extern double distance(Point *p1, Point *p2);
```

äÿĂæŮęä¡ăăĘŹăě¡äžĘæŐěăŘčæŰĞäżűïijŇăřśăŘŕäžěăIJÍăŚ¡äżď'èąŇăůěăĚůäÿ■èřČçŤÍSwigäžĘïijŽ

```
bash % swig -python -py3 sample.i
bash %
```

swigçŽĎè¿ŞăĞžăřśæŸŕäÿď'äÿłæŰĞäżűïijŇsample_wrap.cäŠŇsample.pyãĂĆ
åŘŐéĺćçŽĎæŰĞäżűăřśæŸŕçŤÍæĹůéIJĂèęĄăŕijăĚěçŽĎãĂĆ èĂŇsam-
ple_wrap.cæŰĞäżűæŸŕéIJĂèęĄèćńçijŰèŕŚăĹřăŘ■ăŘń _sample
çŽĎæŤŕæŇĄæĺąăİŮçŽĎCäżčçăĄãĂĆ è£ŹäÿłăŘŕäżěéĂŽè£ĞèŭŞæŹőéĂŽæĽ'ľăśŢæĺąăİŮäÿĂæăůçŽĎæŁĂæ
ä¿ŃăęĆïijŇä¡ăăĹŻăżžäžĘäÿĂäÿłăęĆäÿŃæĽ'Ăçď'žçŽĎ `setup.py` æŰĞäżűïijŽ

```
# setup.py
from distutils.core import setup, Extension

setup(name='sample',
      py_modules=['sample.py'],
      ext_modules=[
        Extension('_sample',
                  ['sample_wrap.c'],
                  include_dirs = [],
                  define_macros = [],

                  undef_macros = [],
                  library_dirs = [],
                  libraries = ['sample']
                  )
```

(continues on next page)

(continued from previous page)

```
        ]
)
```

èęĄçijŰèŕŚåŠŇæţŃèŕŢïijŇåIJĺsetup.pyäÿŁæĽ'ğèąŇpython3ïijŇåęĆäÿŃïijŽ

```
bash % python3 setup.py build_ext --inplace
running build_ext
building '_sample' extension
gcc -fno-strict-aliasing -DNDEBUG -g -fwrapv -O3 -Wall -Wstrict-
↪prototypes
-I/usr/local/include/python3.3m -c sample_wrap.c
 -o build/temp.macosx-10.6-x86_64-3.3/sample_wrap.o
sample_wrap.c: In function âĂŸSWIG_InitializeModuleâĂŹ:
sample_wrap.c:3589: warning: statement with no effect
gcc -bundle -undefined dynamic_lookup build/temp.macosx-10.6-x86_64-
↪3.3/sample.o
 build/temp.macosx-10.6-x86_64-3.3/sample_wrap.o -o _sample.so -
↪lsample
bash %
```

åęĆæđIJäÿĂåĹĞæ■čåÿÿçŽĎèŕİïijŇä¡ăäijŽåŔŚçŐřä¡ăåřśåŔŕäżěå¿ĹæŰźä¿£çŽĎä¡£çŤĺçŤ§æĹŘçŽĎCæĽ

```
>>> import sample
>>> sample.gcd(42,8)
2
>>> sample.divide(42,8)
[5, 2]
>>> p1 = sample.Point(2,3)
>>> p2 = sample.Point(4,5)
>>> sample.distance(p1,p2)
2.8284271247461903
>>> p1.x
2.0
>>> p1.y
3.0
>>> import array
>>> a = array.array('d',[1,2,3])
>>> sample.avg(a)
2.0
>>>
```

èőĺèőž

SwigæŸŕPythonåŐĘåŔšäÿ■æđĎåżžæĽ'ĺåśŢæĺąåİŮçŽĎæIJĂåŔď'èĂĄçŽĎåůěåĚůäźŃäÿĂãĂĆ SwigèČ¡èĞłåŁĺåŇŰå¿Ĺåď'ŽåŇĚèčĚçŤ§æĹŘåŹĺçŽĎåď'ĎçŘĘãĂĆ

æĽ'ĂæIJĽ'SwigæŐěåŘčéČ¡äžěçśżäijijäÿŃéİćè£ŹæăůçŽĎäÿžåijĂåď't'ïijŽ

```
%module sample
%{
#include "sample.h"
%}
```

è£ŹäÿłäżĚäżĚåŔłæŸŕåčŕæŸŐäžĘæĽľåśŢæĺąåİŮçŽĎåŘ■çğřåźűæŇĞåőŽäžĘCåďťæŰĞäżűïijŇ
äÿžäžĘèČ¡èőľçijŰèŕŚéĂŽè£Ğå£Ěéążèę ĄåŇĚåŘńè£ŹäžŻåďťæŰĞäżűïijĹä¡■äžŐ %{ åŠŇ %}
çŽĎäżčçăĄïijĽïijŇ åřĘąőČäżňäźŇéŮťåď■åĹűçšŸèťťåĹřè¿ŞåĞžäżčçăĄäÿ■ïijŇè£Źäź§æŸŕä¡äèęĄæŤ¿ç¡őæl

SwigæŐěåŘčçŽĎåžŢäÿŇéČĺåĹĘæŸŕäÿĂäÿłCåčŕæŸŐåĹŮèąĺïijŇä¡äéIJĂèęĄåIJĺæĽľåśŢäÿ■åŇĚåŘńåő
è£ŹéĂŽåÿÿäżŐåďťæŰĞäżűäÿ■èćńåď■åĹűãĂĆåIJĺæĹŚäżňçŽĎä¿Ňå■Řäÿ■ïijŇæĹŚäżňäżĚäżĚåČŔäÿŇéİćè£

```
%module sample
%{
#include "sample.h"
%}
...
extern int gcd(int, int);
extern int in_mandel(double x0, double y0, int n);
extern int divide(int a, int b, int *remainder);
extern double avg(double *a, int n);

typedef struct Point {
    double x,y;
} Point;

extern double distance(Point *p1, Point *p2);
```

æIJĽäÿĂçĆźéIJĂèęĄåijżèřČçŽĎæŸŕè£ŹäžŻåčŕæŸŐäijŽåŚŁèŕĽSwigä¡äæČşèęĄåIJĺPythonæĺąåİŮäÿ■å
éĂŽåÿÿä¡äéIJĂèęĄcijŰè¿Śè£ŹäÿłäčŕæŸŐåĹŮèąĺæĹŰçŻÿäžŤçŽĎä£őæŤźäÿŇåőČäĂĆ
ä¿ŇåęĆïijŇåęĆæđIJä¡ääÿ■æČşæ§ŘäżŻäčŕæŸŐèćńåŇĚåŘńè£ŻæİěïijŇä¡äèęĄåřĘåőČäżŐäčŕæŸŐåĹŮèąĺäÿ■

ä¡£çŤĺSwigæIJĂåď■æİĆçŽĎåIJŕæŰźæŸŕåőČèČ¡çżŹCäżčçăĄæŘŘä¿Żåď ğéĞŘçŽĎèĞłäőŽäźĽæŞ■ä¡IJ
è£ŹäÿłäÿżéćŸåď łåď ğïijŇè£ŹéĞŇæŮäæşŢåśŢåijĂïijŇä¡ĘæŸŕæĹŚäżňåIJĺæIJňèŁĆè£ŸåĽľåśŢçď žäžĘäÿĂäż

çňňäÿĂäÿłèĞłäőŽäźĽæŸŕ %extend æŇĞäżď åĚĄèőÿæŰźæşŢèćńéŹĎåŁäåĹřåůšå■ŸåIJĺçŽĎçżŞæđĎä
æĹŚä¿Ňå■Řäÿ■ïijŇè£ŹäÿłèćńçŤĺæİěæůżåŁääÿĂäÿłPointçżŞæđĎä¡ŞçŽĎæđĎéĂäåŹĺæŰźæşŢãĂĆ
äőČäŘŕäżěèőľä¡äåČŔäÿŇéİćè£Źæäůä¡£çŤĺè£ŹäÿłçżŞæđĎä¡ŞïijŽ

```
>>> p1 = sample.Point(2,3)
>>>
```

äęĆæđIJçŢěè£ĞçŽĎèŕĺïijŇPointäŕźèśąäřśä£ĚéążäżěæŻťäŁääď■æİĆçŽĎæŰźäijŘæİěèćńäĹŻäżžïijŽ

```
>>> # Usage if %extend Point is omitted
>>> p1 = sample.Point()
>>> p1.x = 2.0
>>> p1.y = 3
```

çňňäžŇäÿłèĞłäőŽäźĽæűĽäŘŁäĹřäŕź `typemaps.i` äžŞçŽĎäijŢäĚěäŠŇ
`%apply` æŇĞäżď ïijŇ äőČäijŽæŇĞçď žSwigäŘĆæŢřç■¿äŘ■ `int *remainder`

èęĄèćńå¡ŞåĄŽæŸŕŕè¿ŞåĞžåĂijãĂĆ è£ŹäŷłåŏđéŹĚäŷŁæŸŕäŷĂäŷłæĺąåijŔåŃźéĚ■èğĎåĹŹãĂĆ
åIJĺæŐĕäŷŃæİĕçŽĎæĽĂæIJĽåčŕæŸŐäŷ■ïijŃäżżä¡ŢæŮűåĂŹåŘłèęĄçćŕäŷŁ int
*remainder ïijŃäżŰåŕśäijŽèćńä¡IJäŷžè¿ŞåĞžãĂĆ è£ŹäŷłèĞłåŏŽäźĽæŰźæşŢåŘŕäżěèőľ
divide() åĞ¡æŢŕè£ŤåŻđäŷd'äŷłåĂijãĂĆ

```
>>> sample.divide(42,8)
[5, 2]
>>>
```

æIJĂåŘŐäŷĂäŷłæűĽåŘŁåĹř %typemap æŃĞäżd'çŽĎèĞłåŏŽäźĽåŘŕèČ¡æŸŕè£ŹéĞŃåśŢçd'žçŽĎæIJĂ
äŷĂäŷłtypemapåřśæŸŕäŷĂäŷłåIJĺè¿ŞåĚěäŷ■çĽźåŏŽåŘĆæŢřæĺąåijŘçŽĎèğĎåĹŹãĂĆ
åIJĺæIJňèŁĆäŷ■ïijŃäŷĂäŷłtypemapèćńåŏŽäźĽäŷžåŃźéĚ■åŘĆæŢřæĺąåijŘ (double *a,
int n). åIJĺtypemapåĘĚéČĺæŸŕäŷĂäŷłCäżčçăĄçĽĞæŏţïijŃåŏČåŚŁèŕĽSwigæĂŐæăůåřĘäŷĂäŷłPythonåŕ
æIJňèŁĆäżčçăĄä¡£çŤĺäžĘPythonçŽĎçijŞå■Ŷå■ŔèőőåŏżåŃźéĚ■äżżä¡ŢçIJŃäŷŁåŐżçśżäijijåŘŃçš¿åžęæŢřçż
ïijĹæŕŤåęĆNumPyæŢřçżĎãĂĄarrayæĺąåİŮåĹŻåżžçŽĎæŢřçżĎç■L'ïijL'ïijŃæŻť'åď'ŽèŕůåŘĆèĂČ15.3åřŘèŁĆ

åIJĺtypemapäżčçăĄåĘĚéČĺïijŃ$1åŠŃ$2è£ŹæăůçŽĎåŘŸéĞŘæŻ£æ■ćäijŽèŐůåŘŰtypemapæĺąåijŘçŽĎC
ïijĹæŕŤåęĆ$1æŸäářĎäŷž double *a ïijL'ãĂĆ$inputæŃĞåŘŚäŷĂäŷłä¡IJäŷžè¿ŞåĚěçŽĎ
PyObject * åŘĆæŢřïijŃ èĂŃ $argnum åřśäżčèąĺåŘĆæŢřçŽĎäŷłæŢřãĂĆ

çijŰåĘŹåŠŃçŘĘèğčtypemapsæŸŕä¡£çŤĺSwigæIJĂå§žæIJňçŽĎåL'■æŘŘãĂĆ
äŷ■äżĚæŸŕèŕť'äżčçăĄæŻť'çěđçğŸïijŃèĂŃäŷŤä¡äéIJĂèęĄçŘĘèğčPython C
APIåŠŃSwigåŠŃåőČäžd'äžŠçŽĎæŰźåijŘãĂĆ SwigæŰĞæąčæIJL'æŻť'åď'Žè£ŹæŰźéİćçŽĎçżĘèŁĆïijŃåŘŕäż

äŷ■è£ĞïijŃåęĆæđIJä¡äæIJL'åď'ğéĞŘçŽĎCäżčçăĄéIJĂèęĄèćńæŻť'éIJšäŷžæL'l'åśŢæĺąåİŮãĂĆ
SwigæŸŕäŷĂäŷłéİđåŷŷåijžåď'ğçŽĎåůěåĚůãĂĆåĚşéŤőçĆźåIJĺäžŐSwigæŸŕäŷĂäŷłåď'ĎçŘĘCåčŕæŸŐçŽĎçij
éĂŽè£Ĝåijžåď'ğçŽĎæĺąåijŘåŃźéĚ■åŠŃèĞłåŏŽäźĽçżĎäżűïijŃåŘŕäżèèőľä¡äæŻť'æŤźåčŕæŸŐæŃĞåŏŽåŠŃçş
æŻť'åď'Žä£ąæĄŕèŕůåŐżæ§ěéŸĚ Swigç¡ŚçńŹ ïijŃ è£ŸæIJL'
çL'źåŏŽäžŐPythonçŽĎçŻŷåĚşæŰĞæąč

17.10 15.10 çŤĺCythonåŃĚèčĚCäżčçăĄ

éŮőécŸ

ä¡äæČşä¡£çŤĺCythonæİěåĹŻåżžäŷĂäŷłPythonæL'l'åśŢæĺąåİŮïijŃçŤĺæİěåŃĚèčĚæ§Řäŷłåůšå■ŸåIJĺçŽĎ

èğčåĘşæŰźæąĹ

ä¡£çŤĺCythonæđĎåżžäŷĂäŷłæL'l'åśŢæĺąåİŮçIJŃäŷŁåŐżå¿ĹæL'ŃåĘźæL'l'åśŢæIJL'äžŻçśżäijijïijŃ
åŻääŷżä¡äéIJĂèęĄåĹŻåżžå¿Ĺåď'ŽåŃĚèčĚåĞ¡æŢřãĂĆäŷ■è£ĞïijŃèů§åL'■éİćäŷ■åŘŃçŽĎæŸŕïijŃä¡ääŷ■éIJĂ

ä¡IJäŷžåĞĘåď'ĞïijŃåĄĞèő¿æIJňçńääżŃçż■éČĺåĹĘçŽĎçď'žä¿ŃäżčçăĄåůšçżŘèćńçijŰèŕŚåĹřæ§Řäŷłåŕ
libsample çŽĎCåĞ¡æŢřåžŞäŷ■äżĘãĂĆ éęŰåĚĹåĹŻåżžäŷĂäŷłåŘ■åŘń csample.pxd
çŽĎæŰĞäżűïijŃåęĆäŷŃæL'Ăçď'žïijŽ

```
# csample.pxd
#
# Declarations of "external" C functions and structures
```

(continues on next page)

(continued from previous page)

```
cdef extern from "sample.h":
    int gcd(int, int)
    bint in_mandel(double, double, int)
    int divide(int, int, int *)
    double avg(double *, int) nogil

    ctypedef struct Point:
         double x
         double y

    double distance(Point *, Point *)
```

è£ŹäÿłæŰĞäżűåIJĺCythonäÿ■çŽĎä¡IJçŤĺåřśèů§CçŽĎåď'ťæŰĞäżűäÿĂæăůãĂĆ åĹİåğŃåčřæŸŐ `cdef extern from "sample.h"` æŇĞåőŽäžĘæĽĂå■ęçŽĎCåď'ťæŰĞäżűãĂĆ æŐěäÿŃæİěçŽĎåčřæŸŐéČ¡æŸŕæİěèĞłäžŐéĆčäÿłåď'ťæŰĞäżűãĂĆæŰĞäżűåŘ■æŸŕ `csample.pxd` ïijŇèĂŇäÿ■æŸŕ `sample.pxd` âĂŤâĂŤè£ŹçĆźå¿ĹéĞ■èęĄãĂĆ

äÿŃäÿĂæ■ěïijŇåĹŻåżžäÿĂäÿłåŘ■äÿž `sample.pyx` çŽĎéŮőécŸãĂĆ èŕěæŰĞäżűäijŽåőŽäźĽåŇĚèčĚåŹĺïijŇçŤĺæİěæąěæŐěPythonèğčéĞŁåŹĺåĹř `csample.pxd` äÿ■åčřæŸŐçŽĎCäžčçăĄãĂĆ

```
# sample.pyx

# Import the low-level C declarations
cimport csample

# Import some functionality from Python and the C stdlib
from cpython.pycapsule cimport *

from libc.stdlib cimport malloc, free

# Wrappers
def gcd(unsigned int x, unsigned int y):
    return csample.gcd(x, y)

def in_mandel(x, y, unsigned int n):
    return csample.in_mandel(x, y, n)

def divide(x, y):
    cdef int rem
    quot = csample.divide(x, y, &rem)
    return quot, rem

def avg(double[:] a):
    cdef:
        int sz
        double result
```

(continues on next page)

(continued from previous page)

```
    sz = a.size
    with nogil:
        result = csample.avg(<double *> &a[0], sz)
    return result

# Destructor for cleaning up Point objects
cdef del_Point(object obj):
    pt = <csample.Point *> PyCapsule_GetPointer(obj,"Point")
    free(<void *> pt)

# Create a Point object and return as a capsule
def Point(double x,double y):
    cdef csample.Point *p
    p = <csample.Point *> malloc(sizeof(csample.Point))
    if p == NULL:
        raise MemoryError("No memory to make a Point")
    p.x = x
    p.y = y
    return PyCapsule_New(<void *>p,"Point",<PyCapsule_Destructor>
↪del_Point)

def distance(p1, p2):
    pt1 = <csample.Point *> PyCapsule_GetPointer(p1,"Point")
    pt2 = <csample.Point *> PyCapsule_GetPointer(p2,"Point")
    return csample.distance(pt1,pt2)
```

èŕĕæŰĞäżűæŻť åď ŽçŽĎçżĘèŁĆéČĺåĹĘäijŽåIJĺèőĺèőžéČĺåĹĘèŕęçżĘåśŢåijĂãĂĆ æIJĂåŘŐïijŇäÿžäžĘæđĎåżžæĽľ åśŢæĺąåİŮïijŇåČŔäÿŃéİćè£ŹæăůåĹŻåżžäÿĂäÿł `setup.py` æŰĞäżűïijŽ

```
from distutils.core import setup
from distutils.extension import Extension
from Cython.Distutils import build_ext

ext_modules = [
    Extension('sample',

              ['sample.pyx'],
              libraries=['sample'],
              library_dirs=['.'])]
setup(
  name = 'Sample extension module',
  cmdclass = {'build_ext': build_ext},
  ext_modules = ext_modules
)
```

èęĄæđĎåżžæĹŚäżňæţŃèŕŢçŽĎçŻőæăĞæĺąåİŮïijŇåČŔäÿŃéİćè£ŹæăůåĄŽïijŽ

```
bash % python3 setup.py build_ext --inplace
```

(continues on next page)

(continued from previous page)

```
running build_ext
cythoning sample.pyx to sample.c
building 'sample' extension
gcc -fno-strict-aliasing -DNDEBUG -g -fwrapv -O3 -Wall -Wstrict-
↪prototypes
 -I/usr/local/include/python3.3m -c sample.c
 -o build/temp.macosx-10.6-x86_64-3.3/sample.o
gcc -bundle -undefined dynamic_lookup build/temp.macosx-10.6-x86_64-
↪3.3/sample.o
  -L. -lsample -o sample.so
bash %
```

åęĆæđIJäÿĂåĹĞéąžåĹľçŽĎèŕĬïijŇä¡ăåžŤèŕěæIJĽäžĘäÿĂäÿłæĽľåśŢæĹąåİŮ `sample.so` ïijŇåŘŕåIJĺäÿŃéİćä¿Ňå■Řäÿ■ä¡£çŤĺïijŽ

```
>>> import sample
>>> sample.gcd(42,10)
2
>>> sample.in_mandel(1,1,400)
False
>>> sample.in_mandel(0,0,400)
True
>>> sample.divide(42,10)
(4, 2)
>>> import array
>>> a = array.array('d',[1,2,3])
>>> sample.avg(a)
2.0
>>> p1 = sample.Point(2,3)
>>> p2 = sample.Point(4,5)
>>> p1
<capsule object "Point" at 0x1005d1e70>
>>> p2
<capsule object "Point" at 0x1005d1ea0>
>>> sample.distance(p1,p2)
2.8284271247461903
>>>
```

èőĺèőž

æIJňèŁĆåŇĚåŘńäžĘå¿Ĺåď ŽåĽ■éİćæĽĂèőšçŽĎéńŸçžğçĽźæĂğïijŇåŇĚæŃňæŢřçżĎæŞ■ä¡IJãĂĄåŇĚ
æŕŘäÿĂéČĺåĹĘéČ¡äijŽéĂŘäÿłèćńèőšè£řåĹřïijŇä¡ĘæŸŕæĹŚäžňæIJĂåě¡èČ¡åď■äźääÿĂäÿŃåĽ■éİćåĞäåřŘèŁ
åIJĺéąűåśĆïijŇä¡£çŤĺCythonæŸŕå§žäžŐCäźŃäÿŁãĂĆ.pxdæŰĞäżűäžĚäżĚåŘłåŇĚåŘńCåőŽäźĽïijĹçśżäijij.h
.pyxæŰĞäżűåŇĚåŘńäžĘåőđçŐřïijĹçśżäijij.cæŰĞäżűïijĽãĂĆcimport
èŕ■åŘěèćńCythonçŤĺæĺěåŕijåĚě.pxdæŰĞäżűäÿ■çŽĎåőŽäźĽãĂĆ
åőČèů§ä¡£çŤĺæŹőéĂŽçŽĎåŁäè¡¡PythonæĺąåİŮçŽĎåŕijåĚěèŕ■åŘěæŸŕäÿ■åŘŇçŽĎãĂĆ

åř¡çőą *.pxd* æŰĞäżűåŇĚåŘńäžĘåőŽäźĽïijŇä¡ĘåőČäžňåźűäÿ■æŸŕçŤĺæĺěèĞłåŁĺåĹŻåżžæĽľåśŢäžčçăĄç

åŻžäæ■ď'ïijŇä¡ăè£ŸæŸŕèęĄåĘźåŇĚèčĚåĞ¡æŢřãĂĆä¿ŃåęĆïijŇåřśçőŮ `csample.pxd` æŰĞäżűåčřæŸŐäžĘ `int gcd(int, int)` åĞ¡æŢřïijŇ ä¡ääż■çĎűéIJĂèęĄåIJÍ `sample.pyx` äÿ■äÿžåőČåĘźäÿĂäÿłåŇĚèčĚåĞ¡æŢřãĂĆä¿ŃåęĆïijŽ

```
cimport csample

def gcd(unsigned int x, unsigned int y):
    return csample.gcd(x,y)
```

åŕźäžŐçőĂå■ŢçŽĎåĞ¡æŢřïijŇä¡ääźűäÿ■éIJĂèęĄåŐżåĄŽåď'łåď'ŽçŽĎæŮűãĂĆ CythonäijŽçŤ§æĹŘåŇĚèčĚäżčçăĄæİěæ■čçąőçŽĎè¡ňæ■ćåŘĆæŢřåŠŇè£ŤåŻđåĂijãĂĆ çżŚåőŽåĹřåśđæĂğäÿŁçŽĎCæŢřæ■őçśżåđŃæŸŕåŘŕéĂĽ'çŽĎãĂĆäÿ■è£ĞïijŇåęĆæđIJä¡ääŇĚåŘńäžĘåőČäżňï ä¿ŃåęĆïijŇåęĆæđIJæIJĽäžžä¡£çŤĺèť'§æŢřæİěěřČçŤĺè£ŹäÿłåĞ¡æŢřïijŇäijŽæŁŻåĞžäÿĂäÿłåijĆåÿÿïijŽ

```
>>> sample.gcd(-10,2)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "sample.pyx", line 7, in sample.gcd (sample.c:1284)
    def gcd(unsigned int x,unsigned int y):
OverflowError: can't convert negative value to unsigned int
>>>
```

åęĆæđIJä¡ăæČşåŕźåŇĚèčĚåĞ¡æŢřåĄŽåŘęåď'ŰçŽĎæčĂæ§ěïijŇåŘłéIJĂèęĄä¡£çŤĺåŘęåď'ŰçŽĎåŇĚèčĚ

```
def gcd(unsigned int x, unsigned int y):
    if x <= 0:
        raise ValueError("x must be > 0")
    if y <= 0:
        raise ValueError("y must be > 0")
    return csample.gcd(x,y)
```

åIJÍcsample.pxdæŰĞäżűäÿ■çŽĎ''in_mandel()'' åčřæŸŐæIJĽäÿłå¿ĹæIJĽèűčä¡ĘæŸŕæŕŤè¿ČéŽ¿çŘĘèğ åIJÍè£ŹäÿłæŰĞäżűäÿ■ïijŇåĞ¡æŢřèćńåčřæŸŐäÿžçĎűåŘŐäÿĂäÿłbintèĂŇäÿ■æŸŕäÿĂäÿłintãĂĆ åőČäijŽèőľåĞ¡æŢřåĹŻåżžäÿĂäÿłæ■čçąőçŽĎBooleanåĂijèĂŇäÿ■æŸŕçőĂå■ŢçŽĎæŢť'æŢřãĂĆ åŻžäæ■ď'ïijŇè£ŤåŻđåĂij0èąĺçď'žFalseèĂŇ1èąĺçď'žTrueãĂĆ

åIJÍCythonåŇĚèčĚåŹĺäÿ■ïijŇä¡ääŕŕäżěéĂĽ'æŃľ'ačřæŸŐCæŢřæ■őçśżåđŃïijŇäź§åŕŕäżěä¡£çŤĺæĽ'ĂæIJ åŕźäžŐ `divide()` çŽĎåŇĚèčĚåŹĺåśŢçď'žäžĘè£ŹæäůäÿĂäÿłä¿Ńå■ŘïijŇåŘŇæŮűè£ŸæIJĽäęĆä¡ŢåŐżåď'Ď

```
def divide(x,y):
    cdef int rem
    quot = csample.divide(x,y,&rem)
    return quot, rem
```

åIJÍè£ŹéĞŇïijŇ`rem` åŘŸéĞŘèćńæŸ¿çď'žçŽĎåčřæŸŐäÿžäÿĂäÿłCæŢť'åđŃåŘŸéĞŘãĂĆ å¡ŞåőČèćńäijäåĚě `divide()` åĞ¡æŢřçŽĎæŮűåĂŹïijŇ`&rem` åĹŻåżžäÿĂäÿłèů§CäÿĂæäůçŽĎæŇĞåŘŚåőČçŽĎæŇĞéŚĹãĂĆ `avg()` åĞ¡æŢřçŽĎäżčçăĄæijŤçď'žäžĘCythonæŻť'éńŸçžğçŽĎçĽźæĂğãĂĆ éęŰåĚĹ `def avg(double[:] a)` åčřæŸŐäžĘ `avg()` æŐěåŘŮäÿĂäÿłäÿĂçžť'çŽĎåŘŇçš¿äžęåĘĚå■ŸèğĘåŻ¿ãĂĆ æIJĂæČŁäěĞçŽĎéČĺåĹĘæŸŕè£ŤåŻđçŽĎçżŞæđ

```
>>> import array
>>> a = array.array('d',[1,2,3])
>>> import numpy
>>> b = numpy.array([1., 2., 3.])
>>> import sample
>>> sample.avg(a)
2.0
>>> sample.avg(b)
2.0
>>>
```

åIJÍæ■ď åŇĚèčĚåŹÍäÿ■ïijŇa.size0 åŠŇ &a[0] åĹĘåĹńåijŢçŤÍæŢřçżĎåĚČçť ääÿłæŢřåŠŇåžŢåśĆæŇ
èŕ■æşŢ <double *> &a[0] æŢŹä¡äæĂŐæăůåřĘæŇĞéŠĹè¡ňæ■ćäÿžäÿ■åŘŇçŽĎçśżåđŃãĂĆ
åĽ■æŘŘæŸŕCäÿ■çŽĎ avg() æŐěåŘŮäÿĂäÿłæ■čçąőçśżåđŃçŽĎæŇĞéŠĹãĂĆ
åŘĆèĂČäÿŃäÿĂèŁĆåĚşäžŐCythonåĘĚå■ŸèğĘåŻ¿çŽĎæŻť éńŸçžğèőšè£řãĂĆ

éŹď ažĘåď ĎçŘĘéĂŽåÿÿçŽĎæŢřçżĎåď ŰïijŇavg() çŽĎè£Źäÿłä¿Ńå■Řè£ŸåśŢçď žäžĘåęĆä¡Ţåď ĎçŘ
èŕ■åŘě with nogil: åčřæŸŐäžĘäÿĂäÿłäÿ■éIJĂèęĄGILåřśèČ¡æĽ ğèąŇçŽĎäżčçăĄåİŮãĂĆ
åIJÍè£ŹäÿłåİŮäÿ■ïijŇäÿ■èČ¡æIJĽäżżä¡ŢçŽĎæŹőéĂŽPythonåŕźèśąâĂŤâĂŤåŘłèČ¡ä¡£çŤÍèćńåčřæŸŐäÿž
cdef çŽĎåŕźèśąåŠŇåĞ¡æŢřãĂĆ åŘęåď ŰïijŇåď ŰéČÍåĞ¡æŢřå£ĚéążçŐřåőđçŽĎåčřæŸŐäőČäžňèČ¡äÿ■ä¿İèţ
åŻăæ■ď ïijŇåIJÍcsample.pxdæŰĞäżűäÿ■ïijŇavg() èćńåčřæŸŐäÿž double avg(double
*, int) nogil.

åŕźPointçżŞæđĎä¡ŞçŽĎåď ĎçŘĘæŸŕäÿĂäÿłæŇŚæĹŸãĂĆæIJňèŁĆä¡£çŤÍèČűåŻŁåŕźèśąåřĘPointåŕźèśąâ
èęĄè£ŹæăůåĄŽçŽĎèŕÍïijŇäžŢåśĆCythonäżčçăĄçÍ■å¿őæIJĽçĆźåď ■æÍĆãĂĆ
éęŰåĚĹïijŇäÿŃéÍćçŽĎåŕijåĚěèćńçŤÍæÍěåijŢåĚěCåĞ¡æŢřåžŞåŠŇPython C
APIäÿ■åőŽäźĽçŽĎåĞ¡æŢřïijŽ

```
from cpython.pycapsule cimport *
from libc.stdlib cimport malloc, free
```

åĞ¡æŢř del_Point() åŠŇ Point() ä¡£çŤÍè£ŹäÿłåŁ§èČ¡æÍěåĹŻåżžäÿĂäÿłèČűåŻŁåŕźèśąïijŇ
åőČäijŽåŇĚèčĚäÿĂäÿł Point * æŇĞéŠĹãĂĆcdef del_Point() åřĘ del_Point()
åčřæŸŐäÿžäÿĂäÿłåĞ¡æŢřïijŇ åŘłèČ¡éĂŽè£ĞCythonèő£éŰőïijŇèĂŇäÿ■èČ¡äżŐPythonäÿ■èő£éŰőãĂĆ
åŻăæ■ď ïijŇè£ŹäÿłåĞ¡æŢřåŕźåď ŰéČÍæŸŕäÿ■åŘŕèğĄçŽĎâĂŤâĂŤåőČèćńçŤÍæÍěå¡ŞåĄŽäÿĂäÿłåŻđèřČåĞ¡æ
åĞ¡æŢřèřČçŤÍæŕŤåęĆ PyCapsule_New() ãĂĄPyCapsule_GetPointer()
çŻť æŐěæÍěèĞłPython C APIåźűäÿŤäżěåŘŇæăůçŽĎæŰźåijŘèćńä¡£çŤÍãĂĆ

distance åĞ¡æŢřäżŐ Point() åĹŻåżžçŽĎèČűåŻŁåŕźèśąäÿ■æŘŘåŘŰæŇĞéŠĹãĂĆ
è£ŹéĞŇèęĄæşÍæĎŘçŽĎæŸŕä¡ääÿ■éIJĂèęĄæŇĚå£ČåijĆåÿÿåď ĎçŘĘãĂĆ
åęĆæđIJäÿĂäÿłéŤŹèŕŕçŽĎåŕźèśąèćńäijäè£ŻæÍěïijŇPyCapsule_GetPointer()
äijŽæŁŻåĞžäÿĂäÿłåijĆåÿÿïijŇ ä¡ĘæŸŕCythonåůšçżŘç§ěéĄŞæĂŐäźĹæ§ěæĽ¿åĹřåőČïijŇåźűåřĘåőČäżŐ
distance() äijäéĂŠåĞžåŐżãĂĆ

åď ĎçŘĘPointçżŞæđĎä¡ŞäÿĂäÿłçijżçĆźæŸŕåőČçŽĎåőđçŐřæŸŕäÿ■åŘŕèğĄçŽĎãĂĆ
ä¡ääÿ■èČ¡èő£éŰőäżżä¡ŢåśđæĂğæÍěæ§ěçIJŃåőČçŽĎåĘĚéČÍãĂĆ
è£ŹéĞŇæIJĽåŘęåď ŰäÿĂçğ■æŰźæşŢåŐżåŇĚèčĚåőČïijŇåřśæŸŕåőŽäźĽäÿĂäÿłæĽľåśŢçśżåđŃïijŇåęĆäÿŃ

```
# sample.pyx
```

(continues on next page)

(continued from previous page)

```
cimport csample
from libc.stdlib cimport malloc, free
...

cdef class Point:
    cdef csample.Point *_c_point
    def __cinit__(self, double x, double y):
        self._c_point = <csample.Point *> malloc(sizeof(csample.
↪Point))
        self._c_point.x = x
        self._c_point.y = y

    def __dealloc__(self):
        free(self._c_point)

    property x:
        def __get__(self):
            return self._c_point.x
        def __set__(self, value):
            self._c_point.x = value

    property y:
        def __get__(self):
            return self._c_point.y
        def __set__(self, value):
            self._c_point.y = value

def distance(Point p1, Point p2):
    return csample.distance(p1._c_point, p2._c_point)
```

åIJÍè£ŹéĞŇïijŇcdifçśż `Point` åřĘPointåčřæŸŐäÿžäÿĂäÿłæĽľåśŢçśżåđŃãĂĆçśżåśđæĂğ `cdef csample.Point *_c_point` åčřæŸŐäžĘäÿĂäÿłåőđä¿ŃåŘŸéĞŘïijŇæŃěæIJĽäÿĂäÿłæŇĞåŘŠåžŢåśĆPointçżŞæđĎä¡ŞçŽĎæŇĞéŠĹãĂĆ `__cinit__()` åŠŇ `__dealloc__()` æŰźæşŢéĂŽè£Ğ `malloc()` åŠŇ `free()` åĹŻåżžåźűéŤĂæŕĄåžŢåśĆCçżŞæđĎä¡ŞãĂĆxåŠŇyåśđæĂğçŽĎåčřæŸŐèőľä¡ăèŐůåŘŰåŠŇèő¿ç¡őåžŢåśĆçżŞæđĎä¡ŞçŽĎåśđæĂğåĂijãĂĆ `distance()` çŽĎåŇĚèčĚåŹĺè£ŸåŘŕäżěèćńä£őæŤźïijŇä¡£å¿ŮåőČèČ¡æŐěåŘŮ `Point` æĽľåśŢçśżåđŃåőđä¿Ńä¡IJäÿžåŘĆæŢřïijŇ èĂŇäijăéĂŠåžŢåśĆæŇĞéŠĹçżŹCåĞ¡æŢřãĂĆ

åĄŽäžĘè£ŹäÿłæŤźåŘŸåŘŐïijŇä¡ăäijŽåŘŚçŐřæŞ■ä¡IJPointåŕźèśąåřśæŸ¿å¿ŮæŻťåŁăèĞłçĎűäžĘïijŽ

```
>>> import sample
>>> p1 = sample.Point(2,3)
>>> p2 = sample.Point(4,5)
>>> p1
<sample.Point object at 0x100447288>
>>> p2
<sample.Point object at 0x1004472a0>
>>> p1.x
```

(continues on next page)

(continued from previous page)

```
2.0
>>> p1.y
3.0
>>> sample.distance(p1,p2)
2.8284271247461903
>>>
```

æIJňèŁĆåůšçżŘæijŤçď žäžĘå¿Ĺåď ŽCythonçŽĎæäÿå£ČçĽźæĂğïijŇä¡ääŘŕäżěäżěæ■ď äÿžå§žåĞĘæİěæ äÿ■è£ĞïijŇä¡ăæIJĂäě¡åĚĹåŐżéŸĚèŕżäÿŃäőŸæŰźæŰĞæąčæİěäžĘèğčæŻť åď Žä£ąæĄŕãĂĆ

æŐěäÿŃæİěåĞäèŁĆè£ŸäijŽçżğçż■æijŤçď žäÿĂäžŻCythonçŽĎåĚűäżŰçĽźæĂğãĂĆ

17.11 15.11 çŤĹCythonåĘŹéńŸæĂğèČ¡çŽĎæŢřçżĎæŞ■ä¡IJ

éŮőécŸ

ä¡äèęĄåĘŹéńŸæĂğèČ¡çŽĎæŞ■ä¡IJæİěèĞłNumPyäźŃçśżçŽĎæŢřçżĎèőąçőŮåĞ¡æŢřãĂĆ ä¡ääůšçżŘç§ěéĄŞäžĘCythonè£Źæäůç ŽĎäůěäĚůäijŽèőľäőČäŘŸå¿ŮçőĂå■ŢïijŇä¡ĘæŸŕäźűäÿ■çąőäőŽèŕěæĂ

èğčåĘşæŰźæąĹ

ä¡IJäÿžäÿĂäÿłä¿Ňå■ŘïijŇäÿŃéİćçŽĎäżčçăĄæijŤçď žäžĘäÿĂäÿłCythonåĞ¡æŢřïijŇçŤĺæİěä£őæŢť äÿĂä

```
# sample.pyx (Cython)

cimport cython

@cython.boundscheck(False)
@cython.wraparound(False)
cpdef clip(double[:] a, double min, double max, double[:] out):
    '''
    Clip the values in a to be between min and max. Result in out
    '''
    if min > max:
        raise ValueError("min must be <= max")
    if a.shape[0] != out.shape[0]:
        raise ValueError("input and output arrays must be the same␣
↪size")
    for i in range(a.shape[0]):
        if a[i] < min:
            out[i] = min
        elif a[i] > max:
            out[i] = max
        else:
            out[i] = a[i]
```

èęĄçijŰèŕŚåŠŇæđĎåżžè£ŹäÿłæĽľåśŢïijŇä¡ăéIJĂèęĄäÿĂäÿłåČŔäÿŃéİćè£ŹæăůçŽĎ `setup.py` æŰĞäżű ïijĹä¡£çŤĺ `python3 setup.py build_ext --inplace` æİěæđĎåżžåőČïijĽïijŽ

```
from distutils.core import setup
from distutils.extension import Extension
from Cython.Distutils import build_ext

ext_modules = [
    Extension('sample',
              ['sample.pyx'])
]

setup(
  name = 'Sample app',
  cmdclass = {'build_ext': build_ext},
  ext_modules = ext_modules
)
```

ä¡ăäijŽåŔŚçŐřçżŞæđIJåĞ¡æŢřçąőåőđåŕźæŢřçżĎè£ŻèąŇçŽĎä£őæ■čïijŇåźűäÿŤåŔŕäżěéĂĆçŤĺäžŐåď Žç

```
>>> # array module example
>>> import sample
>>> import array
>>> a = array.array('d',[1,-3,4,7,2,0])
>>> a

array('d', [1.0, -3.0, 4.0, 7.0, 2.0, 0.0])
>>> sample.clip(a,1,4,a)
>>> a
array('d', [1.0, 1.0, 4.0, 4.0, 2.0, 1.0])

>>> # numpy example
>>> import numpy
>>> b = numpy.random.uniform(-10,10,size=1000000)
>>> b
array([-9.55546017,  7.45599334,  0.69248932, ...,  0.69583148,
       -3.86290931,  2.37266888])
>>> c = numpy.zeros_like(b)
>>> c
array([ 0.,  0.,  0., ...,  0.,  0.,  0.])
>>> sample.clip(b,-5,5,c)
>>> c
array([-5.        ,  5.        ,  0.69248932, ...,  0.69583148,
       -3.86290931,  2.37266888])
>>> min(c)
-5.0
>>> max(c)
5.0
>>>
```

ä¡ăè£ŸäijŽåŔŚçŐřè£ŘèąŇçŤ§æĹŘçżŞæđIJéİđåÿÿçŽĎå£ńăĂĆ
äÿŃéĺćæĹŚäżňåřĘæIJňä¿ŃåŠŇnumpyäÿ■çŽĎåůšå■ŸåIJĺçŽĎ clip()
åĞ¡æŢřåĄŽäÿĂäÿłæĂğèČ¡åŕźæŕŤïijŽ

```
>>> timeit('numpy.clip(b,-5,5,c)','from __main__ import b,c,numpy',
↪number=1000)
8.093049556000551
>>> timeit('sample.clip(b,-5,5,c)','from __main__ import b,c,sample
↪',
...         number=1000)
3.760528204000366
>>>
```

æ■čåęĆä¡äçIJŃåĹřçŽĎïijŃåőČèęĄå£ńå¿Ĺåď ŽâĂŤâĂŤè£ŹæŸŕäÿĂäÿłå¿ĹæIJĽèűčçŽĎçżŞæđIJïijŃåŻäż

èőĺèőž

æIJňèŁĆåĹľçŤĺäžĘCythonçśżåđŃçŽĎåĘĚå■ŸèğĘåŻ¿ïijŃæđĄåď ğçŽĎçőĂåŃŰäžĘæŢřçżĎçŽĎæŞ■ä¡I
cpdef clip() åčřæŸŐäžĘ clip() åŘŃæŮűäÿžCçžğåĹńåĞ¡æŢřäżěåŔŁPythonçžğåĹńåĞ¡æŢřãĂĆ
åIJĺCythonäÿ■ïijŃè£ŹäÿłæŸŕå¿ĹéĞ■èęĄçŽĎïijŃåŽääÿžåőČèąĺçď žæ■ď åĞ¡æŢřěřČçŤĺèęĄæŕŤåĚűäżŰCytho
ïijĹæŕŤåęĆä¡äæČşåIJĺåŘęåď ŰäÿĂäÿłäÿ■åŘŃçŽĎCythonåĞ¡æŢřäÿ■èřČçŤĺclip()ïijL'ãĂĆ

çśżåđŃåŔĆæŢř double[:] a åŠŃ double[:] out
åčřæŸŐè£ŹäžŻåŔĆæŢřäÿžäÿĂçżť çŽĎåŔŃçš¿åžęæŢřçżĎãĂĆ
ä¡IJäÿžè¿ŞåĚěïijŃåőČäżňäijŽèő£éŮőäżżä¡ŢåőđçŐřäžĘåĘĚå■ŸèğĘåŻ¿æŐěåŔčçŽĎæŢřçżĎåŕźèśąïijŃè£ŹäÿŁ
3118æIJL'èŕęçżĘåőŽäźL'ãĂĆ åŃĚæŃňäžĘNumPyäÿ■çŽĎæŢřçżĎåŠŃåĘĚç¡őçŽĎarrayåžŞãĂĆ

å¡Şä¡äçijŰåĘŹçŤ§æĹŘçżŞæđIJäÿžæŢřçżĎçŽĎäżčçăĄæŮűïijŃä¡ääžŤèŕěéĄţå¿łäÿŁéİćçď žä¿ŃéĆčæäůè
åőČäijŽåřĘåĹŻåżžè¿ŞåĞžæŢřçżĎçŽĎèť čäżżçżŹèřČçŤĺèĂĚïijŃäÿ■éIJĂèęĄç§ěéĄŞä¡äæŞ■ä¡IJçŽĎæŢřçżĎç
ïijĹåőČäżĚäżĚåĄĞèő¿æŢřçżĎåůšçżŔåĞĘåď Ğåě¡äžĘïijŃåŔłéIJĂèęĄåĄŽäÿĂäżŽåřŔçŽĎæčĂæ§ěæŕŤåęĆçą
åIJĺåČŔNumPyäźŃçśżçŽĎäžŞäÿ■ïijŃä¡£çŤĺ numpy.zeros() æĹŰ numpy.
zeros_like() åĹŻåżžè¿ŞåĞžæŢřçżĎçŻÿåŕźèĂŃèĺĂæŕŤè¿ČåőźæŸ ŞãĂĆåŔęåď ŰïijŃèęĄåĹŻåżžæIJłåĹİ
ä¡ääŔŕäžěä¡£çŤĺ numpy.empty() æĹŰ numpy.empty_like() .
åęĆæđIJä¡äæČşèęĘçŻŰæŢřçżĎåĘĚåőźä¡IJäÿžçżŞæđIJçŽĎèŕİéĂL'æŃľè£ŹäÿďäÿłäijŽæŕŤè¿Čå£ńçĆźãĂĆ

åIJĺä¡äçŽĎåĞ¡æŢřäőđçŐřäÿ■ïijŃä¡ääŔłéIJĂèęĄçőĂå■ŢçŽĎéĂŽè£ĞäÿŃæăĞè£ŘçőŮåŠŃæŢřçżĎæ§ěæ
CythonäijŽèť §èť čäÿžä¡äçŤ§æĹŘéńŸæŢĹçŽĎäžčçăĄãĂĆ

clip() åőŽäźL'äźŃåL'■çŽĎäÿď äÿłèčĚéěřåŹĺåŔŕäžěäijŸåŃŰäÿŃæĂğèČ¡ãĂĆ
@cython.boundscheck(False) çIJĄåŐżäžĘæL'ĂæIJL'çŽĎæŢřçżĎèűŁçŢŃæčĂæ§ěïijŃ
å¡Şä¡äç§ěéĄŞäÿŃæăĞèő£éŮőäÿ■äijŽèűŁçŢŃçŽĎæŮűåĂŹåŔŕäžěä¡£çŤĺåőČãĂĆ
@cython.wraparound(False) æűĹéŹď äžĘçŻÿåŕźæŢřçżĎåř¿éČĺçŽĎèť §æŢřäÿŃæăĞçŽĎåď ĎçŘĘïijĺ
åijŢåĚěè£ŹäÿďäÿłèčĚéěřåŹĺåŔŕäžěæđĄåď ğçŽĎæŘŘå■ĞæĂğèČ¡ïijĹæţŃèŕŢè£Źäÿłä¿Ńå■ŘçŽĎæŮűåĂŹåď

äżżä¡ŢæŮűåĂŹåď ĎçŘĘæŢřçżĎæŮűïijŃçăŤçľ'ůåźűæŤźåŮĎåžŢåśĆçőŮæşŢåŘŃæăůåŔŕäžěæđĄåď ğçŽ
ä¿ŃåęĆïijŃèĂČèŹŚåŕź clip() åĞ¡æŢřçŽĎåęĆäÿŃä£őæ■čïijŃä¡£çŤĺæİąäžűèąĺè¿¿åijŔïijŽ

```
@cython.boundscheck(False)
@cython.wraparound(False)
cpdef clip(double[:] a, double min, double max, double[:] out):
```

(continues on next page)

(continued from previous page)

```
    if min > max:
        raise ValueError("min must be <= max")
    if a.shape[0] != out.shape[0]:
        raise ValueError("input and output arrays must be the same␣
↪size")
    for i in range(a.shape[0]):
        out[i] = (a[i] if a[i] < max else max) if a[i] > min else␣
↪min
```

åőđéŹĚæţŃèŕŢçżŞæđIJæŸŕïijŇè£ŹäÿłçĽĹæIJňçŽĎäżčçăĄè£ŘèąŇéĂ§åžęèęĄå£ń50%äżëäÿŁïijĹ2.44çğ
`timeit()` æţŃèŕŢçŽĎ3.76çğŠïijĽãĂĆ

åĹřè£ŹéĞŇäÿžæ■ćïijŇä¡ššŘřèČ¡æČşç§ěéĄŞè£Źçğ■äżčçăĄæĂŐäźĹèČ¡èů§æĽŃåĘŹCèŕ■èĺĂPKåŚćïij§
ä¿ŃåęĆïijŇä¡ššŘřèČ¡åĘŹäžĘåęĆäÿŃçŽĎCåĞ¡æŢřåźűä¡£çŤĺåĽ■éİćåĞäèŁĆçŽĎæŁĂæIJŕæİěæĽŃåĘŹæĽľ

```
void clip(double *a, int n, double min, double max, double *out) {
  double x;
  for (; n >= 0; n--, a++, out++) {
    x = *a;

    *out = x > max ? max : (x < min ? min : x);
  }
}
```

æĹŚäżňæšąæIJĽåśŢçď žè£ŹäÿłçŽĎæĽľåśŢäżčçăĄïijŇä¡ĘæŸŕèŕŢéłŇäźŃåŘŐïijŇæĹŚäżňåŘŚçŐřäÿĂä
æIJĂåžŢäÿŃçŽĎäÿĂèąŇæŕŤä¡äæČşèśąçŽĎè£ŘèąŇçŽĎå£ńå¿Ĺåď ŽãĂĆ

ä¡ššŘŕäżěåŕźåőđä¿ŃäżčçăĄæđĎåżžåď ŽäÿłæĽľåśŢãĂĆ åŕźäžŐæ§ŘäžŻæŢřçżĎæŞ■ä¡IJïijŇæIJĂåě¡èęĄ
èęĄè£ŹæŭůåĄŽçŽĎèŕĺïijŇéIJĂèęĄä£őæŤźäżčçăĄïijŇä¡£çŤĺ `with nogil:` èŕ■åŘěïijŽ

```
@cython.boundscheck(False)
@cython.wraparound(False)
cpdef clip(double[:] a, double min, double max, double[:] out):
    if min > max:
        raise ValueError("min must be <= max")
    if a.shape[0] != out.shape[0]:
        raise ValueError("input and output arrays must be the same␣
↪size")
    with nogil:
        for i in range(a.shape[0]):
            out[i] = (a[i] if a[i] < max else max) if a[i] > min␣
↪else min
```

åęĆæđIJä¡æČşåĘŹäÿĂäÿłæŞ■ä¡IJäžŇçżť æŢřçżĎçŽĎçĽĹæIJňïijŇäÿŃéİćæŸŕåŘŕäżěåŘĆèĂČäÿŃïijŽ

```
@cython.boundscheck(False)
@cython.wraparound(False)
cpdef clip2d(double[:,:] a, double min, double max, double[:,:]␣
↪out):
    if min > max:
```

(continues on next page)

(continued from previous page)

```
        raise ValueError("min must be <= max")
    for n in range(a.ndim):
        if a.shape[n] != out.shape[n]:
            raise TypeError("a and out have different shapes")
    for i in range(a.shape[0]):
        for j in range(a.shape[1]):
            if a[i,j] < min:
                out[i,j] = min
            elif a[i,j] > max:
                out[i,j] = max
            else:
                out[i,j] = a[i,j]
```

åÿŇæIJŻèŕżèĂĚäÿ■èęĄå£ŸäžĘæIJňèŁĆæL'ĂæIJL'äžčçăĄéČ¡äÿ■äijŽçżŚåőŽåĹřæ§ŘäÿłçL'źåőŽæŢřçżĎ
è£ŹæăůäžčçăĄåřśæŻť'æIJL'çĄtæť'żæĂğãĂĆäÿ■è£ĞïijŇèęĄæşĺæĎŔçŽĎæŸŕåęĆæđIJåď'ĎçŘĘæŢřçżĎèęĄæ
è£ŹäžŽåĘĚåőźåůšçżŔèűĚåĞžæIJňèŁĆèŇČåŻť'ïijŇæŻť'åď'Žä£ąæĄŕèŕůåŔĆèĂČ PEP 3118 ïijŇ åŘŇæŮűŰ CythonæŰĞæąčäÿ■åĚşäžŐâĂIJçśżåđŃåĘĚå■ŸèğĘåŻ¿âĂİçŕĞäź§åĂijå¿ŮäÿĂèŕżãĂĆ

17.12 15.12 åřĘåĞ¡æŢřæŇĞéŠĹè¡ňæ■ćäÿžåŔŕèřČçŤĺåŕźèśą

éŮőécŸ

ä¡ăåůšçżŔèŐůå¿ŮäžĘäÿĂäÿłèćńçijŰèŕŚåĞ¡æŢřçŽĎåĘĚå■ŸåIJřåİĂïijŇæČşåřĘåőČè¡ňæ■ćæĹŘäÿĂäÿłP
è£ŹæăůçŽĎèŕİä¡ăåřśåŔŕäžěåřĘåőČä¡IJäÿžäÿĂäÿłæL'l'åśŢåĞ¡æŢřä¡£çŤĺäžĘãĂĆ

èğčåĘşæŰźæąĹ

`ctypes` æĺąåİŮåŔŕèćńçŤĺæİěåĹŻåżžåŇĚèčĚäżżæĎŔåĘĚå■ŸåIJřåİĂçŽĎPythonåŔŕèřČçŤĺåŕźèśąãĂĆäÿŇéİćçŽĎä¿Ňå■ŘæijŤçď'žäžĘæĂŐæăůèŐůåŔŰCåĞ¡æŢřçŽĎåŐ§ăgŇãĂĄäžŢåśĆåIJřåİĂïijŇäżěåŔŁąęĆä¡

```
>>> import ctypes
>>> lib = ctypes.cdll.LoadLibrary(None)
>>> # Get the address of sin() from the C math library
>>> addr = ctypes.cast(lib.sin, ctypes.c_void_p).value
>>> addr
140735505915760

>>> # Turn the address into a callable function
>>> functype = ctypes.CFUNCTYPE(ctypes.c_double, ctypes.c_double)
>>> func = functype(addr)
>>> func
<CFunctionType object at 0x1006816d0>

>>> # Call the resulting function
>>> func(2)
```

(continues on next page)

(continued from previous page)

```
0.9092974268256817
>>> func(0)
0.0
>>>
```

èőÍèőž

èęĄæđĎåżžäÿĂäÿłåŘŕèřČçŤÍåŕźèśąïijŇä¡ăéęŰåĚĹéIJĂèęĄåĹŻåżžäÿĂäÿł CFUNCTYPE åőđä¿ŇãĂĆ CFUNCTYPE() çŽĎçňňäÿĂäÿłåŘĆæŢřæŸŕè£ŤåŻđçśżåđŇãĂĆ æŐěäÿŇæİěçŽĎåŘĆæŢřæŸŕåŘĆæŢřçśżåđŇãĂĆäÿĂæŮęä¡ăáőŽäźĽäžĘåĞ¡æŢřçśżåđŃïijŇä¡ăářśèČ¡ářĘáőČ çŤ§æĹŘçŽĎåŕźèśąèćńå¡ŞåĄŽæŹőéĂŽçŽĎåŘŕéĂŽè£Ğ ctypes èő£éŮőçŽĎåĞ¡æŢřæİěä¡£çŤÍãĂĆ

æIJňèŁĆçIJŃäÿŁåŐżåŘŕèČ¡æIJĽçĆźçěđçğŸïijŇåĄŘåžŢåśĆäÿĂçĆźãĂĆ ä¡ĘæŸŕïijŇä¡ĘæŸŕåőČèćńåź£æşŻä¡£çŤÍäžŐåŘĎçğ■énŸçžğäżčçăĄçŤ§æĹŘæŁĂæIJŕæŕŤåęĆå■şæŮűçijŰèŕŚ

ä¿ŃåęĆïijŇäÿŃéİćæŸŕäÿĂäÿłä¡£çŤÍ llvmpy æĽľåśŢçŽĎçőĂå■Ţä¿Ňå■ŘïijŇçŤÍæİěæđĎåżžäÿĂäÿłåřI åźűåřĘåĚűè¡ňæ■ćäÿžäÿĂäÿłPythonåŘŕèřČçŤÍåŕźèśąãĂĆ

```
>>> from llvm.core import Module, Function, Type, Builder
>>> mod = Module.new('example')
>>> f = Function.new(mod,Type.function(Type.double(), \
                     [Type.double(), Type.double()], False), 'foo')
>>> block = f.append_basic_block('entry')
>>> builder = Builder.new(block)
>>> x2 = builder.fmul(f.args[0],f.args[0])
>>> y2 = builder.fmul(f.args[1],f.args[1])
>>> r = builder.fadd(x2,y2)
>>> builder.ret(r)
<llvm.core.Instruction object at 0x10078e990>
>>> from llvm.ee import ExecutionEngine
>>> engine = ExecutionEngine.new(mod)
>>> ptr = engine.get_pointer_to_function(f)
>>> ptr
4325863440
>>> foo = ctypes.CFUNCTYPE(ctypes.c_double, ctypes.c_double, ctypes.
↪c_double)(ptr)

>>> # Call the resulting function
>>> foo(2,3)
13.0
>>> foo(4,5)
41.0
>>> foo(1,2)
5.0
>>>
```

åźűäÿ■æŸŕèŕťåIJÍè£ŹäÿłåśĆéİćçŁŕäžĘäżżä¡ŢéŤŹèŕŕåřśäijŽåŕijèĞťPythonèğčéĞŁåŹÍæŃĆæŐĽãĂĆ èęĄèőŕå¿ŮçŽĎæŸŕä¡ăæŸŕåIJÍçŻťæŐěèů§æIJžåŹÍçžğåĹńçŽĎåĘĚå■ŸåIJŕåİĂåŠŃæIJňåIJŕæIJžåŹÍçăĄæĽŞä

17.13 15.13 äijăéĂŠNULLçżŞář¿çŽĎå■ŮçňęäÿšçżŹCåĞ¡æŢřåžŞ

éŮőéćŸ

ä¡ăèęĄåĘŹäÿĂäÿłæL'ľåśŢæĺąåİŮïijŇéIJĂèęĄäijăéĂŠäÿĂäÿłNULLçżŞář¿çŽĎå■ŮçňęäÿšçżŹCåĞ¡æŢřå
äÿ■è£ĞïijŇä¡ăäÿ■æŸŕå¿ĹçąőåőŽæĂŐæăůä¡£çŤÍPythonçŽĎUnicodeå■Ůçňęäÿšå0żåőđçŐřåőČãĂĆ

èğčåĘşæŰźæąĹ

èőÿåď'ŽCåĞ¡æŢřåžŞåŇĚåŘńäÿĂäžŻæŞ■ä¡IJNULLçżŞář¿çŽĎå■ŮçňęäÿšïijŇèćńåčřæŸŐçśżåđŃäÿž
`char *`. èĂČèŹŚåęĆäÿŃçŽĎCåĞ¡æŢřïijŇæĹŚäżňçŤÍæİěåĄŽæijŤçď'žåŠŇæţŃèŕŢçŤÍçŽĎïijŽ

```
void print_chars(char *s) {
    while (*s) {
        printf("%2x ", (unsigned char) *s);

        s++;
    }
    printf("\n");
}
```

æ■ď'åĞ¡æŢřäijŽæL'Şå■řèćńäijăè£ŻæİěåĄŮçňęäÿšçŽĎæŕŘäÿłå■ŮçňęçŽĎå■ĄåĚ■è£ŻåĹűèąÍçď'žïijŇè£Ź

```
print_chars("Hello");   // Outputs: 48 65 6c 6c 6f
```

åŕźäžŐåIJÍPythonäÿ■èřĆçŤÍè£ŹæăůçŽĎCåĞ¡æŢřïijŇä¡ăæIJL'åĞăçğ■éĂL'æŃl'ãĂĆ
éęŰåĚĹïijŇä¡ăåŘŕäżěéĂŽè£ĞèřĆçŤÍ `PyArg_ParseTuple()`
åźűæŇĞåőŽâĂİyâĂIJè¡ňæ■ćçăĄæİěéŹŘåĹűåőČåŘłèČ¡æŞ■ä¡IJå■ŮèŁĆïijŇåęĆäÿŃïijŽ

```
static PyObject *py_print_chars(PyObject *self, PyObject *args) {
  char *s;

  if (!PyArg_ParseTuple(args, "y", &s)) {
    return NULL;
  }
  print_chars(s);
  Py_RETURN_NONE;
}
```

çżŞæđIJåĞ¡æŢřçŽĎä¡£çŤÍæŰźæşŢåęĆäÿŃãĂĆäżŤçżĘèğĆåŕ§åţŇåĚěäžĘNULLå■ŮèŁĆçŽĎå■Ůçňęäÿš

```
>>> print_chars(b'Hello World')
48 65 6c 6c 6f 20 57 6f 72 6c 64
>>> print_chars(b'Hello\x00World')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: must be bytes without null bytes, not bytes
>>> print_chars('Hello World')
```

(continues on next page)

(continued from previous page)

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'str' does not support the buffer interface
>>>
```

åęĆæđIJä¡ăæČşäijăéĂŠUnicodeå■ŮçňęäÿšïijŇåIJÍ `PyArg_ParseTuple()` äÿ■ä¡£çŤÍâĂİsâĂIJæăijåijŘçăĄïijŇåęĆäÿŃïijŽ

```
static PyObject *py_print_chars(PyObject *self, PyObject *args) {
  char *s;

  if (!PyArg_ParseTuple(args, "s", &s)) {
    return NULL;
  }
  print_chars(s);
  Py_RETURN_NONE;
}
```

å¡Ŝèćńä¡£çŤÍçŽĎæŮűåĂŹïijŇåőČäijŽèĞłåŁÍåřĘæL'ĂæIJL'å■Ůçňęäÿšè¡ňæ■ćäÿžäżěNULLçżŞåř¿çŽĎU 8çijŰçăĄãĂĆä¿ŃåęĆïijŽ

```
>>> print_chars('Hello World')
48 65 6c 6c 6f 20 57 6f 72 6c 64
>>> print_chars('Spicy Jalape\u00f1o')  # Note: UTF-8 encoding
53 70 69 63 79 20 4a 61 6c 61 70 65 c3 b1 6f
>>> print_chars('Hello\x00World')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: must be str without null characters, not str
>>> print_chars(b'Hello World')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: must be str, not bytes
>>>
```

åęĆæđIJåŽääÿžæ§ŘäžŽåŐ§åŽäïijŇä¡äèęĄçŽt'æŐëä¡£çŤÍ `PyObject *` èĂŇäÿ■èČ¡ä¡£çŤÍ `PyArg_ParseTuple()` ïijŇ äÿŃéIćçŽĎä¿Ńå■ŘåŘŚä¡äåśŢçď'žäžĘæĂŐæäůäżŐå■ŮèŁĆåŠŇå■Ůçňęäÿšåŕźèśąäÿ■æčĂæ§ěåŠŇæŘŘåŘŰäÿ `char *` åijŢçŤÍïijŽ

```
/* Some Python Object (obtained somehow) */
PyObject *obj;

/* Conversion from bytes */
{
   char *s;
   s = PyBytes_AsString(o);
   if (!s) {
      return NULL;   /* TypeError already raised */
```

(continues on next page)

(continued from previous page)

```
    }
    print_chars(s);
}

/* Conversion to UTF-8 bytes from a string */
{
    PyObject *bytes;
    char *s;
    if (!PyUnicode_Check(obj)) {
        PyErr_SetString(PyExc_TypeError, "Expected string");
        return NULL;
    }
    bytes = PyUnicode_AsUTF8String(obj);
    s = PyBytes_AsString(bytes);
    print_chars(s);
    Py_DECREF(bytes);
}
```

åL’■éİćäÿď’çğ■è¡ňæ■ćéČ¡åŘŕäżěçąőä£İæŸŕNULLçżŞåř¿çŽĎæŢřæ■őïijŇ
ä¡ĘæŸŕåőČäżňåźűäÿ■æčĂæ§ěå■Ůçňęäÿšäÿ■éŮť’æŸŕåŘęåţŇåĚěäžĘNULLå■ŮèŁĆãĂĆ
åŻăæ■ď’ïijŇåęĆæđIJè£Źäÿłå¿ĹéĞ■èęĄçŽĎèŕİïijŇéĆčä¡ăéIJĂèęĄèĞłåůśåŐżåĄŽæčĂæ§ěäžĘãĂĆ

èőĺèőž

åęĆæđIJåŘŕèČ¡çŽĎèŕİïijŇä¡ăåžŤèŕěéĄ£åĚ■åŐżåĘŹäÿĂäžŻä¿İèţŰäžŐNULLçżŞåř¿çŽĎå■ŮçňęäÿšïijŇ
æIJĂåě¡çżŞåŘĹä¡£çŤĺäÿĂäÿłæŇĞéŠĹåŠŇéŢ£åžęåĂijæİěåď’ĎçŘĘå■ŮçňęäÿšãĂĆ
äÿ■è£ĞïijŇæIJL’æŮűåĂŹä¡åå£ĚéążåŐżåď’ĎçŘĘCèŕ■èĺĂéĄŮçŢŹäżčçăĄæŮűåřśæšąå¿ŮéĂL’æŃl’äžĘãĂĆ

åř¡çőąå¿Ĺåőźæ ŸŞä¡£çŤĺïijŇä¡ĘæŸŕå¿Ĺåőźæ ŸŞå£¡ěğĘçŽĎäÿĂäÿłéŮőéćŸæŸŕåIJĺ
`PyArg_ParseTuple()` äÿ■ä¡£çŤĺâĂIJsâĂİæăijåijŘåŇŰçăĄäijŽæIJL’åĘĚå■Ÿæ■§èĂŮãĂĆ
ä¡Ęä¡ăéIJĂèęĄä¡£çŤĺè£Źçğ■è¡ňæ■ćçŽĎæŮűåĂŹïijŇäÿĂäÿłUTF-
8å■ŮçňęäÿšèćňåĹŻåżžåźűæřÿäźĚéŹĎåŁăåIJĺåŐ§åğŇå■ŮçňęäÿšåŕźèśąäÿŁéİćãĂĆ
åęĆæđIJåŐ§åğŇå■ŮçňęäÿšåŇĚåŘńéİđASCIIå■ŮçňęçŽĎèŕİïijŇåřśäijŽåŕijèĞť’å■ŮçňęäÿšçŽĎåřžåřÿåćđåĹřäÿ

```
>>> import sys
>>> s = 'Spicy Jalape\u00f1o'
>>> sys.getsizeof(s)
87
>>> print_chars(s)      # Passing string
53 70 69 63 79 20 4a 61 6c 61 70 65 c3 b1 6f
>>> sys.getsizeof(s)    # Notice increased size
103
>>>
```

åęĆæđIJä¡ăåIJĺäźŐè£ŹäÿłåĘĚå■ŸçŽĎæ■§èĂŮïijŇä¡ăæIJĂåě¡éĞ■åĘŹä¡ăçŽĎCæL’l’åśŢäżčçăĄïijŇèől’å
`PyUnicode_AsUTF8String()` åĞ¡æŢřãĂĆåęĆäÿŃïijŽ

```
static PyObject *py_print_chars(PyObject *self, PyObject *args) {
  PyObject *o, *bytes;
  char *s;

  if (!PyArg_ParseTuple(args, "U", &o)) {
    return NULL;
  }
  bytes = PyUnicode_AsUTF8String(o);
  s = PyBytes_AsString(bytes);
  print_chars(s);
  Py_DECREF(bytes);
  Py_RETURN_NONE;
}
```

éĂŽè£Ğè£ŹäÿŁä£ŏæŤźïijŇäÿĂäÿŁUTF-8çijŰçăAçŽĎå■Ůçňęäÿšæăźæ■ŏéIJĂèęĄèćňåĹŻåżžïijŇçĎűåŘĈ

```
>>> import sys
>>> s = 'Spicy Jalape\u00f1o'
>>> sys.getsizeof(s)
87
>>> print_chars(s)
53 70 69 63 79 20 4a 61 6c 61 70 65 c3 b1 6f
>>> sys.getsizeof(s)
87
>>>
```

åęĆæđIJä¡ăèŕŢçĺĂäijăéĂŠNULLçżŞåř¿å■ŮçňęäÿšçżŹctypesåŇĚèčĚè£ĞçŽĎåĞ¡æŢřïijŇ
èęĄæşĺæĎŘçŽĎæŸŕctypesåŘłèČ¡åĚĄèőÿäijăéĂŠå■ŮèŁĆïijŇåźűäÿŤåŏČäÿ■äijŽæčĂæ§ĕäÿ■éŮť åţŇåĚĕçŽĎ

```
>>> import ctypes
>>> lib = ctypes.cdll.LoadLibrary("./libsample.so")
>>> print_chars = lib.print_chars
>>> print_chars.argtypes = (ctypes.c_char_p,)
>>> print_chars(b'Hello World')
48 65 6c 6c 6f 20 57 6f 72 6c 64
>>> print_chars(b'Hello\x00World')
48 65 6c 6c 6f
>>> print_chars('Hello World')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ctypes.ArgumentError: argument 1: <class 'TypeError'>: wrong type
>>>
```

åęĆæđIJä¡ăæČşäijăéĂŠå■ŮçňęäÿšèĂŇäÿ■æŸŕå■ŮèŁĆïijŇä¡ăéIJĂèęĄåĚĹæL'ğèąŇæL'ŃåŁĺçŽĎUTF-8çijŰçăĄãĂĆä¿ŃåęĆïijŽ

```
>>> print_chars('Hello World'.encode('utf-8'))
48 65 6c 6c 6f 20 57 6f 72 6c 64
>>>
```

åŕźäžŐåĚűäżŰæL'l'åśŢåůěåĚůïijĹæŕŤåęĆSwigãĂĄCythonïijL'ïijŇ

åIJÍä¡ää¡£çŤÍåőČäžňäijäéĂŠå■ŮçňęäÿšçżŹCäžčçăĄæŮűèęĄåĚĹåě¡åě¡å■ęäźăçŻÿåžŤçŽĎäÿIJèě£äžEãĂĆ

17.14 15.14 äijäéĂŠUnicodeå■ŮçňęäÿšçżŹCåĞ¡æŢřåžŞ

éŮőécŸ

ä¡äèęĄåĘŹäÿĂäÿłæL'ľåśŢæÍąåİŮïijŇéIJĂèęĄåřĘäÿĂäÿłPythonå■ŮçňęäÿšäijäéĂŠçżŹCçŽĎæ§ŘäÿłåžŞ

èğčåĘşæŰźæąĹ

è£ŹéĞŇæĹŚäžňéIJĂèęĄèĂČèŹŚå¿Ĺåď'ŽçŽĎéŮőécŸïijŇä¡ĘæŸŕæIJĂäÿżèęĄçŽĎéŮőécŸæŸŕçŐřå■Ÿç
åŻăæ■ď'ïijŇä¡äçŽĎæŇŚæĹŸæŸŕåřĘPythonå■Ůçňęäÿšè¡ňæ■ćäÿžäÿĂäÿłèČ¡èćńCçŘĘèğčçŽĎå¡ćåijŘãĂĆ

äÿžäžĘæijŤçď'žçŽĎçŻőçŽĎïijŇäÿŃéİćæIJL'äÿď'äÿłCåĞ¡æŢřïijŇçŤÍæĚèæŞ■ä¡IJå■ŮçňęäÿšæŢřæ■őåźűè
äÿĂäÿłä¡£çŤÍå¡ćåijŘäÿž `char *, int` å¡ćåijŘçŽĎå■ŮèŁĆïijŇ
èĂŇåŘęäÿĂäÿłä¡£çŤÍå¡ćåijŘäÿž `wchar_t *, int` çŽĎåő¡å■Ůçňęå¡ćåijŘïijŽ

```
void print_chars(char *s, int len) {
  int n = 0;

  while (n < len) {
    printf("%2x ", (unsigned char) s[n]);
    n++;
  }
  printf("\n");
}

void print_wchars(wchar_t *s, int len) {
  int n = 0;
  while (n < len) {
    printf("%x ", s[n]);
    n++;
  }
  printf("\n");
}
```

åŕźäžŐéİćåŘŚå■ŮèŁĆçŽĎåĞ¡æŢř `print_chars()` ïijŇä¡äéIJĂèęĄåřĘPythonå■Ůçňęäÿšè¡ňæ■ćäÿžäÿ
8. äÿŃéİćæŸŕäÿĂäÿłè£ŹæăůçŽĎæL'ľåśŢåĞ¡æŢřä¿Ňå■ŘïijŽ

```
static PyObject *py_print_chars(PyObject *self, PyObject *args) {
  char *s;
  Py_ssize_t  len;

  if (!PyArg_ParseTuple(args, "s#", &s, &len)) {
    return NULL;
  }
  print_chars(s, len);
```

(continues on next page)

(continued from previous page)

```
  Py_RETURN_NONE;
}
```

åŕźäžŐéĆčäžŻéIJĂèęĄåď ĎçŘĘæIJžåŹĺæIJňåIJř wchar_t çśżåđŃçŽĎåžŞåĞ¡æŢřïijŇä¡ăåŘŕäżěåČŘäÿŃéĺćè£ŹæăůçijŰåĘŹæĽ'ľåśŢäžčçăĄïijŽ

```
static PyObject *py_print_wchars(PyObject *self, PyObject *args) {
  wchar_t *s;
  Py_ssize_t  len;

  if (!PyArg_ParseTuple(args, "u#", &s, &len)) {
    return NULL;
  }
  print_wchars(s,len);
  Py_RETURN_NONE;
}
```

äÿŃéĺćæŸŕäÿĂäÿłäžď äžŠäijŽèŕĺæĺěæijŤçď žè£ŹäÿłåĞ¡æŢřæŸŕåęĆä¡Ţåůěä¡IJçŽĎïijŽ

```
>>> s = 'Spicy Jalape\u00f1o'
>>> print_chars(s)
53 70 69 63 79 20 4a 61 6c 61 70 65 c3 b1 6f
>>> print_wchars(s)
53 70 69 63 79 20 4a 61 6c 61 70 65 f1 6f
>>>
```

äżŤçżĘèğĆåŕ§è£ŹäÿłéĺćåŘŚå■ŮèŁĆçŽĎåĞ¡æŢř print_chars() æŸŕæĂŐæăůæŐěåŘŮUTF-8çijŰçăĄæŢřæ■őçŽĎïijŇ äžěåŘŁ print_wchars() æŸŕæĂŐæăůæŐěåŘŮUnicodeçijŰçăĄåĂijçŽĎ

èőĺèőž

åIJĺçżğçż■æIJňèŁĆäźŃåĽ■ïijŇä¡ăåžŤèŕěéęŰåĚĹå■ęäźăä¡ăèő£éŮőçŽĎCåĞ¡æŢřåžŞçŽĎçĽźå¿ĄãĂĆ åŕźäžŐå¿Ĺåď ŽCåĞ¡æŢřåžŞïijŇéĂŽåÿÿäijăéĂŠå■ŮèŁĆèĂŃäÿ■æŸŕå■ŮçňęäÿšäijŽæŕŤè¿Čåě¡äžŽãĂĆèęĄè£

```
static PyObject *py_print_chars(PyObject *self, PyObject *args) {
  char *s;
  Py_ssize_t  len;

  /* accepts bytes, bytearray, or other byte-like object */
  if (!PyArg_ParseTuple(args, "y#", &s, &len)) {
    return NULL;
  }
  print_chars(s, len);
  Py_RETURN_NONE;
}
```

åęĆæđIJä¡ăäż■çĎűè£ŸæŸŕæČçèęĄäijăéĂŠå■ŮçňęäÿšïijŇ ä¡ăéIJĂèęĄç§ěéĄŞPython 3åŘŕä¡£çŤĺäÿĂäÿłåŘĹéĂĆçŽĎå■Ůçňęäÿšèąĺçď žïijŇ

åőČåźűäÿ■çŻť æŐěæŸäářĎåĹřä¡£çŤÍæäĞåĞĘçśżåđŃ `char *` æĹŰ
`wchar_t *` ïijĹæŻť åď ŽçżĘèŁĆåŘĆèĂČPEP 393ïijĽçŽĎCåĞ¡æŢřåžŞãĂĆ
åŻäæ■ď ïijŇèęĄåIJÍCäÿ■èąÍçď žè£ŹäÿłÅ■ŮçňęäÿšæŢřæ■őïijŇäÿĂäžŻè¡ňæ■ćè£ŸæŸřå£ĚéążèęĄçŽĎãĂĆ
åIJÍ `PyArg_ParseTuple()` äÿ■ä¡£çŤÍâĂİs#âĂİ åŠŇâĂİu#âĂİæäijåijŘåŇŰçäĄåŘřäżěåőĽ åĚÍçŽĎæĽ ğèą

äÿ■è£Ğè£Źçğ■è¡ňæ■ćæIJĽäÿłçijžçĆźåřśæŸřåőČåŘřèČ¡äijŽåříjèĞť åŐ§åğŇå■ŮçňęäÿšåŕźèśąçŽĎåřžåřÿ
äÿĂæŮęè¡ňæ■ćè£ĞåŘŐïijŇäijŽæIJĽäÿĂäÿłè¡ňæ■ćæŢřæ■őçŽĎåď ■åĹűéŹĎåŁäåĹřåŐ§åğŇå■Ůçňęäÿšåŕźèśą
ä¡äåŘřäżěèğĆåŕ§äÿŇè£Źçğ■æŢĹæđIJïijŽ

```
>>> import sys
>>> s = 'Spicy Jalape\u00f1o'
>>> sys.getsizeof(s)
87
>>> print_chars(s)
53 70 69 63 79 20 4a 61 6c 61 70 65 c3 b1 6f
>>> sys.getsizeof(s)
103
>>> print_wchars(s)
53 70 69 63 79 20 4a 61 6c 61 70 65 f1 6f
>>> sys.getsizeof(s)
163
>>>
```

åŕźäžŐåřŚéĞŘçŽĎå■ŮçňęäÿšåŕźèśąïijŇåŘřèČ¡æšąäżĂäźĹå¡śåŞ■ïijŇ
ä¡ĘæŸřåęĆæđIJä¡äéIJĂèęĄåIJÍæĽ ľ åśŢäÿ■åď ĎçŘĘåď ğéĞŘçŽĎæŰĞæIJňïijŇä¡äåŘřèČ¡æČşéĄ£åĚ■è£Źäÿł
äÿŇéİćæŸřäÿĂäÿłä£őèőćçĽ ĹæIJňåŘřäżěéĄ£åĚ■è£Źçğ■åĘĚå■Ÿæ■§èĂŮïijŽ

```
static PyObject *py_print_chars(PyObject *self, PyObject *args) {
  PyObject *obj, *bytes;
  char *s;
  Py_ssize_t   len;

  if (!PyArg_ParseTuple(args, "U", &obj)) {
    return NULL;
  }
  bytes = PyUnicode_AsUTF8String(obj);
  PyBytes_AsStringAndSize(bytes, &s, &len);
  print_chars(s, len);
  Py_DECREF(bytes);
  Py_RETURN_NONE;
}
```

èĂŇåŕź `wchar_t` çŽĎåď ĎçŘĘæŮűæČşèęĄéĄ£åĚ■åĘĚå■Ÿæ■§èĂŮåŕśæŻť åŁäéŽ¿åŁđäžĘäĂĆ
åIJÍåĘĚéČÍïijŇPythonä¡£çŤÍæIJĂéńŸæŢĹçŽĎèąÍçď žæİěå■ŸåĆÍå■ŮçňęäÿšäĂĆ
ä¿ŃåęĆïijŇåŘłåŇĚåŘńASCIIçŽĎå■Ůçňęäÿšèćńå■ŸåĆÍäÿžå■ŮèŁĆæŢřçżĎïijŇ
èĂŇåŇĚåŘńèŇČåŻť äżŐU+0000åĹřU+FFFFçŽĎå■ŮçňęçŽĎå■Ůçňęäÿšä¡£çŤÍåŘŇå■ŮèŁĆèąÍçď žäĂĆ
çŤśäžŐåŕźäžŐæŢřæ■őçŽĎèąÍçď žå¡ćåijŘäÿ■æŸřå■ŢäÿĂçŽĎïijŇä¡ääÿ■èČ¡åřĘåĘĚéČÍæŢřçżĎè¡ňæ■ćäÿž
`wchar_t *` çĎűåŘŐæIJ§æIJŻåőČèČ¡æ■čçąőçŽĎåůěä¡IJãĂĆ ä¡äåžŤèřěåĹŻåżžäÿĂäÿł
`wchar_t` æŢřçżĎåźűåŘŚåĚűäÿ■åď ■åĹűæŰĞæIJňãĂĆ `PyArg_ParseTuple()`
çŽĎâĂİu#âĂİæäijåijŘçăĄåŘřäżěåÿőåŁľä¡äéńŸæŢĹçŽĎåőŇæĹŘåőČïijĹåőČåřĘåď ■åĹűçżŞæđIJéŹĎåŁäåĹ

åęĆæđIJä¡ăæČşéĄ£åĚ■éŢ£æŮűéŮť åĘĚå■Ÿæ■§èĂŮïijŇä¡ăåŤŕäÿĂçŽĎéĂĽæŃľ åřśæŸŕåď■åĹűUnicod
åřĘåőČäijăéĂŠçżŹCåĞ¡æŢřïijŇçĎűåŘŐåŻđæŤűè£ŹäÿłæŢřçżĎçŽĎåĘĚå■ŸãĂĆäÿŃéİćæŸŕäÿĂäÿłåŕŕèČ¡çŽ

```
static PyObject *py_print_wchars(PyObject *self, PyObject *args) {
  PyObject *obj;
  wchar_t *s;
  Py_ssize_t len;

  if (!PyArg_ParseTuple(args, "U", &obj)) {
    return NULL;
  }
  if ((s = PyUnicode_AsWideCharString(obj, &len)) == NULL) {
    return NULL;
  }
  print_wchars(s, len);
  PyMem_Free(s);
  Py_RETURN_NONE;
}
```

åIJÍè£ŹäÿłåőđçŐřäÿ■ïijŇPyUnicode_AsWideCharString()
åĹŻåżžäÿĂäÿłäÿť æŮűçŽĎwchar_tçijŞåĘšåźűåď■åĹűæŢřæ■őè£ŻåŐżãĂĆ
è£Źäÿłçij ŞåĘšèćńäijăéĂŠçżŹCçĎűåŘŐèćńéĞŁæŤ¿æŐL'ãĂĆ ä¡ĘæŸŕæĹŚåĘŹè£ŹæIJňäźęçŽĎæŮűåĂŹïijŇè

åęĆæđIJä¡ăç§ěéĄŞCåĞ¡æŢřåžŞéIJĂèęĄçŽĎå■ŮèŁĆçijŰçăĄåźűäÿ■æŸŕUTF-8ïijŇ
ä¡ăåŘŕäżěåijżåĹűPythonä¡£çŤĺæL'ľåśŢçăĄæİěæL'ğèąŇæ■čçąőçŽĎè¡ňæ■ćïijŇåřśåČŘäÿŃéİćè£Źæăůïij Ž

```
static PyObject *py_print_chars(PyObject *self, PyObject *args) {
  char *s = 0;
  int   len;
  if (!PyArg_ParseTuple(args, "es#", "encoding-name", &s, &len)) {
    return NULL;
  }
  print_chars(s, len);
  PyMem_Free(s);
  Py_RETURN_NONE;
}
```

æIJĂåŘŐïijŇåęĆæđIJä¡ăæČşçŻť æŐěåď ĎçŘĘUnicodeå■ŮçňęäÿšïijŇäÿŃéİćçŽĎæŸŕä¿Ńå■ŘïijŇæijŤç

```
static PyObject *py_print_wchars(PyObject *self, PyObject *args) {
  PyObject *obj;
  int n, len;
  int kind;
  void *data;

  if (!PyArg_ParseTuple(args, "U", &obj)) {
    return NULL;
  }
  if (PyUnicode_READY(obj) < 0) {
    return NULL;
  }
```

(continues on next page)

(continued from previous page)

```
  len = PyUnicode_GET_LENGTH(obj);
  kind = PyUnicode_KIND(obj);
  data = PyUnicode_DATA(obj);

  for (n = 0; n < len; n++) {
    Py_UCS4 ch = PyUnicode_READ(kind, data, n);
    printf("%x ", ch);
  }
  printf("\n");
  Py_RETURN_NONE;
}
```

åIJÍè£ŹäÿłäžčçăĄäÿ■ïijŇPyUnicode_KIND() åŠŇ PyUnicode_DATA() è£Źäÿd'äÿłåőŘåŠŇUnicodeçŽĎåŘŕåŘŸåő¡åžęå■ŸåĆÍæIJL'åĚşïijŇè£ŹäÿłåIJÍPEP 393äÿ■æIJL'æŘŘè£řãĂĆ kind åŘŸéĞŘçijŰçăĄåžŢåśĆå■ŸåĆÍïijĹ8ä¡■ãĂĄ16ä¡■æĹŰ32ä¡■ïijL'äžěåŘŁæŇ åIJÍåőđéŹĚæČĚåĘţäÿ■ïijŇä¡ăåźűäÿ■éIJĂèęĄç§ěéĄŞäżżä¡Ţèů§è£ŹäžŻåĂijæIJL'åĚşçŽĎäÿIJèě£ïijŇ åŘłéIJĂèęĄåIJÍæŘŘåŘŰå■ŮçňęçŽĎæŮűůåĂŹåřĘåőČäžňäijăçżŹ PyUnicode_READ() åőŘãĂĆ

è£ŸæIJL'æIJĂåŘŐåĞăåŘěïijŽå¡ŞäżŐPythonäijăéĂŠUnicodeå■ŮçňęäÿšçżŹCçŽĎæŮűůåĂŹïijŇä¡ăåžŤèŕě åęĆæđIJæIJL'UTF-8åŠŇåő¡å■Ůçňęäÿd'çğ■éĂL'æŃl'ïijŇèŕůéĂL'æŃl'UTF-8. åŕźUTF-8çŽĎæŤŕæŇĄæŻť'åŁăæŹőéĄ■äÿĂäžŻïijŇäź§äÿ■åőźæŸŞçŁŕéŤŹïijŇèğčéĞŁåŹĺäź§èČ¡æŤŕæŇĄçŽĎæŻť'åě æIJĂåŘŐïijŇçąőä£İä¡ăäžŤçżĘéŸĚèŕżäžĘ åĚşäžŐåď'ĎçŘĘUnicodeçŽĎçŻÿåĚşæŰĞæąč

17.15 15.15 Cå■Ůçňęäÿšè¡ňæ■ćäÿžPythonå■Ůçňęäÿš

éŮőéćŸ

æĂŐæăůåřĘCäÿ■çŽĎå■Ůçňęäÿšè¡ňæ■ćäÿžPythonå■ŮèŁĆæĹŰäÿĂäÿłå■Ůçňęäÿšåŕźèśąïij§

èğčåĚşæŰźæąĹ

Cå■Ůçňęäÿšä¡£çŤĺäÿĂåŕź char * åŠŇ int æİěèąĺçď'žïijŇ ä¡ăéIJĂèęĄåĚşåőŽå■Ůçňęäÿšå£ŕåžŢæŸŕçŤĺäÿĂäÿłåŐ§åğŃå■ŮèŁĆå■Ůçňęäÿšè£ŸæŸŕäÿĂäÿłUnicodeå■Ůç å■ŮèŁĆåŕźèśąåŘŕäžěåČŘäÿŃéİćè£ŹæăůäÿĂ¡£çŤĺ Py_BuildValue() æİěæđĎåžžïijŽ

```
char *s;      /* Pointer to C string data */
int   len;    /* Length of data */

/* Make a bytes object */
PyObject *obj = Py_BuildValue("y#", s, len);
```

åęĆæđIJä¡ăèęĄåĹŻåžžäÿĂäÿłUnicodeå■ŮçňęäÿšïijŇåźűäÿŤä¡ăç§ěéĄŞ s æŇĞåŘŚäžĘUTF-8çijŰçăĄçŽĎæŢŕæ■ŏïijŇåŘŕäžěä¡£çŤĺäÿŃéİćçŽĎæŰźåijŘïijŽ

```
PyObject *obj = Py_BuildValue("s#", s, len);
```

åęĆæđIJ s ä¡£çŤÍåĚűäžŰçijŰçăĄæŰźåijŘïijŇéĆčäźĹåŘŕäžěåČŘäÿŃéİćä¡£çŤÍ PyUnicode_Decode() æİěæđĎåżžäÿĂäÿłå■ŮçňęäÿšïijŽ

```
PyObject *obj = PyUnicode_Decode(s, len, "encoding", "errors");

/* Examples /*
obj = PyUnicode_Decode(s, len, "latin-1", "strict");
obj = PyUnicode_Decode(s, len, "ascii", "ignore");
```

åęĆæđIJä¡ăæĄřåě¡æIJL'äÿĂäÿłçŤÍ wchar_t *, len åŕźèąĺçď'žçŽĎåő¡å■ŮçňęäÿšïijŇæIJL'åĞăçğ■éĂL'æŃl'æĂğăĂĆéęŰåĚĹä¡ăåŘŕäžěä¡£çŤÍ Py_BuildValue() ïijŽ

```
wchar_t *w;     /* Wide character string */
int len;        /* Length */

PyObject *obj = Py_BuildValue("u#", w, len);
```

åŘęåď'ŰïijŇä¡ăè£ŸåŘŕäžěä¡£çŤÍ PyUnicode_FromWideChar() :

```
PyObject *obj = PyUnicode_FromWideChar(w, len);
```

åŕźäžŐåő¡å■ŮçňęäÿšïijŇåźűæšąæIJL'åŕźå■ŮçňęæŢŕæ■őè£ŻèąŇèğčæđŘâĂŤâĂŤåőČèćńåĄĞåőŽæŸŕåŐ§

èőÍèőž

åřĘCäÿ■çŽĎå■Ůçňęäÿšè¡ňæ■ćäÿžPythonå■ŮçňęäÿšéĄţå¿łåŠŇI/OåŘŇæăůçŽĎåŐ§åĹŹãĂĆ
äź§åřśæŸŕèŕť'ïijŇæİěèĞłCäÿ■çŽĎæŢŕæ■őå£Ěéążæăźæ■őäÿĂäžŻèğčçăĄåŹÍèćńæŸ¿åijŘçŽĎèğčçăĄäÿžäÿĂä
éĂŽåÿÿçijŰçăĄæäijåijŘåŇĚæŃňASCIIãĂĄLatin-1åŠŇUTF-8.
åęĆæđIJä¡ăåźűäÿ■çąőåőŽçijŰçăĄæŰźåijŘæĹŰèĂĚæŢŕæ■őæŸŕäžŇè£ŻåĹűçŽĎïijŇä¡ăæIJĂåě¡åřĘå■Ůçňęäÿ
å¡ŞæđĎéĂääÿĂäÿłåŕźèśąçŽĎæŰűåĂŹïijŇPythonéĂŽåÿÿäijŽåď'■åĹűä¡ăæŘŘä¿ŻçŽĎå■ŮçňęäÿšæŢŕæ■őãĂĆ
åęĆæđIJæIJL'å£ĚèęĄçŽĎèŕÍïijŇä¡ăéIJĂèęĄåIJÍåŘŐéİćåŐżéĞŁæŤ¿Cå■Ůçňęäÿšã̆ĂĆ
åŘŇæŰűïijŇäÿžäžĘèől'çÍŃåžŘæŻť'åŁäąĄěåčőïijŇä¡ăäžŤèŕěåŘŇæŰűä¡£çŤÍäÿĂäÿłæŇĞéŠĹåŠŇäÿĂäÿłåď'ğ
èĂŇäÿ■æŸŕä¿ÌèţŰNULLçżŞåř¿æŢŕæ■őæİěåĹŻäžžå■Ůçňęäÿšã̆ĂĆ

17.16 15.16 äÿ■çąőåőŽçijŰçăĄæäijåijŘçŽĎCå■Ůçňęäÿš

éŰőécŸ

ä¡ăèęĄåIJÍCåŠŇPythonçŻť'æŐěæİěåŻđè¡ňæ■ćå■ŮçňęäÿšïijŇä¡ĘæŸŕCäÿ■çŽĎçijŰçăĄæäijåijŘåźűäÿ■ç
ä¿ŃåęĆïijŇåŘŕèČ¡Cäÿ■çŽĎæŢŕæ■őæIJ§æIJŻæŸŕUTF-8ïijŇä¡ĘæŸŕåźűæšąæIJL'åijžåĹűåőČå£Ěéążæ̆ŸŕãĂĆ
ä¡ăæČşçijŰåĘŹäžčçăĄæİěäžěäÿĂçğ■äijŸéŹĚçŽĎæŰźåijŘåď'ĎçŘĘè£ŹäžŻäÿ■åŘĹæäijæŢŕæ■őïijŇè£Źæăůå

èğčåĘşæŰźæąĹ

äÿŃéİćæŸŕäÿĂäžŻCçŽĎæŢřæ■ŏåŠŃäÿĂäÿłåĞ¡æŢřæİěæijŤçď žè£ŹäÿłéŮŏéćŸïijŽ

```
/* Some dubious string data (malformed UTF-8) */
const char *sdata = "Spicy Jalape\xc3\xb1o\xae";
int slen = 16;

/* Output character data */
void print_chars(char *s, int len) {
  int n = 0;
  while (n < len) {
    printf("%2x ", (unsigned char) s[n]);
    n++;
  }
  printf("\n");
}
```

åIJÍè£ŹäÿłäžčçăĄäÿ■ïijŇå■Ůçňęäÿš sdata åŇĚåŘńäžĘUTF-8åŠŃäÿ■åŘĹæăijæŢřæ■ŏãĂĆ äÿ■è£ĞïijŇåęĆæđIJçŤÍæĹůåIJÍCäÿ■èřČçŤÍ print_chars(sdata, slen) ïijŇåŏČçijžèČ¡æ■čåÿÿåůëä¡IJãĂĆ çŐřåIJÍåĄĞèő¿ä¡äæČşåřĘ sdata çŽĎåĘĚåŏźè¡ňæ■ćäÿžäÿĂäÿłPythonå■ŮçňęäÿšãĂĆ è£ŻäÿĂæ■ěåĄĞèő¿ä¡äåIJÍåŘŐéİćè£ŸæČşéĂŽè£ĞäÿĂäÿłæĽľåśŢřĘéĆčäÿłå■Ůçňęäÿšäijääÿł print_chars() åĞ¡æŢřãĂĆ äÿŃéİćæŸŕäÿĂçğ■çŤÍæİěä£İæŁďåŐ§åğŃæŢřæ■ŏçŽĎæŰźæşŢïijŇářçőŮåć

```
/* Return the C string back to Python */
static PyObject *py_retstr(PyObject *self, PyObject *args) {
  if (!PyArg_ParseTuple(args, "")) {
    return NULL;
  }
  return PyUnicode_Decode(sdata, slen, "utf-8", "surrogateescape");
}

/* Wrapper for the print_chars() function */
static PyObject *py_print_chars(PyObject *self, PyObject *args) {
  PyObject *obj, *bytes;
  char *s = 0;
  Py_ssize_t   len;

  if (!PyArg_ParseTuple(args, "U", &obj)) {
    return NULL;
  }

  if ((bytes = PyUnicode_AsEncodedString(obj,"utf-8",
↪"surrogateescape"))
        == NULL) {
    return NULL;
  }
  PyBytes_AsStringAndSize(bytes, &s, &len);
  print_chars(s, len);
```

(continues on next page)

(continued from previous page)

```
  Py_DECREF(bytes);
  Py_RETURN_NONE;
}
```

åęĆæđIJä¡ăåIJÍPythonäÿ■åřÌèŕŢè£ŹäžŻåĞ¡æŢřïijŇäÿŃéÌćæŸŕè£ŘèąŇæŢĹæđIJïijŽ

```
>>> s = retstr()
>>> s
'Spicy JalapeÃśo\udcae'
>>> print_chars(s)
53 70 69 63 79 20 4a 61 6c 61 70 65 c3 b1 6f ae
>>>
```

äžŤçżĘèğĆåŕ§çżŞæđIJä¡ääijŽåŘŚçŐřïijŇäÿ■åŘĹæäijå■ŮçňęäÿšèćńçijŰçăĄåĹřäÿĂäÿłPythonå■Ůçňęäÿ
åźűäÿŤå¡ŞåőČèćńåŻđäijăçżŹCçŽĎæŮűåĂŹïijŇèćńè¡ňæ■ćäÿžåŠŇäźŃåĽ■åŐ§ăğŃCå■ŮçňęäÿšäÿĂæăůçŽĎ

èőÍèőž

æIJňèŁĆåśŢçď žäžĘåIJÍæĽľåśŢæÍąåÌŮäÿ■åď ĎçŘĘå■ŮçňęäÿšæŮűäijŽéĚ■åĹřçŽĎäÿĂäÿłæčŸæĽŃåŘÍ
äź§åřśæŸŕèŕť ïijŇåIJÍæĽľåśŢäÿ■çŽĎCå■ŮçňęäÿšåŘŕèČ¡äÿ■äijŽäÿěæäijéĄţå¿łPythonæĽĂæIJ§æIJŻçŽĎUni
åŻäæ■ď ïijŇå¿ĹåŘŕèČ¡äÿĂäžŻäÿ■åŘĹæäijCæŢřæ■őäijäéĂŠåĹřPythonäÿ■åŐżãĂĆ
äÿĂäÿłå¿Ĺåě¡çŽĎä¿Ňå■Řåřśæ Ÿŕæűľ åŘŁåĹřåžŢåśĆçşżçż§èřĆçŤĺæŕŤåęĆæŰĞäżűåŘ■è£ŹæäůçŽĎå■Ůçňęäÿ
ä¿ŃåęĆïijŇåęĆæđIJäÿĂäÿłçşżçż§èřĆçŤĺè£ŤåŻđçżŹèğčéĞŁåŹĺäÿĂäÿłæ■§åÍŘçŽĎå■ŮçňęäÿšïijŇäÿ■èČ¡èćńa

äÿĂèĹňæÍěèőšïijŇåŘŕäžěéĂŽè£ĞåĹűåőŽäÿĂäžŻéŤŹèŕŕç■ŰçŢěæŕŤåęĆäÿěæäijãĂĄå£¡çŢěãĂĄæŻ£äžč
äÿ■è£ĞïijŇè£ŹäžŻç■ŰçŢěçŽĎäÿĂäÿłçijžçĆźæŸŕåőČäžňæřÿäźĚæĂğçăť åÍŘäžĘåŐ§ăğŃå■ŮçňęäÿšçŽĎåĘĚ
ä¿ŃåęĆïijŇåęĆæđIJä¿Ňå■Řäÿ■çŽĎäÿ■åŘĹæäijæŢřæ■őä¡£çŤĺè£ŹäžŻç■ŰçŢěäźŃäÿĂèğčçăĄïijŇä¡ääijŽå¿Ů

```
>>> raw = b'Spicy Jalape\xc3\xb1o\xae'
>>> raw.decode('utf-8','ignore')
'Spicy JalapeÃśo'
>>> raw.decode('utf-8','replace')
'Spicy JalapeÃśo?'
>>>
```

`surrogateescape` éŤŹèŕŕåď ĎçŘĘç■ŰçŢěäijŽåřĘæĽ ĂæIJĽäÿ■åŘŕèğčçăĄå■ŮèŁĆè¡ňåŃŰäÿžäÿĂä
ä¿ŃåęĆïijŽ

```
>>> raw.decode('utf-8','surrogateescape')
'Spicy JalapeÃśo\udcae'
>>>
```

å■ŢçŃňçŽĎä¡Őä¡■äžčçŘĘå■Ůçňęæŕ ŤåęĆ \udcae åIJÍUni-
codeäÿ■æŸŕéÍđæşŢçŽĎãĂĆ åŻäæ■ď ïijŇè£ŹäÿłÅ■ŮçňęäÿšåřśæŸŕäÿĂäÿłéÍđæşŢèąĹçď žãĂĆ
åőđéŹĚäÿŁïijŇåęĆæđIJä¡ăåřĘåőČäijääÿłäÿĂäÿłæĽ ğèąŇè¿ŞåĞžçŽĎåĞ¡æŢřïijŇä¡ääijŽå¿ŮåĹřäÿĂäÿłéŤŹèŕŕ

```
>>> s = raw.decode('utf-8', 'surrogateescape')
>>> print(s)
```

(continues on next page)

(continued from previous page)

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
UnicodeEncodeError: 'utf-8' codec can't encode character '\udcae'
in position 14: surrogates not allowed
>>>
```

çĎűèĂŇïijŇåĚĄèőÿäżčçŘĘè¡ňæ■ćçŽĎåĚşéŤőçĆźåIJÍäžŐäżŐCäijăçżŹPythonåŘĹåŻđäijăçżŹCçŽĎäÿ■å¡Şè£ŹäÿłÅ■Ůçňęäÿšåę■æňąä¡£çŤÍ `surrogateescape` çijŰçăĄæŮűïijŇäżčçŘĘå■ŮçňęäijŽè¡ňæ■ćåŻđåŐ

```
>>> s
'Spicy JalapeÃśo\udcae'
>>> s.encode('utf-8','surrogateescape')
b'Spicy Jalape\xc3\xb1o\xae'
>>>
```

ä¡IJäÿžäÿĂèĹňåĞĘåĹŹïijŇæIJĂåě¡éĄ£åĚ■äżčçŘĘçijŰçăĄâĂŤâĂŤåęĆæđIJä¡ăæ■čçąőçŽĎä¡£çŤÍäžĘçij äÿ■è£ĞïijŇæIJL'æŮűåĂŹçąőåőđäijŽåĞžçŐřä¡ăåźűäÿ■èČ¡æŐğåĹűæŢřæ■őçijŰçăĄåźűäÿŤä¡ăåŘĹäÿ■èČ¡å£¡ éĆčäźĹåřśåŘŕäżěä¡£çŤÍæIJňèŁĆçŽĎæŁĂæIJŕäžĘãĂĆ

æIJĂåŘŐäÿĂçĆźèęĄæşĺæĎŘçŽĎæŸŕïijŇPythonäÿ■èőÿåď'ŽéĺćåŘŚçşżçż§çŽĎåĞ¡æŢřïijŇçL'źåĹńæŸŕå éČ¡äijŽä¡£çŤÍäżčçŘĘçijŰçăĄãĂĆä¿ŃåęĆïijŇåęĆæđIJä¡ăä¡£çŤÍåČŘ `os.listdir()` è£ŹæăůçŽĎåĞ¡æŢřïijŇ äijăåĚěäÿĂäÿłåŇĚåŘńäžĘäÿ■åŘŕèğčçăĄæŰĞäżűåŘ■çŽĎçŽőå¡ŢçŽĎèŕĺïijŇåőČäijŽ åŘĆèĂČ5.15çŽĎçŽÿåĚşçńăèŁĆãĂĆ

PEP 383 äÿ■æIJL'æŻť'åď'ŽåĚşäžŐæIJňæIJžæŘŘåĹřçŽĎäżěåŘŁåŠŇsurrogateescapeéŤŹèŕŕåď'ĎçŘĘçŻÿåĚşçŽĎä£ąæĄŕãĂĆ

17.17 15.17 äijăéĂŠæŰĞäżűåŘ■çżŹCæL'ľåśŢ

éŮőéćŸ

ä¡ăéIJĂèęĄåŘŚCåžŞåĞ¡æŢřäijăéĂŠæŰĞäżűåŘ■ïijŇä¡ĘæŸŕéIJĂèęĄçąőä£ĺæŰĞäżűåŘ■æăźæ■őçşżçż§a

èğčåĘşæŰźæąĹ

åĘŹäÿĂäÿłæŐěåŘŮäÿĂäÿłæŰĞäżűåŘ■äÿžåŘĆæŢřçŽĎæL'ľåśŢåĞ¡æŢřïijŇåęĆäÿŃè£ŹæăůïijŽ

```
static PyObject *py_get_filename(PyObject *self, PyObject *args) {
  PyObject *bytes;
  char *filename;
  Py_ssize_t len;
  if (!PyArg_ParseTuple(args,"O&", PyUnicode_FSConverter, &bytes)) {
    return NULL;
  }
  PyBytes_AsStringAndSize(bytes, &filename, &len);
  /* Use filename */
  ...
```

(continues on next page)

(continued from previous page)

```
  /* Cleanup and return */
  Py_DECREF(bytes)
  Py_RETURN_NONE;
}
```

åęĆæđIJä¡ăåůšçżŘæIJL'äžĘäÿĂäÿł `PyObject *` ïijŇåÿŇæIJŻåřĘåĚűè¡ňæ■ćæĹŘäÿĂäÿłæŰĞäżűåŘ■ï

```
PyObject *obj;    /* Object with the filename */
PyObject *bytes;
char *filename;
Py_ssize_t len;

bytes = PyUnicode_EncodeFSDefault(obj);
PyBytes_AsStringAndSize(bytes, &filename, &len);
/* Use filename */
...

/* Cleanup */
Py_DECREF(bytes);

If you need to return a filename back to Python, use the following␣
↪code:

/* Turn a filename into a Python object */

char *filename;       /* Already set */
int   filename_len;   /* Already set */

PyObject *obj = PyUnicode_DecodeFSDefaultAndSize(filename, filename_
↪len);
```

èőÍèőž

äžěåŘŕçğżæď■æŰźåijŘæİěåď ĎçŘĘæŰĞäżűåŘ■æŸŕäÿĂäÿłå¿ĹæčŸæL'ŃçŽĎéŮőéćŸïijŇæIJĂåŘŐäžď
åęĆæđIJä¡ăåIJÍæL'ľåśŢäžčçăĄäÿ■ä¡£çŤÍæIJňèŁĆçŽĎæŁĂæIJŕïijŇæŰĞäżűåŘ■çŽĎåď ĎçŘĘæŰźåijŘåŠŇåŠ
åŇĚæŃňçijŰçăĄ/çŢŇéİćå■ŮèŁĆïijŇåď ĎçŘĘåİŘå■ŮçňęïijŇäžčçŘĘè¡ňæ■ćåŠŇåĚűäžŰåď ■æİĆæČĚåĘţãĂĆ

17.18 15.18 äijäéĂŠåůšæL'ŞåijĂçŽĎæŰĞäżűçżŹCæL'ľåśŢ

éŮőéćŸ

ä¡ăåIJÍPythonäÿ■æIJL'äÿĂäÿłæL'ŞåijĂçŽĎæŰĞäżűåŕźèśąïijŇä¡ĘæŸŕéIJĂèęĄåřĘåőČäijäçżŹèęĄä¡£çŤÍ

èğčåĘşæŰźæąĹ

èęĄåřĘäÿĂäÿłæŰĞäżűè¡ňæ■ćäÿžäÿĂäÿłæŢť åđŃçŽĎæŰĞäżűæŘŘè£řçňęïijŇä¡£çŤÍ `PyFile_FromFd()` ïijŇåęĆäÿŃïijŽ

```
PyObject *fobj;     /* File object (already obtained somehow) */
int fd = PyObject_AsFileDescriptor(fobj);
if (fd < 0) {
   return NULL;
}
```

çżŞæđIJæŰĞäżűæŘŘè£řçňęæŸŕéĂŽè£ĞèřČçŤÍ `fobj` äÿ■çŽĎ `fileno()` æŰźæşŢèŐůå¿ŮçŽĎãĂĆ åŻăæ■ď ïijŇäżżä¡Ţäžèè£ŹçğŒæŰźåijŘæŽť éIJšçżŹäÿĂäÿłæŘŘè£řåŹÍçŽĎåŕźèšąéČ äÿĂæŮęä¡ăæIJĽäžĘè£ŹäÿłæŘŘè£řåŹÍïijŇåőČåřśèČ¡èćńäijăéĂŠçżŹåď Žäÿłä¡ŐçžğçŽĎåŘŕåď ĎçŘĘæŰĞäżű

åęĆæđIJä¡ăéIJĂèęĄè¡ňæ■ćäÿĂäÿłæŢť åđŃæŰĞäżűæŘŘè£řçňęäÿžäÿĂäÿłPythonåŕźèšąïijŇéĂĆçŤÍäÿŃé `PyFile_FromFd()`:

```
int fd;     /* Existing file descriptor (already open) */
PyObject *fobj = PyFile_FromFd(fd, "filename","r",-1,NULL,NULL,NULL,
↪1);
```

`PyFile_FromFd()` çŽĎåŘĆæŢřåŕźåžŤåĘĚç¡őçŽĎ `open()` åĞ¡æŢřãĂĆ NULL-Lèąĺçď žçijŰçĂąãĂĄéŤŹèŕŕåŠŇæ■ćèąŇåŘĆæŢřä¡£çŤÍéżŸèőď åĂijãĂĆ

èőÍèőž

åęĆæđIJåřĘPythonäÿ■çŽĎæŰĞäżűåŕźèšąäijăçżŹCïijŇæIJĽäÿĂäžŽæşÍæĎŘäžŃéąźãĂĆ éęŰåĚĹïijŇPythonéĂŽè£Ğ `io` æĺąåİŮæĽğèąŇèĞłåůśçŽĎI/OçijŞåĘšãĂĆ åIJÍäijăéĂŠäżżä¡Ţçśżåđ ŃçŽĎæŰĞäżűæŘŘè£řçňęçżŹCäźŃåĽ■ïijŇä¡äéČ¡èęĄéęŰåĚĹåIJÍçŻÿäžŤæŰĞäżűåŕź äÿ■çĎűçŽĎèŕİïijŇä¡ääijŽæĽŞäźśæŰĞäżűçşżçż§äÿŁéĺćçŽĎæŢřæ■őãĂĆ

åĚűæňąïijŇä¡äéIJĂèęĄçĽźåĹńæşÍæĎŘæŰĞäżűçŽĎå¡Šåśđè ĂĚäżěåŘŁåĚşéŮ■æŰĞäżűçŽĎèĄŃèť čãĂĆ åęĆæđIJäÿĂäÿłæŰĞäżűæŘŘè£řçňęèćńäijăçżŹCïijŇä¡ĘæŸŕåIJÍPythonäÿ■è£ŸåIJÍèćńä¡£çŤÍçĺĂïijŇä¡äéIJĂèę çśżäijijçŽĎïijŇåęĆæđIJäÿĂäÿłæŰĞäżűæŘŘè£řçňęèćńè¡ňæ■ćäÿžäÿĂäÿłPythonæŰĞäżűåŕźèšąïijŇä¡äéIJĂèęĄ `PyFile_FromFd()` çŽĎæIJĂåŘŐäÿĂäÿłåŘĆæŢřèćńèő¿ç¡őæĹŘ1ïijŇçŤÍæĺěæŃĞåĞžPythonåžŤèŕěåĚşéŮ

åęĆæđIJä¡ăéIJĂèęĄäżŐCæăĞåĞĘI/OåžŞäÿ■ä¡£çŤÍåęĆãĂĂ `fdopen()` åĞ¡æŢřæĺěåĹŻåżżäÿ■åŘŃçśżåđ ŃçŽĎæŰĞäżűåŕźèšąæŕŤåęĆ `FILE *` åŕźèšąïijŇ ä¡äéIJĂèęĄçĽźåĹńåřŘå£ČäžĘãĂĆè£ŹæăůåĄŽäijŽåIJÍI/OåăĘæăĹäÿ■äžğçŤ§äÿď äÿłåőŃåĚÍäÿ■åŘŃçŽĎI/Oç ïijĹäÿĂäÿłæŸŕæĺěèĞłPythonçŽĎ `io` æĺąåİŮïijŇåŘęäÿĂäÿłæĺěèĞłĆçŽĎ `stdio` ïijĽãĂĆ åČŘĆäÿ■çŽĎ `fclose()` äijŽåĚşéŮ■PythonèęĄ¡£çŤÍçŽĎæŰĞäżűãĂĆ åęĆæđIJèőľä¡äéĂĽçŽĎèŕĺïijŇä¡ääžŤèŕěäijŽéĂĽæŃľåŐżæđĎäżžäÿĂäÿłæĽľåśŢäżčçăĄæĺěåď ĎçŘĘäžŢåśĆ èĂŃäÿ■æŸŕä¡£çŤÍæĺěèĞł<stdio.h>çŽĎéńŸåśĆæŁ¡èśąåŁ§èČ¡ãĂĆ

17.19 15.19 äżŐCèŕ■èÍĂäÿ■èŕżåŘŰçśżæŰĞäżűåŕźèśą

éŮőéćŸ

ä¡ăèęĄåĘŹCæL'l'åśŢæÌĕèŕżåŘŰæÌĕèĞłäżżä¡ŢPythonçśżæŰĞäżűåŕźèśąäÿ■çŽĎæŢřæ■őïijĹæŕŤåęĆæŹőé

èğčåĘşæŰźæąĹ

èęĄèŕżåŘŰäÿĂäÿłçśżæŰĞäżűåŕźèśąçŽĎæŢřæ■őïijŇä¡ăéIJĂèęĄéĞ■åď■èřČçŤÍ
`read()` æŰźæşŢïijŇçĎűåŘŐæ■čçąőçŽĎèğčçăĄèŐůå¿ŮçŽĎæŢřæ■õăĂĆ

äÿŃéÌćæŸŕäÿĂäÿłCæL'l'åśŢåĞ¡æŢřä¿Ńå■ŘïijŇäżĚäżĚåŘłæŸŕèŕżåŘŰäÿĂäÿłçśżæŰĞäżűåŕźèśąäÿ■çŽĎ

```
#define CHUNK_SIZE 8192

/* Consume a "file-like" object and write bytes to stdout */
static PyObject *py_consume_file(PyObject *self, PyObject *args) {
  PyObject *obj;
  PyObject *read_meth;
  PyObject *result = NULL;
  PyObject *read_args;

  if (!PyArg_ParseTuple(args,"O", &obj)) {
    return NULL;
  }

  /* Get the read method of the passed object */
  if ((read_meth = PyObject_GetAttrString(obj, "read")) == NULL) {
    return NULL;
  }

  /* Build the argument list to read() */
  read_args = Py_BuildValue("(i)", CHUNK_SIZE);
  while (1) {
    PyObject *data;
    PyObject *enc_data;
    char *buf;
    Py_ssize_t len;

    /* Call read() */
    if ((data = PyObject_Call(read_meth, read_args, NULL)) == NULL)
↪{
      goto final;
    }

    /* Check for EOF */
    if (PySequence_Length(data) == 0) {
      Py_DECREF(data);
      break;
```

(continues on next page)

(continued from previous page)

```
    }

    /* Encode Unicode as Bytes for C */
    if ((enc_data=PyUnicode_AsEncodedString(data,"utf-8","strict
↪"))==NULL) {
      Py_DECREF(data);
      goto final;
    }

    /* Extract underlying buffer data */
    PyBytes_AsStringAndSize(enc_data, &buf, &len);

    /* Write to stdout (replace with something more useful) */
    write(1, buf, len);

    /* Cleanup */
    Py_DECREF(enc_data);
    Py_DECREF(data);
  }
  result = Py_BuildValue("");

 final:
  /* Cleanup */
  Py_DECREF(read_meth);
  Py_DECREF(read_args);
  return result;
}
```

èęĄæţŃèŕŢè£ŹäÿłäžčçăĄïijŇåĚĹæđĎéĂääÿĂäÿłçśżæŰĞäżůåŕźèśąæŕŤåęĆäÿĂäÿłStringIOåőđä¿ŃïijŇç

```
>>> import io
>>> f = io.StringIO('Hello\nWorld\n')
>>> import sample
>>> sample.consume_file(f)
Hello
World
>>>
```

èőĺèőž

åŠŇæŹőéĂŽçşżçż§æŰĞäżűäÿ■åŘŇçŽĎæŸŕïijŇäÿĂäÿłçśżæŰĞäżůåŕźèśąåźűäÿ■éIJĂèęĄä¡£çŢĺä¡ŐçžğÄ
åŻăæ■ď'ïijŇä¡ääÿ■èČ¡ä¡£çŢĺæŹőéĂŽçŽĎCåžŞåĞ¡æŢřæİěèő£éŮőåőČãĂĆ
ä¡äéIJĂèęĄä¡£çŢĺPythonçŽĎC APIæİěåČŕæŹőéĂŽæŰĞäżűçśżäijijçŽĎéĆčæăůæŞ■ä¡IJçśżæŰĞäżůåŕźèśąãĂ

åIJĺæĹŚäžňçŽĎèğčåĘşæŰźæąĹäÿ■ïijŇ`read()` æŰźæşŢäżŐèćńäijäéĂŠçŽĎåŕźèśąäÿ■æŘŘåŘŰåĞžæİě
äÿĂäÿłåŘĆæŢřåĹŮèąĺèćńæđĎåżžçĎűåŘŐäÿ■æŰ■çŽĎèćńäijäçżŹ `PyObject_Call()`
æİěèřČçŢĺè£ŹäÿłæŰźæşŢãĂĆ èęĄæčĂæ§ěæŰĞäżűæIJńåř¿ïijĹEOFïijL'ïijŇä¡£çŢĺäžĘ
`PySequence_Length()` æİěæ§ěçIJŃæŸŕåŘęè£ŤåŻđåŕźèśąéŢ£åžęäÿž0.

åŕźäžŐæL’ĂæIJL’çŽĎI/OæŞ■ä¡IJïijŇä¡äéIJĂèęĄåĚşæşĺåžŢåśĆçŽĎçijŰçăĄæăijåijŘïijŇè£ŸæIJL’å■ŮèŁ
æIJňèŁĆæijŤçď’žäžĘåęĆä¡ŢäžěæŰĞæIJňæĺąåijŘèŕżåŘŰäÿĂäÿłæŰĞäžűåźűåřĘçżŞæđIJæŰĞæIJňèğčçăĄäÿž
åęĆæđIJä¡äæČşäžěäžŇè£ŽåĹűæĺąåijŘèŕżåŘŰæŰĞäžűïijŇåŘłéIJĂèęĄä£őæŤźäÿĂçĆźçĆźå■şåŘŕïijŇä¿ŃåęĆ

```
...
/* Call read() */
if ((data = PyObject_Call(read_meth, read_args, NULL)) == NULL) {
  goto final;
}

/* Check for EOF */
if (PySequence_Length(data) == 0) {
  Py_DECREF(data);
  break;
}
if (!PyBytes_Check(data)) {
  Py_DECREF(data);
  PyErr_SetString(PyExc_IOError, "File must be in binary mode");
  goto final;
}

/* Extract underlying buffer data */
PyBytes_AsStringAndSize(data, &buf, &len);
...
```

æIJňèŁĆæIJĂéŽ¿çŽĎåIJřæŰźåIJĺäžŐåęĆä¡Ţè£ŻèąŇæ■čçąőçŽĎåĘĚå■ŸçőąçŘĘãĂĆ
å¡Ş åď’ĎçŘĘ `PyObject *` åŘŸéĞŘçŽĎæŮűåĂŹïijŇéIJĂèęĄæşĺæĎŘçőąçŘĘåijŢçŤĺèőąæŢřäžěåŘŁåIJĺäÿ■
åŕź `Py_DECREF()` çŽĎèřČçŤĺåřśæŸŕæĺěåĄŽè£ŹäÿłçŽĎãĂĆ

æIJňèŁĆäžčçăĄäžěäÿĂçğ■éĂŽçŤĺæŰźåijŘçijŰåĘŹïijŇåŽäæ■ď’äžŰäź§èČ¡éĂĆçŤĺäžŐåĚűäżŰçŽĎæŰĆ
ä¿ŃåęĆïijŇèęĄåĘŹæŢřæ■őïijŇåŘłéIJĂèęĄèŐůåŘŰçśżæŰĞäžűåŕźèśąçŽĎ `write()`
æŰźæşŢïijŇåřĘæŢřæ■őè¡ňæ■ćäÿžåŘĹéĂĆçŽĎPythonåŕźèśą ïijĹå■ŮèŁĆæĹŰUni-
codeïijL’ïijŇçĎűåŘŐèřČçŤĺèŕěæŰźæşŢåřĘè¿ŞåĚěåĘŹåĚěåĹřæŰĞäžűãĂĆ

æIJĂåŘŐïijŇåř¡çőąçśżæŰĞäžűåŕźèśąéĂŽåÿÿè£ŸæŘŘä¿ŻåĚűäžŰæŰźæşŢïijĹæŕŤåęĆreadline(),
read_info()ïijL’ïijŇ æĹŚäžňæIJĂåě¡åŘłä¡£çŤĺå§žæIJňçŽĎ `read()` åŠŇ `write()`
æŰźæşŢãĂĆ åIJĺåĘŹCæL’ľåśŢçŽĎæŮűåĂŹïijŇèČ¡çőĂå■Ţåřśåř¡éĞŘçőĂå■ŢãĂĆ

17.20 15.20 åď’ĎçŘĘCèŕ■èĺĂäÿ■çŽĎåŘŕè£■äžčåŕźèśą

éŮőéćŸ

ä¡äæČşåĘŹCæL’ľåśŢäžčçăĄåď’ĎçŘĘæĺěèĞłäżżä¡ŢåŘŕè£■äžčåŕźèśąåęĆåĹŮèąĺãĂĄåĚČçżĎãĂĄæŰĞäż

èğčåĘşæŰźæąĹ

äÿŃéĺćæŸŕäÿĂäÿłCæL’ľåśŢåĞ¡æŢřä¿Ńå■ŘïijŇæijŤçď’žäžĘæĂŐæăůåď’ĎçŘĘåŘŕè£■äžčåŕźèśąäÿ■çŽĎ

```
static PyObject *py_consume_iterable(PyObject *self, PyObject␣
↪*args) {
  PyObject *obj;
  PyObject *iter;
  PyObject *item;

  if (!PyArg_ParseTuple(args, "O", &obj)) {
    return NULL;
  }
  if ((iter = PyObject_GetIter(obj)) == NULL) {
    return NULL;
  }
  while ((item = PyIter_Next(iter)) != NULL) {
    /* Use item */
    ...
    Py_DECREF(item);
  }

  Py_DECREF(iter);
  return Py_BuildValue("");
}
```

èőÍèőž

æIJňèŁĆäÿ■çŽĎäžčçăĄåŠŇPythonäÿ■åŕźåžŤäžčçăĄçśżäijijãĂĆ `PyObject_GetIter()` çŽĎèřČçŤÍåŠŇèřČçŤÍ `iter()` äÿĂæăůåŘŕèŐůå¿ŮäÿĂäÿłè£■äžčåŹÍãĂĆ `PyIter_Next()` åĞ¡æŢřèřČçŤÍ `next` æŰźæşŢè£ŤåŽđäÿŃäÿĂäÿłåĚČçť'äæŁŰNULL(åęĆæđIJæšąæIJL'åĚČçť'äážĘ)ãĂĆ èęĄæşÍæĎŘæ■čçąőçŽĎåĘĚå■ŸçőąçŘĘâĂŤâĂŤ `Py_DECREF()` éIJĂèęĄåŘŇæŮűåIJÍäžğçŤ§çŽĎåĚČçť'äåŠŇè£■äžčåŹÍåŕźèśąæIJňèžńäÿŁåŘŇæŮűèćńèřČçŤÍïijŇ äžěéĄ£åĚ■åĞžçŐřåĘĚå■ŸæşĎéIJšãĂĆ

17.21 15.21 èŕŁæŰ■åŁĘæőţéŤŹèŕŕ

éŮőécŸ

èğčéĞŁåŹÍåŻääÿžæ§ŘäÿłåŁĘæőţéŤŹèŕŕãĂĄæĂżçž£éŤŹèŕŕãĂĄèő£éŮőèűŁçŢŇæŁŰåĚűäżŰèĞť'åŚ¡éŤ ä¡äæČşèŐůå¿ŮPythonåäĘæäŁä£ąæĄŕïijŇäżŐèĂŇæL'¿åĞžåIJÍåŘŚçŤ§éŤŹèŕŕçŽĎæŮűåĂŹä¡äçŽĎçÍŃåžŘè£

èğčåĘşæŰźæąŁ

`faulthandler` æÍąåİŮèČ¡èćńçŤÍæİěåÿőä¡äèğčåĘşè£ŹäÿłéŮőécŸãĂĆ åIJÍä¡äçŽĎçÍŃåžŘäÿ■åijŢåĚěäÿŃåŁŮäžčçăĄïijŽ

```
import faulthandler
faulthandler.enable()
```

åŘęåď Ůè£ŸåŘŕäžěåČŘäÿŃéİćè£Źæăůä¡£çŤÍ -Xfaulthandler æİěè£ŘèąŇPythonïijŽ

```
bash % python3 -Xfaulthandler program.py
```

æIJĂåŘŐïijŇä¡ăåŘŕäžěèő¿ç¡ő PYTHONFAULTHANDLER çŐŕåćČåŘŸéĞŘãĂĆ åijĂåŘŕ-faulthandleråŘŐïijŇåIJÍCæL'ľåśŢäÿ■çŽĎèĞť åŚ¡éŤŹèŕŕäijŽåŕijèĞť äÿĂäÿłPythonéŤŹèŕŕåăĘæăĹèćńæL'Şå■ŕ

```
Fatal Python error: Segmentation fault

Current thread 0x00007fff71106cc0:
  File "example.py", line 6 in foo
  File "example.py", line 10 in bar
  File "example.py", line 14 in spam
  File "example.py", line 19 in <module>
Segmentation fault
```

ăř¡çőąè£Źäÿłåźűäÿ■èČ¡åŚŁèŕL'ä¡ăCäžčçăĄäÿ■åŞłéĞŇåĞžéŤŹäžĘïijŇä¡ĘæŸŕèĞşåřŚèČ¡åŚŁèŕL'ä¡ăPyth

èőÍèőž

faulthandleräijŽåIJÍPythonäžčçăĄæL'ğèąŇåĞžéŤŹçŽĎæŮűåĂŹåŘŚä¡ăåśŢçď žèů§èÿłä£ąæĄŕãĂĆ èĞşåřŚïijŇåőČäijŽåŚŁèŕL'ä¡ăåĞžéŤŹæŮűèćńèřČçŤÍçŽĎæIJĂéąűçžğæL'ľåśŢåĞ¡æŢřæŸŕåŞłäÿłãĂĆ åIJÍpdbåŠŇåĚűäžŰPythonèřČèŕŢåŹÍçŽĎåÿőåŁľäÿŃïijŇä¡ăářśèČ¡è£¡æăźæžŕæžŘæL'¿åĹŕéŤŹèŕŕæL'ĂåIJÍçŽĎ

faulthandleräÿ■äijŽåŚŁèŕL'ä¡ăäżżä¡ŢCèŕ■èÍĂäÿ■çŽĎéŤŹèŕŕä£ąæĄŕãĂĆ åŻăæ■ď'ïijŇä¡ăéIJĂèęĄä¡£çŤÍäijăçż§çŽĎCèřČèŕŢåŹÍïijŇæŕŤåęĆgdbãĂĆ äÿ■è£ĞïijŇåIJÍfaulthandlerè£¡èÿłä£ąæĄŕåŘŕäžěèőľ'ä¡ăåŐżåĹď'æŰ■äžŐåŞłéĞŇçİĂæL'ŃãĂĆ è£ŸèęĄæşÍæĎŘçŽĎæŸŕåIJÍCäÿ■æ§ŘäžŻçśżåđŃçŽĎéŤŹèŕŕåŘŕèČ¡äÿ■åď'łăźæŸŞæĄćåď'■ãĂĆ ä¿ŃåęĆïijŇåęĆæđIJäÿĂäÿłCæL'ľåśŢäÿćåijČäžĘçÍŃåžŘåăĘæăĹä£ąæĄŕïijŇăőČäijŽèőľ'faulthandleräÿ■åŘŕçŤ éĆčäźĹä¡ăäź§å¿Ůäÿ■åĹřäżżä¡Ţè¿ŞåĞžïijĹéŹď'äžĘçÍŃåžŘåěŤæžČåď'ŰïijL'ãĂĆ

18 éŹĎå¡ŢA

18.1 åIJÍçž£èţĎæžŘ

http://docs.python.org

åęĆæđIJä¡ăéIJĂèęĄæůśåĚěäžĘèğčæŐćçľ'ůèŕ■èÍĂåŠŇæÍąåİŮçŽĎçżĘèŁĆïijŇéĆčäźĹäÿ■å£Ěèŕť'ïijŇPyth 3 çŽĎæŰĞæąčèĂŇäÿ■æŸŕäžěåL'■çŽĎèĂĄçL'ĹæIJň

http://www.python.org/dev/peps

åęĆæđIJä¡ăåŘŚçŘĘèğčäÿžpythonèŕ■èÍĂæůżåŁăæŰřçL'źæĂğçŽĎåŁÍæIJžäžěåŘŁåőđçŐřçŽĎçżĘèŁĆïijŇ Enhancement ProposalsâĂŤ-PythonåijĂåŘŚçijŰçăĄèğĎèŇĆïijL'çżÍåŕźæŸŕéÍđåÿÿåőÍèť'ţçŽĎèţĎæžŘãĂĆåŕď'

http://pyvideo.org

è£ŹéĞŇæIJL'æİěèĞłæIJĂè£ŚçŽĎPyConåď'ğäijŽãĂAçŤÍæĹůçżĎèğĄéİćäijŽç■L'çŽĎåď'ğéĞŘèğĘéćŚæi
3äÿ■æůżåŁăçŽĎçŽĎæŰřçL'źæĂğãĂĆ

http://code.activestate.com/recipes/langs/python

éŢ£æIJ§äżěæİěïijŇActiveStateçŽĎPythonçL'ĹåİŮåůšçżŘæĹŘäÿžäÿĂäÿłæL'¿åĹřæŢřäżěå■Čèőąç ŽĎéŠĹ

http://stackoverflow.com/questions/tagged/python

Stack Overflow çŻőåL'■æIJL'èűĚè£Ğ175,000äÿłéŮőéćŸèćńæăĞèőřäÿžPythonçŻÿåĚşïijĹèĂŇåĚűäÿ■åď
3çŽĎïijL'ãĂĆåř¡çőąéŮőéćŸåŠŇåŻđç■ŤçŽĎèť'ĺéĞŘäÿ■åŘŇïijŇä¡ĘæŸřäż■çĎűèČ¡åŘŚçŐřå¿Ĺåď'Žåě¡äijŸçğ

18.2 Pythonå■ęäźžäźęçś■

äÿŃéİćè£ŹäžŻäźęçś■æŘŘä¿ŻäžĘåŕźPythonçijŰçĺŃçŽĎåĚěéŮĺäżŃçż■ïijŇäÿŤéĞ■çĆźæŤ¿åIJĺäžĘPytho
3äÿŁãĂĆ

- *Learning Python*ïijŇçňňåŻŻçL'Ĺ ïijŇä¡IJèĂĚ Mark LutzïijŇ OâĂŹReilly & Associates åĞžçL'Ĺ (2009)ãĂĆ
- *The Quick Python Book*ïijŇä¡IJèĂĚ Vernon Cederïij Ň Manning åĞžçL'Ĺ(2010)ãĂĆ
- *Python Programming for the Absolute Beginner*ïijŇçňňäÿL'çL'ĹïijŇä¡IJèĂĚ Michael DawsonïijŇCourse Technology PTR åĞžçL'Ĺ(2010).
- *Beginning Python: From Novice to Professional*ïijŇçňňäžŇçL'ĹïijŇ ä¡IJèĂĚ Magnus Lie HetâĂŘ landïijŇ Apress åĞžçL'Ĺ(2008).
- *Programming in Python 3*ïijŇçňňäžŇçL'ĹïijŇä¡IJèĂĚ Mark SummerfieldïijŇAddison-Wesley åĞžçL'Ĺ (2010).

18.3 éńŸçžğäźęçś■

äÿŃéİćçŽĎè£ŹäžŻäźęçś■æŘŘä¿ŻäžĘæŻť'åď'ŽéńŸçžğçŽĎèŇČåŻť'ïijŇäź§åŇĚåŘńPython
3æŰźéİćçŽĎåĘĚåőźãĂĆ

- *Programming Python*ïijŇçňňåŻŻçL'Ĺ, by Mark Lutz, OâĂŹReilly & Associates åĞžçL'Ĺ(2010).
- *Python Essential Reference*ïijŇçňňåŻŻçL'ĹïijŇä¡IJèĂĚ David Beazley, Addison-Wesley åĞžçL'Ĺ(2009).
- *Core Python Applications Programming*ïijŇçňňäÿL'çL'ĹïijŇä¡IJèĂĚ Wesley Chun, Prentice Hall åĞžçL'Ĺ(2012).
- *The Python Standard Library by Example* ïijŇ ä¡IJèĂĚ Doug HellmannïijŇAddison-Wesley åĞžçL'Ĺ(2011).
- *Python 3 Object Oriented Programming*ïijŇä¡IJèĂĚ Dusty Phillips, Packt Publishing åĞžçL'Ĺ(2010).

- *Porting to Python 3*ïijŇ ä¡IJèĂĚ Lennart RegebroïijŇCreateSpace åĞžçL'Ĺ(2011), http://python3porting.com.

19 åĚşäžŐèŕŚèĂĚ

åĚşäžŐèŕŚèĂĚ

- åğŞåŘ■ïijŽ çĘŁèČ¡
- å¿őä£ąïijŽ yidao620
- EmailïijŽ yidao620@gmail.com
- å■Žåőćïij Ž https://www.xncoding.com/
- GitHubïijŽ https://github.com/yidao620c

20 Roadmap

2014/08/10 - 2014/08/31:

```
|     githubéąźçŻőæŘ■åżžïijŇreadthedocsæŰĞæąčçŤ§æĹŘãĂĆ
|     æŢt'äÿłéąźçŻőçŽĎæąĘæđűåőŇæĹŘ
```

2014/09/01 - 2014/10/31:

```
|     åL'■4çńăç£żèŕŚåőŇæĹŘ
```

2014/11/01 - 2015/01/31:

```
|     åL'■8çńăç£żèŕŚåőŇæĹŘ
```

2015/02/01 - 2015/03/31:

```
|     åL'■9çńăç£żèŕŚåőŇæĹŘ
```

2015/04/01 - 2015/05/31:

```
|     10çńăç£żèŕŚåőŇæĹŘ
```

2015/06/01 - 2015/06/30:

```
|     11çńăç£żèŕŚåőŇæĹŘ
```

2015/07/01 - 2015/07/31:

```
|     12çńăç£żèŕŚåőŇæĹŘ
```

2015/08/01 - 2015/08/31:

```
|     13çńăç£żèŕŚåőŇæĹŘ
```

2015/09/01 - 2015/11/30:

```
|     14çńăç£żèŕŚåőŇæĹŘ
```

2015/12/01 - 2015/12/20:

```
|     15çńăç£żèŕŚåőŇæĹŘ
```

2015/12/21 - 2015/12/31:

```
|     åŕźåĚÍéČÍç£żèŕŚè£ŻèąŇæăąåŕźäÿĂæňą
```

2016/01/01 - 2016/01/10:

```
|     åŕźåd'ŰåĚňåijĂåŔŚåÿČåőŇæŢt'çL'Ĺ1.
↪0ïijŇåŇĚæŃňè¡ňæ■ćåŘŐçŽĎPDFæŰĞäżű
```