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

ãĂŁLPython CookbookãĂŃçňňäÿL'çL'Ĺ

åŕŚåÿČ 3.0.0

çĘŁèČ¡

2018 åźť 12 æIJĹ 20 æŮě

Contents

Contents:

1 Copyright

äźęåŘ■ïijŽ ãĂŁPython CookbookãĂŃ3rd Edition
ä¡IJèĂĚïijŽ David Beazley, Brian K. Jones
èŕŚèĂĚïijŽ çĘŁèČ¡
çĽĹæIJňïijŽ çňň3çĽĹ
åĞžçĽĹçď'¿ïijŽ OâĂŹReilly Media, Inc.
åĞžçĽĹæŮěæIJ§ïijŽ 2013åźť5æIJĹ08æŮě
Copyright Âl' 2013 David Beazley and Brian Jones. All rights reserved.

æŻť'åď'ŽåŔŚåÿČä£ąæĄŕèŕůåŔĆèĂČ
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 Pythonïij Ą

èŕŚèĂĚäÿĂçŻťåİŽæŇĄä¡£çŤÍ Python 3ïijŇåŻääÿžåőČäżčèąĺäžĘ Python çŽĎæIJłæİěãĂĆèŹ¡çĎűåŘŚåŘŐåĚijåőźæŸŕåőČçŽĎçąñäijďïijŇä¡ĘæŸŕè£ŹäÿłåśĂéİćè£§æŮĺ¨äijŽæŤźåŔŸçŽ èĂŇäÿŤ Python 3 çŽĎæIJłæİěéIJĂèęĄæŕŔäÿłäžžçŽĎåÿőåŁĺ¨åŠŇæŤŕæŇĄãĂĆ çŻőåĽ■åÿĆéİćäÿŁçŽĎæŢŹçÍŃäźęçś■ïijŇç¡Śäÿ ŁçŽĎæĽŃåĘŇåď ğéČĺåĹĘå§žæIJńéČ¡æŸŕ 2.x çşżåĹŮçŽĎïijŇäÿŞéŮĺå§žäžŐ 3.x çşżåĹŮçŽĎäźęçś■åřŚçŽĎåŔŕæĂIJãĂĆ

æIJĂè£ŚçIJŃåĹřäÿĂæIJñãĂŁPython CookbookãĂŃ3rd Edi-tionïijŇåőŇåĚĺå§žäžŐ Python 3ïijŇåĘŹçŽĎäź§å¿Ĺäÿ■éŤŹãĂĆ äÿžäžĘ Python 3 çŽĎæŹőåŔŁïijŇæĹŚäź§äÿ■èĞłéĞŔåŁŻïijŇæČşåĄŽçĆźäżĂäźĹäžŃæČĚãĂĆäžŐæŸŕäźŐïijŇåřśæIJL'äžĘç£ è£Źäÿ■æŸŕäÿĂéąźè¡żæİ¿çŽĎåůěä¡IJïijŇå■ť'æŸŕäÿĂäżűåĂijå¿ŮåĄŽçŽĎåůěä¡IJïijŽäÿ■äżĚæŰźä¿£äžĘåĹńä

èŕŚèĂĚäijŽåİŽæŇĄåŕźèĞłåůśæŕŔäÿĂåŔěçŽĎç£żèŕŚèť§èť čïijŇåŁŻæśĆéńŸèť'léĞŔãĂĆä¡ĘåŔŮèČ¡åŁŽ åęĆæđIJèŕŚæŰĞäÿ■æIJL'äžĂäźĹéŤŹæijŔçŽĎåIJřæŰźèŕůåď ğåőűèğĄèŕËïijŇäź§æňćè£Őåď ğåőűéŽŔæŮűæŇ yidao620@gmail.com

2.3 ä¡IJèĂĚçŽĎèŕİ

èĞłäžŐ 2008 åźť'äžěæİěïijŇPython 3 æĺłçl'žåĞžäÿŰåźűæĚćæĚćè£ŻåŇŰãĂĆPython 3 çŽĎætĄèąŇäÿĂçŻť'ècńèőd'äÿžéIJĂèęĄå¿ĹéŢ£äÿĂæőţæŮűéŮťãĂĆ äžŃåődäÿŁïijŇåĽřæĹŚåĘŹè£ŹæIJñäźęçŽĎ 2013 åźť'ïijŇçżİåď ğéČĺåĹĘçŽĎ Python çÍŃåžŔåŚŸäż■çĎűåIJÍçŤ§äžğçŐŕåćČäÿ■ä¡£çŤÍçŽĎæŸŕçL'ĹæIJñ 2 çşżåĹŮïijŇ æIJĂäÿżèęĄæŸŕåŻääÿž Python 3 äÿ■åŘŚåŘŐåĚijåőźãĂĆæŕńæŮăçŰŚéŰőïijŇåŕźäžŐåůěä¡IJåIJÍéĄŮçŢŹäżčç ä¡ĘæŸŕæŤ¿çIJijæIJłæİěïijŇä¡ăåŕśäijŽåŔŚçŐř Python 3 çżŹä¡ăåÿęæİěäÿ■äÿĂæăůçŽĎæČŁåŰIJãĂĆ

æ■čåęĆ Python 3 äžčèąĺæIJłæİěäÿĂæăůïijŇæŰřçŽĎãĂŁPython Cook-bookãĂŃçL'ĹæIJñçŻÿæŕŤè¿ČäźŃåL'■çŽĎçL'ĹæIJñæIJL'äžĘäÿĂäÿłåĚĺæŰřçŽĎæŤźåŔŸãĂĆ éęŰåĚĹïijŇäź§æŸŕæIJĂéĞ■èęĄçŽĎïijŇè£ŹæĎŘåŠşçİĂæIJñäźęæŸŕäÿĂæIJñéİđåÿÿåL'■æš£çŽĎåŔĆèĂČäź Python 3.3 çL'ĹæIJñäÿŃéİćçijŰåĘŹåŠŇætŃèŕŢçŽĎïijŇ åźűæšąæIJL'èĂČèŹŠäźŃåL'■èĂĄçL'ĹæIJñçŽĎåĚijå ä¡ĘæŸŕæĹŠäžňæIJĂçżĹçŽĎçŻőçŽĎæŸŕåĘŹäÿĂæIJñåőŇåĚĺå§žäžŐçŐřäžčåůěåĚůåŠŇèŕ■èĺĂçŽĎäźęçś■ãĂ æĹŠäžňåÿŇæIJŻæIJñäźęèČ¡åď §æŇĞåŕijäžžäžňä¡£çŤÍ Python 3 çijŰåĘŹæŰřçŽĎäžčçăĄæĹŰèĂĚå■ĞçžğäźŃåL'■çŽĎéĄŮçŢŹäžčçăĄãĂĆ

æŕńæŮăçŰŚéŮőïijŇçijŰåĘŹäÿĂæIJñè£ŹæăůçŽĎäźęçżŹçijŰè¿Śåůěä¡IJåÿęæİěäÿĂåőŽçŽĎæŇŚæĹŸãĂ Python çğŸçś■çŽĎèŕİïijŇäijŽåIJĺèŕÿåęĆ ActiveStateâĂŹs Python recipes æĹŰèĂĚ Stack Overflow çŽĎç¡ŚçńŹäÿŁæŘIJåĹřæŢřäžěå■Čè őąçŽĎæIJL'çŤĺçŽĎçğŸçś■ïijŇä¡ĘæŸŕåĚűäÿ■çżİåď ğéČĺåĹĘéČ è£ŹäžŽçğŸçś■éŹď äžĘæŸŕå§žäžŐ Python 2 çijŰåĘŹäźŃåď ŰïijŇåŔŕèČ¡è£ŸæIJL'å¿Låď Žèğčå Eşæ Ű źæąĹåIJ ïijĹæŕŤåęĆ 2.3 åŠŇ 2.4 çL'ĹæIJñïijL'ãĂĆ åŘęåď ŰïijŇåőČäžňè£ŸäijŽçżŔåÿÿä¡£çŤÍäÿĂäžŻè£ĞæŮűçŽĎæĹA

Python 3.3 éĞŇéÍćåŐżäžĘãĂĆårżæL’¿åőŇåĚĺå§žäžŐ Python 3 çŽĎçğŸçś■çIJ§çŽĎéŽ¿äÿŁåŁăéŽ¿åŢŁãĂĆ

è£ŹæIJñäźęçŽĎæL’ĂæIJL’äÿżécŸéČ¡æŸŕå§žäžŐåůšçżŔå■ŸåIJÍçŽĎäżčçăĄåŠŇæŁĂæIJfŕijŇèĂŇäÿ■æ
Python 3 çL’źæIJL’çŽĎçğŸçś■ãĂĆ åIJÍåŐ§æIJL’äžčçăĄå§žçąĂäÿŁïijŇæĹŚäżňåőŇåĚĺä¡£çŤÍæIJĂæŰřçŽĎ
Python æŁĂæIJŕåŐżæŤźéĂãăĂĆ æL’Ăäžěïij Ňäżżä¡ŢæČşä¡£çŤÍæIJĂæŰřæŁĂæIJŕçijŰåĘŹäżčçăĄçŽĎçÍŃåž

åIJÍéĂL’æŃl’èęĄåŇĚåŔńåŞłäžŽçğŸçś■æŰźéİćïijŇå¿ĹæŸŐæŸ¿äÿ■åŔŕèČ¡çijŰåĘŹäÿĂæIJñäźęåŻŁæŃ
Python écĘå§§æL’ĂæIJL’çŽĎäÿIJèě£ãĂĆ åŻăæ■d’ïijŇæĹŚäżňäijŸåĚĹéĂL’æŃl’äžĘ Python
èŕ■èÍĂæäÿå£ČéČÍåĹĘïijŇäżěåŔŁéĆčäžŻæIJL’çİĂåź£æşŻåžŢçŤÍéćĘå§§çŽĎéŰőécŸãĂĆ
åŔęåd’ŰïijŇåĚűäÿ■æIJL’å¿Ĺåd’ŽçğŸçś■çŤÍæİěåśŢçd’ž Python 3 çŽĎæŰřçL’źæĂğïijŇ
è£Źåŕźäžőå¿Ĺåd’Žäžžæİěèŕť’æŸŕæŕŤè¿ČéŹŇçŤ§çŽĎïijŇåŞłæĂŢæŸŕä¡£çŤÍ
Python èĂĄçL’ĹæIJňçŽĎçżŔélŇäÿřárŇçŽĎçÍŃåžŔåŚŸãĂĆ
è£ŹäžŻçd’žä¿ŃçÍŃåžŔäź§äijŽåĄŔåŘäžŐåśŢçd’žäÿĂäžŻæIJL’çİĂåź£æşŻåžŢçŤÍçŽĎçijŰçÍŃæŁĂæIJŕ
ïijĹå■şçijŰçÍŃæÍąåijŔïijL’ïijŇ èĂŇäÿ■æŸŕäžĚäžĚåőŻä¡■åIJÍäÿĂäžŻåĚůä¡ŞçŽĎéŰőécŸäÿŁãĂĆår¡çőąäź§æ
Python èŕ■èÍĂæäÿå£ČåŠŇæăĞåĞĘåžŞãĂĆ

2.4 è£ŹælJňäźęéĂĆåŘĹèřĄ

è£ŹæIJñäźęçŽĎçŻőæăĞèŕżèĂĚæŸŕéĆčäžŻæČşæůśåĚěçŘĘèğč Python
èŕ■èÍĂæIJžåĹűåŠŇçŐřäżčçijŰçÍŃéčŐæăijçŽĎæIJL’çżŔélŇçŽĎ Python çÍŃåžŔåŚŸãĂĆ
æIJñäźęåd’ğéČÍåĹĘåĘĚåőźéŻĘäÿ■äžŐåIJÍæăĞåĞĘåžŞïijŇæąĘæđűåŠŇåžŢçŤÍçÍŃåžŔäÿ■åź£æşŻä¡£çŤÍçŽĎe
æIJñäźęæL’ĂæIJL’çd’žä¿ŃåİĞåĄĞèő¿èŕżèĂĚåĚůæIJL’äÿĂåőŽçŽĎçijŰçÍŃèČŇæŹŕåźűäÿŤåŘŕäżěèŕżæĞĆçŻ
ïijĹæŕŤåęĆå§žæIJñçŽĎèőąçőŮæIJžçğŚå■ęç§ěèŕĘïijŇæŢřæ■őçżŞædĎç§ěèŕĘïijŇçőŮæşŢåd’■æİĆåžęïijŇçşżç
èŕ■èÍĂçijŰçÍŃç■L’ïijL’ãĂĆ åŔęåd’ŰïijŇæŕŔäÿłçd’žä¿ŃéČ¡åŘłæŸŕäÿĂäÿłåĚěéŮÍæŇĞåŕijïijŇåęĆædIJèŕżèĂĚ
æĹŚäżňåĄĞåőŽèŕżèĂĚåŘŕäżěå¿ĹçĘ§çżČçŽĎä¡£çŤÍæŘIJçt’ćåijŢæŞŐäżěåŘŁç§ěéĄŞæĂŐæăůæ§ěèŕćåIJÍçž£
Python æŰĞæąčĂĆ

æIJL’äÿĂäžŻæŽt’åŁăéńŸçžğçŽĎçğŸçś■ïijŇåęĆædIJèĂŘå£ČéŸĚèŕżïijŇårĘæIJL’åŁl’äžŐçŘĘèğč
Python åžŢåśĆçŽĎåůěä¡IJåŐ§çŘĘãĂĆ äžŐäÿ■ä¡ăårĘå■ęåĹřäÿĂäžŻæŰřçŽĎæŁĂåůğåŠŇæŁĂæIJfŕijŇåźűåž

2.5 è£ŹælJňäźęäÿ■éĂĆåŘĹèřĄ

è£ŹæIJñäźęäÿ■éĂĆåŘĹ Python çŽĎåĹİå■ęèĂĚãĂĆäžŃåőđäÿŁïijŇæIJñäźęåĄĞåőŽèŕżèĂĚåĚůæIJL’
Python æŢŹçÍŃæĹŮåĚěéŮÍäźęçś■äÿ■æL’ĂæŢŹæŐĹçŽĎå§žçąĂç§ěèŕĘãĂĆ
æIJñäźęäź§äÿ■æŸŕéČčçğ■å£ńéĂ§åŔĆèĂČæL’ŃåĘŇ ïijĹä¿ŃåęĆå£ńéĂ§æ§ěèŕćæ§ŘäÿłæĺąåİŮäÿŇçŽĎæ§Řä
æIJñäźęæŮĺåIJÍèĄŽçĎęåĞääÿłæIJĂéĞ■èęĄçŽĎäÿżécŸïijŇæijŤçd’žåĞăçğ■åŔŕèČ¡çŽĎèğčåĘşæŰźæąĹïijŇ
æŔŘä¿Żäÿääÿłèůşæİ£åijŢåŕijèŕżèĂĚè£ŻåĚěäÿĂäžŻæŻt’éńŸçžğçŽĎåĘĚåőźïijĹè£ŹäžŻåŔŕäżěåIJÍç¡Śäÿ ŁæĹ

2.6 åIJÍçž£çd’žä¿Ńäżčçă Ą

æIJñäźęåĞääźŐæL’ĂæIJL’æžŘäżčçăĄåİĞåŔŕäżěåIJÍ http://github.com/dabeaz/python-cookbook äÿŁéİćæL’¿åĹřãĂĆ ä¡IJèĂĚæńćè£ŐåŘĎä¡■èŕżèĂĚä£őæ■č bugïijŇæŤźè£Żäżčçă ĄåŠŇèŕĎèőžãĂĆ

2.7 ä¡£çŤÍçď'žä¿ŃäžčçăĄ

æIJňäźęářśæŸŕåÿőåŁľ'ä¡ăåőŇæĹŘä¡ăçŽĎåůěä¡IJçŽĎãĂĆ
äÿĂèĹňæİěèőšïijŇåŔłèęĄæŸŕæIJňäźęäÿŁéİćçŽĎçď'žä¿ŃäžčçăĄïijŇä¡ăéČ¡åŔŕäžěéŽŔæŮűæŃ£è£ĞåŐžåIJĺä
éŹď'éİđä¡ăä¡£çŤÍäžĘåď'ğéĞŔçŽĎäžčçăĄïijŇåŘęåĹŹäÿ■éIJĂèęĄåŘŚæĹŚäžňçŤşèřůèőÿåŔŕãĂĆ
ä¿ŃåęĆïijŇä¡£çŤÍåĞăäÿłäžčçăĄçL'Ğæőţå0žåőŇæĹŘäÿĂäÿłçÍŃåžŔäÿ■éIJĂèęĄèőÿåŔŕïijŇèt'l'å■ŰæĹŰèĂĚ
åijŢçŤÍæIJňäźęåŠŇçď'žä¿ŃäžčçăĄåŐžç¡ŚäÿŁåŻđç■ŤäÿĂäÿłéŮőécŸäÿ■éIJĂèęĄèőÿåŔŕïijŇä¡ĘæŸŕåŘĹåźűå

æĹŚäžňäÿ■äijŽèęĄæśĆä¡ăæůżåŁăäžčçăĄçŽĎåĞžåď'ĎïijŇä¡ĘæŸŕåęĆæđIJä¡ăè£ŹäźĹåĄŽäžĘïijŇæĹŚäž
åijŢçŤÍéĂŽåÿÿåŇĚåŘńæăĞécŸïijŇä¡IJèĂĚïijŇåĞžçL'Ĺçď¿ïijŇISBNãĂĆ ä¿ŃåęĆïijŽ*Python Cookbook*, 3rd edition, by David Beazley and Brian K. Jones (OâĂŹReilly). Copyright 2013 David Beazley and Brian Jones, 978-1-449-34037-7.

åęĆæđIJä¡ăèğL'å¿Ůä¡ăářźçď'žä¿ŃäžčçăĄçŽĎä¡£çŤÍèűĚåĞžäžĘåŘĹçŘEä¡£çŤÍæĹŰèĂĚäÿŁè£řåĹŮåĞžç
èřůéŽŔæŮűèĄŢçşżæĹŚäžňïijŇæĹŚäžňçŽĎéĆőçőśæŸŕ permissions@oreilly.comãĂĆ

2.8 èĄŢçşżæĹŚäžň

èřůářĘåĚşäžŐæIJňäźęçŽĎèŕĎèőžåŠŇéŮőécŸåŔŚéĂĄçżŹåĞžçL'Ĺçď¿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)

æĹŚäžňäÿžæIJňäźęåžžçńŃäžĘäÿĂäÿłç¡ŚéąţïijŇ åĚűäÿ■åŇĚåŘńåŃŸèŕŕèąĺïijŇçď'žä¿ŃåŠŇäÿĂäžŻåĚűäž
åŘŕäžěéĂŽè£ĞéŞ¿æŐě http://oreil.ly/python_cookbook_3e è£éŮőãĂĆ

åĚşäžŐæIJňäźęçŽĎåžžèőőåŠŇæŁĂæIJŕæĂğéŮőécŸïijŇèřůåŘŚéĂĄéĆőäžűèĞşïijŽ bookquestions@oreilly.com

åĚşäžŐæĹŚäžňçŽĎäźęçś■ïijŇèőĺèőžäijŽïijŇæŰřéŮżçŽĎæŻť'åď'Žä£ąæĄŕïijŇ èřůèő£éŮőæĹŚäžňçŽĎç¡ŚçńŹïijŽ http://www.oreilly.com

åIJÍ Facebook äÿŁæL'¿åĹřæĹŚäžňïijŽhttp://facebook.com/oreilly

åIJÍ Twitter äÿŁåĚşæşĺæĹŚäžňïijŽhttp://twitter.com/oreillymedia

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

2.9 èĞťèřć

æĹŚäžňèąůå£ČæĎ§èřćæIJňäźęçŽĎæŁĂæIJŕæăąåőąäžžåŚŸ Jake Vanderplasïij ÑRobert Kern åŠÑ Andrea Crotti éİđåÿÿæIJL’çŤĺçŽĎèřĎèőžåŠÑåžžèőőïijÑ è£ŸæIJL’ Python çd’¿åŇžçŽĎåÿőåŁl’åŠÑéijŞåŁśãĂĆæĹŚäžňåŘŇæăůæĎ§èřćäÿŁäÿĂäÿłçL’ĹæIJňçŽĎçijŰè¿Ś Alex MartelliïijÑAnna Ravenscroft åŠÑ David AscherãĂĆ åř¡çőąè£Źäÿłç L’ĹæIJňæŸŕæŰřåĹŻä¡IJçŽĎïijÑä¡ĘæŸŕåL’■äÿĂäÿłçL’ĹæIJňäÿžæIJňäźęæŔŘä¿ŻäžĘäÿĂäÿłæÑ æIJĂåŘŐäź§æŸŕæIJĂéĞ■èęĄçŽĎïijÑæĹŚäžňèęĄæĎ§èřćæL’ĂæIJL’æŮl’æIJ§écĎèğĹçL’ĹæIJňçŽĎèŕżèĂĚïij

3 çňňäÿĂçńăïijŽæŢřæ■őçżŞæđĎåŠÑçőŮæşŢ

Python æŔŘä¿ŻäžĘåd’ğéĞŔçŽĎåĘĚç¡őæŢřæ■őçżŞæđĎïijÑåÑĚæŃňåĹŮèąĺïijÑéŻĘåŘĹäžěåŔŁå■ŮåĚ ä¡ĘæŸŕïijÑæĹŚäžňäź§äijŽçżŔåÿÿçćřåĹřåĹřěŕÿåęĆæ§ěèŕćïijÑæŐŠåžŔåŠÑè£Ğæżd’ç■L’ç■L’è£ŹäžŻæŹőéĄ■ åŽăæ■d’ïijÑè£ŹäÿĂçńăçŽĎçŻőçŽĎářśæŸŕèőĺèőžè£ŹäžŻæŕŤè¿Čåÿÿè ğĄçŽĎéŮőécŸåŠÑçőŮæşŢãĂĆ åŔęåd’ŰïijÑæĹŚäžňäź§äijŽçżŹåĞžåIJĺéŻĘåŘĹæĺąåİŮ `collections` å¡Şäÿ■æŞ■ä¡IJè£ŹäžŻæŢřæ■őçżŞæđĎçŽĎæŰźæşŢãĂĆ

3.1 1.1 èğčåŐŃåžŔåĹŮèţŃåĂijçżŹåd’ŽäÿłåŔŸéĞŔ

éŮőécŸ

çŐřåIJĺæIJL’äÿĂäÿłåŇĚåŘń N äÿłåĚČçt’ăçŽĎåĚČçżĎæĹŰèĂĚæŸŕåžŔåĹŮïijÑæĂŐæăůårĘåőČéĞŃéİć N äÿłåŔŸéĞŔïij§

èğčåĘşæŰźæąĹ

äžża¡ŢçŽĎåžŔåĹŮïijĹæĹŰèĂĚæŸŕåŔŕè£■äžčårźèśąïijL’åŔŕäžěéĂŽè£ĞäÿĂäÿłçőĂå■ŢçŽĎèţŃåĂijèŕ■å åŤŕäÿĂçŽĎåL’■æŘŘårśæŸŕåŔŸéĞŔçŽĎæŢřéĞŔå£Ëéążèů§åžŔåĹŮåĚČçt’ăçŽĎæŢřéĞŔæŸŕäÿĂæăůçŽĎãĂ

äžčçăĄçd’ža¿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)

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

åęĆædIJåŔŸéĞŔäýłæŢřåŠŇåžŔåĹŮåĚČçť ăçŽĎäýłæŢřäý■åŇźéĚ■ïijŇäijŽäžğçŤ§äÿĂäýłåijĆåÿÿãĂĆ

äžčçăĄçď 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

åőđéŹĚäýŁïijŇè£Źçğ■èğčåŐŃètŃåĂijåŔŕäžěçŤĺåIJÍäżża¡ŢåŔŕè£■äžčåŕźèśąäýŁéİćïijŇèĂŇäý■äżĚäżĚæåŇĚæŃňå■ŮçňęäÿšïijŇæŰĞäżűåŕźèśąïijŇè£■äžčåŹĺåŠŇçŤ§æĹŘåŹĺãĂĆ

äžčçăĄçď žä¿ŃïijŽ

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

æIJL'æŮűåĂŹïijŇä¡ăåŔŕèČ¡åŔłæČşèğčåŐŃäÿĂéČĺåĹĘïijŇäýćåijČåĚűäžŰçŽĎåĂijãĂĆåŕźäžŐè£Źçğ■a Python åźűæšąæIJL'æŔŘä¿ŻçL'źæőŁçŽĎèí■æşŢãĂĆ ä¡ĘæŸŕä¡ăåŔŕäžěä¡£çŤĺäżżæĎŔåŘŸéĞŔåŘ■åŐżå■ää¡

äžčçăĄçď žä¿ŃïijŽ

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

(continues on next page)

```
91.1
>>>
```

ä¡ăå£Ëéąžä£ÌèŕĄä¡ăéĂL’çŤĺçŽĎéĆčäžŻå■ăä¡■åŔŸéĞŔåŘ■åIJĺåĚűäżŮåIJřæŰźæšąèćńä¡£çŤĺåĹřãĂĆ

3.2 1.2 èğčåŐŃåŔŕè£■äžčåŕźèśąèţŃåĂijçżŹåď’ŽäÿłåŔŸéĞŔ

éŮőécŸ

åęĆæđIJäÿĂäÿłåŔŕè£■äžčåŕźèśąçŽĎåĚČçţ’ăäÿłæŢřèűĚè£ĞåŔŸéĞŔäÿłæŢřæŮűïijŇäijŽæŁŻåĞžäÿĂäÿł
`ValueError` ãĂĆ éĆčäźĹæĂŐæăůæL’■èČ¡äżŐè£ŹäÿłåŔŕè£■äžčåŕźèśąäÿ■èğčåŐŃåĞž N
äÿłåĚČçţ’ăåĞžæÌěïij§

èğčåĘşæŰźæąĹ

Python çŽĎæŸ§åŔůèąĺè¿¿åijŔåŔŕäžěçŤĺæÌěèğčåĘşè£ŹäÿłéŮőécŸãĂĆæŕŤåęĆïijŇä¡ăåIJĺå■ęäźăäÿĂéŮĺ
ä¡ăæČşçż§èőąäÿŃåőűåž■ä¡IJäÿŽçŽĎåźşåİĞæĹŘçżl’ïijŇä¡ĘæŸŕæŐŠéŹď’æŐL’çňňäÿĂäÿłåŠŇæIJĂåŘŐäÿĂä
ä¡ĘåęĆæđIJæIJL’ 24 äÿłåŚćïij§è£ŹæŮűåĂŹæŸ§åŔůèąĺè¿¿åijŔåřśæť¿äÿŁçŤĺåIJžäžĘïijŽ

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

åŘęåď’ŰäÿĂçğ■æČĚåĘţïijŇåĄĞèő¿ä¡ăçŐřåIJĺæIJL’äÿĂäžŻçŤĺæĹůçŽĎèőřå¡ŢåĹŮèąĺïijŇæŕŘæİąèőřå¡Ţ
ä¡ăåŘŕäžěåČŔäÿŃéİćè£ŹæăůåĹĘèğčè£ŹäžŻèőřå¡Ţïij Ž

```
>>> 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¿ŮæşĺæĎŘçŽĎæŸŕäÿŁéİćèğčåŐŃåĞžçŽĎ `phone_numbers`
åŘŸéĞŔæřÿè£IJéČ¡æŸŕåĹŮèąĺçśžåđŃïijŇäÿ■çőąèğčåŐŃçŽĎçŢţèŕİåŘůçăĄæŢřéĞŔæŸŕåď’ŽåřŚïijĹåŃĚæŃ
0 äÿłïijL’ãĂĆ æL’Ääžěïij Ňäżżä¡Ţä¡£çŤĺåĹř `phone_numbers`
åŘŸéĞŔçŽĎäžčçăĄåŕśäÿ■éIJĂèęĄåĄŽåď’Žä¡ŹçŽĎçśżåđŃæčĂæ§ěåŐżçąőèőď’åőČæŸŕåŘęæŸŕåĹŮèąĺçśžåd

æŸ§åŔůèąĺè¿¿åijŔäź§èČ¡çŤĺåIJĺåĹŮèąĺçŽĎåijĂåğŃéČĺåĹĘãĂĆæŕŤåęĆïijŇä¡ăæIJL’äÿĂäÿłåĚňåŘÿåL’■
8 äÿłæIJĹéŤĂåŤőæŢřæ■őçŽĎåžŘåĹŮïijŇ ä¡ĘæŸŕä¡ăæČşçIJŃäÿŃæIJĂè£ŚäÿĂäÿłæIJĹæŢřæ■őåŠŇåL’■éİć
7 äÿłæIJĹçŽĎåźşåİĞåĂijçŽĎåŕźæŕŤãĂĆä¡ååŔŕäžěè£ŹæăůåĄŽïijŽ

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

äÿŃéÍćæŸŕåIJÍ Python èğčéĞŁåŹĺäÿ■æĽğèąŇçŽĎçżŞæđIJïijŽ

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

èőÍèőž

æĽľåśŢçŽĎè£■äžčèğčåŐŃèŕ■æşŢæŸŕäÿŞéŮĺäÿžèğčåŐŃäÿ■çąőåőŽäÿłæŢřæĹŰäżżæĎŔäÿłæŢřåĚČçť
éĂŽåÿÿïijŇè£ŹäžŽåŔŕè£■äžčåŕźèśąçŽĎåĚČçť ăçżŞæđĎæIJĽçąőåőŽçŽĎèğĎåĹŹïijĹæŕŤåęĆçňň
1 äÿłåĚČçť ăåŘŐéÍćéČ¡æŸŕçŢţèŕÍåŔůçăĄïijĽïijŇ æŸ§åŔůèąĺè¿¿åijŔèőľåijĂåŔŠäžžåŠŸåŔŕäżěå¿ĹåőźæŸ§ç
èĂŇäÿ■æŸŕéĂŽè£ĞäÿĂäžŽæŕŤè¿Čåď■æÍĆçŽĎæĽŃæőţåŐżèŐůåŔŰè£ŹäžŽåĚşèĄŢçŽĎåĚČçť ăåĂijãĂĆ

åĂijå¿ŮæşĺæĎŔçŽĎæŸŕïijŇæŸ§åŔůèąĺè¿¿åijŔåIJÍè£■äžčåĚČçť ăäÿžåŔŕåŔŸéŢ£åĚČçżĎçŽĎåžŔåĹŮæ
æŕŤåęĆïijŇäÿŃéÍćæŸŕäÿĂäÿłåÿęæIJĽæăĞç■¿çŽĎåĚČçżĎåžŔåĹŮïijŽ

```
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§åŔůèğčåŐŃèŕ■æşŢåIJÍå■ŮçňęäÿšæŞ■ä¡IJçŽĎæŮűåĂŹäź§äijŽå¿ĹæIJĽçŤĺïijŇæŕŤåęĆå■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)

(çż■äÿŁéąţ)

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

æIJL'æŮűåĂŹïijŇä¡ăæČşèğčåŐŃäÿĂäžŻåĚČçť'ääŘŐäÿćåijČåőČäżñïijŇä¡ăääÿ■èČ¡çőĂå■Ţåřśä¡£çŤÍ * ïijŇ ä¡ĘæŸŕä¡ääŔŕäżěä¡£çŤĺäÿĂäÿłæŹőéĂŽçŽĎåž§åijČåŘ■çğřïijŇæŕŤåęĆ _ æĹŰèĂĚ ign ïijĹignoreïijL'ãĂĆ

äžčçăĄçď'žä¿ŃïijŽ

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

åIJĺå¿Ĺåď'ŽåĞ¡æŢřåijŔèŕ■èĺĂäÿ■ïijŇæŸ§åŔůèğčåŐŃèŕ■æşŢèů§åĹŮèąĺåď'ĎçŘĘæIJL'èőÿåď'ŽçŻÿäijijä ä¡ääŔŕäżěå¿Ĺåőźæ ŸŞçŽĎårĘåőČåĹĘåL'šæĹŘåL'■åŘŐäÿď'éČĺåĹĘïijŽ

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

åęĆæđIJä¡ääď'§èĄłæŸŐçŽĎèŕÏïijŇè£ŸèČ¡çŤĺè£Źçğ■åĹĘåL'šèŕ■æşŢåŐżåůğåęŹçŽĎåőđçŐřéĂŠå¡Šçő

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

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

3.3 1.3 ä£ĺçŢŹælJĂåŘŐ N äÿłåĚČçť'ă

éŮőécŸ

åIJĺè£■äžčæŞ■ä¡IJæĹŰèĂĚåĚűäżŰæŞ■ä¡IJçŽĎæŮűåĂŹïijŇæĂŐæăůåŔłä£ĺçŢŹæIJĂåŘŐæIJL'éŹŘåĞă

èğčåĘşæŰźæąĹ

ä£ÍçŢŹæIJL'éŹŘåŐĘåŔšèőřå¡Ţæ■čæŸŕ `collections.deque` åď'ğæŸ¿èžńæĽŃçŽĎæŮűåĂŹãĂĆærŤåęĆïijŇäÿŃéİćçŽĎäžčçăĄåIJÍåď'ŽèąŇäÿŁéİćåĄŽçőĂå■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¿¿åijŔçŽĎçŤ§æĹŘåŹÍåĞ¡æŢřïijŇäź§åřśæŸŕæĹŚäžňäÿŁéİćçď'žä¿ŃäžčçăĄäÿ■çŽĎéĆčæåůãĂĆ è£ŹæăůåŘŕäžěåřĘæŘIJçť'ćè£ĞçÍŃäžčçăĄåŠŇä¡£çŤÍæŘIJçť'ćçżŞæđIJäžčçăĄèğčèĂęãĂĆåęĆæđIJä¡ăè£Ÿäÿ■a 4.3 èŁĆãĂĆ

ä¡£çŤÍ `deque(maxlen=N)` æđĎéĂååĞ¡æŢřäijŽæŰřåžžäÿĂäÿłåŽžåőŽåď'ğåřŔçŽĎéŸ§åĹŮãĂĆå¡ŞæŰ æIJĂèĂĄçŽĎåĚČçť'ăäijŽèĞłåŁÍèćńçğžéŹď'æŐĽ'ãĂĆ

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§åŔŕäžěæĽ'ŃåŁĺåIJĺäÿĂäÿłåĹÙèąĺäÿŁåðđçŐřè£ŹäÿĂçŽĎæŞ■ä¡IJïijĹæŕŤåęĆåćđåŁããĂąåĹ

æŻť'äÿĂèĹňçŽĎïijŇ deque çśžåŔŕäžěèćńçŤĺåIJĺäżżä¡Ţä¡ääŔłéIJĂèęĄäÿĂäÿłçőĂå■ŢéŸ§åĹÙæŢřæ■őç
åęĆæđIJä¡ääÿ■èő¿ç¡őæIJĂåď'géŸ§åĹÙåď'gářŔïijŇéĆčäźĹářšäijŽå¿ÙåĹřäÿĂäÿłæŮăéŹŘåď'gářŔéŸ§åĹÙïijŇ

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§åĹÙäÿď'çńŕæŔŠåĚěæĹŰåĹăéŹď'åĚČçţ'ăæŮűéŮţ'åď'■æİĆåžęéČ¡æŸŕ O(1) ïijŇåŇžåĹńäžŐåĹÙèąĺïijŇåIJĺåĹÙèąĺçŽĎåijĂåď'ţ'æŔŠåĚěæĹŰåĹăéŹď'åĚČçţ'ăçŽĎæŮűéŮţ'åď'■æİĆåžęäÿž O(N) ãĂĆ

3.4 1.4 æ§ěæĽ¿ælJĂåď'gæĹŰælJĂářŔçŽĎ N äÿłåĚČçť'ă

éŮőécŸ

æĂŐæăůäžŐäÿĂäÿłéŻĘåŘĹäÿ■èŐůå¿ÙæIJĂåď'gæĹŰèĂĚæIJĂářŔçŽĎ N äÿłåĚČçť'ăåĹÙèąĺïij§

èğčåĘşæŰźæąĹ

heapq æĺąåİŮæIJL'äÿď'äÿłåĞ¡æŢřïijŽnlargest() åŠŇ nsmallest() åŔŕäžěåőŇç¿Őèğčåĕşè£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¡æŐěåŔŮäÿĂäÿłåĚşéŤőå■ŮåŔĆæŢřïijŇçŤĺäžŐæŻť'åď'■æİĆçŽĎæŢřæ■őçżŞædĎä

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

(continues on next 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'])
```

èŕŚèĂĚæşĺïijŽäÿŁéİćäžčçăĄåIJĺåŕźæŕŔäÿłåĚČçť'äè£ŻèąŇåŕźæŕŤçŽĎæŮűåĂŹïijŇäijŽäžě `price` çŽĎåĂijè£ŻèąŇæŕŤè¿ČãĂĆ

èőĺèőž

åęĆæđIJä¡ăæČşåIJĺäÿĂäÿłéŻĘåŘĹäÿ■æ§ěæL'¿æIJĂåřŔæĹŰæIJĂåď'ğçŽĎ N äÿłåĚČçť'ăïijŇåźűäÿŤ N åřŔäžŐéŻĘåŘĹåĚČçť'äæŢřéĞŔïijŇéĆčäźĹè£ŹäžŽåĞ¡æŢřæŔŘä¿ŻäžĘå¿Ĺåě¡çŽĎæ åŽääÿžåIJĺåžŢåśĆåőđçŐřéĞŇéİćïijŇéęŮåĚĹäijŽåĚĹåřĘéŻĘåŘĹæŢřæ■őè£ŻèąŇåäĘæŐŠåžŔåŘŐæŤ¿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]
>>>
```

åăĘæŢřæ■őçżŞædĎæIJĂéĞ■èęĄçŽĎçL'źå¿ĄæŸŕ `heap[0]` æřÿè£IJæŸŕæIJĂåřŔçŽĎåĚČçť'ăãĂĆåźűäÿŤåL'ľä¡ŹçŽĎåĚČçť'ääŔŕäžěå¿ĹåőźæŸ$çŽĎéĂŽè£ĞèřČçŤĺ `heapq.heappop()` æŰźæşŢå¿ŮåĹřïijŇ èŕěæŰźæşŢäijŽåĚĹåřĘçňňäÿĂäÿłåĚČçť'ăåijźåĞžæİěïijŇçĎűåŘŐ O(log N)ïijŇN æŸŕääĘåď'ğåřŔïijL'ãĂĆ æŕŤåęĆïijŇåęĆædIJæČşèęĄæ§ěæL'¿æIJĂåřŔçŽĎ 3 äÿłåĚČçť'ăïijŇä¡ääŔŕäžěè£ŹæăůåĄŽïijŽ

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

å¡ŞèęĄæ§ěæL'¿çŽĎåĚČçť'ääÿłæŢřçŻÿåŕźæŕŤè¿ČåřŔçŽĎæŮűåĂŹïijŇåĞ¡æŢř `nlargest()` åŠŇ `nsmallest()` æŸŕå¿ĹåŘĹéĂĆçŽĎãĂĆ åęĆædIJä¡ääžĚäžĚæČşæ§ěæL'¿åŤŕäÿĂçŽĎæIJĂåřŔæĹŰæIJĂåď'ğïijĹN=1ïijL'çŽĎåĚČçť'ăçŽĎèŕİïijŇéĆčäźĹ `min()` åŠŇ `max()` åĞ¡æŢřäijŽæŻť'å£ńäžŽãĂĆ çśžäijijçŽĎïijŇåęĆædIJ N çŽĎåď'ğåřŔåŠŇéŻĘåŘĹåď'ğåřŔæŐèè£ŚçŽĎæŮűåĂŹïijŇéĂŽåÿÿåĚĹæŐŠåžŔè£ŹäÿłéŻĘåŘĹçĎűåŘŐåĘ■ä¡ ïijĹ `sorted(items)[:N]` æĹŰèĂĚæŸŕ `sorted(items)[-N:]` ïijL'ãĂĆ éIJĂèęĄåIJĺæ■čçąőåIJžåŘĹä¡£çŤĺåĞ¡æŢř `nlargest()` åŠŇ `nsmallest()` æL'■èČ¡åŔŚæŇěåőČäžňçŽĎäijŸåŁ£ ïijĹåęĆædIJ N å£ńæŐèè£ŚéŻĘåŘĹåď'ğåřŔäžĘïijŇéĆčäźĹä¡£çŤĺæŐŠåžŔæŞ■ä¡IJäijŽæŻť'åě¡äžŻïijL'ãĂĆ

åř¡çőąä¡ăæšąæIJL’å£ĚèęĄäÿĂåőŽä¡£çŤÍè£ŹéĞŇçŽĎæŰźæşŢïijŇä¡ĘæŸŕåăĘæŢřæ■őçżŞæđĎçŽĎåőđçČ
å§žæIJňäÿŁåŔłèęĄæŸŕæŢřæ■őçżŞæđĎåŠŇçőŮæşŢäźęçś■éĞŇéİćéČ¡äijŽæIJL’æŔŘåŔŁåĹřãĂĆ
heapq æÍąåİŮçŽĎåőŸæŰźæŰĞæąčéĞŇéİćäź§èŕęçżĘçŽĎäżŇçż■äžĘåăĘæŢřæ■őçżŞæđĎåžŢåśĆçŽĎåőđçČ

3.5 1.5 åőđçŐřäÿĂäÿłäijŸåĚĹçžğéŸ§åĹŮ

éŮőécŸ

æĂŐæăůåőđçŐřäÿĂäÿłæŇL’äijŸåĚĹçžğæŐŠåžŔçŽĎéŸ§åĹŮïij§
åźűäÿŤåIJÍè£ŹäÿłéŸ§åĹŮäÿŁéİćæŕŔæňą pop æŞ■ä¡IJæĂżæŸŕè£ŤåŽđäijŸåĚĹçžğæIJĂéńŸçŽĎéĆčäÿłåĚČç

èğčåĘşæŰźæąĹ

äÿŃéİćçŽĎçśżåĹ’çŤÍ heapq æÍąåİŮåőđçŐřäžĘäÿĂäÿłçőĂå■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]
```

äÿŃéİćæŸŕåőČçŽĎä¡£çŤÍæŰźåijŔïijŽ

```
>>> 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)

(çż■äÿŁéąţ)

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

äżŤçżĘèğĆåŕ§åŔŕäżĕåŔŠçŐŕïijŇçňňäÿĂäÿł pop() æŞ■ä¡IJè£ŤåŽđäijŸåĚĹçžğæIJĂéńŸçŽĎåĚČçť äãĂ åŔęåďŰæşĺæĎŔåĹřåęĆæđIJäÿďäÿłæIJĽçĪĂçŻÿåŔŇäijŸåĚĹçžğçŽĎåĚČçť äïijĹ foo åŠŇ grok ïijĽïijŇpop æŞ■ä¡IJæŇĽçĚğåőČäżňèćńæŔŠåĚěåĹřéŸ§åĹŮçŽĎéąžåžŔè£ŤåŽđçŽĎãĂĆ

èőĺèőž

è£ŹäÿĂåřŔèŁĆæĹŚäżňäÿżèęĄåĚşæşĺ heapq æĺąåİŮçŽĎä¡£çŤĺãĂĆ åĞ¡æŢř heapq.heappush() åŠŇ heapq.heappop() åĹĘåĹńåIJĺéŸ§åĹŮ _queue äÿŁæŔŠåĚěåŠŇåĹăéŹď çňňäÿĂäÿłåĚČçť äïijŇ åźűäÿŤéŸ§åĹŮ _queue ä£İèŕĄçňňäÿĂäÿłåĚČçť äæŃěæIJĽæIJĂéńŸäijŸåĚĹçžğïijĽ 1.4 èŁĆåůšçżŔèőĺèőžè£Ğè£ŹäÿłéŮőéćŸïijĽãĂĆ heappop() åĞ¡æŢřæĂżæŸŕè£ŤåŽđâĂİæIJĂåřŔçŽĎâĂİçŽĎåĚČçť äïijŇè£Źåřśæŕä£İèŕĄéŸ§åĹŮpopæŞ■ä¡IJè£ŤåŽđæ åŔęåďŰïijŇçŤśäžŐ push åŠŇ pop æŞ■ä¡IJæŮűéŮť åď■æİĆåžęäÿž O(log N)ïijŇåĚűäÿ■ N æŸŕåăEçŽĎåď ğåřŔïijŇåŽăæ■ď åřśçőŮæŸŕ N å¿Ĺåď ğçŽĎæŮűåĂŹåőČäżňè£ŘèąŇéĂ§åžęäź§ä¿İæŮğå¿Ĺå£ńãĂĆ

åIJĺäÿŁéİćäżčçăĄäÿ■ïijŇéŸ§åĹŮåŇĚåŘńäžĘäÿĂäÿł (-priority, index, item) çŽĎåĚČçżĎãĂĆ äijŸåĚĹçžğäÿžèť§æŢřçŽĎçŻőçŽĎæŸŕä¡£å¿ŮåĚČçť äæŇĽçĚğäijŸåĚĹçžğäżŐéńŸ è£Źäÿłèů§æŹőéĂŽçŽĎæŇĽäijŸåĚĹçžğäżŐä¡ŐåĹřéńŸæŐŠåžŔçŽĎåăEæŐŠåžŔæAřåůğçŻÿåŔ■ãĂĆ

index åŔŸéĞŔçŽĎä¡IJçŤĺæŸŕä£İèŕĄåŘŇç■Ľ äijŸåĚĹçžğåĚČçť äçŽĎæ■čçąőæŐŠåžŔãĂĆ éĂŽè£Ğä£Ią■ŸäÿĂäÿłäÿ■æŰ■åćđåŁăçŽĎ index äÿŃæăĞåŔŸéĞŔïijŇåŔŕäżěçąőä£İåĚČçť äæŇĽçĚğåőČäż èĂŇäÿŤïijŇ index åŔŸéĞŔäź§åIJĺçŻÿåŔŇäijŸåĚĹçžğåĚČçť äæŕŤè¿ČçŽĎæŮűåĂŹèţůåĹřéĞ■èęĄä¡IJçŤĺã

äÿžäžĘéŸŘæŸŐè£ŹäžŻïijŇåĚĹåĄĞåőŽ Item åőđä¿ŃæŸŕäÿ■æŤřæŇĄæŐŠåžŔçŽĎïijŽ

```
>>> 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) ïijŇåŔłèęĄäÿďäÿłåĚČçť äçŽĎäijŸåĚĹçžğäÿ■åŘŇåř ä¡ĘæŸŕåęĆæđIJäÿďäÿłåĚČçť äăijŸåĚĹçžğäÿĂæăůçŽĎèŕİïijŇéĆčäźĹæŕŤè¿ČæŞ■ä¡IJåřśäijŽèů§äźŃåĹ■äÿĂ

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

(continues on next page)

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

éĂŽè£ĞåijŢåĚěåŔęåď'ŰçŽĎ `index` åŔŸéĞŔçżĎæĹŘäÿL'åĚČçżĎ `(priority, index, item)` ïijŇářśèČ¡å¿Ĺåě¡çŽĎéĄ£åĚ■äÿŁéİćçŽĎéŤŹèŕŕïijŇ åŻääÿžäÿ■åŔŕèČ¡æIJL'äÿď'äÿłåĚČçť'ăæIJL'çŻÿåŘŇçŽĎ `index` åĂijãĂĆPython åIJĺåĄŽåĚČçżĎæŕŤè¿ČæŮűåĂŹïijŇåęĆæđIJåL'■éİćçŽĎæŕŤè¿ČåůšçżŕåŕŕäžěçąőåőŽçżŞæđIJäžĘïijŇ åŘŐéİćçŽĎæŕŤè¿ČæŞ■ä¡IJářśäÿ■äijŽåŔŠçŤ§äžĘïijŽ

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

åęĆæđIJä¡ăæČşåIJĺåď'Žäÿłçž£çĺŃäÿ■ä¡£çŤĺåŘŇäÿĂäÿłéŸ§åĹŮïijŇéĆčäźĹä¡ăéIJĂèęĄåćđåŁăéĂĆå¡ŞçŽ åŔŕäžěæ§ěçIJŃ 12.3 åřŔèŁĆçŽĎä¿Ńå■ŘæijŤçď'žæŸŕæĂŐæăůåĄŽçŽĎãĂĆ

`heapq` æĺąåİŮçŽĎåőŸæŰźæŰĞæąčæIJL'æŻť'èŕęçżĘçŽĎä¿Ńå■ŘçĺŃåžŔäžěåŘŁåŕźäžŐåăĘçŘĘèőžåŘŁ

3.6 1.6 å■ŮåĚÿäÿ■çŽĎéŤőæŸăåřĎåď'Žäÿłå Äij

éŮőécŸ

æĂŐæăůåőđçŐřäÿĂäÿłéŤőåŕźåžŤåď'Žäÿłå ÄijçŽĎå■ŮåĚÿïijĹäź§åŔń `multidict`ïijL'ïij§

èğčåĘşæŰźæąĹ

äÿĂäÿłå■ŮåĚÿåŕśæŸŕäÿĂäÿłéŤőåŕźåžŤäÿĂäÿłå■ŢåÄijçŽĎæŸăåřĎãĂĆåęĆæđIJä¡ăæČşèęĄäÿĂäÿłéŤőå æŕŤåęĆåĹŮèąĺæĹŰèĂĚéŻĘåŘĹéĞŇéİćãĂĆæŕŤåęĆïijŇä¡ăåŔŕäžěåČŔäÿŇéİćè£ŹæăůæđĎéĂăè£ŹæăůçŽĎå■

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

éĂL’æŃl’ä¡£çŤĺåĹŮèąĺè£ŸæŸréŻĘåŘĹåŔŰåEşäžŐä¡ăçŽĎåőđéŹĚéIJĂæśĆãĂĆåęĆæđIJä¡ăæČşä£ÌæŇA
åęĆæđIJæČşåŐżæŐL’éĞ■åd’■åĚČçt’ăářśä¡£çŤĺéŻĘåŘĹïijĹåźűäyŤäy■åĚşå£ČåĚČçt’ăçŽĎéąžåžŔéŮőécŸïijL

ä¡ăåŔŕäżěå¿ĹæŰźä¿£çŽĎä¡£çŤĺ `collections` æĺąåİŮäy■çŽĎ `defaultdict` æİěædĎéĂăè£Źæăůç ŽĎå■ŮåĚÿãĂĆ `defaultdict` çŽĎäyĂäyłçL’źå¿ĄæŸŕåőČäijŽèĞłåŁĺåĹİåğŃåŇŰæŕŔäył `key` åĹŽåijĂåğŃåŕźåžŤçŽĎåĂijïijŇæL’Ăäžěä¡ăåŔłéIJĂèęĄåĚşæşĺæůżåŁăăĚČçt’ăæŞ■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)
```

éIJĂèęĄæşĺæĎŔçŽĎæŸŕïijŇ `defaultdict` äijŽèĞłåŁĺäyžåřĘèęĄè£éŮőçŽĎéŤőïijĹåřśçőŮçŻőåL’■å
åęĆæđIJä¡ăåźűäy■éIJĂèęĄè£ŹæăůçŽĎçL’źæĂğïijŇä¡ăåŔŕäżěåIJĺäyĂäyłæŹőéĂŽçŽĎå■ŮåĚÿäyŁä¡£çŤĺ `setdefault()` æŰźæşŢæİěäžčæŻ£ãĂĆæŕŤåęĆïijŽ

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

ä¡ĘæŸŕå¿Ĺåd’ŽçÍŃåžŔåŚŸèğL’å¿Ů `setdefault()` çŤĺèţůæİěæIJL’çĆźåĹńæL’■ãĂĆåŻääyžæŕŔæňąèř `[]` ïijL’ãĂĆ

èőĺèőž

äÿĂèĹňæİěèőšïijŇåĹŻåžžäyĂäyłåd’ŽåĂijæŸăářĎå■ŮåĚÿæŸŕå¿ĹçőĂå■ŢçŽĎãĂĆä¡ĘæŸŕïijŇåęĆæđIJä
ä¡ăåŔŕèČ¡äijŽåČŔäyŃéİćè£ŹæăůæİěåőđçŐřïijŽ

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

åęĆæđIJä¡£çŤĺ `defaultdict` çŽĎèŕİäžčçăĄåřśæŻt’åŁăçőĂæt’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¡ăæČşåĹŻåžžäÿĂäÿłå■ŮåĚÿïijŇåźűäÿŤåIJÍè£■äžčæĹŰåžŔåĹŮåŇŰè£Źäÿłå■ŮåĚÿçŽĎæŮűåĂŹèČ¡åd

èğčåĘşæŰźæąĹ

äÿžäžĘèČ¡æŐğåĹűäÿĂäÿłå■ŮåĚÿäÿ■åĚČçt'ăçŽĎéąžåžŔïijŇä¡ååŔŕäżěä¡£çŤÍ `collections` æÍąåİŮäÿ■çŽĎ `OrderedDict` çśżãĂĆ åIJÍè£■äžčæŞ■ä¡IJçŽĎæŮűåĂŹåőČäijŽä£ÍæŇĄåĚČçt'ăèćńæŔŠåĚěæŮűçŽĎéąžåžŔïijŇçd'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¡ăæČşèęĄæđĎåžžäÿĂäÿłåřĘæİěéIJĂèęĄåžŔåĹŮåŇŰæĹŰçijŰçăĄæĹŘåĚűäžŰæăijåijŔçŽĎæŸăåřI `OrderedDict` æŸŕéİđåÿÿæIJĽçŤÍçŽĎãĂĆ ærŤåęĆïijŇä¡ăæČşçš¿çąőæŐğåĹűäžě JSON çijŰçăĄåŘŐå■Ůæőţç ŽĎéąžåžŔïijŇä¡ăåŔŕäżěåĚĹä¡£çŤÍ `OrderedDict` æİěæđĎåžžè£ŹæăůçŽĎæŢřæ■őïijŽ

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

èőÍèőž

`OrderedDict` åĘĚéČÍçżt'æŁd'çİĂäÿĂäÿłæăźæ■őéŤőæŔŠåĚěéąžåžŔæŐŠåžŔçŽĎåŔŇåŘŠéŞ¿èąÍãĂĆ åőČäijŽèćńæŤ¿åĹřéŞ¿èąÍçŽĎår¿éČÍãĂĆårźäžŐäÿĂäÿłåůšçżŔå■ŸåIJÍçŽĎéŤőçŽĎéĞ■åď■ètŇåĂijäÿ■äijŽæ

éIJĂèęĄæşÍæĎŔçŽĎæŸŕïijŇäÿĂäÿł `OrderedDict` çŽĎåď ğårŔæŸŕäÿĂäÿłæŹőéĂŽå■ŮåĚÿçŽĎäÿd'å æĽĂäżěåęĆæđIJä¡äèęĄæđĎåžžäÿĂäÿłéIJĂèęĄåď ğéĞŔ `OrderedDict` åőđä¿ŃçŽĎæŢřæ■őçżŞæđĎçŽĎæŮűåŹïijĹærŤåęĆèŕźåŔŰ 100,000 èąŇ CSV æŢřæ■őåĹřäÿĂäÿł `OrderedDict` åĹŮèąÍäÿ■åŐżïijL'ïijŇ éĆčäźĹä¡ååŕšå¿ŮäžŤçżĘæİČèąąäÿĂäÿŃæŸŕåŘęä¡£çŤÍ `OrderedDict` åÿęæİěçŽĎåě¡åd'ĎèęĄåd'ğè£Ğéćİåd'ŰåĘĚå■ŸæűĹèĂŮçŽĎå¡šåŞ■ãĂĆ

3.8 1.8 å■ŮåĚÿçŽĎè£ŘçőŮ

éŮőécŸ

æĂŐæăůåIJÍæŢřæ■őå■ŮåĚÿäÿ■æĽğèąŇäÿĂäžŻèőąçőŮæŞ■ä¡IJïijĹæŕŤåęĆæśĆæIJĂåřŔåĂijãĂĄæIJĂå

èğčåĘşæŰźæąĹ

èĂČèŹŚäÿŃéİćçŽĎèĆąçěĺåŘ■åŠŇäžůæăijæŸăåřĎå■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£ĞæİěãĂĆ æŕŤåęĆïijŇäÿŃéİćæŸŕæ§ěæĽ¿æIJĂåřŔåŠŇæIJĂåd'ğèĆąçěĺ

```
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')
```

çśžäijijçŽĎïijŇåŔŕäžěä¡£çŤÍ `zip()` åŠŇ `sorted()` åĞ¡æŢřæİěæŐŠåĹŮå■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£ŹäžŻèőąçőŮçŽĎæŮűåĂŹïijŇéIJĂèęĄæşĺæĎŔçŽĎæŸŕ `zip()` åĞ¡æŢřåĹŻåžžçŽĎæŸŕäÿłåŔłèČ¡èő£éŮőäÿĂæňąçŽĎè£■äžčåŹĺãĂĆ æŕŤåęĆïijŇäÿŃéİćçŽĎäžčçăĄåřśäijŽäžğçŤ§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
```

èőÍèőž

åęĆæđIJä¡ăåIJÍäÿĂäÿłå■ŮåĚÿäÿŁæĽğèąŇæŹőéĂŽçŽĎæŢřå■ęè£ŘçőŮïijŇä¡ăäijŽåŔŚçŐřåőČäžňäžĚäž

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

è£ŹäÿłçżŞæđIJåźűäÿ■æŸŕä¡ăæČşèęĄçŽĎïijŇåŻăäÿžä¡ăæČşèęĄåIJĺå■ŮåĚÿçŽĎåĂijéŻĘåŘĹäÿŁæL'ğèą
æĹŰèőÿä¡ăäijŽåřİèŕŢçİĂä¡£çŤĺå■ŮåĚÿçŽĎ `values()` æŰźæşŢæİêèğčåĘşè£ŹäÿłéŮőécŸïijŽ

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

äÿ■åźÿçŽĎæŸŕïijŇéĂŽåÿÿè£ŹäÿłçżŞæđIJåŘŇæăůäź§äÿ■æŸŕä¡ăæČşèęĄçŽĎãĂĆ
ä¡ăåŘŕèČ¡è£ŸæČşèęĄç§ěéĄŞåŕźåžŢçŽĎéŤőçŽĎä£ąæĄŕïijĹæŕŤåęĆéĆčçğ■èĆąçĚĺäžůæăijæŸŕæIJĂä¡ŐçŽĎï

ä¡ăåŘŕäžěåIJÍ `min()` åŠŇ `max()` åĞ¡æŢřäÿ■æŘŘä¿Ż `key` åĞ¡æŢřåŔĆæŢřæİêèŐůåŘŰæIJĂåřŘåĂijæĹŰæIJĂåď ğåĂijåŕźåžŢçŽĎéŤőçŽĎä£ąæĄŕãĂĆæŕŤåęĆïijŽ

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

ä¡ĘæŸŕïijŇåęĆæđIJè£ŸæČşèęĄå¿ŮåĹřæIJĂåřŘåĂijïijŇä¡ăåŘĹå¿ŮæL'ğèąŇäÿĂæňąæ§ěæL'¿æŞ■ä¡IJãĂ

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

åL'■éİćçŽĎ `zip()` åĞ¡æŢřæŰźæąĹéĂŽè£ĞåřĘå■ŮåĚÿâĂİåŘ■è¡ňâĂİäÿž
(åĂijïijŇéŤő) åĚČçżĎåžŘåĹŮæİèèğčåĘşäžĘäÿŁè£řéŮőécŸãĂĆ
å¡ŞærŤè¿Čäÿď äÿłåĚČçżĎçŽĎæŮűåĂŹïijŇåĂijäijŽåĚĹè£ŻèąŇærŤè¿ČïijŇçĎűåŘŐæL'■æŸŕéŤőãĂĆ
è£ŹæăůçŽĎèŕİä¡ăåřśèČ¡éĂŽè£ĞäÿĂæİąçőĂå■ŢçŽĎèŕ■åŘěåřśèČ¡å¿Ĺè¡żæİ¿çŽĎåőđçŐřåIJĺå■ŮåĚÿäÿŁçŽĎ

éIJĂèęĄæşĺæĎŘçŽĎæŸŕåIJĺèőąçőŮæŞ■ä¡IJäÿ■ä¡£çŤĺåĹřäžĘ (åĂijïijŇéŤő)
åŕźãĂĆå¡Şåď Žäÿłåőđä¡ŞæŇěæIJL'çŽÿåŘŇçŽĎåĂijçŽĎæŮűåĂŹïijŇéŤőäijŽåĘşåőŽè£ŤåŻđçżŞæđIJãĂĆ
ærŤåęĆïijŇåIJĺæL'ğèąŇ `min()` åŠŇ `max()` æŞ■ä¡IJçŽĎæŮűåĂŹïijŇåęĆæđIJæĄřåůğæIJĂåřŘæĹŰæIJĂåď

```
>>> 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§ěæL'¿äÿď å■ŮåĚÿçŽĎçŻÿåŘŇçĆź

éŮőécŸ

æĂŐæăůåIJĺäÿď äÿłå■ŮåĚÿäÿ■åŕźåŕżæL'¿çŻÿåŘŇçĆźïijĹærŤåęĆçŻÿåŘŇçŽĎéŤőãĂĄçŻÿåŘŇçŽĎåĂij

èğčåĘşæŰźæąĹ

èĂČèŹŚäÿŃéİćäÿď äÿłå■ŮåĚÿïijŽ

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

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

äÿžäžĘåŕżæĽ¿äÿďäÿłå■ŮåĚÿçŽĎçŻÿåŘŇçĆźïijŇåŔŕäžěçőĂå■ŢçŽĎåIJÍäÿďå■ŮåĚÿçŽĎ keys() æĹŰèĂĚ items() æŰźæşŢè£ŤåŻđçżŞæđIJäÿŁæĽğèąŇéŻĘåŘĹæŞ■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§åŔŕäžěçŤÍäžŐä£őæŤźæĹŰèĂĚè£Ğæżďå■ŮåĚÿåĚČçťăãĂĆ æŕŤåęĆïijŇåĄĞåęĆä¡ăæČşäžěçŐřæIJL'å■ŮåĚÿædĎéĂääÿĂäÿłæŐŠéŹďåĞääÿłæŇĞåőŽéŤőçŽĎæŰřå■ŮåĚÿ äÿŃéÍćåĹľçŤÍå■ŮåĚÿæŐÍåŕijæÍěåőđçŐřè£Źæăůç ŽĎéIJĂæśĆïijŽ

```
# 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■ŮåĚÿåŕśæŸŕäÿĂäÿłéŤőéŻĘåŘĹäÿŐåĂijéŻĘåŘĹçŽĎæŸăåŕĎåĚşçşżãĂĆ å■ŮåĚÿçŽĎ keys() æŰźæşŢè£ŤåŻđäÿĂäÿłåśŢçŐřéŤőéŻĘåŘĹçŽĎéŤőèğĘåŻ¿åŕźèśąãĂĆ éŤőèğĘåŻ¿çŽĎäÿĂäÿłå¿ĹåřŚèćńäžĘèğčçŽĎçL'źæĂğåŕśæŸŕåőČäžňäź§æŤŕæŇĄéŻĘåŘĹæŞ■ä¡IJïijŇæŕŤåęĆ æL'Ääžěïij Ňåę ĆæđIJä¡ăæČşåŕźéŻĘåŘĹçŽĎéŤőæL'ğèąŇäÿĂäžŻæŹőéĂŽçŽĎéŻĘåŘĹæŞ■ä¡IJïijŇåŔŕäžěçŻt' setãĂĆ

å■ŮåĚÿçŽĎ items() æŰźæşŢè£ŤåŻđäÿĂäÿłåŇĚåŘń (éŤőïijŇåĂij) åŕźçŽĎåĚČçťăèğĘåŻ¿åŕźèśąãĂĆ è£ŹäÿłåŕźèśąåŘŇæăůäź§æŤŕæŇĄéŻĘåŘĹæŞ■ä¡IJïijŇåźűäÿŤåŔŕäžěèćńçŤ

åř¡çőąå■ŮåĚÿçŽĎ values() æŰźæşŢäź§æŸŕçśžäijijïijŇä¡ĘæŸŕåőČåźűäÿ■æŤŕæŇĄè£ŹéĞŇäžŃçż■ç æ§Řçğ■çÍŃåžęäÿŁæŸŕåŻääÿžåĂijèğĘåŻ¿äÿ■èČ¡ä£ÌèŕĄæL'ĂæIJL'çŽĎåĂijäžŠäÿ■çŻÿåŘŇïijŇè£ŹæăůäijŽåŕ äÿ■è£ĞïijŇåęĆæđIJä¡ăçąňèęĄåIJĺåĂijäÿŁéÍćæL'ğèąŇè£ŹäžŻéŻĘåŘĹæŞ■ä¡IJçŽĎèŕÍïijŇä¡ääŔŕäžěåĚĹåřĘåĂ setïijŇçĎűåŘŐåĘ■æL'ğèąŇéŻĘåŘĹè£ŘçőŮåřśèąŇäžĘãĂĆ

3.10 1.10 åĹǎéŹďåžŔåĹŮçŻÿåŘŇåĚČçťǎåźűä£ÌæŇĄéąžåžŔ

éŮőécŸ

æĂŐæăůåIJÍäÿĂäÿłåžŔåĹŮäÿŁéÍćä£ÍæŇĄåĚČçťǎéąžåžŔçŽĎåŘŇæŮűæűĹéŹď'éĞ■åď'■çŽĎåĂijïij§

èğčåĘşæŰźæąĹ

åęĆæđIJåžŔåĹŮäÿŁçŽĎåĂijéČ¡æŸŕ hashable çśžåđŃïijŇéĆčäźĹåŔŕäžěå¿ĹçőĂå■ŢçŽĎåĹl'çŤĺéŻĘ

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

äÿŃéÍćæŸŕä¡£çŤĺäÿŁè£ŕåĞ¡æŢřçŽĎä¿Ńå■ŘïijŽ

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

è£ŹäÿłæŰźæşŢäžĚäžĚåIJÍåžŔåĹŮäÿ■åĚČçť'ääÿž hashable çŽĎæŮűåĂŹæL'■çőąçŤĺãĂĆ åęĆæđIJä¡ăæČşæúĹéŹď'åĚČçť'ääÿ■åŔŕåŞĹåÿŇïijĹærŤåęĆ dict çśžåđŃïijL'çŽĎåžŔåĹŮäÿ■éĞ■åď'■åĚČçť'ăçŽĎèŕÍïijŇä¡ăéIJĂèęĄåřĘäÿŁè£ŕäžčçăĄçÍ■å¿őæŤźåŔŸäÿĂ

```
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£ŹéĞŇçŽĎkeyåŔĆæŢřæŇĞåőŽäžĘäÿĂäÿłåĞ¡æŢřïijŇårĘåžŔåĹŮåĚČçť'ăè¡ňæ■ćæĹŘ hashable çśžåđŃãĂĆäÿŃéÍćæŸŕåőČçŽĎçŤĺæşŢçď'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¡ăæČşå§žäžŐå■Ţäÿłå■Ůæőţã ĂĄåśđæĂğæĹŰèĂĚæ§ŘäÿłæŻť'åď'ğçŽĎæŢřæ■őçżŞæđĎæİěæűĺ

èőĺèőž

åęĆæđIJä¡ääżĔäżĔåřśæŸŕæČşæűĹéŹď éĞ■åď■åĔČçť äïijŇéĂŽåÿÿåŔŕäżěçőĂå■ŢçŽĎæđĎéĂăäÿĂäÿłé

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

çĎűèĂŇïijŇè£Źçğ■æŰźæşŢäÿ■èČ¡çżť æŁď åĔČçť äçŽĎéąžåžŔïijŇçŤ§æĹŘçŽĎçżŞæđIJäÿ■çŽĎåĔČçť

åIJÍæIJňèŁĆäÿ■æĹŚäżňä¡£çŤÍäžĘçŤ§æĹŘåŹÍåĞ¡æŢřèőľ æĹŚäżňçŽĎåĞ¡æŢřæŻť åŁăéĂŽçŤÍïijŇäÿ■äžĘ æŕŤåęĆïijŇåęĆæđIJåęĆæđIJä¡ääæČşèŕżåŔŰäÿĂäÿłæŰĞäżűïijŇæűĹéŹď éĞ■åď■èąŇïijŇä¡ääåŔŕäżěå¿Ĺåőźæ

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

äÿŁè£řkeyåĞ¡æŢřåŔĆæŢřæĺąäż£äžĘ sorted() , min() åŠŇ max() çŋĽåĘĚç¡őåĞ¡æŢřçŽĎçŽÿäijijåŁ§èČ¡ãĂĆ åŔŕäżěåŔĆèĂČ 1.8 åŠŇ 1.13 åřŘèŁĆäžĘèğčæŻť åď ŽãĂĆ

3.11 1.11 åŚ¡åŘ■åĹĞçĽĞ

éŮőéćŸ

åęĆæđIJä¡äçŽĎçĺŃåžŔåŇĚåŘńäžĘåď ğéĞŔæŮăæşŢçŻť èğĘçŽĎçąňçijŰçăĄåĹĞçĽĞïijŇåźűäÿŤä¡ăæČ

èğčåĘşæŰźæąĹ

åĄĞåőŽä¡ăèęĄäżŐäÿĂäÿłèőřå¡ŢïijĹæŕŤåęĆæŰĞäżűæĹŰåĔűäżŰçśżäijijæäijåijŔïijĽäÿ■çŽĎæ§ŘäžŻåž

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

äÿŐåĔűéĆčæăůåĘŹïijŇäÿžäżĂäźĹäÿ■æČşè£Źæăůåš¡åŘ■åĹĞçĽĞåŚćïijŽ

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

åIJÍè£Źäÿłçĺ ĹæIJňäÿ■ïijŇä¡ăéĄ£åĔ■äžĘä¡£çŤÍåď ğéĞŔéŽ¿äżěçŘĘèğčçŽĎçąňçijŰçăĄäÿŃæăĞãĂĆè£

èőÍèőž

äÿĂèĹňæİěèőšïijŇäžčçăĄäÿ■åęĆæđIJåĞžçŐřåď'ğéĞŔçŽĎçąňçijŰçăĄäÿŃæăĞäijŽä¡£å¿ŮäžčçăĄçŽĎå
æŕŤåęĆïijŇåęĆæđIJä¡ăåŻđè£ĞæİěçIJŃçIJŃäÿĂåźť'åL'■ä¡ăåĘŹçŽĎäžčçăĄïijŇä¡ăäijŽæŚÿçİĂèĎŚècŃæČşéĆ
è£ŹæŸŕäÿĂäÿłå¿ĹçőĂå■ŢçŽĎèğčåĘşæŰźæąĹïijŇåőČèől'ä¡ăæŻť'åŁăæÿĚæŹřçŽĎèąĺè¿¿äžčçăĄçŽĎçŽőçŽĎ

åĘĚç¡őçŽĎ `slice()` åĞ¡æŢřåĹŻåžžäžĘäÿĂäÿłåĹĞçL'ĞåŕźèśąaĂĆæL'ĂæIJL'ä¡£çŤĺåĹĞçL'ĞçŽĎåIJřa

```
>>> 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¡ăæIJL'äÿĂäÿłåĹĞçL'ĞåŕźèśąaïijŇä¡ăåŔŕäžěåĹĘåĹńèřČçŤĺåőČçŽĎ `a.start`, `a.stop`, `a.step` åśđæĂğæİěèŐůåŔŰæŻť'åď'ŽçŽĎä£ąæĄŕãĂĆæŕŤåęĆïijŽ

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

åŔęåď'ŰïijŇä¡ăè£ŸåŔŕäžěéĂŽè£ĞèřČçŤĺåĹĞçL'ĞçŽĎ `indices(size)` æŰźæşŢåřĘåőČæŸăåřĎåĹŕäÿĂäÿłåůšç§ěåď'ğåřŔçŽĎåžŔåĹŮäÿŁãĂĆ è£ŹäÿłæŰźæşŢè£ŤåŻđäÿĂäÿłäÿL'åĚČçżĎ `(start, stop, step)` ïijŇæL'ĂæIJL'çŽĎåĂijéČ¡äijŽècńçijl'åŕŔïijŇçŽť'åĹřéĂĆåŘĹè£Źäÿłåůšç§ěåžŔåĹŮçŽĎè¿źçŢŇäÿžæ■ćãĂĆ è£ŹæăůïijŇä¡£çŤĺçŽĎæŮűåřšäÿ■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 åžŔåĹŮäÿ■åĞžçŐřæňąæŢřæIJÄåď'ŽçŽĎåĚČçť'ă

éŮőécŸ

æĂŐæăůæL'¿åĞžäÿĂäÿłåžŔåĹŮäÿ■åĞžçŐřæňąæŢřæIJÄåď'ŽçŽĎåĚČçť'ääŚćïij§

èğčåĘşæŰźæąĹ

collections.Counter çśžåřśæŸŕäÿŞéŮĺäÿžè£ŹçśżéŮőécŸèĂŇèő¿èőąçŽĎïijŇ åőČçŤŽèĞşæIJL'äÿĂäÿłæIJL'çŤĺçŽĎ most_common() æŰźæşŢçŻť'æŐěçżŹäžĘä¡äç■ŤæąĹãĂĆ

äÿžäžĘæijŤçď'žïijŇåĚĹåĄĞèő¿ä¡ăæIJL'äÿĂäÿłå■Ţèŕ■åĹŮèąĺåźűäÿŤæČşæL'¿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)
# åĞžçŐřécŚçŐĞæIJÄéńŸçŽĎ3äÿłå■Ţèŕ■
top_three = word_counts.most_common(3)
print(top_three)
# Outputs [('eyes', 8), ('the', 5), ('look', 4)]
```

èőĺèőž

ä¡IJäÿžè¿ŞåĚëïijŇ Counter åŕźèśąåŔŕäžěæŐěåŔŮäżżæĎŔçŽĎçŤśåŔŕåŞĹåÿŇïijĹhashableïijL'åĚČ åIJĺåžŢåśĆåőđçŐŕäÿŁïijŇäÿĂäÿł Counter åŕźèśąåřśæŸŕäÿĂäÿłå■ŮåĚÿïijŇåřĘåĚČçť'äæŸäåřĎåĹŕåőČåĞžç

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

åęĆæđIJä¡äæČşæL'ŃåŁĺåćđåŁäèőąæŢřijŇåŔŕäžěçőĂå■ŢçŽĎçŤĺåŁäæşŢïijŽ

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

æĹŰèĂĚä¡ăåŔŕäžěä¡£çŤĺ update() æŰźæşŢïijŽ

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

Counter åőđä¿ŃäÿĂäÿłéšIJäÿžäžžç§ěçŽĎçL'źæĂğæŸŕåőČäžňåŔŕäžěå¿ĹåőźæŸ§çŽĎèů§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})
>>>
```

æŕńæŮăçŰŚéŮőïijŇ Counter åŕźèśąåIJĺåĞäáźŐæL'ĂæIJL'éIJÄèęĄåĹűèąĺæĹŰèĂĚèőąæŢřæŢřæ■őçŽÌ
åIJĺèğčåEşè£ŹçśźéŮőécŸçŽĎæŮűåĂŹä¡ăåžŤèŕěäijŸåĚĹéĂL'æŃl'åőČïijŇèĂŇäÿ■æŸŕæL'ŃåŁĺçŽĎåĹl'çŤĺå

3.13 1.13 éĂŽè£Ğæ§ŘäÿłåĚşéŤőå■ŮæŐŠåžŔäÿĂäÿłå■ŮåĚÿåĹŮèąĺ

éŮőécŸ

ä¡ăæIJL'äÿĂäÿłå■ŮåĚÿåĹŮèąĺïijŇä¡ăæČşæăźæ■őæ§Řäÿłæĺ Ű æ§ŘåĞääÿłå■ŮåĚÿå■ŮæőţæİěæŐŠåžŔè£

èğčåEşæŰźæąĹ

éĂŽè£Gä¡£çŤĺ operator æĺąåİŮçŽĎ itemgetter
åĞ¡æŢřijŇåŔŕäžěéİđåÿÿåőźæŸŞçŽĎæŐŠåžŔè£ŹæăůçŽĎæŢřæ■őçżŞæđĎãĂĆ
åĄĞèő¿ä¡ăäžŐæŢřæ■őåžŞäÿ■æčĂçť'ćåĞžæİěç¡ŚçńŹäijŽåŚŸä£ąæĄŕåĹŮèąĺïijŇåźűäÿŤäžěäÿŃåĹŮçŽĎæŢřa

```
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■őäżżæĎŔçŽĎå■ŮåĚÿå■ŮæőţæİěæŐŠåžŔè¿ŞåĚěçżŞæđIJèąŇæŸŕå¿Ĺåőźæ ŸŞåőđçŐřçŽĎïijŇäžč

```
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 keysïijŇæŕŤåęĆäÿŃéİćçŽĎäžčçăĄ

```
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` èćńäijåéĂŠçżŹæŐěåŔŮäÿĂäÿłåĚşéŤőå■ŮåŔĆæŢřçŽĎ `sorted()` åĘĚç¡őåĞ¡æŢřãĂĆ è£ŹäÿłåŔĆæŢřæŸŕ `callable` çśžåđŇïijŇåźűäÿŤäžŐ `rows` äÿ■æŐěåŔŮäÿĂäÿłå■ŢäÿĂåĚČçť'äïijŇçĎůåŘŐè£ŤåŻđèćńçŤÍæİěæŐŠåžŔçŽĎåĂijãĂĆ `itemgetter()` åĞ¡æŢřářśæŸŕèť'§èť'čåĹŻåžžè£Źäÿł `callable` årźèśąçŽĎãĂĆ

`operator.itemgetter()` åĞ¡æŢřæIJL'äÿĂäÿłèćń `rows` äÿ■çŽĎèőřå¡ŢçŤÍæİěæ§ěæL'¿åĂijçŽĎçť'ćåijŢåŔĆæŢřãĂĆåŔŕäžěæŸŕäÿĂäÿłå■ŮåĚÿéŤőåŘ■çğřïijŇ äÿĂäÿłæŢť'å¡ćåĂijæĹŰèĂĚäżżä¡ŢèČ¡åď'§äijååĚěäÿĂäÿłårźèśąçŽĎ `__getitem__()` æŰźæşŢçŽĎåĂijãĂĆ åęĆæđIJä¡ăäijååĚěåď'Žäÿłçť'ćåijŢåŔĆæŢřçżŹ `itemgetter()`

ïijŇåőČçŤ§æĹŘçŽĎ `callable` åŕźèśąäijŽè£ŤåŻđäÿĂäÿłåŇĚåŘńæĽĂæIJĽåĚČçť'ääĂijçŽĎåĚČçżĎïijŇ åźűäÿŤ `sorted()` åĞ¡æŢřäijŽæäźæ■őè£ŹäÿłåĚČçżĎäÿ■åĚČçť'äéąžåžŘåŐżæŐŠåžŘãĂĆ ä¡Ęä¡äæČşèęĄåŘŇæŮűůåIJĺåĞääÿłå■Ůæőţäÿ£éÍćè£ŻèąŇæŐŠåžŔïijĹæŕŤåęĆéĂŽè£ĞåğŞåŠŇåŘ■æÍěæŐŠåžĺ

`itemgetter()` æIJĽæŮűůåĂŹäź§åŔŕäžěçŤÍ `lambda` èąÍè¿¿åijŔäžčæŽ£ïijŇæŕŤåęĆïijŽ

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

è£Źçğ■æŰźæąĹäź§äÿ■éŤŹãĂĆä¡ĘæŸŕïijŇä¡£çŤÍ `itemgetter()` æŰźåijŔäijŽè£ŘèąŇçŽĎçÍ■å¿ő å£ńçĆźãĂĆåŽäæ■d'ïijŇåęĆæđIJä¡ääŕźæĂğèČ¡èęĄæśĆæŕŤè¿ČénŸçŽĎèŕÍåřš `itemgetter()` æŰźåijŔãĂĆ

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 æŐŠåžŔäÿ■æŤŕæŇĄåŐ§çŤ§æŕŤè¿ČçŽĎåŕźèśą

éŮőécŸ

ä¡ääČşæŐŠåžŔçśżåđŇçŻÿåŘŇçŽĎåŕźèśąïijŇä¡ĘæŸŕäžŰäžñäÿ■æŤŕæŇĄåŐ§çŤ§çŽĎæŕŤè¿ČæŞ■ä¡IJã

èğčåĘşæŰźæąĹ

åĘĚç¡őçŽĎ `sorted()` åĞ¡æŢřæIJĽäÿĂäÿłåĚşéŤőå■ŮåŔĆæŢř `key` ïijŇåŔŕäžěäijääĚěäÿĂäÿł `callable` åŕźèśąçżŹåőČïijŇ è£Źäÿł `callable` åŕźèśąåŕźæŕŔäÿłäijääĚěçŽĎåŕźèśąè£ŤåŻđäÿĂäÿłåĂijïijŇè£ŹäÿłåĂijäijŽèćń `sorted` çŤÍæÍěæŐŠåžŔè£ŹäžŽåŕźèśąãĂĆ æŕŤåęĆïijŇåęĆæđIJä¡ääIJĺåžŢçŤÍçÍŃåžŔéĞŇéÍćæIJĽäÿĂäÿł `User` åőđä¿ŃåžŔåĽŮïijŇåźűäÿŤä¡ääÿŇæIJŻéĂŽè£ĞäżŰäžňçŽĎ `user_id` åśđæĂğè£ŻèąŇæŐŠåžŔïijŇ ä¡ääŔŕäžěæŔŘä¿ŻäÿĂäÿłäžě `User` åőđä¿Ńä¡IJäÿžè¿ŞåĚěåźűè¿ŞåĞžåŕźåžŤ `user_id` åĂijçŽĎ `callable` åŕźèśąĂĆæŕŤåęĆïijŽ

```
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)

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

åŔęåď Űäÿ Ăçğ■æŰźåijŔæŸŕä¡£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()` åŔŕèČ¡åŔŰåĘşäžŐäÿłäžžåŰIJåě¡ãĂĆ ä¡ĘæŸŕïijŇ `attrgetter()` åĞ¡æŢřéĂŽåÿÿäijŽè£ŘèąŇçŽĎå£ńçĆźïijŇåźűäÿŤè£ŸèČ¡åŘŇæŮűåĚąèőÿåď Žäÿłå■Ůæőţè£ŻèąŇæŕŤè¿ČãĂ è£Źäÿłèů§ `operator.itemgetter()` åĞ¡æŢřä¡IJçŤÍäžŐå■ŮåĚÿçśžåđŇå¿ŁçśžäijijïijĺåŔĆèĂČ1.13åřŔè ä¿ŇåęĆïijŇåęĆæđIJ `User` åőđä¿Ńè£ŸæIJL'äÿĂäÿł `first_name` åŠŇ `last_name` åśđæĂğïijŇéĆčäźĹåŔŕäžěåŘŚäÿŇéİćè£ŹæăůæŐŠåžŔïijŽ

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

åŘŇæăůéIJĂèęĄæşÍæĎŔçŽĎæŸŕïijŇè£ŹäÿĂåřŔèŁĆçŤÍåĹřçŽĎæŁĂæIJŕåŘŇæăůéĂĆçŤÍäžŐåČŔ `min()` åŠŇ `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■Ůæőţå řĘèőřå¡ŢåĹĘçżĎ

éŮőécŸ

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

èğčåĘşæŰźæąĹ

`itertools.groupby()` åĞ¡æŢřåŕźäžŐè£ŹæăůçŽĎæŢřæ■őåĹĘçżĎæŞ■ä¡IJéİđåÿÿåőđçŤÍãĂĆ äÿžäžĘæijŤçd'žïijŇåĄĞèő¿ä¡ăåůšçżŔæIJL'äžĘäÿŇåĹŮçŽĎå■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¡ăæČşåIJÍæŇL’ date åĹEçżĎåŘŐçŽĎæŢřæ■őåİŮäÿŁè£Żèą Ňè£■äžčãĂĆäÿžäžĘè£Źæăůà
`date`)æŐŠåžŔïijŇ çĎűåŘŐèřČçŤÍ `itertools.groupby()` åĞ¡æŢřijŽ

```
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£ŘèąŇçżŞædIJïijŽ

```
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¡æŢřæL’ńæŔŕæŢt’äÿłåžŔåĹŮåźűäÿŤæ§ěæL’¿è£đçż■çŻÿåŘŇåĂijïijĹæĹŰèĂĚæăźæ■ő
key åĞ¡æŢře£ŤåŻđåĂijçŻÿåŘŇïijL’çŽĎåĚČçt’ăåžŔåĹŮãĂĆ
åIJÍæŕŔæňąè£■äžčçŽĎæŮűåĂŹïijŇåőČäijŽè£ŤåŻđäÿĂäÿłåĂijåŠŇäÿĂäÿłè£■äžčåŹĺåŕźèšąïijŇ
è£Źäÿłè£■äžčåŹĺåŕźèšąåŔŕäžěçŤ§æĹŘåĚČçt’ăåĂijåĚĺéČĺç■L’äžŐäÿŁéĺćéĆčäÿłåĂijçŽĎçżĎäÿ■æL’ĂæIJL’år

äÿĂäÿłéİđåÿÿéĞ■èęĄçŽĎåĞĘåd’Ğæ■ěéłd’æŸŕèęĄæăźæ■őæŇĞåőŽçŽĎå■Ůæőţ åřĘæŢřæ■őæŐŠåžŔãĂ
åŻăäÿz `groupby()` äžĚäžĚæčĂæ§ěè£đçż■çŽĎåĚČçt’ăïijŇåęĆæđIJäžŇåĚĹåźűæšąæIJL’æŐŠåžŔåőŇæĹŘç

åęĆæđIJä¡ăäżĚäżĚåŔłæŸŕæČşæăźæ■ő `date` å■ŮæőţåřĘæŢřæ■őåĹĘçżĎåĹřäÿĂäÿłåď˙ğçŽĎæŢřæ■őçżŞ
éĆčäźĹä¡ăæIJĂåě¡ä¡£çŤĺ `defaultdict()` æİěæđĎåžžäÿĂäÿłåď˙ŽåĂijå■ŮåĚÿïijŇåĚşäžŐåď˙ŽåĂijå■ŮåĚÿ
1.6 åřŔèŁĆæIJL'è£Ğèŕęçżę Ç ŽĎäżŃçż■ãĂĆæŕŤåęĆïijŽ

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

è£Źæăůç ŽĎèŕÍä¡ăåŔŕäžěå¿Ĺè¡żæİ¿çŽĎåřśèČ¡åŕźæŕŔäÿłæŇĞåőŽæŮěæIJ§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'}
>>>
```

åIJÍäÿŁéİćè£ŹäÿłäżŃå■ŘäÿïijŇæĹŚäžňæšąæIJL'å£ĚèęĄåĚĹåřĘèőŕå¡ŢæŐŠåžŔãĂĆåŻăæ■d'ïijŇåęĆæđ
è£Źçğ■æŰźåijŔäijŽæŕŤåĚĹæŐŠåžŔçĎűåŘŐåĘ■éĂŽè£Ğ `groupby()`
åĞ¡æŢřè£■äžčçŽĎæŰźåijŔè£ŘèąŇå¿Ůå£ńäÿĂäžŻãĂĆ

3.16 1.16 è£Ğæžď˙åžŔåĹŮåĚČçť˙ă

éŮőécŸ

ä¡ăæIJL'äÿĂäÿłæŢřæ■őåžŔåĹŮïijŇæČşåĹľçŤĺäÿĂäžŻèğĎåĹŹäžŐäÿ■æŘŘåŘŰåĞžéIJĂèęĄçŽĎåĂijæĺ

èğčåĘşæŰźæąĹ

æIJĂçőĂå■ŢçŽĎè£Ğæžď˙åžŔåĹŮåĚČçť˙ăçŽĎæŰźæşŢåřśæŸŕä¡£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¡IJåIJÍçijžéŹůåřśæŸŕåęĆæđIJè¿ŞåĚéĺđåÿÿåď˙ğçŽĎæŮűåĂŹäijŽäžğç
åęĆæđIJä¡ååŕźåĘĚå■ŸæŕŤè¿ĈæŢŔæĎ§ïijŇéĆčäźĹä¡ăåŔŕäžěä¡£çŤĺçŤ§æĹŘåŹĺèąĺè¿¿åijŔè£■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)

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

æIJL’æŮűåĂŹïijŇè£ĞæżďˇèğĎåĹŹæŕŤè¿Čåď’■æİĆïijŇäÿ■èČ¡çőĂå■ŢçŽĎåIJĺåĹŮèąĺæŐĺåŕijæĹŰèĂĚçæŕŤåęĆïijŇåĄĞèő¿è£ĞæżďˇçŽĎæŮűåĂŹéIJĂèęĄåďˇĎçŘĘäÿĂäžŽåijĆåÿÿæĹŰèĂĚåĚűäżŰåď’■æİĆæČĚåĘţçĎűåŘŐä¡£çŤĺåĘĚåžžçŽĎ `filter()` åĞ¡æŢřãĂĆçďˇžä¿ŃåęĆäÿŇïijŽ

```
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¡æŢřåĹŽåżžäžĘäÿĂäÿłè£■äżčåŹĺïijŇåŽäæ■ď’åęĆæđIJä¡ăæČşå¿ŮåĹřäÿĂäÿłåĹŮèąĺçŽĎè `list()` åŐżè¡ňæ■ćãĂĆ

èőĺèőž

åĹŮèąĺæŐĺåŕijåŠŇçŤ§æĹŘåŹĺèąĺè¿¿åijŔéĂŽåÿÿæČĚåĘţäÿŃæŸŕè£ĞæżďˇæŢřæ■őæIJĂçőĂå■ŢçŽĎæŰ åĚűåőđåőČäžňè£ŸèČ¡åIJĺè£ĞæżďˇçŽĎæŮűåĂŹè¡ňæ■ćæŢřæ■őãĂĆæŕŤåęĆïijŽ

```
>>> 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£ĞæżďˇæŞ■ä¡IJçŽĎäÿĂäÿłåŔŸçğ■åřśæŸŕåřĘäÿ■çňęåŘĹæİąäžűçŽĎåĂijçŤĺæŰřçŽĎåĂijäżčæŻ£ïijŇè æŕŤåęĆïijŇåIJĺäÿĂäĹŮæŢřæ■őäÿ■ä¡ååŔŕèČ¡äÿ■äžĚæČşæL’¿åĹřæ■čæŢřïijŇèĂŇäÿŤè£ŸæČşåřĘäÿ■æŸŕæ■ éĂŽè£ĞåřĘè£ĞæżďˇæİąäžűæŤ¿åĹřæİąäžűèąĺè¿¿åijŔäÿ■åŐżïijŇåŔŕäžěå¿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)

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

åŘęåď Űäÿ Ăäÿłå Ăijå¿ ŮåĚşæşÍçŽĎè£ĞæżďåůěåĚůåřśæŸŕ itertools.compress() ïijŇ åőČäžëäÿ Ăäÿł iterable åŕźèśąåŠŇäÿ Ăäÿł çŻÿåŕźåžŤçŽĎ Boolean éĂĽ'æŃľ åŹĺåžŔåĹŮä¡IJäÿžè¿ŞåĚěåŘĆæŢřãĂĆ çĎŰåŘŐè¿ŞåĞž iterable åŕźèśąäÿ■åŕźåžŤéĂĽ'æŃľ åŹĺäÿž True çŽĎåĚČçť ăãĂĆ å¡Şä¡ăéIJĂèęĄçŤĺåŘęåď Űäÿ Ăäÿł çŻÿåĚşèĄŢçŽĎåžŔåĹŮæÌěè£Ğæżďæ§Řäÿł åžŔåĹŮçŽĎæŮűåĂŹïijŇè£ŹäærŤåęĆïijŇåĄĞåęĆçŐřåIJĺä¡ăæIJL'äÿŃéÍćäÿďåĹŮæŢřæ■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¡ăæČşářĘéĆčäžŻářźåžŤ count åĂijåď ğäžŐ5çŽĎåIJřåİĂåĚÍéČÍè¿ŞåĞžïijŇéĆčäźĹä¡ăåŔŕäżèè£

```
>>> 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£ŹéĞŇçŽĎåĚşéŤőçĆźåIJÍäžŐåĚĹåĹŻåžžäÿĂäÿł Boolean åžŔåĹŮïijŇæŇĞçď žåŞłäžŻåĚČçť ăçňęåŘĹæİąäżűãĂĆ çĎŰåŘŐ compress() åĞ¡æŢřæäźæ■őè£ŹäÿłåŹŔåĹŮåŐžéĂĽ'æŃľè¿ŞåĞžåŕźåžŤä¡■ç¡őäÿž True çŽĎåĚČçť ăãĂĆ

åŠŇ filter() åĞ¡æŢřçśžäijijïijŇ compress() äźş§æŸŕè£ŤåŻđçŽĎäÿĂäÿłè£■äžčåŹĺãĂĆåŻäæ■d'ïij éĆčäźĹä¡ăéIJĂèęĄä¡£çŤĺ list() æİěåřĘçżŞæđIJè¡ñæ■ćäÿžåĹŮèąÍçśžåđŃãĂĆ

3.17 1.17 äžŐå■ŮåĚÿäÿ■æŔŘåŘŰå■ŘéŻĘ

éŮőécŸ

ä¡ăæČşæđĎéĂääÿĂäÿłå■ŮåĚÿïijŇåőČæŸŕåŘęåď Űäÿ Ăäÿłå■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}
```

èőÍèőž

åď ğåď ŽæŢřæČĚåĘţäÿŃå■ŮåĚÿæŐĺåŕijèČ¡åĄŽåĹřçŽĎïijŇéĂŽè£ĞåĹŻåžäÿĂäÿłåĚČçżĎåžŔåĹŮçĎű
`dict()` åĞ¡æŢřäź§èČ¡åőđçŐřãĂĆæŕŤåęĆïijŽ

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

ä¡ĘæŸŕïijŇå■ŮåĚÿæŐĺåŕijæŰźåijŔèąÍæĎŔæŻť æŸĚæŹřïijŇåźűäÿŤåőđéŹĚäÿŁäź§äijŽè£ŘèąŇçŽĎæŻť
ïijĹåIJÍè£ŹäÿłäżŃå■Řäÿ■ïijŇåőđéŹĚæţŃèŕŢåĞääźŐæŕŤ `dict()`
åĞ¡æŢřæŰźåijŔå£ńæŢť æŢť äÿĂåĂ■ïijL ãĂĆ

æIJL æŮűåĂŹåőŇæĹŘåŘŇäÿĂäżűäžŇäijŽæIJL åď ŽçğæŰźåijŔãĂĆæŕŤåęĆïijŇçňňäžŇäÿłä¿Ńå■ŘçÍŃ

```
# 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 æţŃèŕŢçżŞædIJæŸ¿çď žè£ŹçğæŰźæąĹåď ğæęĆæŕŤçňňäÿĂçğæŰźæąĹæ
1.6 åĂ■ãĂĆ åęĆædIJåŕźçÍŃåžŔè£ŘèąŇæĂğèČ¡èęĄæśĆæŕŤè¿ČéńŸçŽĎèŕİïijŇéIJĂèęĄèŁśçĆźæŮűéŮť åŐžå
åĚşäžŐæŻť åď Žèőąæ ŮűåŠŇæĂğèČ¡æţŃèŕŢïijŇåŔŕäžěåŔĆèĂČ 14.13 åŕŘèŁĆãĂĆ

3.18 1.18 æŸäåřĎåŘ■çğřåĹřåžŔåĹŮåĚČçť ă

éŮőécŸ

ä¡ăæIJL äÿĂæőţéĂŽè£ĞäÿŃæăĞèő£éŮőåĹŮèąÍæĹŰèĂĚåĚČçżĎäÿ■åĚČçť ăçŽĎäžčçăĄïijŇä¡ĘæŸŕè£Ź
äžŐæŸŕä¡ăæČşéĂŽè£ĞåŘ■çğřæİèèő£éŮőåĚČçť ããĂĆ

èğčåĘşæŰźæąĹ

collections.namedtuple() åĞ¡æŢřéĂŽè£Ğä¡£çŤĺäÿĂäÿłæŹőéĂŽçŽĎåĚČçżĎåŕźèśąæİěåÿőä¡ă
è£ŹäÿłåĞ¡æŢřåőđéŹĚäÿŁæŸŕäÿĂäÿłè£ŤåŻđ Python äÿ■æăĞåĞĘåĚČçżĎçśżåđŃå■ŘçśżçŽĎäÿĂäÿłåůěåŐĆă
ä¡ăéIJĂèęĄäijăéĂŠäÿĂäÿłçśżåđŃåŘ■åŠŇä¡ăéIJĂèęĄçŽĎå■ŮæőţçżŹåőČïijŇçĎűåŘŐåőČåŕśäijŽè£ŤåŻđäÿĂ
äžčçăĄçď´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'
>>>
```

åř¡çőą namedtuple çŽĎåőđä¿ŃçIJŃèţůæİěåČŔäÿĂäÿłæŹőéĂŽçŽĎçśżåőđä¿ŃïijŇä¡ĘæŸŕåőČèů§åĚČ
æŕŤåęĆïijŽ

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

åŠ¡åŘ■åĚČçżĎçŽĎäÿĂäÿłäÿżèęĄçŤĺéĂŤæŸŕåřĘä¡ăçŽĎäžčçăĄäžŐäÿŃæăĞæŞ■ä¡IJäÿ■èğčèĎśåĞžæİě
åŻăæ■ď¨ïijŇåęĆæđIJä¡ääžŐæŢřæ■őåžŞèřČçŤĺäÿ■è£ŤåŻđäžĘäÿĂäÿłå¿Ĺåď´ğçŽĎåĚČçżĎåĹŮèąĺïijŇéĂŽè£Ğ
å¡Şä¡ääIJĺèąĺäÿ■æůżåŁääžĘæŰřçŽĎåĹŮçŽĎæŮűåĂŹä¡ăçŽĎäžčçăĄåŕèČ¡åŕśäijŽåĞžéŤŹäžĘãĆä¡ĘæŸŕå

äÿžäžĘèŕť´æŸŐæÿĚæěŽïijŇäÿŃéİćæŸŕä¡£çŤĺæŹőéĂŽåĚČçżĎçŽĎäžčçăĄïijŽ

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

äÿŃæăĞæŞ■ä¡IJéĂŽåÿÿäijŽèőĺ´äžčçăĄèąĺæĎŔäÿ■æÿĚæŹřïijŇåźűäÿŤéİđåÿÿä¿ÌètŰèőřå¡ŢçŽĎçżŞæđĎă
äÿŃéİćæŸŕä¡£çŤĺåŠ¡åŘ■åĚČçżĎçŽĎçL´ĹæIJñïijŽ

```
from collections import namedtuple

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

(continues on next page)

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

èőÍèőž

åŠ¡åŘ■åĚČçżĎåŔęäÿĂäÿłçŤĺéĂŤåřśæŸŕä¡IJäÿžå■ŮåĚÿçŽĎæŻ£äżčïijŇåŻääÿžå■ŮåĚÿå■ŸåĆĺéIJĂèęĄ
åęĆæđIJä¡ăéIJĂèęĄæđĎåżžäÿĂäÿłéİđåÿÿåď ğçŽĎåŇĚåŘńå■ŮåĚÿçŽĎæŢřæ■őçżŞæđĎïijŇéĆčäźĹä¡£çŤĺåŠ¡
ä¡ĘæŸŕéIJĂèęĄæşÍæĎŔçŽĎæŸŕïijŇäÿ■åČŔå■ŮåĚÿéĆčæăůïijŇäÿĂäÿłåŠ¡åŘ■åĚČçżĎæŸŕäÿ■åŔŕæŻťæŤźç

```
>>> 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¡ăçIJ§çŽĎéIJĂèęĄæŤźåŔŸåśđæĂğçŽĎåĂijïijŇéĆčäźĹåŔŕäżěä¡£çŤĺåŠ¡åŘ■åĚČçżĎåőđä¿Ńç
_replace() æŰźæşŢïijŇ åőČäijŽåĹŻåżžäÿĂäÿłåĚĺæŰřçŽĎåŠ¡åŘ■åĚČçżĎåźűåřĘåŕźåžŤçŽĎå■ŮæőţçŤĺæ

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

_replace() æŰźæşŢè£ŸæIJL'äÿĂäÿłå¿ĹæIJL'çŤĺçŽĎçL'źæĂğåřśæŸŕå¡Şä¡ăçŽĎåŠ¡åŘ■åĚČçżĎæŃě
åőČæŸŕäÿĂäÿłéİđåÿÿæŰźä¿£çŽĎåąńåĚĚæŢřæ■őçŽĎæŰźæşŢãĂĆ
ä¡ääŔŕäżěåĚĹåĹŻåżžäÿĂäÿłåŇĚåŘńçijžçIJĄåĂijçŽĎåŐ§ådŃåĚČçżĎïijŇçĎűåŘŐä¡£çŤĺ
_replace() æŰźæşŢåĹŻåżžæŰřçŽĎåĂijècńæŻťæŰrè£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¡£çŤĺæŰźæşŢïijŽ

```
>>> 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)
>>>
```

æIJĂåŘŐèęĄèŕťçŽĎæŸŕïijŇåęĆæđIJä¡ăçŽĎçŻőæăĞæŸŕåőŽäźL'äÿĂäÿłéIJĂèęĄæŻť'æŰřå¿Ĺåď'Žåőđä¿è£ŹæŮűåĂŹä¡ăåžŤèŕěèĂČèŹŠåőŽäźL'äÿĂäÿłåŇĚåŘń __slots__ æŰźæşŢçŽĎçśżïijĹåŘĆèĂČ8.4åřŔèŁĆïijL'ãĂĆ

3.19 1.19 è¡ňæ■ćåźűåŘŇæŮűèőąçőŮæŢřæ■ő

éŮőécŸ

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

èğčåĘşæŰźæąĹ

äÿĂäÿłéİđåÿÿäijŸéŻĚçŽĎæŰźåijŔåŐżçżŞåŘĹæŢřæ■őèőąçőŮäÿŐè¡ňæ■ćåřśæŸŕä¡£çŤĺäÿĂäÿłçŤ§æĹŘæŕŤåęĆïijŇåęĆæđIJä¡ăæČşèőąçőŮåźşæŰźåŠŇïijŇåŔŕäžěåČŔäÿŃéİćè£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)

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

èőÍèőž

äÿŁéÍćçŽĎçď'žä¿ŃåŘŚä¡ăæijŤçď'žäžĘå¡ŞçŤ§æĹŘåŹÍèąÍè¿¿åijŔä¡IJäÿžäÿĂäÿłå■ŢçŃňåŔĆæŢřäijăéĂŠç
æŕŤåęĆïijŇäÿŃéÍćè£ŹäžŻèŕ■åŘěæŸŕç■L'æŢĹçŽĎïijŽ

```
s = sum((x * x for x in nums)) #␣
↪æŸ¿åijŔçŽĎäijăéĂŠäÿĂäÿłçŤŞæĹŘåŹÍèąÍè¿¿åijŔåŕźèśą
s = sum(x * x for x in nums) #␣
↪æŻt'åŁääijŸéŽĚçŽĎåőđçŐřæŰźåijŔïijŇçIJĄçŢĕäžĘæŃňåŔů
```

ä¡£çŤÍäÿĂäÿłçŤ§æĹŘåŹÍèąÍè¿¿åijŔä¡IJäÿžåŔĆæŢřäijŽæŕŤåĚĹåĹŻåžžäÿĂäÿłäÿť'æŮűåĹŮèąÍæŻť'åŁăér
æŕŤåęĆïijŇåęĆædIJä¡ääÿ■ä¡£çŤÍçŤ§æĹŘåŹÍèąÍè¿¿åijŔçŽĎèŕÍïijŇä¡ăåŔŕèČ¡äijŽèĂČèŹŚä¡£çŤÍäÿŃéÍćçŽĎå

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

è£Źçğ■æŰźåijŔåŘŇæăůåŔŕäžěè¿¿åĹřæČşèęĄçŽĎæŢĹædIJïijŇä¡ĘæŸŕåőČäijŽåď'ŽäÿĂäÿłæ■ěéłď'ïijŇå
åŕźäžŐåřŘådŃåĹŮèąÍåŔŕèČ¡æšąäžĂäźĹåĚşçşżïijŇä¡ĘæŸŕåęĆædIJåĚČçť'ăæŢřéĞŘéÍđåÿÿåď'ğçŽĎæŮűåĂŹï
åőČäijŽåĹŻåžžäÿĂäÿłåůÍåď'ğçŽĎäžĚäžĚèćńä¡£çŤÍäÿĂæňąåřśèćńäÿćåijČçŽĎäÿť'æŮűæŢřæ■őçżŞædĎãĂĆè

åIJÍä¡£çŤÍäÿĂäžŻèĄŽéŻĘåĞ¡æŢřæŕŤåęĆ `min()` åŠŇ `max()`
çŽĎæŮűåĂŹä¡ăåŔŕèČ¡æŻť'åŁăåĂ¿åŘŚäžŐä¡£çŤÍçŤ§æĹŘåŹÍçL'ĹæIJňïijŇ
åőČäžňæŐěåŔŮçŽĎäÿĂäÿł key åĚşéŤőå■ŮåŔĆæŢřæĹŰèőÿåŕźä¡ăå¿ĹæIJL'åÿőåŁl'ãĂĆ
æŕŤåęĆïijŇåIJÍäÿŁéÍćçŽĎèŕĄåĹÿä¿Ńå■Řäÿ■ïijŇä¡ăåŔŕèČ¡äijŽèĂČèŹŚäÿŃéÍćçŽĎåőđçŐřçL'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

çŐřåIJÍæIJL'åď'Žäÿłå■ŮåĚÿæĹŰèĂĚæŸăåřĎïijŇä¡ăæČşåřĘåőČäžňäžŐéĂżè¿ŚäÿŁåŘĹåźűäÿžäÿĂäÿłå■
æŕŤåęĆæ§ěæL'¿åĂijæĹŰèĂĚæčĂæ§ěæ§ŘäžŻéŤőæŸŕåŘęå■ŸåIJÍãĂĆ

èğčåĘşæŰźæąĹ

åĄĞåęĆä¡ăæIJL'åęĆäÿŃäÿď'äÿłå■ŮåĚÿ:

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

çŐřåIJÍåĄĞèő¿ä¡ää£Ëéążå IJÍäÿd'äÿłå■ŮåĚÿäÿ■æL'ğèąŇæ§ěæL'¿æŞ■ä¡IJïijĹæŕŤåęĆåĚĹäżŐ a äÿ■æL'¿ïijŇåęĆæđIJæL'¿äÿ■åĹřåĘ■åIJÍ b äÿ■æL'¿ïijL'ãĂĆ äÿĂäÿłéÍđåÿÿçőĂå■ŢçŽĎèğčåEşæŰźæąĹářśæŸŕä¡£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 æŐěåŔŮåd'Žäÿłå■ŮåĚÿåźűåřĘåőČäżňåIJÍéÀżè¿ŚäÿŁåŔŸäÿžäÿĂäÿłå■ŮåĚÿãĂĆ çĎűåŘŐïijŇè£ŹäžŻå■ŮåĚÿåźűäÿ■æŸŕçIJ§çŽĎåŘĹåźűåIJÍäÿĂètůäžĘïijŇ ChainMap çśżåŔłæŸŕåIJÍåĘĚéČÍåĹŻåżžäžĘäÿĂäÿłåőźçžşè£ŹäžŻå■ŮåĚÿçŽĎåĹŮèąÍ åźűéĞ■æŰřåőŽäźL'äžĘäÿĂäžŻåÿÿèğAçŽĎå■ŮåĚÿæŞ■ä¡IJæÌěéĄ■åŐĘè£ŹäÿłåĹŮèąÍãĂĆåd'ğéČÍåĹĘå■ŮåĚ

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

åęĆæđIJåĞžçŐřéĞ■åd'■éŤőïijŇéĆčäźĹçňňäÿĂæňąåĞžçŐřçŽĎæŸăářĎåĂijäijŽèćńè£ŤåŻđãĂĆ åŻăæ■d'ïijŇä¿Ńå■ŘçÍŃåžŔäÿ■çŽĎ c['z'] æĂżæŸŕäijŽè£ŤåŻđå■ŮåĚÿ a äÿ■åŕźåžŤçŽĎåĂijïijŇèĂŇäÿ■æŸŕ b äÿ■åŕźåžŤçŽĎåĂijãĂĆ

åŕźäžŐå■ŮåĚÿçŽĎæŻt'æŰřæĹŰåĹăéŹd'æŞ■ä¡IJæĂżæŸŕå¡śåŞ■çŽĎæŸŕåĹŮèąÍäÿ■çňňäÿĂäÿłå■ŮåĚÿã

```
>>> 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¡IJçŤÍèŇČåŻt'åŔŸéĞŔïijĹæŕŤåęĆ globals , locals ç■L'ïijL'æŸŕéÍđåÿÿæIJL'çŤÍçŽĎãĂĆ äžŃåőđäÿŁïijŇæIJL'äÿĂäžŻæŰźæşŢåŔŕäžëä¡£åőČåŔŸå¿ŰçőĂå■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¡ăåŔŕèČ¡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¡ĘæŸŕåőČéIJÀèęĄä¡ăåĹŻåżžäÿĂäÿłåőŇåĚĺäÿ■åŘŇçŽĎå■ŮåĚÿåŕźèśąï
åŘŇæŮűïijŇåęĆæđIJåŐ§å■ŮåĚÿåĄŽäžĘæŻťæŰřïijŇè£Źçğ■æŤźåŔŸäÿ■äijŽåŔ■åžŤåĹřæŰřçŽĎåŘĹåźűå■Ů

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

`ChainMap` ä¡£çŤĺåŐ§æİěçŽĎå■ŮåĚÿïijŇåőČèĞłåůśäÿ■åĹŻåżžæŰřçŽĎå■ŮåĚÿãĂĆæL'ĂäžěåőČåźűäÿ

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

(continues on next page)

(çż■äÿŁéąţ)

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

4 çňňäžŇçńăïijŽå■ŮçňęäÿšåŠŇæŰĞæIJň

åĞäăźŐæĽĂæIJĽæIJĽçŤĺçŽĎçĺŇåžŔéČ¡äijŽæűĽåŔŁåĹřæ§ŘäžŽæŰĞæIJňåďĎçŘĘïijŇäÿ■çőąæŸŕèğ
è£ŹäÿĂçńăåřĘéĞ■çĆźåĚşæşĺæŰĞæIJňçŽĎæŞ■ä¡IJåďĎçŘĘïijŇæŕŤåęĆæŘŘåŔŰå■ŮçňęäÿšïijŇæŘIJçť ćïijŇ
åďğéČĺåĹĘçŽĎéŮőéćŸéČ¡èČ¡çőĂå■ŢçŽĎèřČçŤĺå■ŮçňęäÿšçŽĎåĘĚåžžæŰźæşŢåőŇæĹŘãĂĆ
ä¡ĘæŸŕïijŇäÿĂäžŻæŻť äÿžåď■æİćçŽĎæŞ■ä¡IJåŔŕèČ¡éIJĂèęĄæ■čåĹŹèąĺè¿¿åijŔæĹŰèĂĚåijžåďğçŽĎèğčæ
åźűäÿŤåIJĺæŞ■ä¡IJUnicodeæŮűåĂŹçćřåĹřçŽĎäÿĂäžŻæčŸæĽŃçŽĎéŮőéćŸåIJĺè£ŹéĞŇäź§äijŽèćńæŘŘåŔ

Contents:

4.1 2.1 ä¡£çŤĺåďŽäÿłçŢŇåőŽçňęåĹĘåĽšå■Ůçňęäÿš

éŮőéćŸ

ä¡ăéIJĂèęĄåřĘäÿĂäÿłå■ŮçňęäÿšåĹĘåĽšäÿžåďŽäÿłå■ŮæőţïijŇä¡ĘæŸŕåĹĘéŽŤçňę(è£ŸæIJĽåŚĺåŻť çŽĎ

èğčåĘşæŰźæąĹ

`string` åŕźèśąçŽĎ `split()` æŰźæşŢåŔłéĂĆåžŤäžŐéİđåÿÿçőĂå■ŢçŽĎå■ŮçňęäÿšåĹĘåĽšæČĚå¡ćïij
åőČåźűäÿ■åĚĄèőÿæIJĽåďŽäÿłåĹĘéŽŤçňęæĹŰèĂĚæŸŕåĹĘéŽŤçňęåŚĺåŻť äÿ■çąőåőŽçŽĎçľžæăijãĂĆ
å¡Şä¡ăéIJĂèęĄæŻť åŁăçĄţæť żçŽĎåĹĞåĽšå■ŮçňęäÿšçŽĎæŮűåĂŹïijŇæIJĂåě¡ä¡£çŤĺ `re.split()` æŰźæşŢïijŽ

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

èőĺèőž

åĞ¡æŢř `re.split()` æŸŕéİđåÿÿåőđçŤĺçŽĎïijŇåŽäåÿžåőČåĚĄèőÿä¡ăäÿžåĹĘéŽŤçňęæŇĞåőŽåďŽäÿł
æŕŤåęĆïijŇåIJĺäÿŁéİćçŽĎä¿Ńå■Řäÿ■ïijŇåĹĘéŽŤçňęåŔŕäżěæŸŕéĂŮåŔůïijŇåĹĘåŔůæĹŰèĂĚæŸŕçľžæăijïijŇ
åŔłèęĄè£ŹäÿłæĺąåijŕèćńæĽ¿åĹřïijŇéĆčäźĹåŇźéĚ■çŽĎåĹĘéŽŤçňęäÿďè¿źçŽĎåőđä¡ŞéČ¡äijŽèćńå¡ŞæĹŘæ
è£ŤåŽđçżŞæđIJäÿžäÿĂäÿłå■ŮæőţåĺůèąĺïijŇè£Źäÿłèů§ `str.split()`
è£ŤåŽđåĂijçśżåđŃæŸŕäÿĂæăůçŽĎãĂĆ

å¡Şä¡ää¡£çŤÍ `re.split()` åĞ¡æŢřæŮűåĂŹïijŇéIJĂèęĄçL'źåĹńæşĺæĎŔçŽĎæŸŕæ■čåĹŹèąĺè¿¿åijŔäÿ
åęĆæđIJä¡£çŤĺäžĘæ■ŢèŐůåĹĘçżĎïijŇéĆčäźĹèćńåŇźéĚ■çŽĎæŰĞæIJňäź§ářĘåĞžçŐřåIJĺçżŞæđIJåĹŮèąĺäÿ

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

èŐůåŔŰåĹĘåL'šå■Ůçňęåijĺæ§ŘäžŻæČĚåĘţäÿŃäź§æŸŕæIJL'çŤĺçŽĎãĂĆ
æŕŤåęĆïijŇä¡ăåŔŕèČ¡æČşä£ĺçŢŹåĹĘåL'šå■ŮçňęäÿšïijŇçŤĺæĺěåIJĺåŘŐéĺćéĞ■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£ĺçŢŹåĹĘåL'šå■ŮçňęäÿšåĹřçżŞæđIJåĹŮèąĺäÿ■åŐżïijŇä¡Ęäż■çĎűéIJĂèęĄä¡£çŤĺåĹ
çąőä£Ïä¡ăçŽĎåĹĘçżĎæŸŕéĺđæ■ŢèŐůåĹĘçżĎïijŇå¡ćåęĆ `(?:...)` ãĂĆæŕŤåęĆïijŽ

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

4.2 2.2 å■ŮçňęäÿšåijĂåď'æĹŰçżŞář¿åŇźéĚ■

éŮőécŸ

ä¡ăéIJĂèęĄéĂŽè£ĞæŇĞåőŽçŽĎæŰĞæIJňæĺąåijŔåŐżæčĂæ§ěå■ŮçňęäÿšçŽĎåijĂåď'æĹŰèĂĚçżŞář¿
Schemeç■L'ç■L'ãĂĆ

èğčåĘşæŰźæąĹ

æčĂæ§ěå■ŮçňęäÿšåijĂåď'æĹŰçżŞář¿çŽĎäÿĂäÿłçőĂå■ŢæŰźæşŢæŸŕä¡£çŤĺ `str.startswith()` æĹŰèĂĚæŸŕ `str.endswith()` æŰźæşŢãĂĆæŕŤåęĆïijŽ

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

(continues on next page)

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

åęĆæđIJä¡ăæČşæčĂæ§ěåď'Žçğ■åŇźéĚ■åŔŕèČ¡ïijŇåŔłéIJĂèęĄåřĘæL'ĂæIJL'çŽĎåŇźéĚ■éąźæŤ¿åĚěåĹi
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
>>>
```

äÿŃéİćæŸŕåŔęäÿĂäÿłä¿Ńå■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()
```

åěĞæĂłçŽĎæŸŕïijŃè£ŹäÿłæŰźæşŢäÿ■å£Ěéążèęą¿ŞåĚěäÿĂäÿłåĚČçżĎä¡IJäÿžåŔĆæŢřãĂĆ
åęĆæđIJä¡ăæĄřåůğæIJL'äÿĂäÿł `list` æĹŰèĂĚ `set` çśżåđŃçŽĎéĂL'æŃl'éąźïijŃ
èęĄçąőä£İäijăéĂŠåŔĆæŢřåL'■åĚĹèřČçŤĺ `tuple()` åřĘåĚűè¡ñæ■ćäÿžåĚČçżĎçśżåđŃãĂĆæŕŤåęĆïijŽ

```
>>> 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()` æŰźæşŢæŔŘä¿ŻäžĘäÿĂäÿłéİđåÿÿæŰźä¿£çŽĎæŰźåijŔåŐżåĄŽå
çśżäijijçŽĎæŞ■ä¡IJäź§åŔŕäżěä¡£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¡ååŔŕäžěèČ¡è£ŸæČşä¡£çŤĺæ■čåĹŹèąĺè¿¿åijŔåŐżåőđçŐřïijŇæŕŤåęĆïijŽ

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

è£Źçğ■æŰźåijŔäź§èąŇå¿ŮéĂŽïijŇä¡ĘæŸŕåŕźäžŐçőĂå■ŢçŽĎåŇźéĚ■åőđåIJĺæŸŕæIJL’çĆźåřŔæİŘåd’ğ

æIJĂåŘŐæŔŘäÿĂäÿŃïijŇå¡ŞåŠŇåĚűäżŰæŞ■ä¡IJæŕŤåęĆæŹőéĂŽæŢřæ■őèĄŽåŘĹçŻÿçżŞåŘĹçŽĎæŮű
`startswith()` åŠŇ `endswith()` æŰźæşŢæŸŕå¿Ĺäÿ■éŤŹçŽĎãĂĆ
æŕŤåęĆïijŇäÿŃéİćè£Źäÿłèŕ■åŘěæčĂæ§ěæ§Řäÿłæ ŰĞäžűåď źäÿ■æŸŕåŖęå■ŸåIJĺæŇĞåőŽçŽĎæŰĞäžűçśžåđ

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

4.3 2.3 çŤĺShelléĂŽéĚ■çňęåŇźéĚ■å■Ůçňęäÿš

éŮőécŸ

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

èğčåĘşæŰźæąĹ

`fnmatch` æĺąåİŮæŔŘä¿ŻäžĘäÿd’äÿłåĞ¡æŢřâĂŤâĂŤ `fnmatch()` åŠŇ `fnmatchcase()` ïijŇåŔŕäžęçŤĺæİěåőđçŐřè£ŹæăůçŽĎåŇźéĚ■ãĂĆçŤĺæşŢåęĆäÿŃïijŽ

```
>>> 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)

(çż■äÿŁéąţ)

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

fnmatch() åĞ¡æŢřä¡£çŤĺåžŢåśĆæŞ■ä¡IJçşżçż§çŽĎåď ğåřŔåĘŹæŢŔæĎ§èğĎåĹŹ(äÿ■åŘŇçŽĎçşżçż§

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

åęĆæđIJä¡ăåŕźè£ŹäÿłåŇžåĹńå¿ĹåIJĺæĎŔïijŇåŔŕäžěä¡£çŤĺ fnmatchcase() æİěäžčæŻ£ãĂĆåőČåőŇåĚĺä¡£çŤĺä¡ăçŽĎæĺąåijŔåď ğåřŔåĘŹåŇźéĚ■ãĂĆæŕŤåęĆïijŽ

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

è£Źäÿď äÿłåĞ¡æŢřéĂŽåÿÿäijŽèćńå£¡çŢěçŽĎäÿĂäÿłçL'źæĂğæŸŕåIJĺåď ĎçŘĘéİđæŰĞäžűåŘ■çŽĎå■Ůç æŕŤåęĆïijŇåĄĞèő¿ä¡ăæIJL'äÿĂäÿłèąÛéĄŞåIJŕåİĂçŽĎåĹÛèąĺæŢřæ■őïijŽ

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

ä¡ăåŔŕäžěåČŔè£ŹæăůåĘŹåĹÛèąĺæŐĺåŕijïijŽ

```
>>> 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']
>>>
```

èőÍèőž

fnmatch() åĞ¡æŢřåŇźéĚ■èČ¡åŁŻäżŇäžŐçőĂå■ŢçŽĎå■ŮçňęäÿšæŰźæşŢåŠŇåijžåď ğçŽĎæ■čåĹŹèą åęĆæđIJåIJĺæŢřæ■őåď ĎçŘĘæŞ■ä¡IJäÿ■åŔłéIJĂèęĄçőĂå■ŢçŽĎéĂŽéĚ■çňęářśèČ¡åőŇæĹŘçŽĎæŮűåĂŹïijŇ

åęĆæđIJä¡ăçŽĎäžčçăĄéIJĂèęĄåĄŽæŰĞäžűåŘ■çŽĎåŇźéĚ■ïijŇæIJĂåě¡ä¡£çŤĺ glob æĺąåİŮãĂĆåŔĆèĂČ5.13åřŔèŁĆãĂĆ

4.4 2.4 å■Ůçňęäÿšå ŇźéĚ■åŠŇæŘIJçťć

éŮőécŸ

ä¡ăæČşåŇźéĚ■æĹŰèĂĚæŘIJçťćçĽźåőŽæĺąåijŔçŽĎæŰĞæIJň

èğčåĘşæŰźæąĹ

åęĆædIJä¡ăæČşåŇźéĚ■çŽĎæŸŕå■ŮéIćå■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■æIĆçŽĎåŇźéĚ■éIJĂèęĄä¡£çŤĺæ■čåĹŹèąĺè¿¿åijŔåŠŇ `re` æĺąåİŮãĂĆ
äÿžäžĘèğčéĞŁæ■čåĹŹèąĺè¿¿åijŔçŽĎå§žæIJňåŐ§çŘĘïijŇåĄĞèő¿ä¡ăæČşåŇźéĚ■æŢřå■ŮæăijåijŔçŽĎæŮěæ
`11/27/2012` ïijŇä¡ăåŔŕäžěè£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
>>>
```

åęĆædIJä¡ăæČşä¡£çŤĺåŘŇäÿĂäÿłæĺąåijŔåŐżåĄŽåď Žæňąå ŇźéĚ■ïijŇä¡ååžŤèŕěåĚĹářĘæĺąåijŔå■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■ŮçňęäÿšåijĂåğŇåŐżåŇźéĚ■ïijŇåęĆædIJä¡ăæČşæ§ěæL'¿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']
>>>
```

åIJĺåőŽäźL'æ■čåĹŹåijŔçŽĎæŮűåĂŹïijŇéĂŽåÿÿäijŽåĹl'çŤĺæŇňåŔůåŐżæ■ŢèŐůåĹĘçżĎãĂĆæŕŤåęĆïij

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

æ■ŢèŐůåĹĘçżĎåŔŕäżěä¡£å¿ŮåŘŐéİćçŽĎåď ĎçŘĘæŻt'åŁäçőĂå■ŢïijŇåŻääÿžåŔŕäżěåĹĘåĹńářĘæŕŔäÿ

```
>>> 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)

(çż■äÿŁéąţ)

```
>>> 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 ćæŰĞæIJňåźűäżĕåĹŮèąĺå¡ćåijŔè£ŤåŻđæĽ ĂæIJĽçŽĎåŃźéĔ■ãĂĆ åęĆæđIJä¡ăæČşäżĕè£■äżčæŰźåijŔè£ŤåŻđåŃźéĔ■ïijŇåŘŕäżĕä¡£çŤÍ finditer() æŰźæşŢæĬĕäżčæŻ£ïijŇæŕŤåęĆïijŽ

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

èőÍèőž

åĔşäžŐæ■čåĹŹèąÍè¿¿åijŔçŘĘèőžçŽĎæŢŹçÍŃåůšçżŔèűĔåĞžäžĘæIJňäźęçŽĎèŇČåŻť ãĂĆ äÿ■è£ĞïijŇè£Źäÿ ĂèŁĆéŸŘè£řäžĘä¡£çŤÍreæĺąåİŮè£Žèąŋåŋžé■åŠŇæŘIJçť ćæŰĞæIJňçŽĎæIJĂå§žæIJňæ æäÿå£Čæ■ěéłď åřśæŸŕåĔĹä¡£çŤÍ re.compile() çijŰèŕŚæ■čåĹŹèąÍè¿¿åijŔå■ŮçňęäÿšïijŇ çĎűåŘŐä¡£çŤÍ match() , findall() æĹŰèĂĚ finditer() ç■Ľ æŰźæşŢãĂĆ

å¡ŞåĘŹæ■čåĹŹåijŔå■Ůçňęäÿšç ŽĎæŮűåĂŹïijŇçŻÿåŕźæŹőéĄ■çŽĎåĄŽæşŢæŸŕä¡£çŤÍåŐ§åğŃå■Ůçňęä r'(\d+)/(\d+)/(\d+)' ãĂĆ è£Źçğ■å■Ůçňęäÿšåř Ęäÿ■åŐżèğčæđŘåŔ■æŰIJæİăïijŇè£ŹåIJÍæ■čåĹŹèąÍè åęĆæđIJäÿ■è£Źæăůå ĄŽçŽĎèŕİïijŇä¡ăå£Ěéążä¡£çŤÍäÿď äÿłåŔ■æŰIJæİăïijŇçśżäijij '(\\d+)/(\\d+)/(\\d+)' ãĂĆ

éIJĂèęĄæşÍæĎŔçŽĎæŸŕ match() æŰźæşŢäżĔäżĔæčĂæ§ěå■Ůçňęäÿšç ŽĎåijĂåğŃéČĺåĹĘãĂĆåőČçŽ

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

åęĆæđIJä¡ăæČşçš¿çąőåŃźéĔ■ïijŇçąőä£İä¡ăçŽĎæ■čåĹŹèąÍè¿¿åijŔäžě$çżŞåř¿ïijŇåřśåČŔè£ŹäźĹè£Źæăů

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

æIJÄåŘŐïijŇåęĆæđIJä¡ăäžĚäžĚæŸŕåĄŽäÿĂæňąçőĂå■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■ŸètůæİěïijŇåŻăæ■ď'åźűäÿ■äijŽæűĹ
ä¡ĘæŸŕåęĆæđIJä¡£çŤĺécĎçijŰèŕŚæĺąåijŘçŽĎèŕİïijŇä¡ăářĘäijŽåĞŘářŚæ§ěæĽ'¿åŠŇäÿĂäžŽécİåď'ŰçŽĎåď'Ď

4.5 2.5 å■ŮçňęäÿšæŘIJçť´ćåŠŇæŻ£æ■ć

éŮőécŸ

ä¡ăæČşåIJĺå■Ůçňęäÿšäÿ■æŘIJçť´ćåŠŇåŇźéĚ■æŇĞåőŽçŽĎæŰĞæIJňæĺąåijŘ

èğčåĘşæŰźæąĹ

åŕźäžŐçőĂå■ŢçŽĎå■ŮéİćæĺąåijŘïijŇçŻť´æŐěä¡£çŤÍ `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¡£çŤÍ `re` æĺąåİŮäÿ■çŽĎ `sub()` åĞ¡æŢřãĂĆ äÿžäžĘęŕť´æŸŐè£ŹäÿłïijŇåĄĞèő¿ä¡äæČşåřĘå¡ćåijŘäÿž `11/27/2012` çŽĎæŮěæIJ§å■ŮçňęäÿšæŤźæĹŘ `2012-11-27` ãĂĆçď'žä¿ŃåęĆäÿŇïijŽ

```
>>> 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■çŽĎçňňäÿĂäÿłåŘĆæŢřæŸŕèćńåŇźéĚ■çŽĎæĺąåijŘïijŇçňňäžŇäÿłåŘĆæŢřæŸŕæŻ£æ
`\3` æŇĞåŘŠåĽ■éİćæĺąåijŘçŽĎæ■ŢèŐůçżĎåŮãĂĆ

åęĆæđIJä¡ăæĽ'ŞçőŮçŤÍçŻÿåŘŇçŽĎæĺąåijŘåĄŽåď'Žæňąæ Ż£æ■ćïijŇèĂČèŹŠåĚĹçijŰèŕŚåőČæİěæŘŘå■

```
>>> 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´åŁåđ'■æİĆçŽĎæŻ£æ■ćïijŇåŘŕäžěäijăéĂŠäÿĂäÿłæŻ£æ■ćåŻđèřČåĞ¡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¡æŢřçŽĎåŔĆæŢřæŸŕäÿĂäÿł match åŕźèśąïijŇäź§åřśæŸŕ match() æĹŰèĂĚ find() è£ŤåŻđçŽĎåŕźèśąãĂĆ ä¡£çŤĺ group() æŰźæşŢæİěæŔŘåŔŰçL'źåőŽçŽĎåŇźéĚ■éČĺåĹĘãĂĆåŻđèřČåĞ¡æŢřæIJĂåŘŐè£ŤåŻđæŻ£æ■ćå■Ůçňęäÿšã

åęĆæđIJéŹđ'äžĘæŻ£æ■ćåŘŐçŽĎçżŞæđIJåđ'ŰïijŇä¡äè£ŸæČşç§ěéĄŞæIJL'åđ'ŽåŕŚæŻ£æ■ćåŔŚçŤ§äžĘï 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
>>>
```

èőĺèőž

åĚşäžŐæ■čåĹŹèąĺè¿¿åijŔæŘIJçt'ćåŠŇæŻ£æ■ćïijŇäÿŁéİćæijŤçđ'žçŽĎ sub() æŰźæşŢå§žæIJňåůšçżŔæűţçŻŰäžĘæL'ĂæIJL'ãĂĆ åĚűåőđæIJĂéŽ¿çŽĎéČĺåĹĘåŕśæŸŕçijŰåĘŹæ■čåĹŹèąĺè¿

4.6 2.6 å■Ůçňęäÿšå£¡çŢěåđ'ğåŕŔåĘŹçŽĎæŘIJçt'ćæŻ£æ■ć

éŮőécŸ

ä¡ăéIJĂèęĄäżěå£¡çŢěåđ'ğåŕŔåĘŹçŽĎæŰźåijŔæŘIJçt'ćäÿŐæŻ£æ■ćæŰĞæIJňå■Ůçňęäÿš

èğčåĘşæŰźæąĹ

äÿžäžĘåIJĺæŰĞæIJňæŞ■ä¡IJæŮűå£¡çŢěåđ'ğåŕŔåĘŹïijŇä¡äéIJĂèęĄåIJĺä¡£çŤĺ re æĺąåİŮçŽĎæŮűåĂŹçżŹè£ŹäžŻæŞ■ä¡IJæŔŘä¿Ż re.IGNORECASE æăĞå£ŮåŔĆæŢřãĂĆæŕŤåęĆïijŽ

```
>>> 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'
>>>
```

æIJÄåŘŐçŽĎéĆčäÿłä¿Ńå■ŘæŔ■çď žäžĘäÿĂäÿłářŔçijžéŹůïijŇæŻ£æ■ćå■Ůçňęäÿšåźűäÿ■äijŽèĞłåŁĺèů§äÿžäžĘä£őåď■è£ŹäÿłïijŇä¡ääŔŕèČ¡éIJĂèęĄäÿĂäÿłè¿ĚåŁ'åĞ¡æŢřïijŇářšåČŔäÿŃéİćçŽĎè£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¡£çŤĺäÿŁè£řåĞ¡æŢřçŽĎæŰźæşŢïijŽ

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

èŕŚèĂĚæşĺïijŽ `matchcase('snake')` è£ŤåŻđäžĘäÿĂäÿłåŻđèřČåĞ¡æŢř(åŔĆæŢřå£ĚéążæŸŕ `match` åŕźèśą)ïijŇåĽ■éİćäÿĂèŁĆæŔŘåĹřè£ĞïijŇ `sub()` åĞ¡æŢřéŹď'äžĘæŐěåŔŮæŻ£æ■ćå■Ůçňęäÿšåď'ŰïijŇè£ŸèČ¡æŐěåŔŮäÿĂäÿłåŻđèřČåĞ¡æŢřãĂĆ

èőĺèőž

åŕźäžŐäÿĂèĹňçŽĎå£¡çŢěåď'ğåřŔåĘŹçŽĎåŃźéĚ■æŞ■ä¡IJïijŇçőĂå■ŢçŽĎäijäéĂŠäÿĂäÿł `re.IGNORECASE` æăĞå£ŮåŔĆæŢřåřšåůšçżŔèűşåď'§äžĘãĂĆä¡ĘæŸŕéIJĂèęĄæşĺæĎŔçŽĎæŸŕïijŇè£ŹäÿłåŕźäžŐæ§ŘäžŻéIJĂèęĄåď'ğåřŔåĘŹè¡ñæ■ćçŽĎUnicodeåŃźéĚ■ååŔĆèĂČ2.10åřŔèŁĆäžĘèğčæŻť'åď'ŽçżĘèŁĆãĂĆ

4.7 2.7 æIJĂç§■åŃźéĚ■æĺąåijŔ

éŮőécŸ

ä¡ăæ■čåIJĺèŕŢçİĂçŤĺæ■čåĹŹèąĺè¿¿åijŔåŃźéĚ■æ§ŘäÿłæŰĞæIJñæĺąåijŔïijŇä¡ĘæŸŕåőČæĽ¿åĹřçŽĎæŸèĂŃä¡ăæČşä£őæŤźåőČåŔŸæĹŘæ§ěæĽ¿æIJĂç§■çŽĎåŔŕèČ¡åŃźéĚ■ãĂĆ

èğčåĘşæŰźæąĹ

è£ŹäÿłéŮőécŸäÿĂèĹňåĞžçŐřåIJĺéIJĂèęĄåŃźéĚ■äÿĂåŕźåĹĘéŽŤçňęäźŃéŮť'çŽĎæŰĞæIJñçŽĎæŮűåĂäÿžäžĘèŕť'æŸŐæÿĚæěŽïijŇèĂČèŹŚåęĆäÿŃçŽĎä¿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■ïijŇæÍąåijŔ r'\"(.*)\"' çŽĎæĎŔåŻ¿æŸŕåŇźéĚ■èćńåŔŇåijŢåŘůåŇĚåŘńçŽĎ
ä¡ĘæŸŕåIJÍæ■čåĹŹèąÍè¿¿åijŔäÿ■*æŞ■ä¡IJçňęæŸŕèťłål'łçŽĎïijŇåŻăæ■ď'åŇźéĚ■æŞ■ä¡IJäijŽæ§ěæL'¿æIJÁéţ
äžŐæŸŕåIJÍçňňäžŇäÿłä¿Ńå■Řäÿ■æŘIJçť'ć text2 çŽĎæŮűåĂŹè£ŤåŻđçżŞæđIJåźűäÿ■æŸŕæĹŚäžňæČşèęĄç

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

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

è£Źæăůåřśä¡£å¿ŮåŇźéĚ■åŘŸæĹŘéİđèť'łål'łæÍąåijŔïijŇäžŐèĂŇå¿ŮåĹřæIJĂç§■çŽĎåŇźéĚ■ïijŇäź§åřśa

èőÍèőž

è£ŹäÿĂèŁĆåśŢçd'žäžĘåIJÍåĘŹåŇĚåŘńçĆź(.)å■ŮçňęçŽĎæ■čåĹŹèąÍè¿¿åijŔçŽĎæŮűåĂŹéĄĞåĹřçŽĎäÿ
åIJÍäÿĂäÿłæÍąåijŔå■Ůçňęäÿšäÿ■ïijŇçĆź(.)åŇźéĚ■éŹd'äžĘæ■ćèąŇåd'ŰçŽĎäżżä¡Ţå■Ůçňęã ĂĆ
çĎűèĂŇïijŇåęĆæđIJä¡ăářĘçĆź(.)åŔůæŤ¿åIJÍåijĂåğŃäÿŐçżŞæİ§çňę(æŕŤåęĆåijŢåŔů)äźŃéŮt'çŽĎæŮűåĂŹi
è£ŹæăúéĂŽåÿÿäijŽářijèĞt'å¿Ĺåd'Žäÿ■éŮt'çŽĎèćńåijĂåğŃäÿŐçżŞæİ§çňęåŇĚåŘńçŽĎæŰĞæIJňèćńå£¡çŢěa
éĂŽè£ĞåIJÍ * æĹŰèĂĚ + è£ŹæăůçŽĎæŞ■ä¡IJçňęåŘŐéİćæůżåŁääÿĂäÿł ?
åŘŕäžěåijžåĹűåŇźéĚ■çőŮæşŢæŤźæĹŘářżæL'¿æIJĂç§■çŽĎåŔŕèČ¡åŇźéĚ■ãĂĆ

4.8 2.8 åď'ŽèąŇåŇźéĚ■æÍąåijŔ

éŮőécŸ

ä¡ăæ■čåIJÍèŕŢçİĂä¡£çŤÍæ■čåĹŹèąÍè¿¿åijŔåŐżåŇźéĚ■äÿĂåd'ğåİŮçŽĎæŰĞæIJňïijŇèĂŇä¡ăéIJĂèęĄèů

èğčåĘşæŰźæąĹ

è£ŹäÿłéŮőécŸå¿ĹåĚÿåđŃçŽĎåĞžçŐřåIJÍå¡Şä¡ăçŤÍçĆź(.)åŐżåŇźéĚ■äżżæĎŔå■ŮçňęçŽĎæŮűåĂŹïijŇa
æŕŤåęĆïijŇåĄĞèő¿ä¡ăæČşèŕŢçİĂåŐżåŇźéĚ■Cèŕ■èÍĂåĹĘåL'šçŽĎæşÍéĞŁïijŽ

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

(continues on next page)

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

äÿžäžĘä£őæ■čè£ŹäÿłéŮőécŸïijŇä¡ăåŘŕäžěä£őæŤźæÍąåijŔå■ŮçňęäÿšïijŇåćđåŁåářźæ■ćèąŇçŽĎæŤŕæŇ

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

åIJÍè£ŹäÿłæÍąåijŔäÿ■ïijŇ (?:.|\n) æŇĞåőŽäžĘäÿĂäÿłéİđæ■ŢèŐůçżĎ (äź§åřśæŸŕåőČåőŽäźL'äžĘäÿĂäÿłäżĚäżĚçŤÍæİěåĄŽåŇźéĚ■ïijŇèĂŇäÿ■èČ¡éĂŽè£Ğå■ŢçŃňæ■ŢèŐůæŁŰèĂ

èőÍèőž

re.compile() åĞ¡æŢřæŐěåŘŮäÿĂäÿłæăĞå£ŮåŘĆæŢřåŔń re.DOTALL ïijŇåIJÍè£ŹéĞŇéİđåÿÿæIJL'çŤÍãĂĆ åőČåŘŕäžěèől'æ■čåŁŹèąÍè¿¿åijŔäÿ■çŽĎçĆź(.)åŇźéĚ■åŇĚæŃňæ■ćèąŇ

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

åŕźäžŐçőĂå■ŢçŽĎæČĚåĘţä¡£çŤÍ re.DOTALL æăĞèőřåŘĆæŢřåůěä¡IJçŽĎå¿Ĺåě¡ïijŇ ä¡ĘæŸŕåęĆædIJæÍąåijŔéİđåÿÿåd'■æİĆæĹŰèĂĚæŸŕäÿžäžĘædĎéĂăå■Ůçňęäÿšäžd'çL'ŇèĂŇårĘåd'Žäÿłæ Íąåi è£ŹæŮűåĂŹä¡£çŤÍè£ŹäÿłæăĞèőřåŘĆæŢřåršåŔŕèČ¡åĞžçŐřäÿĂäžŻéŮőécŸãĂĆ åęĆædIJèől'ä¡ăéĂL'æŃl'çŽĎèŕÍïijŇæIJĂåě¡è£ŸæŸŕåőŽäźL'èĞłåůśçŽĎæ■čåŁŹèąÍè¿¿åijŔæÍąåijŔïijŇè£Źæă

4.9 2.9 åřĘUnicodeæŰĞælJňæăĞåĞĘåŇŰ

éŮőécŸ

ä¡ăæ■čåIJÍåd'ĎçŘĘUnicodeå■ŮçňęäÿšïijŇéIJĂèęAçąőä£ÍæL'ĂæIJL'å■ŮçňęäÿšåİJÍåžŢåśĆæIJL'çŻÿåŘŇ

èğčåĘşæŰźæąĹ

åIJÍUnicodeäÿ■ïijŇæ§ŘäžŻå■Ůçňęèč¡åd'§çŤÍåd'Žäÿłå ŘĹæşŢçŽĎçijŰçăĄèąÍçd'žãĂĆäÿžäžĘèŕt'æŸŐïij

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

(continues on next page)

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

è£ŹéĞŇçŽĎæŰĞæIJňâĂİSpicy JalapeÃśoâĂİä¡£çŤĺäžĘäÿď˙çğ■å¡ćåijŔæİěèąĺçď˙žãĂĆ çňňäÿĂçğ■ä¡£çŤĺæŢt˙ä¡Şå■Ůçňęâ ĂİÃśâĂİ(U+00F1)ïijŇçňňäžŇçğ■ä¡£çŤĺæŃL˙äÿĄå■Ůæŕ■âĂİnâĂİåŘŐéİćè

åIJĺéIJÄèęĄæŕŤè¿Čå■ŮçňęäÿšçŽĎçĺŃåžŔäÿ■ä¡£çŤĺå■Ůçňęç ŽĎåď˙Žçğ■èąĺçď˙žäijŽäžğçŤ§éŮőécŸãĂĆ äÿžäžĘä£őæ■čè£ŹäÿłéŮőécŸïijŇä¡ääŔŕäżěä¡£çŤĺunicodedataæĺąåİŮåĚĹářĘæŰĞæIJňæăĞåĞĘåŇŰïijŽ

```
>>> 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() çňňäÿĂäÿłåŔĆæŢřæŇĞåőŽå■ŮçňęäÿšæăĞåĞĘåŇŰçŽĎæŰźåijŔãĂĆ NFCèąĺçď˙žå■ŮçňęåžŤèŕěæŸŕæŢt˙ä¡ŞçżĎæĹŘ(æŕŤåęĆåŔŕèČ¡çŽĎèŕİåřśä¡£çŤĺå■ŢäÿĂçijŰçăA)ïijŇèĂŇNFI

PythonåŘŇæăůæŤŕæŇĄæĽ˙l˙åśŢçŽĎæăĞåĞĘåŇŰå¡ćåijŔNFKCåŠŇNFKDïijŇåőČäžňåIJĺåď˙DçŘĘæ§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

æăĞåĞĘåŇŰåŕžäŐäżżä¡ŢéIJĂèęĄäžěäÿĂèĞťçŽĎæŰźåijŔåď ĎçŘĘUnicodeæŰĞæIJñçŽĎçÍŇåžŔéČ¡å¡Şåď ĎçŘĘæÍěèĞłçŤÍæĹůè¿ŞåĚěçŽĎå■Ůçňęäÿšè ĂŇä¡ăå¿ĹéŽ¿åŐżæŐğåĹűçijŰçăĄçŽĎæŮűåĂŹåřď åĚűåę

åIJÍæÿĚçŘĘåŠŇè£Ğæžď æŰĞæIJñçŽĎæŮűåĂŹå■Ůçňę çŽĎæăĞåĞĘåŇŰäź§æŸŕå¿ĹéĞ■èęĄçŽĎãĂĆ æŕŤåęĆïijŇåĄĞèő¿ä¡ăæČşæÿĚéŹď æŐL'äÿĂäžŻæŰĞæIJňäÿŁéÍćçŽĎåŔŸé§şçňęçŽĎæŮűåĂŹ(åŔŕèČ¡æŸŕäÿ

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

æIJĂåŘŐäÿĂäÿłä¿Ńå■ŘåśŢçď žäžĘ `unicodedata` æĺąåİŮçŽĎåŘęäÿĂäÿłéĞ■èęĄæŰźéÍćïijŇäź§åřśæ `combining()` åĞ¡æŢřåŔŕäžěætŃèŕŢäÿĂäÿłå■ŮçňęæŸŕåŘęäÿžåŠŇé§şå■Ůçňęã ĂĆ åIJÍè£ŹäÿłæĺąåİŮäÿ■è£ŸæIJL'åĚűäžŰåĞ¡æŢřçŤÍäžŐæ§ěæL'¿å■ŮçňęçśżåĺńïijŇætŃèŕŢæŸŕåŘęäÿžæŢřå■Ůå

UnicodeæŸ¿çĎűæŸŕäÿĂäÿłå¿Ĺåď ğçŽĎäÿżécŸãĂĆåęĆæđIJæČşæŻť æůśåĚěçŽĎäžĘèğčåĚşäžŐæăĞåČ èŕůçIJŃèĂČ UnicodeåőŸç¡Śäÿ■åĚşäžŐè£ŹéČĺåĹĘçŽĎèŕťæŸŐ Ned BatchelderåIJÍ äżŰçŽĎç¡ŚçńŹ äÿŁåŕźPythonçŽĎUnicodeåď ĎçŘĘéŮőécŸäź§æIJL'äÿĂäÿłå¿Ĺåě¡çŽĎäżŃçż■ãĂĆ

4.10 2.10 åIJÍæ■čåĹŹåijŔäÿ■ä¡£çŤÍUnicode

éŮőécŸ

ä¡ăæ■čåIJÍä¡£çŤÍæ■čåĹŹèąÍè¿¿åijŔåď ĎçŘĘæŰĞæIJñïijŇä¡ĘæŸŕåĚşæşÍçŽĎæŸŕUnicodeå■Ůçňęåď Ďç

èğčåĘşæŰźæąĹ

éżŸèőď æČĚåĘţäÿŃ `re` æĺąåİŮåůšçżŘåŕźäÿĂäžŻUnicodeå■Ůçňęçśżæ IJL'äžĘå§žæIJñçŽĎæŤŕæŇĄãĂĆ æŕŤåęĆïijŃ `\\d` åůšçżŘåŇźéĚ■äżżæĎŘçŽĎunicodeæŢřå■Ůå■Ůçňęäžĕïijž

```
>>> 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¡ăæČşåIJÍæĺąåijŔäÿ■åŇĚåŘńæŇĞåőŽçŽĎUnicodeå■ŮçňęïijŇä¡ååŔŕäžěä¡£çŤÍUnicodeå■Ůçň `\uFFF` æĹŰèĂĚ `\UFFFFFFF`)ãĂĆ æŕŤåęĆïijŇäÿŃéİćæŸŕäÿĂäÿłåŇźéĚ■åĞääÿłäÿ■åŘŇéŸ£æŃL'äijŕçijŰç

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

å¡ŞæĽğèąŇåŇźéĚ■åŠŇæŘIJçť'ćæŞ■ä¡IJçŽĎæŮűůåĂŹïijŇæIJĂåě¡æŸŕåĚĹæăĞåĞĘåŇŰåźűäÿŤæÿĚçŘĚ
ä¡ĘæŸŕåŘŇæăůäź§åžŤèřěæşĺæĎŔäÿĂäžŻçĽźæőŁæČĚåĘţïijŇæŕŤåęĆåIJĺå£¡çŢěåď'ğářŔåĘŹåŇźéĚ■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'
>>>
```

èőÍèőž

æůůåŘĹä¡£çŤÍUnicodeåŠŇæ■čåĹŹèąÍè¿¿åijŔéĂŽåÿÿäijŽèőľä¡ăæŁŞçŃĆãĂĆ
åęĆæđIJä¡ăçIJ§çŽĎæĽŞçőŮè£Źæăůå ĄŽçŽĎèŕĬïijŇæIJĂåě¡èĂČèŹŚäÿŇåőĽèčĚçňňäÿĽæŰźæ■čåĹŹåijŔåžŞ
åőČäžňäijŽäÿžUnicodeçŽĎåď'ğářŔåĘŹè¡ňæ■ćåŠŇåĚűäžŰåď'ğéĞŔæIJĽèűčçĽźæĂğæŔŘä¿ŻåĚÍéİćçŽĎæŤŕ

4.11 2.11 åĹăéŹď'å■Ůçňęäÿšäÿ■äÿ■éIJĂèęĄçŽĎå■Ůçňę

éŮőécŸ

ä¡ăæČşåŐżæŐĽ'æŰĞæIJňå■ŮçňęäÿšåijĂåď'ťïijŇçżŞář¿æĹŰèĂĚäÿ■éŮť'äÿ■æČşèęĄçŽĎå■ŮçňęïijŇæŕ

èğčåĘşæŰźæąĹ

strip() æŰźæşŢèČ¡çŤĺäžŐåĹăéŹď'åijĂåğŇæĹŰçżŞář¿çŽĎå■Ůçňęã ĂĆ
lstrip() åŠŇ rstrip() åĹĘåĹńäžŐåůęåŠŇäžŐåŔşæĽ'ğèąŇåĹăéŹď'æŞ■ä¡IJãĂĆ
éżŸèőď'æČĚåĘţäÿŃïijŇè£ŹäžŻæŰźæşŢäijŽåŐżéŹď'çľ'žçŹ¡å■ŮçňęïijŇä¡Ęæ Ÿŕä¡ääź§åŔŕäžěæŇĞåőŽåĚűäžŰ

```
>>> # 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)

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

èőÍèőž

è£ŹäžŻ `strip()` æŰźæşŢåIJÍèŕżåŔŰåŠŇæÿĚçŘĘæŢřæ■őäžěåď ĞåŘŐçż■åď ĎçŘĘçŽĎæŮűåĂŹæŸŕç
æŕŤåęĆïijŇä¡ääŔŕäžěçŤÍåőČäžňæÍěåŐżæŐL'çl'žæăijïijŇåijŢåŔůåŠŇåőŇæĹŘåĚűäżŰäżżåŁąãĂĆ

ä¡ĘæŸŕéIJĂèęĄæşÍæĎŔçŽĎæŸŕåŐżéŹď'æŞ■ä¡IJäÿ■äijŽåŕźå■Ůçňęäÿšç ŽĎäÿ■éŮť'çŽĎæŰĞæIJňäžğçŤ

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

åęĆæđIJä¡ăæČşåď ĎçŘĘäÿ■éŮť'çŽĎçl'žæăijïijŇéĆčäźĹä¡ăéIJĂèęĄæśĆåŁl'åĚűäžŰæŁĂæIJŕãĂĆæŕŤåę
`replace()` æŰźæşŢæĹŰèĂĚæŸŕçŤÍæ■čåĹŹèąÍè¿¿åijŔæŻ£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£ŹæăůçŽĎèŕÍïijŇéĆčäźĹçŤ§æĹŘåŹÍèąÍè¿¿åijŔåřšåŔŕäžěåď ğæŸ¿èžńæL'ŃäžĘãĂĆæŕŤåęĆïijŽ

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

åIJÍè£ŹéĞŇïijŇèąÍè¿¿åijŔ `lines = (line.strip() for line in f)`
æL'ğèąŇæŢřæ■őè¡ňæ■ćæŞ■ä¡IJãĂĆ è£ŹçğæŰźåijŔéÍđåÿÿénŸæŢĹïijŇåŽääÿžåőČäÿ■éIJĂèęĄéćĎåĚĹèŕžåŀ
åőČäžĚäžĚåŔłæŸŕåĹŻåžžäÿĂäÿłçŤ§æĹŘåŹÍïijŇázűäÿŤæŕŔæňąè£ŤåŽđèąŇäźŃåL'■äijŽåĚĹæL'ğèąŇ
`strip` æŞ■ä¡IJãĂĆ

åŕźäžŐæŻť'énŸéŸűçŽĎstripïijŇä¡ääŔŕèČ¡éIJĂèęĄä¡£çŤÍ `translate()`
æŰźæşŢãĂĆèŕůåŔĆéŸĚäÿŃäÿĂèŁĆäžĘèğčæŻť'åď ŽåĚşäžŐå■ŮçňęäÿšæÿĚçŘĘçŽĎåĘĚåőźãĂĆ

4.12 2.12 åőąæ§ěæÿĚçŘĘæŰĞæIJňå■Ůçňęäÿš

éŮőécŸ

äÿĂäžŻæŮăèĄŁçŽĎåźijçÍŽéżŠåőćåIJÍä¡ăçŽĎç¡ŚçńŹéąţéÍćèąĺå■Ţäÿ■è¿ŞåĚěæŰĞæIJňâĂİpÃ¡tÄěÃűÃś

èğčåĘşæŰźæąĹ

æŰĞæIJňæÿĚçŘĘéŮőécŸäijŽæűL'åŔŁåĹřåŇĚæŃňæŰĞæIJňèğčædŘäÿŐæŢřæ■őåď'ĎçŘĘç■L'äÿĂçşżå
åIJÍéÍđåÿÿçőĂå■ŢçŽĎæČĚå¡ćäÿŇïijŇä¡ăåŔŕèČ¡äijŽéĂL'æŃl'ä¡£çŤĺå■Ůçňęäÿšå Ğ¡æŢř(æŕŤåęĆ
`str.upper()` åŠŇ `str.lower()`)åřĘæŰĞæIJňè¡ňäÿžæăĞåĞĘæăijåijŔãĂĆ ä¡£çŤĺ
`str.replace()` æĹŰèĂĚ `re.sub()` çŽĎçőĂå■ŢæŻ£æ■ćæŞ■ä¡IJèČ¡åĹăéŹď'æĹŰèĂĚæŤźåŔŸæŇĞåőŹ
ä¡ăåŘŇæăůè£ŸåŔŕäžěä¡£çŤĺ2.9åřŔèŁĆçŽĎ `unicodedata.normalize()`
åĞ¡æŢřåřĘunicodeæŰĞæIJňæăĞåĞĘåŇŰãĂĆ

çĎűåŘŐïijŇæIJL'æŮűåĂŹä¡ăåŔŕèČ¡è£ŸæČşåIJÍæÿĚçŘĘæŞ■ä¡IJäÿŁæŻt'è£ŻäÿĂæ■ěãĂĆæŕŤåęĆïijŇä¡
äÿžäžĘè£ŹæăůåĄŽïijŇä¡ăåŔŕäžěä¡£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■ŮçňęãĂĆäÿžäžĘè£ŹæăůåĄŽïijŇåĚĹåĹŻåžžäÿĂäÿłåřŘçŽĎè¡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'žæăijãĂĆåŻđè¡ęå■ŮçňęrçŻt'æŐěèćńåĹăéŹď'ãĂĆ

ä¡ăåŔŕäžěäžěè£Źäÿłèąĺæăijäÿžå§žçąÄè£ŻäÿĂæ■ěædĎåżžæŻt'åď'gçŽĎèąĺæăijãĂĆæŕŤåęĆïijŇèől'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)

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

äÿŁéİćä¿Ńå■Řäÿ■ïijŇéĂŽè£Ğä¡£çŤÍ dict.fromkeys() æŰźæşŢædĎéĂääÿĂäÿłå■ŮåĚÿïijŇæŕŔäÿłUnicodeåŠŇé§şçňęä¡IJäÿžéŤőïijŇåŕźåžŤçŽĎåĂijåĚĺéČĺäÿž None ãĂĆ

çĎűåŘŐä¡£çŤÍ unicodedata.normalize() åřĘåŐ§ågŃè¿ŞåĚěæăĞåĞĘåŇŰäÿžåĹĘèğčå¡ćåijŔå■Ů çĎűåŘŐåĘ■èřČçŤÍ translate åĞ¡æŢřåĹăéŹd'æL'ĂæIJL'éĞ■é§şçňęãĂĆ åŘŇæăůçŽĎæŁĂæIJŕäź§åŔŕäżěèćńçŤÍæİèåĹăéŹd'åĚűäžŰçśżåđŃçŽĎå■Ůçňę(æŕŤåęĆæŐğåĹűå■Ůçňęç■L')ã

ä¡IJäÿžåĶęäÿĂäÿłä¿Ńå■ŘïijŇè£ŹéĞŇæđĎéĂääÿĂäÿłåřĘæL'ĂæIJL'UnicodeæŢřå■Ůå■ŮçňęæŸăåřĎåĹ

```
>>> 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'
>>>
```

åŔęäÿĂçğ■æÿĚçŘĘæŰĞæIJňçŽĎæŁĂæIJŕæűL'åŔŁåĹřI/Oèğčçă ĄäÿŐçij Ůçă Ąåğ¡æŢřăĂĆè£ŹéĞŇçŽĎ çĎűåŘŐåĘ■çżŞåŘĹ encode() æĹŰèĂĚ decode() æŞ■ä¡IJæİěæÿĚéŹd'æĹŰä£őæŤźåőČãĂĆæŕŤåęĆïijŽ

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

è£ŹéĞŇçŽĎæăĞåĞĘåŇŰæŞ■ä¡IJåřĘåŐ§æİèçŽĎæŰĞæIJňåĹĘèğčäÿžå■ŢçŃňçŽĎåŠŇé§şçňęãĂĆæŐěă ą¡ŞçĎűïijŇè£Źçğ■æŰźæşŢäżĚäżĚåŔłåIJĺæIJĂåŘŐçŽĎçŽőæăĞåřśæŸŕèŐůåŘŰåĹřæŰĞæIJňåŕźåžŤACSIIèą

èőĺèőž

æŰĞæIJňå■Ůçňęæÿ ĚçŘĘäÿĂäÿłæIJĂäÿżèęĄçŽĎéŮőécŸåžŤèŕěæŸŕè£ŘèąŇçŽĎæĂğèČ¡ãĂĆäÿĂèĹňæ åŕźäžŐçőĂå■ŢçŽĎæŻ£æ■ćæŞ■ä¡IJïijŇ str.replace() æŰźæşŢéĂŽåÿÿæŸŕæIJĂå£ńçŽĎïijŇçŤŽèĞşåIJ æŕŤåęĆïijŇäÿžäžĘæÿĚçŘĘçl'žçŹ¡å■Ůçňęïij Ňä¡ăåŔŕäżěè£ŹæăůåĄŽïijŽ

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

(continues on next page)

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

åęĆædIJä¡ăåŐżæţŃèŕŢçŽĎèŕÍïijŇä¡ăåřśäijŽåŔŞçŐřè£Źçğ■æŰźåijŔäijŽæŕŤä¡£çŤÍ `translate()` æĹŰèĂĚæ■čåĹŹèąÍè¿¿åijŔèęĄå£ńå¿Ĺåď˙ŽãĂĆ

åŘęäýĂæŰźéÍćïijŇåęĆædIJä¡ăéIJĂèęĄæL˙ğèąŇäżżä¡Ţåď˙■æÍĆå■Ůçňęářźå■ŮçňęŽĎéĞ■æŰřæŸăářĎæ `tanslate()` æŰźæşŢäijŽéİđåýÿçŽĎå£ńãĂĆ

äżŐåď˙ğçŽĎæŰźéÍćæİèèőšïijŇářźäžŐä¡ăçŽĎåžŤçŤÍçÍŃåžŔæİèèŕť˙æĂğèČ¡æŸŕä¡ääý■å¿Ůäÿ■åŐżèĞłåů äÿ■åźÿçŽĎæŸŕïijŇæĹŚäżňäÿ■åŘŕèČ¡çżŹä¡ääżżèőőäÿĂäýłçL˙źåőŽçŽĎæŁĂæIJŕïijŇä¡£åőČèČ¡åď˙§éĂĆåžŤa åŽăæ■ď˙åőđéŹĚæČĚåĘţäÿ■éIJĂèęĄä¡ăèĞłåůšåŐżåřİèŕŢäÿ■åŘŇçŽĎæŰźæşŢåźűèŕĎäijřåőĆãĂĆ

ář¡çőąè£ŹäÿĂèŁĆéŻĘäÿ■èőÍèőžçŽĎæŸŕæŰĞæIJňïijŇä¡ĘæŸŕçśżäijijçŽĎæŁĂæIJŕäź§åŘŕäžěéĂĆçŤÍäž

4.13 2.13 å■Ůçňęäÿšářźé¡Ř

éŮőécŸ

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

èğčåĘşæŰźæąĹ

åŕźäžŐå§žæIJńçŽĎå■Ůçňęäÿšářźé¡ŘæŞ■ä¡IJïijŇåŘŕäžěä¡£çŤÍå■ŮçňęäÿšçŽĎ `ljust()` , `rjust()` åŠŇ `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¡æŐěåŘŮäÿĂäýłåŘŕéĂL˙çŽĎåąńåĚĚå■ŮçňęãĆæŕŤåęĆïijŽ

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

åĞ¡æŢř `format()` åŘŇæăůåŘŕäžěçŤÍæÍěå¿Ĺåőźæ˙ŸşçŽĎåŕźé¡Řå■ŮçňęäÿšãĆ ä¡ăèęĄåĄŽçŽĎåřśæ˙Ÿŕä¡£çŤÍ <, > æĹŰèĂĚ ^ å■ŮçňęåŐéÍćçt˙ğèů§äÿĂäýłæŇĞåőŽçŽĎåő¡åžęãĆæŕŤåęĆ

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

åęĆæđIJä¡ăæČşæŇĞåőŽäÿĂäÿłéIđçľžæăijçŽĎåąńåĚĚå■Ůçňęïij Ňår Ęåő ČåĘŹåĹřářźé¡Řå■Ůçňęç ŽĎåL'■

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

å¡Şæăijåij ŔåŇŰåď Žäÿłå Ăijç ŽĎæŮűåĂŹïijŇè£ŹäžŻæăijåijŔäžčçăĄäź§åŔŕäżěèćńçŤĺåIJÍ format() æŰźæşŢäÿ■ãĂĆæŕŤåęĆïijŽ

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

format() åĞ¡æŢřçŽĎäÿĂäÿłåě¡åď ĎæŸŕåőČäÿ■äžĚéĂĆçŤĺäžŐå■Ůçňęäÿšã ĂĆåőČåŔŕäżěçŤĺæİěæăijæŕŤåęĆïijŇä¡ăåŔŕäżěçŤĺåőČæİěæăijåijŔåŇŰæŢřå■ŮïijŽ

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

èőÍèőž

åIJÍèĂĄçŽĎäžčçăĄäÿ■ïijŇä¡ăçżŔåÿÿäijŽçIJŃåĹřèćńçŤĺæİěæăijåijŔåŇŰæŰĞæIJňçŽĎ % æŞ■ä¡IJçňęãĂĆæŕŤåęĆïijŽ

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

ä¡ĘæŸŕïijŇåIJÍæŰřçĽĹæIJňäžčçăĄäÿ■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 åŽääÿžåőČåŔŕäżěçŤĺæİěæăijåijŔåŇŰäżżæĎŔåŕźèśąïijŇèĂŇäÿ■äžĚäžĚæŸŕå■Ůçňęäÿšã ĂĆ

åęĆæđIJæČşèęĄåőŇåĚíäžĘèğč `format()` åĞ¡æŢřçŽĎæIJL'çŤÍçL'źæĂğïijŇèŕůåŔĆèĂČ åIJÍçž£PythonæŰĞæąč

4.14 2.14 åŘĹåźűæŃijæŐěå■Ůçňęäÿš

éŮőécŸ

ä¡ăæČşåřĘåĞăäÿłåřŔçŽĎå■Ůçňęäÿšå ŘĹåźűäÿžäÿĂäÿłåď ğçŽĎå■Ůçňęäÿš

èğčåĘşæŰźæąĹ

åęĆæđIJä¡ăæČşèęĄåŘĹåźűçŽĎå■Ůçňęäÿšæ Ÿŕå IJÍäÿĂäÿłåžŔåĹŮæĹŰèĂĚ `iterable` äÿ■ïijŇéĆčäźĹæIJĂå£ńçŽĎæŰźåijŔåřśæŸŕä¡£çŤÍ `join()` æŰźæşŢãĂĆærŤåęĆïijŽ

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

åĹİçIJŃètůæİěïijŇè£Źçğ■èí■æşŢçIJŃäÿŁåŐżäijŽæŕŤè¿ČæĂłïijŇä¡ĘæŸŕ `join()` èćńæŇĞåőŽäÿžå■Ůçňęäÿšç ŽĎäÿĂäÿłæŰźæşŢãĂĆ è£ŹæăůåĄŽçŽĎéČĺåĹĘåŐ§åŻăæŸŕä¡ăæČşåŐżè£đæŐěçŽĎåŕźèśąåŔŕèČ¡æİěèĞłåŘĎçğ■äÿ■åŘŇçŽĎæŢřæ■ åęĆæđIJåIJÍæL'ĂæIJL'è£ŹäžŽåŕźèśąäÿŁéČ¡åőŽäźL'äÿĂäÿł `join()` æŰźæşŢæŸŐæŸ¿æŸŕåĘŮä¡ŹçŽĎãĂĆ åŻăæ■d'ä¡ăåŘłéIJĂèęĄæŇĞåőŽä¡ăæČşèęĄçŽĎåĹĘåL'šå■Ůçňęäÿšåź `join()` æŰźæşŢåŐżåřĘæŰĞæIJňçL'ĞæőţçżĎå ŘĹètůæİěãĂĆ

åęĆæđIJä¡ăäźĚäźĚåŘłæŸŕåŘĹåźűåřŚæŢřåĞăäÿłå■ŮçňęäÿšïijŇä¡£çŤÍåŁăåŔů(+)éĂŽåÿÿåůšçżŔèűşåď§

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

åŁăåŔů(+)æŞ■ä¡IJçňęåIJÍä¡IJäÿžäÿĂäžŽåď ■æİĆå■ŮçňęäÿšæăijåijŔåŇŰçŽĎæŻ£äžčæŰźæąĹçŽĎæŮűå

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

åęĆæđIJä¡ăæČşåIJÍæžŘçăĄäÿ■årĘäÿd'äÿłå■ŮéİĆå■Ůçňęäÿšå ŘĹåźűètůæİěïijŇä¡ăåŘłéIJĂèęĄçőĂå■ŢçŽ

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

èőÍèőž

å■Ůçňęäÿšå ŘĹåźűåŘŕèČ¡çIJŃäÿŁåŐżåźűäÿ■éIJĂèęĄçŤÍäÿĂæŢť'èŁĆæİěèőÍèőžãĂĆ
ä¡ĘæŸŕäÿ■åžŤèŕěåřŘçIJŃè£ŹäÿłéŮőécŸïijŇçÍŃåžŘåŚŸéĂŽåÿÿåIJÍå■ŮçňęäÿšæäijåijŘåŇŰçŽĎæŮűåĂŹåŻ

æIJĂéĞ■èęĄçŽĎéIJĂèęĄåijŢèţůæşÍæĎŘçŽĎæŸŕïijŇå¡ŞæĹŚäżňä¡£çŤÍåŁååŘů(+)æŞ■ä¡IJçňęåŐżè£đæ
åŻääÿžåŁäåŘůè£đæŐěäijŽåijŢèţůåĘĚå■Ÿåď'■åĹűäžěåŘŁåđČåIJ¿åŻđæŤűæŞ■ä¡IJãĂĆ
çĽźåĹńçŽĎïijŇä¡ăæřÿè£IJéČ¡äÿ■åžŤåČŘäÿŃéÍćè£ŹæäůåĘŹå■Ůçňęäÿšè£đæŐěäžčçăĄïijŽ

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

è£Źçğ■åĘŹæşŢäijŽæŕŤä¡£çŤÍ `join()` æŰźæşŢè£ŘèąŇçŽĎèęĄæĚćäÿĂäžŻïijŇåŻääÿžæŕŘäÿĂæňąæĽ
ä¡ăæIJĂåě¡æŸŕåĚĹæŤűéŻĘæĽ'ĂæIJĽ'çŽĎå■ŮçňęäÿšçĽĞæőţçĎůåŘŐåĘ■åřĘåőČäžňè£đæŐěèţůæÍěãĂĆ

äÿĂäÿłçŻÿåŕźæŕŤè¿ČèĄłæŸŐçŽĎæŁĂåůğæŸŕåĹľçŤÍçŤ§æĹŘåŹÍèąÍè¿¿åijŔ(åŔĆèĂČ1.19åřŔèŁĆ)è¡ňa

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

åŘŇæăůè£Ÿå¿ŮæşÍæĎŔäÿ■å£ĚèęĄçŽĎå■Ůçňęäÿšè£đæŐěæŞ■ä¡IJãĂĆæIJĽ'æŮűåĂŹçÍŃåžŔåŚŸåIJÍæ

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

å¡Şæůůå ŘĹä¡£çŤÍI/OæŞ■ä¡IJåŠŇå■Ůçňęäÿšè£đæŐěæŞ■ä¡IJçŽĎæŮűåĂŹïijŇæIJĽ'æŮűåĂŹéIJĂèęĄäż
ærŤåęĆïijŇèĂČèŹŚäÿŃéÍćçŽĎäÿď'çńŕäžčçăĄçĽ'Ğæőţïij Ž

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

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

åęĆæđIJäÿď'äÿłå■Ůçňęäÿšå¿ĹåřŔïijŇéĆčäźĹçňňäÿĂäÿłçĽ'ĹæIJňæĂğèČ¡äijŽæŻť'åě¡äžŻïijŇåŻääÿžI/Oç
åŔęåď'ŰäÿĂæŰźéÍćïijŇåęĆæđIJäÿď'äÿłå■Ůçňęäÿšå¿Ĺåď'ğïijŇéĆčäźĹçňňäžŇäÿłçĽ'ĹæIJňåŔŕèČ¡äijŽæŻť'åŁ
åŻääÿžåőČéĄ£åĚ■äžĘåĹŻåżžäÿĂäÿłå¿Ĺåď'ğçŽĎäÿť'æŮűçżŞæđIJåźűäÿŤèęĄåď'■åĹűåď'ğéĞŔçŽĎåĘĚå■Ÿå
è£ŸæŸŕéĆčåŔěèŕÍïijŇæIJĽ'æŮűåĂŹæŸŕéIJĂèęĄæäźæ■őä¡ăçŽĎåžŤçŤÍçÍŃåžŔçĽ'źçĆźæÍěåĘşåőŽåžŤèŕěä¡£

æIJĂåŘŐèřĹäÿĂäÿŃïijŇåęĆæđIJä¡ăåĞĘåď'ĞçijŰåĘŹæđĎåžžåď'ğéĞŔåřŔå■ŮçňęäÿšçŽĎè¿ŞåĞžäżčçăA
ä¡ăæIJĂåě¡èĂČèŹŚäÿŃä¡£çŤĺçŤ§æĹŘåŹĺåĞ¡æŢřïijŇåĹl'çŤĺyieldèŕ■åŘëäžğçŤ§è¿ŞåĞžçL'Ğæőţã ĂĆæŕŤåęĆ

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

è£Źçğ■æŰźæşŢäÿĂäÿłæIJL'èűčçŽĎæŰźéĺćæŸŕåőČåźűæšąæIJL'åŕźè¿ŞåĞžçL'Ğæőţå Ĺřåž Ţèę ĄæĂŐæăů
ä¿ŃåęĆïijŇä¡ăåŔŕäżěçőĂå■ŢçŽĎä¡£çŤĺ `join()` æŰźæşŢårĘè£ŹäžŽçL'Ğæőţå ŘĹåźűèţůæĺëïijŽ

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

æĹŰèĂĚä¡ăäź§åŔŕäżěårĘå■Ůçňęäÿšç L'Ğæőţé Ğ■åőŽåŘŠåĹřI/Oïij Ž

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

åĘ■æĹŰèĂĚä¡ăè£ŸåŔŕäżěåĘŹåĞžäÿĂäžŽçżŞåŘĹI/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£ŹéĞŇçŽĎåĚşéŤőçĆźåIJĺäžŐåŐ§ågŃçŽĎçŤ§æĹŘåŹĺåĞ¡æŢřåźűäÿ■éIJĂèęAç§ěéĄŞä¡£çŤĺçżĘèŁĆïij

4.15 2.15 å■Ůçňęäÿšäÿ■æŔŠåĚěåŔŸéĞŔ

éŮőécŸ

ä¡ăæČşåĹŻåžžäÿĂäÿłåĘĚåţŇåŔŸéĞŔçŽĎå■Ůçňęäÿšïij ŇåŔŸéĞŔèćńåőČçŽĎåĂijæL'Ăèą Íçd'žçŽĎå■Ů

èğčåĘşæŰźæąĹ

PythonåźűæšąæIJL'åŕźåIJĺå■Ůçňęäÿšäÿ■çőĂå■ŢæŻ£æ■ćåŔŸéĞŔåĂijæŔŘä¿ŻçŻt'æŐěçŽĎæŤŕæŇĄãĂä¡ĘæŸŕéĂŽè£Ğä¡£çŤĺå■Ůçňęäÿšç ŽĎ `format()` æŰźæşŢæİěèğčåĘşè£ŹäÿłéŮőécŸãĂĆæŕŤåęĆïijŽ

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

æĹŰèĂĚïijŇåęĆæđIJèęĄèćńæŻ£æ■ćçŽĎåŔŸéĞŔèČ¡åIJĺåŔŸéĞŔå§§äÿ■æL'¿åĹřïijŇéĆčäźĹä¡ääŔŕäżěçżŞåŘĹä¡£çŤĺ `format_map()` åŠŇ `vars()` ãĂĆåřśåČŔäÿŇéİćè£ŹæăůïijŽ

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

`vars()` è£ŸæIJL'äÿĂäÿłæIJL'æĎŔæĂİçŽĎçL'źæĂğåřśæŸŕåőČäź§éĂĆçŤĺäžŐåŕźèśąåőđä¿ŇãĂĆæŕŤå

```
>>> 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'ĎçŘĘåŔŸéĞ

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

äÿĂçğ■éĄ£åĚ■è£Źçğ■éŤŹèŕŕçŽĎæŰźæşŢæŸŕåŔęåď'ŰåőŽäźL'äÿĂäÿłåŘńæIJL' `__missing__()` æŰźæşŢçŽĎå■ŮåĚÿåŕźèśąïijŇåřśåČŔäÿŇéİćè£ŹæăůïijŽ

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

çŐřåIJĺä¡ääŔŕäżěåĹl'çŤĺè£Źäÿłçśżå ŇĚèčĚè¿ŞåĚěåŘŐäijäéĂŠçżŹ `format_map()` ïijŽ

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

åęĆæđIJä¡ăåŔŠçŐřèĞłåůśåIJÍäžčçăĄäÿ■éćŚçźĄçŽĎæĽğèąŇè£ŹäžŻæ■ěéłď’ïijŇä¡ăåŔŕäžěåřĘåŔŸéĞŔa

```
import sys

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

çŐřåIJÍä¡ăåŔŕäžěåČŔäÿŃéİćè£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’äžěæİěçŤśäžŐPythonçijžäźŔåŕźåŔŸéĞŔæŻ£æ■ćçŽĎåĘĚç¡őæŤŕæŇĄèĂŇåŕijèĞť’äžĘåŘĎçğ■äÿ
ä¡IJäÿžæIJňèŁĆäÿ■åśŢçď’žçŽĎäÿĂäÿłåŔŕèČ¡çŽĎèğčåĘşæŰźæąĹïijŇä¡ăåŔŕäžěæIJL’æŮűåĂŹäijŽçIJŃåĹřåČ

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

ä¡ăåŔŕèČ¡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’åŁăăĚĹ
ä¡£çŤĺ `format()` æŰźæşŢè£ŸæIJL’äÿĂäÿłåě¡åď’ĎåřśæŸŕä¡ăåŔŕäžěèŐůå¿Ůåŕźå■ŮçňęäÿšæăijåijŔåŇŰçŽĎ
èĂŇè£ŹäžŻçL’źæĂgæŸŕä¡£çŤĺåČŔæĺąæİ£å■ŮçňęäÿšäźŇçśżçŽĎæŰźæąĹäÿ■åŔŕèČ¡èŐůå¿ŮçŽĎãĂĆ

æIJňæIJžè£ŸéČĺåĹĘäžŃçż■äžĘäÿĂäžŻénŸçžğçL’źæĂgãĂĆæŸăåřĎæĹŰèĂĚå■ŮåĚÿçśżäÿ■éšIJäÿžäžž
`__missing__()` æŰźæşŢåŔŕäžěèőľä¡ăåőŽäźL’ąęĆä¡Ţåď’ĎçŘĘçijžåď’śçŽĎåĂijãĂĆ åIJÍ

SafeSub çśżäÿ■ïijŇè£Źäÿ łæŰźæşŢèćńåőŽäźL'äÿžåŕźçijžåď'śçŽĎåĂijè£ŤåŻđäÿĂäÿłå■ăä¡■çňęãĂĆ
ä¡ăåŔŕäžěåŔŚçŐřçijžåď'śçŽĎåĂijäijŽåĞžçŐřåIJĺçżŞæđIJå■Ůçňęäÿšäÿ■(åIJĺèřČèŕŢçŽĎæŮűåĂŹåŔŕèČ¡å¿Ła
KeyError åijĆåÿÿãĂĆ

sub() åĞ¡æŢřä¡£çŤĺ sys._getframe(1) è£ŤåŻđèřČçŤĺèĂĚçŽĎæăĹåÿğãĂĆåŔŕäžěäžŐäÿ■èő£éŮ
f_locals æĺěèŐůå¿ŮåśĂéČĺåŔŸéĞŔãĂĆ æŕńæŮăçŰŚéŮőçżĺåď'ğéČĺåĹĘæČĚåĘţäÿŇåIJĺäžčçăĄäÿ■åŐżç
ä¡ĘæŸŕïijŇåŕźäžŐåČŔå■ŮçňęäÿšæŻ£æ■ćåůěåĚůåĞ¡æŢřèĂŇèĺĂåőČæŸŕéİđåÿÿæIJL'çŤĺçŽĎãĂĆ
åŔęåď'ŰïijŇåĂijå¿ŮæşĺæĎŔçŽĎæŸŕ f_locals æŸŕäÿĂäÿłåď'■åĹűěřČçŤĺåĞ¡æŢřçŽĎæIJňåIJřåŔŸéĞŔçŽ
åř¡çőąä¡ăåŔŕäžěæŤźåŔŸ f_locals çŽĎåĘĚåőźïijŇä¡ĘæŸŕè£ŹäÿłäŁőæŤźåŕźäžŐåŘŐéİćçŽĎåŔŸéĞŔèő£é
æL'Ăäžěïij ŇèŹ¡èŕt'èő£éŮőäÿĂäÿłæăĹåÿğçIJŃäÿŁåŐżå¿ĹéĆłæĄűïijŇä¡ĘæŸŕåŕźåőČçŽĎäżżä¡ŢæŞ■ä¡IJäÿ■ä

4.16 2.16 äžěæŇĞåőŽåĹŮåő¡æăijåijŔåŇŰå■Ůçňęäÿš

éŮőécŸ

ä¡ăæIJL'äÿĂäžŻéŢ£å■Ůçňęäÿšïij ŇæČşäžěæŇĞåőŽçŽĎåĹŮåő¡ářĘåőČäżňéĞ■æŰřæăijåijŔåŇŰãĂĆ

èğčåĘşæŰźæąĹ

ä¡£çŤĺ textwrap æĺąåİŮæİěæăijåijŔåŇŰå■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."
```

äÿŃéİćæijŤçď'žä¡£çŤĺ textwrap æăijåijŔåŇŰå■ŮçňęäÿšçŽĎåď'Žçğ■æŰźåijŔïijŽ

```
>>> 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)

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

èőÍèőž

textwrap æÍąåİŮåŕźäžŐå■ŮçňęäÿšæL'Şå■řæŸŕéİđåÿÿæIJL'çŤÍçŽĎïijŇçL'źåĹńæŸŕå¡Şä¡ăåÿŇæIJŻè¿Ş
ä¡ăåŔŕäžěä¡£çŤÍ os.get_terminal_size() æŰźæşŢæİèèŐůåŔŰçżĹçńŕçŽĎåď ğářŔářžářÿãĂĆæŕŤåęĆ

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

fill() æŰźæşŢæŐěåŔŮäÿĂäžŻåĚűäżŰåŔŕéĂL'åŔĆæŢřæİěæŐğåĹűtabïijŇèŕ■åŔěçżŞář¿ç■L'ãĂĆ
åŔĆéŸĚ textwrap.TextWrapperæŰĞæąč èŐůåŔŰæŻť'åď'ŽåĘĚåőźãĂĆ

4.17 2.17 åIJĺå■Ůçňęäÿšäÿ■åď'ĎçŘĘhtmlåŠŇxml

éŮőécŸ

ä¡ăæČşåřĘHTMLæĹŰèĂĚXMLåőđä¡ŞåęĆ &entity; æĹŰ &#code;
æŻ£æ■ćäÿžåŕźåžŢçŽĎæŰĞæIJñãĂĆ åĘ■èĂĚïijŇä¡ăéIJĂèęĄè¡ňæ■ćæŰĞæIJñäÿ■çL'źåőŽçŽĎå■Ůçňę(æŕŤåę
>, æĹŰ &)ãĂĆ

èğčåĘşæŰźæąĹ

åęĆæđIJä¡ăæČşæŻ£æ■ćæŰĞæIJňå■Ůçňęäÿšäÿ■çŽĎ âĂŸ<âĂŹ æĹŰèĂĚ âĂŸ>âĂŹ
ïijŇä¡£çŤÍ html.escape() åĞ¡æŢřåŔŕäžěå¿Ĺåőźæ ŸŞçŽĎåőŇæĹŘãĂĆæŕŤåęĆïijŽ

```
>>> 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'ĎçŘĘçŽĎæŸŕASCIIæŰĞæIJñïijŇåźűäÿŤæČşåřĘéİđASCIIæŰĞæIJňåŕźåžŤçŽĎç
åŔŕäžěçżŹæ§ŘäžŻI/OåĞ¡æŢřäijăéĂŠåŔĆæŢř errors='xmlcharrefreplace'
æİèè¿¿åĹřè£ŹäÿłçŻőãĂĆæŕŤåęĆïijŽ

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

äÿžäžĘæŻ£æ■ćæŰĞæIJñäÿ■çŽĎçijŰçăĄåőđä¡ŞïijŇä¡ăéIJĂèęĄä¡£çŤĺåŔęåđ'ŰäÿĂçğ■æŰźæşŢãĂĆ
åęĆæđIJä¡ăæ■čåIJĺåđ'ĎçŘĘHTMLæĹŰèĂĚXMLæŰĞæIJñïijŇèŕŢçİĂåĚĹä¡£çŤĺäÿĂäÿłåŘĹéĂĆçŽĎHTML
éĂŽåÿÿæČĚåĘţäÿŃïijŇè£ŹäžŻåůěåĚůäijŽèĞłåŁĺæŻ£æ■ćè£ŹäžŻçijŰçăĄåĂijïijŇä¡ăæŮăéIJĂæŃĚå£ČãĂĆ

æIJL'æŮűåĂŹïijŇåęĆæđIJä¡ăæŐěæŤűåĹřäžĘäÿĂäžŻåŘńæIJL'çijŰçăĄåĂijçŽĎåŐ§ågŃæŰĞæIJñïijŇél
éĂŽåÿÿä¡ăåŘłéIJĂèęĄä¡£çŤĺHTMLæĹŰèĂĚXMLèğčæđŘåŹĺçŽĎäÿĂäžŽçŻÿåĚşåůěåĚůåĞ¡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 >>>'
>>>
```

èőÍèőž

åIJÍçŤ§æĹŘHTMLæĹŰèĂĚXMLæŰĞæIJñçŽĎæŮűåĂŹïijŇåęĆæđIJæ■čçąőçŽĎè¡ñæ■ćçL'źæőŁæăĞèć
çL'źåĹńæŸŕå¡Şä¡ăä¡£çŤĺ `print()` åĞ¡æŢřæĹŰèĂĚåĚűäžŰå■ŮçňęäÿšæăijåijŔåŇŰæİëäžğçŤ§è¿ŞåĞžçŽĎa
ä¡£çŤĺåČŔ `html.escape()` çŽĎåůěåĚůåĞ¡æŢřåŕäžěå¿Ĺåőźæ ŸŞçŽĎèğčåĘşè£ŹçšźéŰőécŸãĂĆ

åęĆæđIJä¡ăæČşäžěåĚűäžŰæŰźåijŔåď'ĎçŘĘæŰĞæIJñïijŇè£ŸæIJL'äÿĂäžŻåĚűäžŰçŽĎåůěåĚůåĞ¡æŢřa
`xml.sax.saxutils.unescape()` åŔŕäžěåÿőåŁľä¡ăãĂĆ
çĎűèĂŇïijŇä¡ăåžŤèŕěåĚĹèŕČçăŤæÿĚæěŽæĂŐæăůä¡£çŤĺäÿĂäÿłåŘĹéĂĆçŽĎèğčæđŘåŹĺãĂĆ
æŕŤåęĆïijŇåęĆæđIJä¡ăåIJĺåď'ĎçŘĘHTMLæĹŰXMLæŰĞæIJñïijŇ
ä¡£çŤĺæ§ŘäÿłèğčæđŘæĺąåİŮæŕŤåęĆ `html.parse` æĹŰ `xml.etree.ElementTree`
åůšçżŔåÿőä¡ăèĞłåŁĺåď'ĎçŘĘäžĘçŻÿåĚşçŽĎæŻ£æ■ćçżĘèŁĆãĂĆ

4.18 2.18 å■Ůçňęäÿšäžď'çL'ŇèğčæđŘ

éŮőécŸ

ä¡ăæIJL'äÿĂäÿłå■ŮçňęäÿšïijŇæČşäžŐåůęèĞşåŘşářĘåĚűèğčæđŘäÿžäÿĂäÿłäžď'çL'ŇætĄãĂĆ

èğčåĘşæŰźæąĹ

åĄĞåęĆä¡ăæIJL'äÿŃéİćè£Źæăůäÿ ĂäÿłæŰĞæIJñå■Ůçňęäÿšïijž

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

äÿžäžĘäżď'çL'ŇåŇŰå■ŮçňęäÿšïijŇä¡ääÿ■äżĚéIJĂèęĄåŇźéĚ■æĺąåijŕïijŇè£Ÿå¿ŮæŇĞåőŽæĺąåijŔçŽĎç
æŕŤåęĆïijŇä¡ääŔŕèČ¡æČşářĘå■Ůçňęäÿšå ČŔäÿŃéİćè£Źæăůè¡ňæ■ćäÿžåžŔåĹŮåŕźïijŽ

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

äÿžäžĘæL'ğèąŇè£ŹæăůçŽĎåĹĞåĹĘïijŇçňňäÿĂæ■ěåřśæŸŕåČŔäÿŃéİćè£ŹæăůåĹl'çŤĺåŠ¡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]))
```

åIJĺäÿŁéİćçŽĎæĺąåijŔäÿ■ïijŇ `?P<TOKENNAME>` çŤĺäžŐçżŹäÿĂäÿłæĺąåijŔåŠ¡åŘ■ïijŇä¿ŻåŘŐéİćä¡£çŤ

äÿŃäÿĂæ■ěïijŇäÿžäžĘäżď'çL'ŇåŇŰïijŇä¡£çŤĺæĺąåijŔåŕźèśąå¿ĹåřŠèćńäžžç§ěéĄŞçŽĎ
`scanner()` æŰźæşŢãĂĆ è£ŹäÿłæŰźæşŢäijŽåĹŻåžžäÿĂäÿł
`scanner` åŕźèśąïijŇ åIJĺè£ŹäÿłåŕźèśąäÿŁäÿ■æŰ■çŽĎèřČçŤĺ `match()`
æŰźæşŢäijŽäÿĂæ■ěæ■ěçŽĎæL'ńæŔŔçŻőæăĞæŰĞæIJñïijŇæŕŔæ■ěäÿĂäÿłåŇźéĚ■ãĂĆ
äÿŃéİćæŸŕæijŤçď'žäÿĂäÿł `scanner` åŕźèśąåęĆä¡Ţåůëä¡IJçŽĎäžď'äžŠåijŔä¿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)

(çż■äÿŁéąţ)

```
>>>
```

åőđéŹĚä¡£çŤĺè£Źçğ■æŁĂæIJŕçŽĎæŮűůåĂŹïijŇåŔŕäžěå¿Ĺåőźæ ŸŞçŽĎåČŔäÿŃéİćè£Źæăůå řĘäÿŁè£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¡ăåŔŕäžěåőŽäźL'æŻť'åď'ŽçŽĎçŤ§æĹŘåŹĺåĞ¡æŢřæĹŰèĂĚä¡ æŕŤåęĆïijŇäÿŃéİćeijŤçď'žæĂŐæăůè£ĞæżďæL'ĂæIJL'çŽĎçľžçŹ¡äżďçL'Ňïijž

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

èőÍèőž

éĂŽåÿÿæİěèőšäżďçL'ŇåŇŰæŸŕå¿Ĺåď'ŽéńŸçžğæŰĞæIJňèğčæđŘäÿŐåď'ĎçŘEçŽĎçňňäÿĂæ■ěãĂĆ äÿžäžĘä¡£çŤĺäÿŁéİćçŽĎæL'ńæŔŘæŰźæşŢïijŇä¡ăéIJĂèęĄèőŕä¡Ŕè£ŹéĞŇäÿĂäžŻéĞ■èęĄçŽĎåĞăçĆźãĂĆ çňňäÿĂçĆźåřśæŸŕä¡ăå£Ěéążçąőèőď'ä¡ăä¡£çŤĺæ■čåĹŹèąĺè¿¿åijŔæŇĞåőŽäžĘąæL'ĂæIJL'è¿ŞåĚěäÿ■åŔŕèČ¡åĞ åęĆæđIJæIJL'äżżä¡Ţäÿ■åŔŕåŇźéĚ■çŽĎæŰĞæIJňåĞžçŐřäžĘïijŇæL'ńæŔŘåřśäijŽçŻť'æŐěåĄIJæ■ćãĂĆè£Źä

äżď'çL'ŇçŽĎéąžåžŔäź§æŸŕæIJL'å¡śåŞ■çŽĎãĂĆ `re` æĺąåİŮäijŽæŇL'çĚğæŇĞåőŽåě¡çŽĎéąžåžŔåŐżåĄ åŻäæ■ď'ïijŇåęĆæđIJäÿĂäÿłæĺąåijŔæĄřåě¡æŸŕåŔęäÿĂäÿłæŻť'éŢ£æĺąåijŔçŽĎå■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
```

çňňäžŇäÿłæĺąåijŔæŸŕéŤŹçŽĎïijŇåŽääÿžåőČäijŽářĘæŰĞæIJň<=åŇźéĚ■äÿžäżď'çL'ŇLTçť'ğèů§çİĂEQï

æIJĂåŘŐïijŇä¡ăéIJĂèęĄçŢŹæĎŔäÿŇå■Řå■Ůçňęäÿšå¡ćåijŔçŽĎæĺąåijŔãĆĆæŕŤåęĆïijŇåĄĞèő¿ä¡ăæIJL

```
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')
```

åĚşäžŐæŻťéńŸéŸűçŽĎäžďçĽŇåŇŰæŁĂæIJŕïijŇä¡ăåŘŕèČ¡éIJĂèęĄæ§ěçIJŃ PyParsing æĹŰèĂĚ PLY åŇĚãĂĆ äÿĂäÿłèřČçŤÍPLYçŽĎä¿Ńå■ŘåIJĺäÿŃäÿĂèŁĆäijŽæIJL'æijŤçd'žãĂĆ

4.19 2.19 åőđçŐřäÿĂäÿłçőĂå■ŢçŽĎéĂŠå¡ŠäÿŃéŹ■åĹĘæđŘåŹÍ

éŮőécŸ

ä¡ăæČşæăźæ■őäÿĂçżĎèŕ■æşŢèğĎåĹŹèğčæđŘæŰĞæIJňåźűæL'ğèąŇåŚ¡äžd'ïijŇæĹŰèĂĚæđĎéĂăäÿĂäåęĆæđIJèŕ■æşŢéİđåÿÿçőĂå■ŢïijŇä¡ăåŘŕäžěäÿ■åŐżä¡£çŤĺäÿĂäžŻæąĘæđűïijŇèĂŇæŸŕèĞłåůśåĘŹè£Źäÿłèğč

èğčåĘşæŰźæąĹ

åIJĺè£ŹäÿłéŮőécŸäÿ■ïijŇæĹŚäžňéŻĘäÿ■èőĺèőžæăźæ■őçL'źæőŁèŕ■æşŢåŐżèğčæđŘæŰĞæIJňçŽĎéŮőéäÿžäžĘè£ŹæăůåĄŽïijŇä¡ăéęŰåĚĹèęĄäžěBNFæĹŰèĂĚEBNFå¡ćåijŔæŇĞåőŽäÿĂäÿłæăĞåĞĘèŕ■æşŢãĆ æŕŤåęĆïijŇäÿĂäÿłçőĂå■ŢæŢřå■ęèąĺè¿¿åijŔèŕ■æşŢåŘŕèČ¡åČŔäÿŃéİćè£Źæăůïijž

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

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

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

æĹŰèĂĚïijŇäžěEBNFå¡ćåijŔïijž

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

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

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

åIJÍEBNFäÿ■ïijÑèćńåŇĚåŘńåIJÍ { . . . }* äÿ■çŽĎèğĎåĹŹæŸŕåŘŕéĂĽ'çŽĎãĂĆ*äžčèąĺ0æňąæĹŮåď'Žæ

çŐřåIJÍïijÑåęĆæđIJä¡ăåŕźBNFçŽĎåůěä¡IJæIJžåĹűè£Ÿäÿ■æŸŕå¿ĹæŸŐçŹ¡çŽĎèŕÍïijÑåřśæŁŁåőČå¡ŞåĄ
äÿĂèĹňæÍěèőšïijÑèğčæđŘçŽĎåŐ§çŘĘåřśæŸŕä¡ăåĹl'çŤÍBNFåőŇæĹŘåď'ŽäÿłæŻ£æ■ćåŠŇæL'l'åśŢäžěåŇźéĚ
äÿžäžĘæijŤçď'žïijÑåĄĞèő¿ä¡ăæ■čåIJÍèğčæđŘå¡ćåęĆ 3 + 4 * 5 çŽĎèąĺè¿¿åijŔãĂĆ
è£Źäÿłèąĺè¿¿åijŔåĚĹèęĄéĂŽè£Ğä¡£çŤÍ2.18èŁĆäÿ■äżÑçż■çŽĎæŁĂæIJŕåĹĘèğčäÿžäÿĂçżĎäžď'çL'ŇæţĄãĂ
çżŞæđIJåŘŕèČ¡æŸŕåČŔäÿŇåĹŮè£ŹæăůçŽĎäžď'çL'ŇåžŔåĹŮïijŽ

```
NUM + NUM * NUM
```

åIJÍæ■d'å§žçąĂäÿŁïijÑ èğčæđŘåŁĺä¡IJäijŽèŕŢçİĂåŐžéĂŽè£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
```

äÿŃéİćæL'ĂæIJL'çŽĎèğčæđŘæ■ěéłd'åŔŕèČ¡éIJĂèęĄèŁśçĆźæŮűéŮt'åijĎæŸŐçŹ¡ïijÑä¡ĘæŸŕåőČäžňåĆ
çňňäÿĂäÿłè¿ŞåĚěäžď'çL'ŇæŸŕfNUMïijÑåŽăæ■d'æŻ£æ■ćéęŰůåĚĹäijŽåŇźéĚ■éČčäÿłéČĺåĹĘãĂĆ
äÿĂæŮęåŇźéĚ■æĹŘåŁ§ïijÑåřśäijŽè£ŻåĚěäÿŃäÿĂäÿłäžď'çL'Ň+ïijÑäžěæ■d'çśźæŐĺãĂĆ
å¡ŞåůšçżŕçąőåőŽäÿ■èČ¡åŇźéĚ■äÿŃäÿĂäÿłäžď'çL'ŇçŽĎæŮűůåĂŹïijÑåŕşè¿źçŽĎéČĺåĹĘ(æŕŤåęĆ
{ (*/) factor }*)åřśäijŽèćńæÿĚçŘĘæŐL'ãĂĆ åIJĺäÿĂäÿłæĹŘåŁ§çŽĎèğčæđŘäÿ■ïijÑæŢt'äÿłåŕşè¿źé

æIJL'äžĘåL'■éİćçŽĎç§ěèŕĘèČŇæŹŕïijÑäÿŃéİćæĹŚäžňäÿ¿äÿĂäÿłçőĂå■Ţçď'žä¿ŃæÍěåśŢçď'žåęĆä¡ŢæđI

```
#!/usr/bin/env python
# -*- encoding: utf-8 -*-
"""
Topic: äÿŃéŹ■èğčæđŘåŹÍ
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)

```
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)

```
        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)

```
    #     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 ğçŽĎäÿžéćŸïijŇ äÿĂèĹňäijŽå■ăçŤĺå■ęçŤ§å■ęäźăçijŰèŕŚèŕ¿çÍŃæŮ åęĆæđIJä¡ååIJÍæĽ¿åŕżåĚşäžŐèŕ■æşŢïijŇèğčæđŘçőŮæşŢç■Ľ'çŻÿåĚşçŽĎèČŇæŹŕç§ěèŕĘçŽĎèŕÎïijŇä¡ååžŤèŕ å¿ĹæŸ¿çĎűïijŇåĚşäžŐè£ŹæŰźéÍćçŽĎåĘĚåőźåď łåď ŽïijŇäÿ■åŔŕèČ¡åIJÍè£ŹéĞŇåĚÍéČĺåśŢåijĂãĂĆ

åř¡çőąåęĆæ■ď'ïijŇçijŰåĘŹäÿĂäÿłéĂŠå¡ŠäÿŃéŹ■èğčæđŘåŹÍçŽĎæŢť'ä¡ŞæĂİèůŕæŸŕæŕŤè¿ČçőĂå■ŢçŽ åijĂåğŃçŽĎæŮűåĂŹïijŇä¡ååĚĹèŐůå¿ŮæĽĂæIJL'çŽĎèŕ■æşŢèğĎåĹŹïijŇçĎűåŘŐářĘåĚűè¡ňæ■ćäÿžäÿĂäÿ åŻăæ■ď'åęĆæđIJä¡ăçŽĎèŕ■æşŢçśżäijijè£Źæăůïijž

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

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

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

ä¡ååžŤèŕěéęŰåĚĹářĘåőČäżňè¡ňæ■ćæĹŘäÿĂçżĎåČŔäÿŃéİćè£ŹæăůçŽĎæŰźæşŢïijž

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

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

äžŐæ§ŘçğæĎŘäźĽäÿŁèőšïijŇæŰźæşŢçŽĎçŻőçŽĎåřśæŸŕèęĄäźĹåď˙ĎçŘĘåőŇèí■æşŢèğĎåĹŹïijŇèęĄäźĹ
äÿžäžĘè£ŹæăůåĄŽïijŇéIJĂéĞĞçŤĺäÿŇéÍćçŽĎè£ŹäžŽåőđçŐřæŰźæşŢïijŽ

- åęĆæđIJèğĎåĹŹäÿ■çŽĎäÿŇäÿłçňęåŘůæŸŕåŘęåď˙ŰäÿĂäÿłèí■æşŢèğĎåĹŹçŽĎåŘ■å■Ů(æŕŤåęĆtermæ
  è£ŹåřśæŸŕèŕěçőŮæşŢäÿ■âĂİäÿŇéŹ■âĂİçŽĎçŤśæİě -
  æŐğåĹűäÿŇéŹ■åĹřåŘęäÿĂäÿłèí■æşŢèğĎåĹŹäÿ■åŐżãĂĆ
  æIJĽæŮűåĂŹèğĎåĹŹäijŽèřČçŤĺåůšçżŘæĽğèąŇçŽĎæŰźæşŢ(æŕŤåęĆïijŇåIJĺ
  `factor ::= '('expr ')'` äÿ■åŕźexprçŽĎèřČçŤĺ)ãĂĆ
  è£ŹåřśæŸŕçőŮæşŢäÿ■âĂİéĂŠå¡ŠâĂİçŽĎçŤśæİěãĂĆ
- åęĆæđIJèğĎåĹŹäÿ■äÿŇäÿĂäÿłçňęåŘůæŸŕäÿłçĽźæőŁçňęåŘů(æŕŤåęĆ()ïijŇä¡åå¿Ůæ§ěæĽ¿äÿŇäÿĂäÿ
  åęĆæđIJäÿ■åŇźéĚ■ïijŇåřśäžğçŤ§äÿĂäÿłèí■æşŢéŤŹèŕŕãĂĆè£ŹäÿĂèŁĆäÿ■çŽĎ
  `_expect()` æŰźæşŢåřśæŸŕçŤĺæİěåĄŽè£ŹäÿĂæ■ěçŽĎãĂĆ
- åęĆæđIJèğĎåĹŹäÿ■äÿŇäÿĂäÿłçňęåŘůäÿžäÿĂäžŽåŘŕèČ¡çŽĎéĂĽ'æŃľéąź(æŕŤåęĆ +
  æĹŰ -)ïijŇ ä¡åå£ĚéążåŕźæŕŔäÿĂçğ■åŘŕèČ¡æČĚåĘţæčĂæ§ěäÿŇäÿĂäÿłäžd˙çĽ'ŇïijŇåŘłæIJĽå¡ŞåőČåŇ
  è£Źäź§æŸŕæIJňèŁĆçd˙žä¿Ňäÿ■ `_accept()` æŰźæşŢçŽĎçŽőçŽĎãĂĆ
  åőČçŻÿå¡ŞäžŐ_expect()æŰźæşŢçŽĎåijśåŇŰçĽ'ĹæIJňïijŇåŽăäÿžåęĆæđIJäÿĂäÿłåŇźéĚ■æĽ'¿åĹřäžĘåč
  ä¡ĘæŸŕåęĆæđIJæšąæĽ'¿åĹřïijŇåőČäÿ■äijŽäžğçŤ§éŤŹèŕŕèĂŇæŸŕåŽđæżŽ(åĚĄèőÿåŘŐçż■çŽĎæčĂæ§
- åŕźäžŐæIJĽéĞ■åď˙■éČĺåĽĘçŽĎèğĎåĹŹ(æŕŤåęĆåIJĺèğĎåĹŹèąĺè¿¿åijŔ `::= term {
  ('+'|'-') term }*` äÿ■)ïijŇ éĞ■åď˙■åŁĺä¡IJéĂŽè£ĞäÿĂäÿłwhileå¿łçŐŕæİěåődçŐřãĂĆ
  å¿łçŐŕäÿżä¡ŞäijŽæŤűéŽĘæĹŰåď˙ĎçŘĘæĽ'ĂæIJĽçŽĎéĞ■åď˙■åĚČçt'ăçŻt'åĹřęšąæIJĽåĚűäžŰåĚČçt'ă
- äÿĂæŮęæŢt'äÿłèí■æşŢèğĎåĹŹåď˙ĎçŘĘåőŇæĹŘïijŇærŔäÿłæŰźæşŢäijŽè£ŤåŻđæ§Řçğ■çżŞæđIJçżŹè
  è£ŹåřśæŸŕåIJĺèğčæđŘè£ĞçÍŇäÿ■åĂijæŸŕæĂŐæăůçt'ŕåŁăçŽĎåŐ§çŘĘãĂĆ
  æŕŤåęĆïijŇåIJĺèąĺè¿¿åijŔæśĆåĂijçÍŇåžŔäÿ■ïijŇè£ŤåŻđåĂijäžčèąĺèąĺè¿¿åijŔèğčæđŘåŘŐçŽĎéČĺåĹĘç
  æIJĂåŘŐæĽ'ĂæIJĽåĂijäijŽåIJĺæIJĂéąűåśĆçŽĎèí■æşŢèğĎåĹŹæŰźæşŢäÿ■åŘĹåźűèţůæİěãĂĆ

åř¡çőąåŘŚä¡ăæijŢçd˙žçŽĎæŸŕäÿĂäÿłçőĂå■ŢçŽĎä¿Ńå■ŘïijŇéĂŠå¡ŠäÿŇéŹ■èğčæđŘåŹĺåŔŕäžěçŤĺæİěå
æŕŤåęĆïijŇPythonèŕ■èĺĂæIJňèžńåřśæŸŕéĂŽè£ĞäÿĂäÿłéĂŠå¡ŠäÿŇéŹ■èğčæđŘåŹĺåŐžèğčéĞŁçŽĎãĂĆ
åęĆæđIJä¡ååŕźæ■d˙æĎ§åĚt'èűčïijŇä¡ăåŔŕäžěéĂŽè£Ğæ§ěçIJŃPythonæžŘçăĄæŰĞäžűGrammar/Grammaræĺ
çIJŃåőŇä¡ăäijŽåŘŚçŐřïijŇéĂŽè£ĞæĽŃåŁĺæŰźåijŔåŐžåőđçŐřäÿĂäÿłèğčæđŘåŹĺåĚűåőđäijŽæIJĽå¿Ĺåd˙Žç

åĚűäÿ■äÿĂäÿłåśĂéŹŘåřśæŸŕåőČäžňäÿ■èČ¡èćńçŤĺäžŐåŇĚåŘńäžżä¡ŢåůęéĂŠå¡ŠçŽĎèí■æşŢèğĎåĹŹäÿ

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

äÿžäžĘè£ŹæăůåĄŽïijŇä¡ăåŔŕèČ¡äijŽåČŔäÿŇéÍćè£ŹæăůäŁçŤĺ `items()` æŰźæşŢïijŽ

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

åŤŕäÿĂçŽĎéŮőécŸæŸŕè£ŹäÿłæŰźæşŢæăźæIJňäÿ■èČ¡åůěä¡IJïijŇäžŃåőđäÿŁïijŇåőČäijŽäžğçŤ§äÿĂäÿ

åĚşäžŐèí■æşŢèğĎåĹŹæIJňèžńä¡ăåŔŕèČ¡äź§äijŽçćřåĹřäÿĂäžŽæčŸæĽ'ŃçŽĎéŮőécŸãĂĆ
æŕŤåęĆïijŇä¡ăåŔŕèČ¡æČşç§ěéĄŞäÿŇéÍćè£ŹäÿłçőĂå■ŢæĽ'ijèŕ■æşŢæŸŕåŘęèąĺè£řå¿Ůå¡ŞïijŽ

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

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

è£Źäÿłèŕ■æşŢçIJŃäÿŁåŐżæšąåŢěéŮőéćŸïijŇä¡ĘæŸŕåőČå■t’äÿ■èČ¡åŕ§ègL’åĹřæăĞåĞĘåŻŻåĹŹè£ŘçőŮ
æŕŤåęĆïijŇèąĺè¿¿åijŔ "3 + 4 * 5" äijŽå¿ŮåĹř35èĂŇäÿ■æŸŕæIJ§æIJŻçŽĎ23.
åĹĘåijĂä¡£çŤĺâĂİexprâĂİåŠŇâĂİtermâĂİèğĎåĹŹåŔŕäžěèől’åőČæ■čçąőçŽĎåůëä¡IJãĂĆ

åŕźäžŐåď’■æİĆçŽĎèŕ■æşŢïijŇä¡ăæIJĂåě¡æŸŕéĂL’æŃl’æ§ŘäÿłèğčæđŘåůëåĚůæŕŤåęĆPyParsingæĹŰèĂ
äÿŃéİćæŸŕä¡£çŤĺPLYæİěéĞ■åĘŹèąĺè¿¿åijŔæśĆåĂijçĺŃåžŔçŽĎäžčçăĄïijŽ

```
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)

```
    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£Źäÿłçĺ Ńåžŕäÿ■ïijŇæĽ ĂæIJĽäžčçăĄéČ¡ä¡■äžŐäÿĂäÿłæŕŤè¿ČéńŸçŽĎåśĆæňąãĂĆä¡ääŔłéIJĂèęĄäÿ
èĂŇåőđéŹĚçŽĎè£Řèą NèğčæđŕåŹĺïijŇæŐěåŔŮäžđ’çĽŇç■Ľ’ç■Ľ’åžŢåśĆåŁĺä¡IJåůšçżŔèćńåžŞåĞ¡æŢřåőđçŐ

äÿŃéİćæŸŕäÿĂäÿłæĂŐæăůä¡£çŤĺå¿ŮåĹřçŽĎèğčæđŔåŕźèśąçŽĎä¿Ńå■ŘïijŽ

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

åęĆædIJä¡ăæČşåIJĺä¡ăçŽĎçijŰçÍŃè£ĞçÍŃäÿ■æİěçĆźæŇŚæĹŸåŠŇåĹžæ£ĂïijŇçijŰåĘŹèğčædŘåŹĺåŠŇ
åĘ■æňąïijŇäÿĂæIJňçijŰèřŚåŹÍçŽĎäźęçś■äijŽåŇĚåŘńå¿Ĺåď˙ŽåžŢåśĆçŽĎçŘĘèőžç§ěèřĘãĂĆäÿ■è£Ğå¿Ĺåď˙
PythonèĞłåůśçŽĎastæĺąåİŮäź§åĂijå¿ŮåŐžçIJŃäÿĂäÿŃãĂĆ

4.20 2.20 å■ŮèŁĆå■Ůçňęäÿšäÿ£çŽĎå■ŮçňęäÿšæŞ■ä¡IJ

éŮőécŸ

ä¡ăæČşåIJĺå■ŮèŁĆå■Ůçňęäÿšäÿ£æL'ğèąŇæŹőéĂŽçŽĎæŰĞæIJňæŞ■ä¡IJ(æŕŤåęĆçğżéŹď˙ijŇæŘIJçť'ćå

èğčåĘşæŰźæąĹ

å■ŮèŁĆå■ŮçňęäÿšåŕŇŇæăůäź§æŤŕæŇĄåď˙ğéČĺåĹĘåŠŇæŰĞæIJňå■ŮçňęäÿšäÿĂæăůçŽĎåĘĚç¡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■ŮèŁĆæŢřçżĎãĂĆæŕŤåęĆïijŽ

```
>>> 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¿¿åijŔåŇźéĚ■å■ŮèŁĆå■ŮçňęäÿšïijŇä¡ĘæŸŕæ■čåĹŹèąĺè¿¿åijŔæIJňèžńå£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■Ůçňęäÿšäÿ ŁçŽĎæŞ■ä¡IJåİĞåŔŕçŤĺäžŐå■ÙèŁĆå■Ůçňęäÿš çĎűčĂŇïijŇè£ŹéĞŇäź§æIJL’äÿĂäžŻéIJĂèęĄæşÍæĎŔçŽĎäÿ■åŘŇçĆźãĂĆéęŰåĚĹïijŇå■ÙèŁĆå■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 ĎçŘĘéİćåŘŠå■ÙèŁĆçŽĎå■Ůçňęæ Ţřæ■őæIJL’å¡śåŞ■ãĂĆ

çňňäžŇçĆźïijŇå■ÙèŁĆå■Ůçňęäÿšäÿ■äijŽæŔŘä¿ŻäÿĂäÿłç¿ÓèğĆçŽĎå■Ůçňęäÿšèą Íçď 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¡ŢéĂĆçŤĺäžŐå■ÙèŁĆå■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)

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

åęĆædIJä¡ăæČşæăijåijŔåŇŰå■ÛèŁĆå■ÛçňęäÿšïijŇä¡ăå¿ÛåĚĹä¡£çŤĺæăĞåĞĘçŽĎæŰĞæIJňå■Ûçňęäÿš

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

æIJĂåŘŐéIJĂèęĄæşĺæĎŔçŽĎæŸŕïijŇä¡£çŤĺå■ÛèŁĆå■ÛçňęäÿšåŔŕèČ¡äijŽæŤźåŔŸäÿĂäžŻæŞ■ä¡IJçŽĚ
æŕŤåęĆïijŇåęĆædIJä¡ăä¡£çŤĺäÿĂäÿłçijÚçăĄäÿžå■ÛèŁĆçŽĎæŰĞäžűåŘ■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¿Ńå■Řäÿ■çŽĎæIJĂåŘŐéČĺåĹĘçżŹçŻőå¡ŢåŘ■äijăéĂŠäÿĂäÿłå■ÛèŁĆå■ÛçňęäÿšæŸŕæĂŐæăů
åIJĺçŻőå¡Ţäÿ■çŽĎæŰĞäžűåŘ■åŇĚåŘńåŐ§ågŃçŽĎUTF-8çijÚçăĄãĂĆ
åŔĆèĂČ5.15åřŔèŁĆèŐůåŔŰæŻťåď˙ŽæŰĞäžűåŘ■çŻÿåĚşçŽĎåĘĚåőźãĂĆ

æIJĂåŘŐæŔŘäÿĂçĆźijÑäÿĂäžŻçĺŃåžŔåŚŸäÿžäžĘæŔŘå■ĞçĺŃåžŔæĽğèąŇçŽĎéĂ§åžęäijŽåĂ¿åŘŚäž
åř¡çőąæŞ■ä¡IJå■ÛèŁĆå■ÛçňęäÿšçąőåőđäijŽæŕŤæŰĞæIJňæŻť˙åŁäéńŸæŢĹ(åŻääÿžåď˙ĎçŘĘæŰĞæIJňåŻžæI
è£Žæăůå˛ĄŽéĂŽåÿÿäijŽåŕijèĞť˙éİđåÿÿæİĆäźśçŽĎäžčçăĄãĂĆä¡ääijŽçżŔåÿÿåŘŚçŐřå■ÛèŁĆå■Ûçňęäÿšåźűäÿ
åźűäÿŤä¡äè£Ÿå¿ÛæĽŃåŁĺåď˙ĎçŘĘæĽĂæIJĽçŽĎçijŰçăĄ/èğčçăĄæŞ■ä¡IJãĂĆ
åİęçŹ¡èőšïijŇåęĆædIJä¡ăåIJĺåď˙ĎçŘĘæŰĞæIJňçŽĎèŕİïijŇåřśçŻť˙æŐěåIJĺçĺŃåžŔäÿ■ä¡£çŤĺæŹőéĂŽçŽĎæŰĆ

5 çňňäÿĽçńäïijŽæŢřå■ÛæÙěæIJ§åŠŇæÙűéÙť

åIJĺPythonäÿ■æĽğèąŇæŢť˙æŢřåŠŇæţőçĆźæŢřçŽĎæŢřå■ęè£ŘçőÛæÙűå¿ĹçőĂå■ŢçŽĎãĂĆ
åř¡çőąåęĆæ■ď˙ïijŇåęĆædIJä¡ăéIJĂèęĄæĽğèąŇåĹĘæŢřãĂĄæŢřçżĎæĹŰèĂĚæŸŕæÙěæIJ§åŠŇæÙűéÙť˙çŽĎ
æIJňçńäéŻĘäÿ■èőĺèőžçŽĎåřśæŸŕè£ŹäžŻäÿžécŸãĂĆ

Contents:

5.1 3.1 æŢřå■ŮçŽĎåŻŻèĹ■ážŤåĚě

éŮőécŸ

ä¡ăæČşåŕźæţőçĆźæŢřæĽğèąŇæŇĞåőŽçš¿åžęçŽĎèĹ■åĚěè£ŘçőŮãĂĆ

èğčåĘşæŰźæąĹ

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

```
>>> 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¡åIJÍäÿď äÿłè¿źçŢŇçŽĎäÿ■éŮť çŽĎæŮűåĂŹïijŇ `round` åĞ¡æŢřè£ŤåŻđçężåőČæIJĂè£ŞçŽĎåĄűæŢřãĂĆ äźş§åŕśæŸŕèŕťïijŇåŕź1.5æĹŰèĂĚ2.5çŽĎèĹ■åĚěè£ŘçőŮéČ¡

äijăçżŹ `round()` åĞ¡æŢřçŽĎ `ndigits` åŔĆæŢřåŔŕäžěæŸŕèť §æŢřïijŇè£Źçğ■æČĚåĘţäÿŃïijŇèĹ■åĚěè£ŘçőŮäijŽä¡IJçŤĺåIJÍå■Ąä¡■ãĂĄçŹ¿ä¡■ãĂĄå■Čä¡■ç■L'äÿŁéİćãĂĆærŤåęĆïijŽ

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

èőÍèőž

äÿ■èęĄåřĘèĹ■åĚěåŠŇæäijåijŔåŇŰè¿ŞåĞžæŘđæůůæůĘäžĘãĂĆ åęĆæđIJä¡ăçŽĎçŽőçŽĎåŔłæŸŕçőĂå■ŢçŽĎè¿ŞåĞžäÿĂåőŽåő¡åžęçŽĎæŢřïijŇä¡ăäÿ■éIJĂèęĄä¡£çŤÍ `round()` åĞ¡æŢřãĂĆ èĂŇäžĚäžĚåŔłéIJĂèęĄåIJÍæäijåijŔåŇŰçŽĎæŮűåĂŹæŇĞåőŽçš¿åžęå■şåŔŕãĂĆærŤ

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

(continues on next 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¡£çŤĺåĹřæţőçĆźçŽĎçĺŃåžŔïijŇæšąæIJL’å£ĚèęĄäź§äÿ■æŐĺè■Řè£ŹæăůåĄŽãĂĆ
år¡çőąåIJĺèőąçőŮçŽĎæŮűůåĂŹäijŽæIJL’äÿĂçĆźçĆźåřŔçŽĎèŕřåůőïijŇä¡ĘæŸŕè£ŹäžŽåřŔçŽĎèŕřåůőæŸŕèČ¡è
åęĆæđIJäÿ■èČ¡åĚĄèőÿè£ŹæăůçŽĎåřŔèŕřåůő(æŕŤåęĆæűĽ’åŔŁåĹřéĞŚèđ■écĘå§§)ïijŇéĆčäźĹåřšå¿ŮèĂČèŹ
`decimal` æĺąåİŮäžĘïijŇäÿŃäÿĂèŁĆæĹŚäžňäijŽèŕęçżĘèőĺèőžãĂĆ

5.2 3.2 æĽ’ğèąŇçš¿çąőçŽĎæţőçĆźæŢřè£ŘçőŮ

éŮőécŸ

ä¡ăéIJĂèęĄåŕźæţőçĆźæŢřæĽ’ğèąŇçš¿çąőçŽĎèőąçőŮæŞ■ä¡IJïijŇåźűäÿŤäÿ■åÿŃæIJŽæIJL’äżżä¡ŢåŕŔèŕŕ

èğčåĘşæŰźæąĹ

æţőçĆźæŢřçŽĎäÿĂäÿłæŹőéĄ■éŮőécŸæŸŕåőČäżňåźűäÿ■èČ¡çš¿çąőçŽĎèąĺçď žå■Ąè£ŽåĹűæŢřãĂĆ
åźűäÿŤïijŇå■şä¡£æŸŕæIJĂçőĂå■ŢçŽĎæŢřå■ęè£ŘçőŮäź§äijŽäžğçŤ§åŕŔçŽĎèŕřåůőïijŇæŕŤåęĆïijŽ

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

è£ŹäžŽéŤŹèŕŕæŸŕçŤšåžŢåšĆCPUåŠŇIEEE 754æäĞåĞĘéĂŽè£ĞèĞłåůšçŽĎæţőçĆźå■Ţä¡■åŐżæĽ’ğèąŇ
çŤšäžŐPythonçŽĎæţőçĆźæŢřæ■őçšžåđŃä¡£çŤĺåžŢåšĆèąĺçď žå■ŸåĆĺæŢřæ■őïijŇåŽäæ■ď ä¡äæšąåŁđæşŢåŐ

åęĆæđIJä¡äæČşæŻť’åŁäçš¿çąő(åźűèČ¡åőźå£■äÿĂåőŽçŽĎæĂğèČ¡æ■§èĂŮ)ïijŇä¡ääŔŕäžëä¡£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
```

åĹİçIJŃèţůæİěïijŇäÿŁéİćçŽĎäžčçăĄåě¡åČŔæIJL’çĆźåěĞæĂłïijŇæŕŤåęĆæĹŚäžňçŤĺå■Ůçňęäÿšæİěèąĺçc
çĎűěĂŇïijŇ `Decimal` åŕźèśąäijŽåČŔæŹőéĂŽæţőçĆźæŢřäÿĂæăůçŽĎåůěä¡IJ(æŤŕæŇĄæĽĂæIJL’çŽĎåÿÿçT
åęĆæđIJä¡ăæĽŞå■řáőČäžňæĹŰèĂĚåIJĺå■ŮçňęäÿšæăijåijŔåŇŰåĞ¡æŢřäÿ■ä¡£çŤĺáőČäžňïijŇçIJŃèţůæİěèů§a

`decimal` æĺąåİŮçŽĎäÿĂäÿłäÿżèęĄçĽźå¿ĄæŸŕåĚĄèőÿä¡ăæŐğåĹűèőąçőŮçŽĎæŕŔäÿĂæŰźéİćïijŇåŇ
äÿžäžĘè£ŹæăůåĄŽïijŇä¡ăåĚĹå¿ŮåĹŻåžžäÿĂäÿłæIJňåIJřäÿŁäÿŃæŰĞåźűæŻť’æŤźáőČçŽĎèő¿ç¡őïijŇæŕŤåęC

```
>>> from decimal import localcontext
>>> a = Decimal('1.3')
>>> b = Decimal('1.7')
>>> print(a / b)
0.76470588235294117647058823 53
>>> with localcontext() as ctx:
...     ctx.prec = 3
...     print(a / b)
...
0.765
>>> with localcontext() as ctx:
...     ctx.prec = 50
...     print(a / b)
...
0.76470588235294117647058823529411764705882352941176
>>>
```

èőÍèőž

`decimal` æĺąåİŮåőđçŐřäžĘIBMçŽĎâĂİéĂŽçŤĺářŔæŢřè£ŘçőŮèğĎèŇČâĂİãĂĆäÿ■çŤĺèŕť’ïijŇæIJL’å¿

PythonæŰřæL’ŃäijŽåĂ¿åŘŚäžŐä¡£çŤĺ `decimal` æĺąåİŮæİěåď’ĎçŘĘæţőçĆźæŢřçŽĎçš¿çąőè£ŘçőŮãĂ
çĎűěĂŇïijŇåĚĹçŘĘèğčä¡ăçŽĎåžŤçŤĺçĺŃåžŔçŻőçŽĎæŸŕéİđåÿÿéĞ■èęĄçŽĎãĂĆ
åęĆæđIJä¡ăæŸŕåIJĺåĄŽçğŚå■ęèőąçőŮæĹŰåůěçĺŃéćĘå§§çŽĎèőąçőŮãĂĄçŤţèĎŚçżŸåŻ¿ïijŇæĹŰèĂĚæŸŕç
éĆčäźĹä¡£çŤĺæŹőéĂŽçŽĎæţőçĆźçśżåđŇæŸŕæŕŤè¿ČæŹőéĄ■çŽĎåĄŽæşŢãĂĆ
åĚűäÿ■äÿĂäÿłåŐ§åŻăæŸŕïijŇåIJĺçIJ§åőđäÿŰçŢŃäÿ■å¿ĹářŚäijŽèęĄæśĆçš¿çąőåĽřæŹőéĂŽæţőçĆźæŢřèČ¡æ
åŽăæ■d’ïijŃèőąçőŮè£ĞçĺŃäÿ■çŽĎéĆčäźĹäÿĂçĆźçĆźçŽĎèŕŕåůőæŸŕèćńåĚĄèőÿçŽĎãĂĆ
çňňäžŇçĆźåřśæŸŕïijŇåŐ§çŤ§çŽĎæţőçĆźæŢřèőąçőŮèęĄå£ńçŽĎåď’Ž-
æIJL’æŮűåĂŹä¡ăåIJĺæL’ğèąŇåď’ğéĞŔè£ŘçőŮçŽĎæŮűåĂŹéĂ§åžęäź§æŸŕéİđåÿÿéĞ■èęĄçŽĎãĂĆ

å■şä¿£åęĆæ■d’ïijŇä¡ăå■t’äÿ■èČ¡åőŇåĚĺå£¡çŤèèŕŕåůõãĂĆæŢřå■ęåőűèŁśäžĘåď’ğéĞŔæŮűéŮt’åŐżçăŤçl
ä¡ăäź§å¿ŮæşĺæĎŔäÿŃåĞŔæşŢåĹäéŹd’äžěåŔŁåd’ğæŢřåŠŃåřŔæŢřçŽĎåŁăåĹĘè£ŘçőŮæL’ĂåÿęæİěçŽĎå¡śå

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

äÿŁéİćçŽĎéŤŹèŕŕåŔŕäžěåĽľçŤÍ math.fsum() æĽĂæŔŘä¿ŻçŽĎæŻť çš¿çąőèőąçőŮèČ¡åŁŻæİěèğčåĘ

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

çĎűèĂŃïijŃåŕźäžŐåĚűäžŰçŽĎçőŮæşŢïijŃä¡ăäžŤèŕěäžŤçżĘçăŤçľ űåőČåźűçŘĘèğčåőČçŽĎèŕŕåůőäžğçŤ

æĂżçŽĎæİěèŕť ïijŃ decimal æĺąåİŮäÿżèęAçŤĺåIJĺæűĽåŔŁåĽŕéĞŚèđ■çŽĎéćĘå§§ãĂĆ
åIJĺè£ŹçśżçĺŃåžŔäÿ■ïijŃåŞłæĂŢæŸŕäÿĂçĆźåřŔåřŔçŽĎèŕŕåůőåIJĺèőąçőŮè£ĞçĺŃäÿ■èŤŞåżűéČ¡æŸŕäÿ■åĚA
åŽăæ■ďïijŃ decimal æĺąåİŮäÿžèğčåĘşè£ŹçśżéŮőéćŸæŔŘä¿ŻäžĘæŰźæşŢãĂĆ
å¡ŞPythonåŠŇæŢřæ■őåžŞæĽŞäžďéĄŞçŽĎæŮűåĂŹäź§éĂŽåÿÿäijŽéĄĞåĹř Decimal
åŕźèśąïijŃåźűäÿŤïijŃéĂŽåÿÿäź§æŸŕåIJĺåďĎçŘĘéĞŚèđ■æŢřæ■őçŽĎæŮűåĂŹãĂĆ

5.3 3.3 æŢřå■ŮçŽĎæăijåijŔåŇŰè¿ŞåĞž

éŮőéćŸ

ä¡ăéIJĂèęAåřĘæŢřå■Ůæăijåijŕåˇ

ä¡ăéIJĂèęAåřĘæŢřå■ŮæăijåijŔåŇŰåŘŐè¿ŞåĞžïijŃåźűæŐğåĹűæŢřå■ŮçŽĎä¡■æŢřãĂĄåŕźé¡ŘãĂĄå■Č

èğčåĘşæŰźæąĹ

æăijåijŔåŇŰè¿ŞåĞžå■ŢäÿłæŢřå■ŮçŽĎæŮűåĂŹïijŃåŔŕäžěä¡£çŤĺåĘĚç¡őçŽĎ
format() åĞ¡æŢřïijŃæŕŤåęĆïijŽ

```
>>> 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)

```
'  1234.6  '

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

åęĆæđIJä¡ăæČşä¡£çŤÍæŇĞæŢřèőřæşŢïijŇářĘfæŤźæĹŘeæĹŰèĂĚE(åŔŰåĘşäžŐæŇĞæŢřè¿ŞåĞžçŽĎåd

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

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

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

èőÍèőž

æŢřå■Ůæăijåij ŔåŇŰè¿ŞåĞžéĂŽåÿÿæŸŕæŕŤè¿ČçőĂå■ŢçŽĎãĂĆäÿŁéÍćæijŤçď'žçŽĎæŁĂæIJŕåŘŇæŮű `decimal` æĺąåİŮäÿ■çŽĎ `Decimal` æŢřå■Ůåŕźèśąã ĂĆ

å¡ŞæŇĞåőŽæŢřå■ŮçŽĎä¡■æŢřåŘŐïijŇçżŞædIJåĂijäijŽæăźæ■ő `round()` åĞ¡æŢřåŘŇæăůçŽĎèğĎåĹŹè£ŻèąŇåŻŻèĹ■äžŤåĚěåŘŐè£ŤåŻđãĂĆæŕŤåęĆïijŽ

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

åŇĚåŘńå■Čä¡■çňęçŽĎæăijåijŔåŇŰèů§æIJňåIJřåŇŰæšąæIJL'åĚşçşžãĂĆ åęĆæđIJä¡ăéIJĂèęĄæăźæ■őåIJřåŇžæİěæŸ¿çď'žå■Čä¡■çňę¨ijŇä¡ăéIJĂèęĄèĞłåůšåŐżèřČæ§ěäÿŃ `locale` æĺąåİŮäÿ■çŽĎåĞ¡æŢřäžĘãĂĆ ä¡ăåŘŇæăůäź§åŔŕäžěä¡£çŤĺå■ŮçňęäÿšçŽĎ `translate()` æŰźæşŢæİěäžď'æ■ćå■Čä¡■çňęãĂĆæŕŤåęĆïijŽ

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

åIJÍå¿Ĺåď'ŽPythonäžčçžĂąäÿ■äijŽçIJŃåĹřä¡£çŤÍ%æİěæäijåijŔåŇŰæŢřå■ŮçŽĎïijŇæŕŤåęĆïijŽ

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

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

5.4 3.4 äžŇåĚńå■ĄåĚ■è£ŻåĹűæŢť'æŢř

éŮőécŸ

ä¡ăéIJĂèęĄè¡ňæ■ćæĹŰèĂĚè¿ŞåĞžä¡£çŤÍäžŇè£ŻåĹűïijŇåĚńè£ŻåĹűæĹŰå■ĄåĚ■è£ŻåĹűèąÍçd'žçŽĎæŢ

èğčåĘşæŰźæąĹ

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

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

åŔęåď'ŰïijŇåęĆæđIJä¡ăäÿ■æČşè¿ŞåĞž `0b`，`0o` æĹŰèĂĚ `0x` çŽĎåL'■çijĂçŽĎèŕÏïijŇåŔŕäžěä¡£çŤÍ `format()` åĞ¡æŢřãĂĆæŕŤåęĆïijŽ

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

(continues on next page)

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

æŢťˇæŢřæŸŕæIJL’çňęåŔůçŽĎïijŇæL’Ăäžěåę Ćæđ IJä¡ăåIJĺåď’ĎçŘĘèť§æŢřçŽĎèŕİïijŇè¿ŞåĞžçżŞæđ IJäij

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

åęĆæđIJä¡ăæČşäžğçŤ§äÿĂäÿłæŮăçňęåŔůåĂijïijŇä¡ăéIJĂèęĄåćđåŁääÿĂäÿłæŇĞçď’žæIJĂåď’ğä¡■éŢ£åž

```
>>> x = -1234
>>> format(2**32 + x, 'b')
'11111111111111111111011001011110'
>>> format(2**32 + x, 'x')
'fffffb2e'
>>>
```

äÿžäžĘäžěäÿ■åŘŇçŽĎè£ŻåĹűè¡ňæ■ćæŢťˇæŢřå■ŮçňęäÿšïijŇçőĂå■ŢçŽĎä¡£çŤĺåÿęæIJL’è£ŻåĹűçŽĎ `int()` åĞ¡æŢřå■şåŔŕïijŽ

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

èőĺèőž

åď’ğåď’ŽæŢřæČĚåĘţäÿŃåď’ĎçŘĘäžŇè£ŻåĹűãĂĄåĚńè£ŻåĹűåŠŇå■ĄåĚ■è£ŻåĹűæŢťˇæŢřæŸŕå¿ĹçőĂå
åŘłèęĄèőřä¡Ŕè£ŹäžŻè¡ňæ■ćåśđäžŐæŢťˇæŢřåŠŇåĚűåŕźåžŢçŽĎæŰĞæIJňèąĺçď’žäźŃéŮťˇçŽĎè¡ňæ■ćå■şåŔŕã

æIJĂåŘŐïijŇä¡£çŤĺåĚńè£ŻåĹűçŽĎçĺŃåžŘåŚŸæIJL’äÿĂçĆźéIJĂèęĄæşĺæĎŔäÿŃãĂĆ
PythonæŇĞåőŽåĚńè£ŻåĹűæŢřçŽĎèŕ■æşŢèů§åĚűäžŰèŕ■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£İåĚńè£ŻåĹűæŢřçŽĎåL’■çijĂæŸŕ 0○ ïijŇářśåČŔäÿŃéİćè£Źæăůïijž

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

5.5 3.5 å■ÙèŁĆåĹřåď ğæŢťæŢřçŽĎæĽŞåŇĚäÿŐèğčåŇĚ

éŮőécŸ

ä¡ăæIJL'äÿĂäÿłå■ÙèŁĆå■ŮçňęäÿšåźűæČşåřĘåőČèğčåŐŃæĹŘäÿĂäÿłæŢťæŢřăĂĆæĹŰèĂĚïijŇä¡ăéIJĂ

èğčåĘşæŰźæąĹ

åĄĞèő¿ä¡ăçŽĎçÍŃåžŔéIJĂèęĄåď ĎçŘEäÿĂäÿłæŃěæIJL'128ä¡■éŢ£çŽĎ16äÿłåĚČçt'ăçŽĎå■ÙèŁĆå■Ůç

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

äÿžäžĘåřĘbytesèğčæđŘäÿžæŢťæŢřïijŇä¡£çŤÍ `int.from_bytes()` æŰźæşŢïijŇåźűåČŔäÿŃéÍć è£ŹæăůæŇĞåőŽå■ÙèŁĆéąžåžŔïijŽ

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

äÿžäžĘåřĘäÿĂäÿłåď ğæŢťæŢřè¡ňæ■ćäÿžäÿĂäÿłå■ÙèŁĆå■ŮçňęäÿšïijŇä¡£çŤÍ `int.to_bytes()` æŰźæşŢïijŇåźűåČŔäÿŃéÍć è£ŹæăůæŇĞåőŽå■ÙèŁĆæŢřåŠŇå■ÙèŁĆéąžåžŔïijŽ

```
>>> 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

åď ğæŢťæŢřåŠŇå■ÙèŁĆå■ŮçňęäÿšäźŃéŮťçŽĎè¡ňæ■ćæŞ■ä¡IJåźűäÿ■åÿÿèğĄãĂĆ çĎűèĂŇïijŇåIJÍäÿĂäžŽåžŢçŤÍéćĘå§§æIJL'æŮűåĂŹäź§äijŽåĞžçŐřïijŇærŤåęĆårĘçăĄå■ęæĹŰèĂĚç¡ŚçżIJã ä¿ŃåęĆïijŇIPv6ç¡ŚçżIJåIJřåİĂä¡£çŤÍäÿĂäÿł128ä¡■çŽĎæŢťæŢřèąÍçď žãĂĆ åęĆæđIJä¡ăèęĄäžŐäÿĂäÿłæŢřæ■őèőřå¡Ţäÿ■æŔŘåŔŰè£ŹæăůçŽĎåĂijçŽĎæŮűåĂŹïijŇä¡ăářśäijŽéÍćářźè£Ź

ä¡IJäÿžäÿĂçğ■æŻ£äžčæŰźæąĹïijŇä¡ăåŔŕèČ¡æČşä¡£çŤÍ6.11ářŔèŁĆäÿ■æĽĂäżŃçż■çŽĎ `struct` æÍąåİŮæİěèğčåŐŃå■ÙèŁĆãĂĆ è£ŹæăůäźšèąŇå¿ŮéĂŽïijŇäÿ■è£ĞåĹľçŤÍ

struct æÍąåİŮæİěèğčåŐŃåŕźäžŐæŢťæŢřçŽĎåď ğářŔæŸŕæIJL'éŹŘåĹűçŽĎãĂĆ
åŻžæ■d'ïijŇä¡ăåŔŕèČ¡æČşèğčåŐŃåď'Žäÿłå■ŮèŁĆäÿšåźűåřĘçżŞæđIJåŘĹåźűäÿžæIJĂçżĹçŽĎçżŞæđIJïijŇåřś

```
>>> 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■ŮèŁĆéąžåžŔèğĎåĹŹ(littleæĹŰbig)äżĚäżĚæŇĞåőŽäžĘæđĎåżžæŢťæŢřæŮűçŽĎå■ŮèŁĆçŽĎä¡Őä¡■
æĹŚäžñäżŐäÿŃéİćçš¿å£ČæđĎéĂäçŽĎ16è£ŻåĹűæŢřçŽĎèąĺçď'žäÿ■åŔŕäžěå¿Ĺåőźæ ŸŞçŽĎçIJŃåĞžæİěïijŽ

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

åęĆæđIJä¡ăèŕŢçİÅåřĘäÿĂäÿłæŢťæŢřæL'ŞåŇĚäÿžå■ŮèŁĆå■ŮçňęäÿšïijŇéĆčäźĹåőČåřśäÿ■åŘĹéĂĆäžĚ
åęĆæđIJéIJĂèęĄçŽĎèŕÏïijŇä¡ăåŔŕäžěä¡£çŤÍ `int.bit_length()`
æŰźæşŢæİěåĘşåőŽéIJĂèęĄåď'ŽåřŚå■ŮèŁĆä¡■æİěå■ŸåĆÍè£Źäÿłå Ăijã ĂĆ

```
>>> x = 523 ** 23
>>> x
335381300113661875107536852714019056160355655333978849017944067
>>> 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¡ăåĘŹçŽĎæIJĂæŰřçŽĎç¡ŚçżIJèőď'èŕĄæŰźæąĹäžčçăĄéĄĞåĹřäžĘäÿĂäÿłéŽ¿écŸïijŇåźűäÿŤä¡ăåŤŕäÿ
åĘ■æĹŰèĂĚæŸŕä¡ăäžĚäžĚéIJĂèęĄä¡£çŤĺåď'■æŢřæİěæL'ğèąŇäÿĂäžŻèőąçőŮæŞ■ä¡IJãĂĆ

èğčåĘşæŰźæąĹ

åď■æŢřåŔŕäžěçŤĺä¡£çŤĺåĞ¡æŢř `complex(real, imag)` æĹŰèĂĚæŸŕåÿęæIJĽåŘŐçijĂjçŽĎætőçĆźæŢřæİěæŇĞåőŽãĂĆæŕŤåęĆïijŽ

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

åŕźåžŤçŽĎåőđéČĺãĂąèŹŽéČĺåŠŇåĚśè¡■åď■æŢřåŔŕäžěå¿Ĺåőźæ ŸŞçŽĎèŐůåŔŰãĂĆåřśåČŔäÿŃéİćè£Ź

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

åŘęåď Űïij ŇæĽĂæIJĽåÿÿèğĄçŽĎæŢřå■ęè£ŘçőŮéČ¡åŕŕäžěåůěä¡IJïijŽ

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

åęĆæđIJèęĄæĽğèąŇåĚűäžŰçŽĎåď■æŢřåĞ¡æŢřæŕŤåęĆæ■čåijęãĂĄä¡Źåijęæ ĹŰåźşæŰźæăźïijŇä¡£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 ïijŇåŔŕäžěå¿Ĺåőźæ ŸŞçŽĎæđĎéĂääÿĂäÿłåď ■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¡ăçŽĹ

```
>>> 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 ŤřæŇĄåď ■æŢřçŽĎåžŞäÿ■åčřæŸŐåď ■æŢřçśžåđŃçŽĎä¡£çŤÍãĂĆæ

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

5.7 3.7 æŮăçĺůåď ğäÿŐNaN

éŮőécŸ

ä¡ăæČşåĹŻåžžæĹŰæţŃèŕŢæ■čæŮăçĺůãĂĄèť §æŮăçĺůæĹŰNaN(éİđæŢřå■Ů)çŽĎæţőçĆźæŢřãĂĆ

èğčåĘşæŰźæąĹ

PythonåźűæšąæIJL’çL’źæőŁçŽĎèŕ■æşŢæİěèąÍçď žè£ŹäžŻçL’źæőŁçŽĎæţőçĆźåĂijïijŇä¡ĘæŸŕåŔŕäžěä¡£ float() æİěåĹŻåžžåőČäžñãĂĆæŕŤåęĆïijŽ

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

(continues on next page)

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

äÿžäžĘæţŃèŕŢè£ŹäžŻåĂijçŽĎå■ŸåIJĺïijŇä¡£çŤĺ `math.isinf()` åŠŇ `math.isnan()` åĞ¡æŢřãĂĆæŕŤåęĆïijŽ

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

èőĺèőž

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

æŮăçl'ůåď'ğæŢřåIJĺæL'ğèąŇæŢřå■ęèőąçőŮçŽĎæŮűåĂŹäijŽäijăæŠ■ïijŇæŕŤåęĆïijŽ

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

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

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

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

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

(continues on next page)

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

NaNåĂijçŽĎäýĂäýłçL'źåĹńçŽĎåIJřæŰźæŮűåőČäžňäźŃéŮt'çŽĎæŕŤè¿ČæŞ■ä¡IJæĂżæŸŕè£ŤåŻđFalseâ

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

çŤśäžŐè£Źäýłå§åŻăïijŇæţŃèŕŢäýĂäýłNaNåĂijå¿ŮåŤŕäýĂåőL'åĚĺçŽĎæŰźæşŢářśæŸŕä¡£çŤĺ `math.isnan()` ïijŇäź§ářśæŸŕäýŁéİćæijŤçd'žçŽĎéĆčæăůãĂĆ

æIJL'æŮűåĂŹçĺŃåžŔåŚŸæČşæŤźåŔŸPythonézŸèőd'èąŇäýžïijŇåIJĺè£ŤåŻđæŮăçl'ůåd'ğæĹŰNaNçżŞæ `fpectl` æĺąåİŮåŔŕäžěçŤĺæİěæŤźåŔŸè£Źçğ■èąŇäýžïijŇä¡ĘæŸŕåőČåIJĺæăĞåĞĘçŽĎPythonæđĎåžžäý■åźűåźűäýŤéŠĹářźçŽĎæŸŕäýŞåőűçžğçĺŃåžŔåŚŸãĂĆåŔŕäžěåŔĆèĂČåIJĺçž£çŽĎPythonæŰĞæąčèŐůåŔŰæŻt'åd

5.8 3.8 åĹĘæŢřè£ŘçőŮ

éŮőécŸ

ä¡äè£ŻåĚěæŮűéŮt'æIJžåŹĺïijŇçłĄçĎűåŔŚçŐřä¡äæ■čåIJĺåĄŽářŔå■ęåőűåž■ä¡IJäýŽïijŇåźűæűL'åŔŁåĹřă æĹŰèĂĚä¡äåŔŕèČ¡éIJĂèęĄåĘŹäžčçăĄåŐżèőąçőŮåIJĺä¡ăçŽĎæIJĺåůěåůěåŐĆäÿ■çŽĎæţŃéĞŔåĂijãĂĆ

èğčåĘşæŰźæąĹ

`fractions` æĺąåİŮåŔŕäžěèćńçŤĺæİěæL'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)

```
>>> 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 ŽæŢřçÍŃåžŔäý■äýĂèĹňäý■äijŽåĞžçŐřåĹĘæŢřçŽĎèőąçőŮéŮőécŸïijŇä¡ĘæŸŕæIJL’æŮűåĂŽ
ærŤåęĆïijŇåIJÍäýĂäýłåĚĄèőýæŐěåŔŮåĹĘæŢřå¡ćåijŔçŽĎæţŃèŕŢå■Ţä¡■åźűäžěåĹĘæŢřå¡ćåijŔæL’ğèąŇè£Ř
çŻť æŐěä¡£çŤĺåĹĘæŢřåŔŕäžěåĞŔåřŚæL’ŃåŁÍè¡ňæ■ćäýžářŔæŢřæĹŮæţőçĆźæŢřçŽĎåůěä¡IJãĂĆ

5.9 3.9 åď ğåđŃæŢřçżĎè£ŘçőŮ

éŮőécŸ

ä¡ăéIJĂèęĄåIJÍåď ğæŢřæ■őéŻĘ(ærŤåęĆæŢřçżĎæĹŰç¡Śæăij)äýŁéİćæL’ğèąŇèőąçőŮãĂĆ

èğčåĘşæŰźæąĹ

æűL’åŔŁåĹřæŢřçżĎçŽĎéĞ■éĞŔçžğè£ŘçőŮæŞ■ä¡IJïijŇåŔŕäžěä¡£çŤÍ `NumPy` åžŞãĂĆ
`NumPy` çŽĎäýĂäýłäýżèęĄçL’źå¿ĄæŸŕåőĆäijŽçżŹPythonæŔŘä¿ŻäýĂäýłæŢřçżĎåŕźèśąïijŇçŻÿærŤæăĞåĞĘ
äýŃéİćæŸŕäýĂäýłçőĂå■ŢçŽĎåřŔä¿Ńå■ŘïijŇåŘŚä¡ăåśŢçd’žæăĞåĞĘåĹŮèąĺåŕźèśąåŠŇ
`NumPy` æŢřçżĎåŕźèśąáźŃéŮt’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)

```
>>> 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■čåęĆæL’ĂèğĄïijŇäÿd’çğ■æŰźæąĹäÿ■æŢřçżĎçŽĎå§žæIJňæŢřå■ęè£ŘçőŮçżŞæđIJåźűäÿ■çŻÿåŘŇãĂçL’źåĹńçŽĎïijŇ NumPy äÿ■çŽĎæăĞéĞŔè£ŘçőŮ(æŕŤåęĆ ax * 2 æĹŰ ax + 10)äijŽä¡IJçŤĺåIJĺæŕŔäÿĂäÿłåĚČçt’ääÿŁãĂĆ åŘęåd’ŰïijŇå¡Şäÿd’äÿłæŞ■ä¡IJæŢřéČ¡æŸŕæŢřçżĎçŽĎæŮűåĂŹæL’ğèąŇåĚČçt’ääŕźç■L’ä¡■ç¡ôëőąçőŮïijŇåźű

åŕźæŢt’äÿłæŢřçżĎäÿ■æL’ĂæIJL’åĚČçt’ääŘŇæŮűæL’ğèąŇæŢřå■ęè£ŘçőŮåŔŕäżěä¡£å¿Ůä¡IJçŤĺåIJĺæŢt’ä
æŕŤåęĆïijŇåęĆæđIJä¡ăæČşèőąçőŮåd’Žéąźåijŕ̨çŽĎåĂijïijŇåŔŕäżěè£ŹæăůåĄŽïijŽ

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

NumPy è£ŸäÿžæŢřçżĎæŞ■ä¡IJæŔŘä¿ŻäžĘåd’ğéĞŔçŽĎéĂŽçŤĺåĞ¡æŢřïijŇè£ŹäžŻåĞ¡æŢřåŔŕäżěä¡IJäÿ
math æĺąåİŮäÿ■çśžäijijåĞ¡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¡æŢřæL’ğèąŇèőąçőŮèęĄå£ńçŽĎåd’ŽãĂĆ
åŻăæ■d’ïijŇåŘłèęĄæIJL’åŔŕèČ¡çŽĎèŕĺår¡éĞŔéĂL’æŇl’ NumPy çŽĎæŢřçżĎæŰźæąĹãĂĆ

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

äź§åřśæŸŕèŕťïijŇåőČäžňæŸŕäÿĂäÿłéİđåÿÿåď˙ğçŽĎè£đçż■çŽĎåźűçŤśåŘŇçśżåđŃæŢřæ■őçżĎæĹŘçŽĎåĘĚå
æĽĂäżěïijŇä¡ăåŔŕäžěæđĎéĂääÿĂäÿłæŕŤæŹőéĂŽPythonåĹŮèąĺåď˙ğçŽĎåď˙ŽçŽĎæŢřçżĎãĂĆ
æŕŤåęĆïijŇåęĆæđIJä¡ăæČşæđĎéĂääÿĂäÿł10,000*10,000çŽĎæţőçĆźæŢřäžŇçżť˙ç¡ŚæäijïijŇå¿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.]])
>>>
```

æĽĂæIJL˙çŽĎæŹőéĂŽæŞ■ä¡IJè£ŸæŸŕäijŽåŘŇæŮűä¡IJçŤĺåIJĺæĽĂæIJL˙åĚČçť˙ääÿŁïijŽ

```
>>> 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 æIJL˙äÿĂçĆźéIJĂèęĄçL˙źåĹńçŽĎäÿżæĎŔïijŇéĆčåřśæŸŕåőČæL˙ľåśŢPythonåĹŮèąĺçŽĎ
- çL˙źåĹńæŸŕåŕźäžŐåď˙Žçżť˙æŢřçżĎãĂĆ äÿžäžĘèŕť˙æŸŐæÿĚæěŽïijŇåĚĹæđĎéĂääÿĂäÿłçőĂå■ŢçŽĎäžŇçżt

```
>>> 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)

```
>>> # 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 æŸŕPythonécĘå§§äÿ■å¿Ĺåď'ŽçğŚå■ęäÿŐåůěçÍŃåžŞçŽĎå§žçąĂïijŇåŘŇæŮűäź§æŸŕèćńåź£æşŽ
å■şä¿£åęĆæ■ď'ïijŇåIJĺåĹŽåijĂågŃçŽĎæŮűåĂŹéĂŽè£ĞäÿĂäžŽçőĂå■ŢçŽĎä¿Ńå■ŘåŠŇçŐľ'åĚůçÍŃåžŔäź§

éĂŽåÿÿæĹŚäžňåŕijåĚě NumPy æĺąåİŮçŽĎæŮűåĂŹäijŽä¡£çŤĺèŕ■åŔě `import numpy as np` ãĂĆ è£ŹæäůçŽĎèŕÍä¡ăåřśäÿ■çŤĺåĘ■ä¡ăçŽĎçÍŃåžŔéĞŇéİćäÿĂéĄ■éĄ■çŽĎæŢšåĚě numpy ïijŇåŘłéIJĂèęĄè¿ŞåĚě np åřśèąŇäžĘïijŇèŁĆçIJĄäžĘäÿ■åřŚæŮűéŮť'ãĂĆ

åęĆæđIJæČşèŐůåŘŰæŻť'åď'ŽçŽĎä£ąæĄŕïijŇä¡ăå¡ŞçĎűå¿ŮåŐż NumPy åőŸç¡ŚéĂŽéĂŽäžĘïijŇç¡ŚåİĂæŸŕïijŽ http://www.numpy.org

5.10 3.10 ç§ľéŸţäÿŐçž£æĂğäżčæŢřè£ŘçőŮ

éŮőécŸ

ä¡ăéIJĂèęĄæL'ğèąŇç§l'éŸţåŠŇçž£æĂğäżčæŢřè£ŘçőŮïijŇæŕŤåęĆç§l'éŸţäźŸæşŢãĂĄåŕżæL'¿èąŇåĹŮå

èğčåĘşæŰźæąĹ

NumPy åžŞæIJL'äÿĂäÿłç§l'éŸţåŕźèśąåŔŕäżěçŤĺæİěèğčåĘşè£ŹäÿłéŮőécŸãĂĆ

ç§l'éŸţçśżäijijäžŐ3.9åřŔèŁĆäÿ■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]])
>>>
```

åŔŕäżěåIJĺ numpy.linalg å■ŘåŇĚäÿ■æL'¿åĹřæŻt'åď'ŽçŽĎæŞ■ä¡IJåĞ¡æŢřijŇæŕŤåęĆijŽ

```
>>> import numpy.linalg

>>> # Determinant
>>> numpy.linalg.det(m)
```

(continues on next 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¿ĹæŸ¿çĎűçž£æĂğäžčæŢřæŸŕäÿłéÍđåÿÿåď˙ğçŽĎäÿžécŸïijŇåůšçżŔèűĚåĞžäžĘæIJňäźęèČ¡èőÍèőžçŽĎè
ä¡ĘæŸŕïijŇåęĆæđIJä¡ăéIJĂèęĄæŞ■ä¡IJæŢřçżĎåŠŇåŘŚéĞŔçŽĎèŕÍïijŇ NumPy
æŸŕäÿĂäÿłäÿ■éŤŹçŽĎåĚěåŔčçĆźãĂĆ åŔŕäžěèő£éŮő NumPy åőŸç¡Ś http://www.numpy.org
èŐůåŘŰæŻť åď˙Žä£ąæĄŕãĂĆ

5.11 3.11 éŽŔæIJžéĂĽæŃľ

éŮőécŸ

ä¡ăæČşäżŐäÿĂäÿłåžŔåĹŮäÿ■éŽŔæIJžæŁ¡åŘŰèŃěźšåĚČçť˙ăïijŇæĹŰèĂĚæČşçŢ§æĹŘåĞääÿłéŽŔæIJ

èğčåĘşæŰźæąĹ

random æÍąåİŮæIJL˙åď˙ğéĞŔçŽĎåĞ¡æŢřçŤÍæİěäžğçŤ§éŽŔæIJžæŢřåŠŇéŽŔæIJžéĂĽæŃľ˙åĚČçť˙ăãĂĆ
ærŤåęĆïijŇèęĄæČşäżŐäÿĂäÿłåžŔåĹŮäÿ■éŽŔæIJžçŽĎæŁ¡åŘŰäÿĂäÿłåĚČçť˙ăïijŇåŔŕäžěä¡£çŤÍ
random.choice() ïijŽ

```
>>> import random
>>> values = [1, 2, 3, 4, 5, 6]
>>> random.choice(values)
```

(continues on next 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¡ăäżĚäżĚåŔłæŸŕæČşæL'Şäźśåžŕ̊åĹŮäÿ■åĚČçť'ăçŽĎéąžåžŔïijŇåŔŕäżěä¡£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
>>>
```

äÿžäžEçŤ§æĹŘ0åĹř1èŇČåŻťåĘĚåİĞåŇĂåĹĘåÿČçŽĎæţőçĆźæŢřïijŇä¡£çŤÍ `random.random()` ïijŽ

```
>>> random.random()
0.9406677561675867
>>> random.random()
0.133129581343897
>>> random.random()
0.4144991136919316
>>>
```

åęĆæđIJèęĄèŐůåŘŰNä¡■éŽŔæIJžä¡■(äžŇè£ŻåĹű)çŽĎæŢť æŢřïijŇä¡£çŤÍ `random.getrandbits()` ïijŽ

```
>>> random.getrandbits(200)
335837000776573622800628485064121869519521710558559406913275
>>>
```

èőÍèőž

`random` æĺąåİŮä¡£çŤÍ *Mersenne Twister* çőŮæşŢæİěèőąçőŮçŤ§æĹŘéŽŔæIJžæŢřãĂĆè£ŹæŸŕäÿĂäÿłçää¡ĘæŸŕä¡ăåŘŕäżěéĂŽè£Ğ `random.seed()` åĞ¡æŢřä£őæŤźåĹİåğŃåŇŰçğ■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
```

éŹď'äžĘäÿŁè£řäżŃçż■çŽĎåŁ§èČ¡ïijŇrandomæÍąåİŮè£ŸåŇĚåŘńå§žäžŐåİĞåŇĂåĹĘåÿČãĂĄéńŸæŰŕåærŤåęĆïijŇ `random.uniform()` èőąçŐŮåİĞåŇĂåĹĘåÿČéŽŔæIJžæŢřïijŇ `random.gauss()` èőąçŐŮæ■čæĂĄåĹĘåÿČéŽŔæIJžæŢřãĂĆ åŕźäžŐåĚűäżŰçŽĎåĹĘåÿČæČĚåĘţèřůåŘĆèĂČåIJÍçž£æŰĞæąčãĂĆ

åIJÍ `random` æĺąåİŮäÿ■çŽĎåĞ¡æŢřäÿ■åžŤèŕěçŤÍåIJÍåŠŇårEçăĄå■ęçŻÿåĚşçŽĎçÍŃåžŔäÿ■ãĂĆ åęĆæđIJä¡ăçąőåőđéIJĂèęĄçśżäijijçŽĎåŁ§èČ¡ïijŇåŘŕäżěä¡£çŤÍsslæÍąåİŮäÿ■çŻÿåžŤçŽĎåĞ¡æŢřãĂĆ ærŤåęĆïijŇ `ssl.RAND_bytes()` åŘŕäżěçŤÍæİěçŤ§æĹŘäÿĂäÿłåőĽåĚÍçŽĎéŽŔæIJžå■ŮèŁĆåžŔåĹŮãĂĆ

5.12 3.12 å§žælJňçŽĎæŮěælJ§äÿŐæŮűéŮť è¡ňæ■ć

éŮőécŸ

ä¡ăéIJĂèęĄæĽğèąŇçőĂå■ŢçŽĎæŮűéŮť è¡ňæ■ćïijŇærŤåęĆåd'l'åĹrçğŠïijŇářŔæŮűåĹřåĹĘéŠ§ç■L'çŽĎ

èğčåĘşæŰźæąĹ

äÿžäžEæĽğèąŇäÿ■åŘŇæŮűéŮť å■Ţä¡■çŽĎè¡ňæ■ćåŠŇèőąçőŮïijŇèŕůä¡£çŤÍ `datetime` æĺąåİŮãĂĆ ærŤåęĆïijŇäÿžäžEèąÍçd'žäÿĂäÿłæŮűéŮť æőţïijŇåŔŕäżěåĹŻåžžäÿĂäÿł `timedelta` åőđä¿ŃïijŇářšåČŔäÿŃéİćè£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
>>>
```

åęĆædIJä¡ăæČşèąĺçd'žæŇĞåőŽçŽĎæŮěæIJ§åŠŇæŮűéŮt'ïijŇåĚĹåĹŻåżžäÿĂäÿł datetime åőđä¿ŃçĎűåŘŐä¡£çŤĺæăĞåĞĘçŽĎæŢřå■ęè£ŘçőŮæİěæŞ■ä¡IJåőČäżñãĂĆærŤåęĆïijŽ

```
>>> 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
>>>
```

åIJĺèőąçőŮçŽĎæŮűåĂŹïijŇéIJĂèęĄæşĺæĎŔçŽĎæŸŕ datetime äijŽèĞłåŁĺåd'ĎçŘĘéŮřåźt'ãĂĆærŤåęĆïijŽ

```
>>> 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¡ăéIJĂèęĄæĽ ğèąŇæŻť åŁăåď ■æİĆçŽĎæŮěæIJ§æŞ■ä¡IJïijŇæŕŤåęĆ åŘŕäžěèĂČèŹŠä¡£çŤÍ dateutilæÍąåİŮ

èőÿåď ŽçśžäijijçŽĎæŮűéŮť èőąçőŮåŘŕäžěä¡£çŤÍ `dateutil.relativedelta()` åĞ¡æŢřäžčæŻ£ãĂĆ ä¡ĘæŸŕïijŇæIJL'äÿĂçĆźéIJĂèęĄæşÍæĎŘçŽĎåřśæŸŕïijŇåőČäijŽåIJÍåď 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¡ăéIJĂèęĄæ§ěæL'¿æŸ§æIJ§äÿ■æ§ŘäÿĂåď l'æIJĂåŘŐåĞžçŐřçŽĎæŮěæIJ§ïijŇæŕŤåęĆæŸ§æIJ§äžŤãĂ

èğčåĘşæŰźæąĹ

PythonçŽĎ `datetime` æÍąåİŮäÿ■æIJL'åůěåĚůåĞ¡æŢřåŠŇçśžåŘŕäžěåÿőåŁl'ä¡ăæL'ğèąŇè£ŹæăůçŽĎèőă äÿŃéIćæŸŕåŕźçśžäijijè£ŹæăůçŽĎéŮőécŸçŽĎäÿĂäÿłéĂŽçŤÍèğčåĘşæŰźæąĹïijŽ

```
#!/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
```

åIJÍäžd'äžŠåijŔèğčéĞŁåŹÍäÿ■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` åŘĆæŢřåŘŕäžěçŤśåŔęåd'ŰäÿĂäÿł `datetime` åőđä¿ŃæÌěæŘŘä¿ŻãĂĆærŤåęĆïijŽ

```
>>> get_previous_byday('Sunday', datetime(2012, 12, 21))
datetime.datetime(2012, 12, 16, 0, 0)
>>>
```

èőÍèőž

äÿŁéÌćçŽĎçőŮæşŢåŐ§çŘĘæŸŕè£Źæăůç ŽĎïijŽåĚĹářĘåijĂåğŃæŮěæIJ§åŠŇçŽőæăĞæŮěæIJ§æŸăářĎ çĎűåŘŐéĂŽè£ĞæÍąè£ŘçőŮèőąçőŮåĞžçŽőæăĞæŮěæIJ§èęAçżŔè£Ğåd'ŽářŚåd'l'æL'■èČ¡åĹřè¿¿åijĂåğŃæŮ

åęĆædIJä¡ăèęĄåČŔè£ŹæăůæL'ğèąŇåd'ğéĞŔçŽĎæŮěæIJ§èőąçőŮçŽĎèŕÍïijŇä¡ăæIJĂåě¡åőL'èčĚçňňäÿL

`python-dateutil` æÌěäžčæŻ£ãĂĆ ærŤåęĆïijŇäÿŃéÌćæŸŕæŸŕä¡£çŤÍ `dateutil`

æÍąåİŮäÿ■çŽĎ `relativedelta()` åĞ¡æŢřæL'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¡ŞåL'■æIJĹäż¡çŽĎæŮěæIJ§èŇČåŻť

éŮőécŸ

ä¡ăçŽĎäżčçăĄéIJĂèęĄåIJĺå¡ŞåL'■æIJĹäż¡äÿ■å¿łçŐŕæŕŔäÿĂåď'l'ïijŇæČşæL'¿åĹřäÿĂäÿłèőąçőŮè£Źäÿła

èğčåĘşæŰźæąĹ

åIJĺè£ŹæăůçŽĎæŮěæIJ§äÿŁå¿łçŐŕåźűéIJĂèęĄäžŇåĚĹæđĎéĂăäÿĂäÿłåŇĚåŘńæL'ĂæIJL'æŮěæIJ§çŽĎ
ä¡ăåŔŕäżčåĚĹèőąçőŮåĞžåijĂåğŇæŮěæIJ§åŠŇçżŞæİ§æŮěæIJ§ïijŇ
çĎűåŘŐåIJĺä¡ăæ■ěè£ŻçŽĎæŮűåĂŹä¡£çŤĺ `datetime.timedelta`
åŕźèśąéĂŠåćđè£Źäÿłæ̊ŮěæIJ§åŔŸéĞŔå■şåŔŕãĂĆ

äÿŇéİćæŸŕäÿĂäÿłæŐěåŘŮäżżæĎŔ `datetime` åŕźèśąåźűè£ŤåŻđäÿĂäÿłçŤśå¡ŞåL'■æIJĹäż¡åijĂåğŇæŮ

```
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)
```

æIJL'äžĘè£Źäÿłåŕśåŕŕäżěå¿Ĺåőźæ ŸŞçŽĎåIJĺè£ŤåŻđçŽĎæŮěæIJ§èŇČåŻť'äÿŁéİćåĄŽå¿łçŐŕæŞ■ä¡IJäž

```
>>> 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

äÿŁéÍćçŽĎäżčçăĄåĚĹèőąçőŮåĞžäÿĂäÿłåŕźåžŤæIJĹäż¡çňňäÿĂåď'l'çŽĎæŮěæIJ§ãĂĆ äÿĂäÿłå£ńéĂ§çŽĎæŰźæşŢåřśæŸŕä¡£çŤĺ `date` æĹŰ `datetime` åŕźèśąçŽĎ `replace()` æŰźæşŢçőĂå■ŢçŽĎåřĘ `days` åśđæĂğèő¿ç¡őæĹŘ1å■şåŘŕãĂĆ `replace()` æŰźæşŢäÿĂäÿłåě¡åď'ĎåřśæŸŕåőČäijŽåĹŻåžžåŠŇä¡ăåijĂåğŃäijăåĚěåŕźèśąçśžåđŃçŻÿåŘŇçŽĎåŕźèśąãĂĆ æL'Ăäżěïij Ňåę Ćæđ IJè¿ŞåĚěåŘĆæŢřæŸŕäÿĂäÿł `date` åőđä¿Ńïij ŇéĆčäźĹçżŞæđIJäź§æŸŕäÿĂäÿł `date` åőđä¿ÑãĂĆ åŘŇæăůçŽĎïijŇåęĆæđIJè¿ŞåĚěæŸŕäÿĂäÿł `datetime` åőđä¿ÑïijŇéĆčäźĹä¡ăå¿ŮåĹřçŽĎåřśæŸŕäÿĂäÿł `datetime` åőđä¿ÑãĂĆ

çĎűåŘŐïijŇä¡£çŤĺ `calendar.monthrange()` åĞ¡æŢřæİěæL'¿åĞžèŕěæIJĹçŽĎæĂżåď'l'æŢřãĂĆ äżżä¡ŢæŮűåĂŹåŘłèęĄä¡ăæČşèŐůå¿ŮæŮěåŐĘä£ąæĄŕïijŇéĆčäźĹ `calendar` æĺąåİŮåřśéİđåÿÿæIJL'çŤĺäžĘãĂĆ `monthrange()` åĞ¡æŢřäijŽè£ŤåŽđåŇĚåŘńæŸ§æIJ§åŠŇèŕěæIJĹåď'l'æŢřçŽĎåĚČçżĎãĂĆ

äÿĂæŮęèŕěæIJĹçŽĎåď'l'æŢřåůšç§ěäžĘïijŇéĆčäźĹçżŞæİ§æŮěæIJ§åřśåŘŕäżěéĂŽè£ĞåIJĺåijĂåğŃæŮěæ æIJL'äÿłéIJĂèęĄæşĺæĎŘçŽĎæŸŕçżŞæİ§æŮěæIJ§åźűäÿ■åŇĚåŘńåIJĺè£ŹäÿłæŮěæIJ§èŇČåŽť'åĘĚ(äžŃåőđäÿ è£Źäÿłå ŠŇPythonçŽĎ `slice` äÿŐ `range` æŞ■ä¡IJèąŇäÿžä£ÍæŇĄäÿĂèĞt'ïijŇåŘŇæăůäź§äÿ■åŇĚåŘńçżŞåi

äÿžäžĘåIJĺæŮěæIJ§èŇČåŽť'äŸŁå¿łçŐŕïijŇèęĄä¡£çŤĺåĹřæăĞåĞĘçŽĎæŢřå■ęåŠŇæŕŤè¿ČæŞ■ä¡IJãĂĆ æŕŤåęĆïijŇåŘŕäżěåĹl'çŤĺ `timedelta` åőđä¿ÑæİěéĂŠåćđæŮěæIJ§ïijŇåŕŘäžŐåŘů<çŤĺæİěæčĂæ§ëäÿĂäÿła

çŘĘæČşæČĚåĘţäÿŃïijŇåęĆæđIJèČ¡äÿžæŮěæIJ§è£■äżčåĹŻåžžäÿĂäÿłåŘŇåĘĚç¡őçŽĎ `range()` åĞ¡æŢřäÿĂæăůçŽĎåĞ¡æŢřåŕśåě¡äžĘãĂĆ åźÿè£ŘçŽĎæŸŕïijŇåŘŕäżěä¡£çŤĺäÿĂäÿłçŤ§æĹŘåŹĺçŽĎä¿Ńå■ŘïijŽ

```
def date_range(start, stop, step):
    while start < stop:
        yield start
        start += step
```

äÿŃéİćæŸŕä¡£çŤĺè£ŹäÿłçŤ§æĹŘåŹĺçŽĎä¿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■åőđçŐřäźŃæL’Ăäžěè£ŹäźĹçőĂå■ŢïijŇè£Ÿå¿Ůå¡ŠåŁ§äžŐPythonäÿ■çŽĎæŮěæIJ§åŠŇæŮűéŮť

5.15 3.15 å■Ůçňęäÿšè¡ňæ■ćäÿžæŮěæIJ§

éŮőécŸ

ä¡ăçŽĎåžŤçŤĺçÍŃåžŔæŐěåŔŮå■ŮçňęäÿšæăijåijŔçŽĎè¿ŞåĚěïijŇä¡ĘæŸŕä¡ăæČşåřĘåőČäżňè¡ňæ■ćäÿž `datetime` åŕźèśąäžëä¿£åIJĺäÿŁéİćæL’ğèąŇéİđå■Ůçňęäÿšæş■ä¡IJãĂĆ

èğčåĘşæŰźæąĹ

ä¡£çŤÍPythonçŽĎæăĞåĞĘæĺąåİŮ `datetime` åŔŕäžěå¿Ĺåőźæ ŸŞçŽĎèğčåĘşè£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¿Ĺåd’ŽçŽĎæăijåijŔåŇŰäžčçăĄïijŇ æŕŤåęĆ `%Y` äžčèąĺ4ä¡■æŢřåźt’äž¡ïijŇ `%m` äžčèąĺäÿd’ä¡■æŢřæIJĹäž¡ãĂĆ è£ŸæIJL’äÿĂçĆźåĂijå¿ŮæşĺæĎŔçŽĎæŸŕè£ŹäžŻæăijåijŔåŇŰå■ăä¡■çňęäź§åŔŕäžěåŔ■è£Ğæİěä¡£çŤĺïijŇářE

æŕŤåęĆïijŇåĄĞèő¿ä¡ăçŽĎäžčçăĄäÿ■çŤ§æĹŘäžĘäÿĂäÿł `datetime` åŕźèśąïijŇ ä¡ăæČşåřĘåőČæăijåijŔåŇŰäÿžæijĆäžőæŸŞèŕźå¡ćåijŔåŘŐæŤ¿åIJĺèĞłåŁĺçŤ§æĹŘçŽĎä£ąäžűæĹŰèĂĚæŁěå

```
>>> 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'äÿĂçĆźéIJĂèęĄæşĺæĎŔçŽĎæŸŕïijŇ strptime() çŽĎæĂğèČ¡èęĄæŕŤä¡ăæČşèśąäÿ■çŽĎåůőå¿Ĺåď'ŽïijŇ åŻăäÿžåőČæŸŕä¡£çŤĺçžŕPythonåőđçŐŕïijŇåźűäÿŤå£Ì åęĆæđIJä¡ăèęĄåIJĺäžčçăĄäÿ■éIJĂèęĄèğčæđŘåď'ğéĞŔçŽĎæŮĕæIJ§åźűäÿŤåůšçżŔç§ěéĄŞäžĘæŮĕæIJ§å■Ůç æŕŤåęĆïijŇåęĆæđIJä¡ăåůšçżŔç§ěéĄŞæĽĂäžěæŮĕæIJ§æăijåijŔæŸŕ YYYY-MM-DD ïijŇä¡ăåŔŕäžěåČŔäÿŃéÍćè£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))
```

åőđéŹĚætŃèŕŢäÿ■ïijŇè£ŹäÿłåĞ¡æŢřæŕŤ datetime.strptime() å£ń7åĂ■åď'ŽãĂĆ åęĆæđIJä¡ăèęĄåď'ĎçŘĘåď'ğéĞŔçŽĎæűĽåŔŁåĹřæŮĕæIJ§çŽĎæŢřæ■őçŽĎèŕİïijŇéĆčäźĹæIJĂåě¡èĂČèŹŚäÿ

5.16 3.16 çżŞåŘĹæŮűåŇžçŽĎæŮĕæIJ§æŞ■ä¡IJ

éŮőécŸ

ä¡ăæIJL'äÿĂäÿłåőĽæŐŠåIJĺ2012åźť12æIJĹ21æŮĕæŮľäÿŁ9:30çŽĎçŤţèŕİäijŽèőőïijŇåIJřçĆźåIJĺèŁİåŁă èĂŇä¡ăçŽĎæIJŃåŔŃåIJĺå■řåžęçŽĎçŔ■åŁăç¡ŮåřŤïijŇéĆčäźĹäžŰåžŤèŕěåIJĺå¡ŞåIJřæŮűéŮť åĞăçĆźåŔĆåŁă

èğčåĘşæŰźæąĹ

åŕźåĞääźŐæĽĂæIJL'æűĽåŔŁåĹřæŮűåŇžçŽĎéŮőécŸïijŇä¡ăéČ¡åžŤèŕěä¡£çŤĺ pytz æĺąåİŮãĂĆè£ŹäÿłåŇĚæŔŘä¿ŻäžĘOlsonæŮűåŇžæŢřæ■őåžŞïijŇ åőČæŸŕæŮűåŇžä£ąæĄŕçŽĎäžŃåőđäÿŁçŽĎæăĞåĞĘïijŇåIJĺå¿Ĺåď'Žèŕ■èÍĂåŠŇæŞ■ä¡IJçşżçż§éĞŇéÍćéČ¡åŔ

pytz æĺąåİŮäÿĂäÿłäÿżèęĄçŤĺéĂŤæŸŕårĘ datetime åžŞåĹŻåžžçŽĎçőĂå■ŢæŮĕæIJ§årźèśąæIJñåIJřåŇŰãĂĆ æŕŤåęĆïijŇäÿŃéÍcåęĆä¡Ţèąĺçď'žäÿĂäÿłèŁİåŁăåŞěa

```
>>> 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)

```
>>> print(loc_d)
2012-12-21 09:30:00-06:00
>>>
```

äÿĂæŮęæŮěæIJ§èćńæIJňåIJřåŇŰäžĘïijŇ åőČåřśåŔŕäżěè¡ňæ■ćäÿžåĚűäżŰæŮűåŇžçŽĎæŮűéŮť äžĘąĂĆ äÿžäžĘå¿ŮåĹřçŔ■åŁăç¡ŮåřŤåŕźåžŤçŽĎæŮűéŮť ïijŇä¡ăåŔŕäżěè£Źæăůå ĄŽïijŽ

```
>>> # 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’ŞçőŮåIJĺæIJňåIJřåŇŰæŮěæIJ§äÿŁæL’ğèąŇèőąçőŮïijŇä¡ăéIJĂèęĄçL’źåĹńæşĺæĎŔåď’Ŕä ærŤåęĆïijŇåIJĺ2013åźť ïijŇç¿ŐåŻ¡æăĞåĞĘåď’ŔäżďæŮűæŮűéŮť åijĂåğŃäžŐæIJňåIJřæŮűéŮť 3æIJĹ13æŮè åęĆæđIJä¡ăæ■čåIJĺæL’ğèąŇæIJňåIJřèőąçőŮïijŇä¡ääijŽå¿ŮåĹřäÿĂäÿłéŤŹèŕŕãĂĆærŤåęĆïijŽ

```
>>> 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!
>>>
```

çżŞæđIJéŤŹèŕŕæŸŕåŽääÿžåőČåźűæšąæIJL’èĂČèŹŚåIJĺæIJňåIJřæŮűéŮť äÿ■æIJL’äÿĂářŔæŮűçŽĎèůşèů äÿžäžĘä£őæ■čè£ŹäÿłéŤŹèŕŕïijŇåŔŕäżěä¡£çŤĺæŮűåŇžářźèšą `normalize()` æŰźæşŢãĂĆærŤåęĆïijŽ

```
>>> from datetime import timedelta
>>> later = central.normalize(loc_d + timedelta(minutes=30))
>>> print(later)
2013-03-10 03:15:00-05:00
>>>
```

èőĺèőž

äÿžäžĘäÿ■èőľä¡ăèćńè£ŹäžŻäÿIJäÿIJåijĎçŽĎæŹŢåď’t’è¡ňåŘŚïijŇåď’ĎçŘĘæIJňåIJřåŇŰæŮěæIJ§çŽĎéĂ åźűçŤĺåőČæÌěæL’ğèąŇæL’ĂæIJL’çŽĎäÿ■éŮť’å■ŸåĆĺåŠŇæŞ■ä¡IJãĂĆærŤåęĆïijŽ

```
>>> 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¡ñæ■ćäÿžUTCïijÑä¡ăåřšäÿ■çŤĺåŐżæŃĚå£Čèů§åď Ŕäžď æŮűçŻÿåĚşçŽĎéŮőécŸäžĘãĂĆ
åŻăæ■ď ïijÑä¡ăåŔřäžěèů§äźŃåĽ■äÿĂæăůæŤ¿å£ČçŽĎæĽğèąŇåÿÿèğĄçŽĎæŮěæIJ§èőąçőŮãĂĆ
å¡Şä¡ăæČşářĘè¿ŞåĞžåŔŸäÿžæIJñåIJřæŮűéÚť çŽĎæŮűåĂŹïijÑä¡£çŤĺåŘĹéĂĆçŽĎæŮűåŃžåŐżè¡ñæ■ćäÿŃå

```
>>> later_utc = utc_d + timedelta(minutes=30)
>>> print(later_utc.astimezone(central))
2013-03-10 03:15:00-05:00
>>>
```

å¡ŞæűĽåŔŁåĹřæŮűåŃžæŞ■ä¡IJçŽĎæŮűåĂŹïijÑæIJĽäÿłéŮőécŸåřśæŸŕæĹŚäžňåęĆä¡Ţå¿ŮåĹřæŮűåŃ
ærŤåęĆïijÑåIJÍè£Źäÿŀäż¿Ńå■Řäÿ■ïijÑæĹŚäžňåęĆä¡Ţç§ééĄŞâĂIJAsia/KolkataâĂİåřśæŸŕå■řäžęářźäžŤçŽĎæ
äÿžäžĘæ§ěæĽ¿ïijÑåŔŕäžěä¡£çŤĺISO 3166åŻ¡åőűäžčçăĄä¡IJäÿžåĚşéŤőå■ŮåŐżæ§ěéŸĚå■ŮåĚÿ
`pytz.country_timezones` ãĂĆærŤåęĆïijŽ

```
>>> pytz.country_timezones['IN']
['Asia/Kolkata']
>>>
```

æşĺïijŽå¡Şä¡ăéŸĚèŕżåĹřè£ŹéĞÑçŽĎæŮűåĂŹïijÑæIJĽåŔŕèČ¡ `pytz`
æĺąåİŮåůšçżŔäÿ■åĘ■åžžèőőä¡£çŤĺäžĘïijÑåŻääÿžPEP431æŔŘåĞžäžĘæŻť åĚĹè£ŻçŽĎæŮűåŃžæŤŕæŇĄãĂĆ
ä¡ĘæŸŕè£ŹéĞÑèřĹåĹřçŽĎå¿Ĺåď ŽéŮőécŸè£ŸæŸŕæIJĽåŔĆèĂČäżůåĂijçŽĎ(ærŤåęĆä¡£çŤĺUTCæŮěæIJ§ç

6 çňňåŻŻçńäïijŽè£■äžčåŹĺäÿŐçŤ§æĹŘåŹĺ

è£■äžčæŸŕPythonæIJĂåijžåď ğçŽĎåŁ§èČ¡äźŃäÿĂãĂĆåĹİçIJŃèţůæİěïijÑä¡ăåŔŕèČ¡äijŽçőĂå■ŢçŽĎèőc
çĎűèĂŇïijÑçżIéİđäžĚäžĚåřśæŸŕåęĆæ■ď ïijÑè£ŸæIJĽå¿Ĺåď Žä¡ăåŔŕèČ¡äÿ■ç§ěéĄŞçŽĎïijÑ
ærŤåęĆåĹŻåžžä¡ăèĞłåůśçŽĎè£■äžčåŹĺåŕźèśąïijÑåIJĺitertoolsæĺąåİŮäÿ■ä¡£çŤĺæIJĽçŤĺçŽĎè£■äžčæĺąåijŔïijĺ
è£ŹäÿĂçńäçŽőçŽĎåřśæŸŕåŘŚä¡ăåśŢçď žèů§è£■äžčæIJĽåĚşçŽĎåŘĎçğ■åÿÿèğĄéŮőécŸãĂĆ

Contents:

6.1 4.1 æĽŃåŁĺéĄ■åŐĘè£■äžčåŹĺ

éŮőécŸ

ä¡ăæČşéĄ■åŐĘäÿĂäÿłåŔŕè£■äžčåŕźèśąäÿ■çŽĎæĽĂæIJĽåĚČçť ăïijÑä¡ĘæŸŕå■ť äÿ■æČşä¡£çŤĺforå¿łçČ

èğčåĘşæŰźæąĹ

äÿžäžĘæĽŃåŁĺçŽĎéĄ■åŐĘåŔŕè£■äžčåŕźèśąïijÑä¡£çŤĺ `next()`
åĞ¡æŢřåźűåIJĺäžčçăĄäÿ■æ■ŢèŐů `StopIteration` åijĆåÿÿãĂĆ
ærŤåęĆïijÑäÿŃéİćçŽĎä¿Ńå■ŘæĽŃåŁĺèŕżåŘŰäÿĂäÿłæŰĞäžűäÿ■çŽĎæĽĂæIJĽèąŇïijŽ

```
def manual_iter():
    with open('/etc/passwd') as f:
```

(continues on next page)

```
        try:
            while True:
                line = next(f)
                print(line, end='')
        except StopIteration:
            pass
```

éĂŽåÿÿæİěèőšïijŇ StopIteration çŤÍæİěæŇĞçđ'žè£■äžčçŽĎçżŞář¿ãĂĆ çĎűèĂŇïijŇåęĆæđIJä¡ăæĽ'ŃåŁÍä¡£çŤÍäÿŁéİćæijŤçđ'žçŽĎ next() åĞ¡æŢřçŽĎèŕÍïijŇä¡ăè£ŸåŘŕäžěéĂŽè£Ğè£ŤåŽđäÿĂäÿłæŇĞåőŽåĂijæİěæăĞèőřçżŞář¿ïijŇæŕŤåęĆ None ãĂĆ äÿŃéİćæŸŕçđ'žä¿ŃïijŽ

```
with open('/etc/passwd') as f:
    while True:
        line = next(f, None)
        if line is None:
            break
        print(line, end='')
```

èőÍèőž

åđ'ğåđ'ŽæŢřæČĚåĘţäÿŃïijŇæĹŚäžňäijŽä¡£çŤÍ for å¿łçŐŕèŕ■åŘěçŤÍæİěéĄ■åŐĘäÿĂäÿłåŘŕè£■äžčåŕźè
ä¡ĘæŸŕïijŇåĄűåřŤäź§éIJĂèęĄåŕźè£■äžčåĄŽæŻť'åŁăçš¿çąőçŽĎæŐğåĹűïijŇè£ŹæŮűåĂŹäžĘèğčåžŢåśĆè£■ä

äÿŃéİćçŽĎäžđ'äžŠçđ'žä¿ŃåŘŚæĹŚäžňæijŤçđ'žäžĘè£■äžčæIJ§éŮť'æĽ'ĂåŘŚçŤ§çŽĎå§žæIJňçżĘèŁĆïijŽ

```
>>> 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
>>>
```

æIJňçńăæŐěäÿŃæİěåĞăåřŘèŁĆäijŽæŻť'æůśåĚěçŽĎèőšèğčè£■äžčçŻýåĚşæŁĂæIJŕïijŇåL'■æŘŘæŸŕä¡ăă
æĽ'Ăäžěçąőä£Ïä¡ăåůšçżŔæŁŁè£Źçńăç ŽĎåĘĚåőźçL'ćçL'ćèőřåIJĺå£ČäÿăĂĆ

6.2 4.2 äžčçŘĘè£■äžč

éŮőécŸ

ä¡ăæđĎåžžäžĘäÿĂäÿłèĞłåőŽäźL’åőźåŹĺåŕźèśąïijŇéĞŇéİćåŇĚåŘńæIJL’åĹŮèąĺãĂĄåĚČçżĎæĹŰåĚűäžÚ
ä¡ăæČşçŻť’æŐěåIJĺä¡äçŽĎè£ŹäÿłæŰřåőźåŹĺåŕźèśąäÿŁæL’ğèąŇè£■äžčæŞ■ä¡IJãĂĆ

èğčåĘşæŰźæąĹ

åőđéŹĚäÿŁä¡ääŘłéIJĂèęĄåőŽäźL’äÿĂäÿł `__iter__()`
æŰźæşŢïijŇářĘè£■äžčæŞ■ä¡IJäžčçŘĘåĹřåőźåŹĺåĘĚéČĺçŽĎåŕźèśąäÿŁåŐżãĂĆæŕŤåęĆïijŽ

```
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£■äžčèŕůæśĆäijåéĂŠçżŹåĘĚé
`_children` åśđæĂğãĂĆ

èőĺèőž

PythonçŽĎè£■äžčåŹĺå■ŔèőőéIJĂèęĄ `__iter__()` æŰźæşŢè£ŤåŽđäÿĂäÿłåőđçŐřäžĘ
`__next__()` æŰźæşŢçŽĎè£■äžčåŹĺåŕźèśąãĂĆ åęĆæđIJä¡ääŘłæŸŕè£■äžčéĄ■åŐĘåĚűäžŰåőźåŹĺçŽĎåĘĚå

è£ŹéĞŇçŽĎ `iter()` åĞ¡æŢřçŽĎä¡£çŤĺçőĂåŇŰäžĘäžčçăĄijŇ
`iter(s)` åŘłæŸŕçőĂå■ŢçŽĎéĂŽè£ĞèřČçŤĺ `s.__iter__()`

æŰźæşŢæİěè£ŤåŻđåŕźåžŢçŽĎè£■äžčåŹĺåŕźèšąïijŇ åřśèů§ `len(s)` äijŽèřČçŤĺ `s.__len__()` åŐ§çŘĘæŸŕäÿĂæăůçŽĎãĂĆ

6.3 4.3 ä¡£çŤíçŤ§æĹŘåŹíåĹŻåžžæŰřçŽĎè£■äžčæĺąåijŔ

éŮőécŸ

ä¡ăæČşåőđçŐřäÿĂäÿłèĞłåőŽäźL'è£■äžčæĺąåijŔïijŇèů§æŹőéĂŽçŽĎåĘĚç¡őåĞ¡æŢřæŕŤåęĆ `range()`，`reversed()` äÿ■äÿĂæăůãĂĆ

èğčåĘşæŰźæąĹ

åęĆæđIJä¡ăæČşåőđçŐřäÿĂçğ■æŰřçŽĎè£■äžčæĺąåijŔïijŇä¡£çŤĺäÿĂäÿłçŤ§æĹŘåŹíåĞ¡æŢřæİěåőŽäźL'åäÿŃéIćæŸŕäÿĂäÿłçŤ§äžğæ§ŘäÿłèŇČåŻ'åĘĚæţőçĆźæŢřçŽĎçŤ§æĹŘåŹíijŽ

```
def frange(start, stop, increment):
    x = start
    while x < stop:
        yield x
        x += increment
```

äÿžäžĘä¡£çŤíè£ŹäÿłåĞ¡æŢřijŇ ä¡ăåŔŕäžěçŤĺforå¿łçŐŕè£■äžčåőČæĹŰèĂĚä¡£çŤĺåĚűäžŰæŐěåŔŮäÿĂä`sum()`，`list()` ç■L')ãĂĆçd'žä¿ŃåęĆäÿŇïijŽ

```
>>> 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■åŔěå■şåŔŕåřĘåĚűè¡ñæ■ćäÿžäÿĂäÿłçŤ§æĹŘåŹíãĂĆèů§æŹőéĂŽåĞ¡æŢřäÿ■åŘŇçŽĎæŸŕïijŇçŤ§æĹŘåŹíåŔłèČ¡çŤĺäžŐè£■äžčæŞ■ä¡IJãĂĆäÿŃéIćæŸŕäÿĂäÿłåőđéłŇïijŇåŘŚä¡ăåśŢçd'žè£ŹæăůçŽĎåĞ¡æŢřåžŢåśĆåůěä¡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¡æŢřäÿżèęAçĽźå¿ĄæŸŕåőČåŔłäijŽåŻđåžŤåIJĺè£■äžčäÿ■ä¡£çŤĺåĹřçŽĎ
next æŞ■ä¡IJãĂĆ äÿĂæŮęçŤ§æĹŘåŹĺåĞ¡æŢřè£ŤåŻđéĂĂåĞžïijŇè£■äžčçżĹæ■ćãĂĆæĹŚäžňåIJĺè£■äžčäÿ■é

6.4 4.4 åőđçŐřè£■äžčåŹĺå■Ŕèőő

éŮőécŸ

ä¡ăæČşæđĎåžžäÿĂäÿłèČ¡æŤŕæŇĄè£■äžčæŞ■ä¡IJçŽĎèĞłåőŽäźL’åŕźèśąïijŇåźűåÿŇæIJŻæL’¿åĹřäÿĂäÿł

èğčåĘşæŰźæąĹ

çŻőåL’■äÿžæ■ćïijŇåIJĺäÿĂäÿłåŕźèśąäÿŁåőđçŐřè£■äžčæIJĂçőĂå■ŢçŽĎæŰźåijŔæŸŕä¡£çŤĺäÿĂäÿłçŤ§æ
åIJĺ4.2åřŔèŁĆäÿ■ïijŇä¡£çŤĺNodeçśżæİěèąĺçđ’žæăŚå¡ćæŢřæ■őçżŞæđĎãĂĆä¡ăåŔŕèČ¡æČşåőđçŐřäÿĂäÿłäžěa

äÿŃéÌćæŸŕäžčçăĄçďžä¿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£Źæőţäžčçă Ąäÿ■ïijŇdepth_first() æŰźæşŢçőĂå■ŢçŻťˇèğĆãĂĆ åőČéęŰåĚĹè£ŤåŻđèĞłåůśæIJňèžńåźűè£■äžčæŕŔäÿĂäÿłå■ŘèŁĆçĆźåźű éĂŽè£ĞèřČçŤĺå■ŘèŁĆçĆźçŽĎ depth_first() æŰźæşŢ(ä¡£çŤÍ yield from èŕ■åŔě)è£ŤåŻđåŕźåžŤåĚČçťˇäãĂĆ

èőÍèőž

PythonçŽĎè£■äžčå■Ŕèőőèę Ąæ śĆäÿĂäÿł __iter__() æŰźæşŢè£ŤåŻđäÿĂäÿłçL’źæőŁçŽĎè£■äžčåŹĺåŕźèśąïijŇ è£Źäÿłè£■äžčåŹĺåŕźèśąåőđçŐřäžĘ __next__() æŰźæşŢåźűéĂŽè£Ğ StopIteration åijĆåÿÿæăĞèŕĘè£■äžčçŽĎåőŇæĹŘãĂĆ ä¡ĘæŸŕïijŇåőđçŐřè£ŹäžŻéĂŽåÿÿäijŽæŕŤè¿ČçźĄçŘŘãĂĆ äÿŃéÌćæĹŚäžňæijŤçď’žäÿŃè£Źçğ■æŰźåijŔïijŇå 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 çšžåŠÑäÿŁéİćä¡£çŤÍçŤ§æĹŘåŹÍçŽĎçL'ĹæIJñåůëä¡IJåŐ§çŘEçšžäijijïijÑ
ä¡ĘæŸŕåőČåĘŹèţůæİěå¿ĹçźAçŘŘïijÑåŽääÿžè£■äžčåŹĺå£Ěéąžå IJÍè£■äžčåď'ĎçŘEè£ĞçÍÑäÿ■çžť'æŁď'åď'ğéČ
åİęçŹ¡æİěëőšïijÑæšąäžžæĎ£æĎŔåĘŹè£ŹäźĹæŹęæűl'çŽĎäžčçăĄãĂĆåřEä¡ăçŽĎè£■äžčåŹĺåőŽäźL'äÿžäÿĂäÿ

6.5 4.5 åŔ■åŘŚè£■äžč

éŮőécŸ

ä¡ăæČşåŔ■æŰźåŘŚè£■äžčäÿĂäÿłåžŔåĹŮ

èğčåĘşæŰźæąĹ

ä¡£çŤÍåĘĚç¡őçŽĎ reversed() åĞ¡æŢřïijÑærŤåęĆïijŽ

```
>>> a = [1, 2, 3, 4]
>>> for x in reversed(a):
...     print(x)
...
4
3
2
1
```

åŔ■åŘŚè£■äžčäžĚäžĚå¡ŞåŕźèšąçŽĎåď'ğåřŔåŔŕécĎåĚÍçąőåőŽæĹŰèĂĚåŕźèšąåőđçŐřäžĘ
__reversed__() çŽĎçL'źæőŁæŰźæşŢæŮűæL'■èČ¡çŤ§æŢĹãĂĆ
åęĆæđIJäÿd'èĂĚéČ¡äÿ■çňęåŘĹïijÑéĆčä¡ăå£Ěéąžå ĚĹåřĘåŕźèšąè¡ňæ■ćäÿžäÿĂäÿłåĹŮèąÍæL'■èąÑïijÑærŤåęĆ

```
# Print a file backwards
f = open('somefile')
for line in reversed(list(f)):
    print(line, end='')
```

èęĄæşÍæĎŘçŽĎæŸŕåęĆæđIJåŔŕè£■äžčåŕźèšąåĚČçť'ăå¿Ĺåď'ŽçŽĎèŕÍïijÑåřĘåĚűécĎåĚÍè¡ňæ■ćäÿžäÿĂ

èőÍèőž

å¿Ĺåď'ŽçÍÑåžŘåŚŸåźűäÿ■ç§ěéĄŞåŔŕäžěéĂŽè£ĞåIJÍèĞłåőŽäźL'çšžäÿŁåőđçŐř
__reversed__() æŰźæşŢæİěåőđçŐřåŔ■åŘŚè£■äžčãĂĆærŤåęĆïijŽ

```
class Countdown:
    def __init__(self, start):
        self.start = start

    # Forward iterator
    def __iter__(self):
        n = self.start
```

(continues on next 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£■äžčåŹĺåŔŕäżěä¡£å¿Ůäžčçă Ąéİđåÿÿç ŽĎéńŸæŢĹïijŇ
åŻăäÿžåőČäÿ■åĘ■éIJĂèęĄåřĘæŢřæ■őåąńåĚĚåĹřäÿĂäÿłåĹŮèąĺäÿ■çĎűåŘŐåĘ■åŐżåŔ■åŘŚè£■äžčè£ŹäÿłåĹ

6.6 4.6 åÿęælJL’åď Ű éČÍçŁűæĂĄçŽĎçŤ§æĹŘåŹĺåĞ¡æŢř

éŮőécŸ

ä¡ăæČşåőŽäźL’äÿĂäÿłçŤ§æĹŘåŹĺåĞ¡æŢřïijÑä¡ĘæŸŕåőČäijŽèřČçŤÍæ§Řäÿłä¡ăæČşæŽť éIJšçżŹçŤÍæĹůä

èğčåĘşæŰźæąĹ

åęĆædIJä¡ăæČşèől’ä¡ăçŽĎçŤ§æĹŘåŹĺæŽť éIJšåď Ű éČÍçŁűæĂĄçżŹçŤÍæĹůïijŇ
åĹńå£ŸäžĘä¡ăåŔŕäżěçőĂå■ŢçŽĎářĘåőČåőđçŐřäÿžäÿĂäÿłçśżïijŇçĎűåŘŐæŁŁçŤ§æĹŘåŹĺåĞ¡æŢřæŤ¿åĹř
__iter__() æŰźæşŢäÿ■è£ĞåŐżãĂĆærŤåęĆïijŽ

```
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¡ååŔŕäżěåřĘåőČå¡ŞåĄŽæŸŕäÿĂäÿłæŹőéĂŽçŽĎçŤ§æĹŘåŹÍåĞ¡æŢřãĂĆ çĎűèĂŇïijŇçŤśäžŐåŔŕäżěåĹŻåżžäÿĂäÿłåőđä¿Ńåŕźèśąïij Ňäž Őæ Ÿŕä¡ååŔŕäżěèő£éŮőåĘĚéČÍåśđæĂğåĂijïijŇ æŕŤåęĆ `history` åśđæĂğæĹŰèĂĚæŸŕ `clear()` æŰźæşŢãĂĆäžčçăĄçđ'žä¿ŃåęĆäÿŇïijŽ

```
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

åĚşäžŐçŤ§æĹŘåŹÍïijŇå¿Ĺåőźæ ŸŞæŐL'è£ŻåĞ¡æŢřæŮăæL'Ăäÿ■èČ¡çŽĎéŹůéŸśãĂĆ åęĆæđIJçŤ§æĹŘåŹÍåĞ¡æŢřéIJĂèęĄèů§ä¡ăçŽĎçÍŃåžŔåĚűäżŰéČÍåĹĘæL'Şäžd'éĄŞçŽĎèŕİ(æŕŤåęĆæŽt'éIJšă åŔŕèČ¡äijŽåŕijèĞt'ä¡ăçŽĎäžčçăĄåijĆåÿÿçŽĎåd'■æÍĆãĂĆ åęĆæđIJæŸŕè£Źçğ■æČĚåĘţçŽĎèŕİïijŇåŔŕäżěèĂĆ åIJÍ `__iter__()` æŰźæşŢäÿ■åőŽäźL'ä¡ăçŽĎçŤ§æĹŘåŹÍäÿ■äijŽæŤźåŔŸä¡ääżä¡ŢçŽĎçőŮæşŢéĂżè¿ŚãĂĆ çŤśäžŐåőĆæŸŕçśżçŽĎäÿĂéČÍåĹĘïijŇæL'ĂäžěåĚĄèőÿä¡ääőŽäźL'åŘĎçğ■åśđæĂğåŠŇæŰźæşŢæÍěä¿ŽçŤÍæÍ

äÿĂäÿłéIJĂèęĄæşÍæĎŔçŽĎåŕŔåIJŕæŰźæŸŕïijŇåęĆæđIJä¡ääIJÍè£■äžčæŞ■ä¡IJæŮűäÿ■ä¡£çŤÍforå¿łçŐŕè `iter()` åĞ¡æŢřãĂĆæŕŤåęĆïijŽ

```
>>> 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£■äžčåŹÍåĹĞçL'Ğ

éŮőécŸ

ä¡äæČşå¿ŮåĹřäÿĂäÿłçŤśè£■äžčåŹÍçŤ§æĹŘçŽĎåĹĞçL'Ğåŕźèśąïij Ňä¡ĘæŸŕæăĞåĞĘåĹĞçL'ĞæŞ■ä¡IJåź

èğčåĘşæŰźæąĹ

åĞ¡æŢř `itertools.islice()` æ■čåě¡éĂĆçŤÍäžŐåIJÍè£■äžčåŹÍåŠŇçŤ§æĹŘåŹÍäÿŁåĄŽåĹĞçL'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£■äżčåŹÍåŠŇçŤ§æĹŘåŹÍäÿ■èČ¡ä¡£çŤÍæăĞåĞĘçŽĎåĹĞçL’ĞæŞ■ä¡IJïijŇåŻăäÿžåőČäżňçŽĎéŢ£åžęäžŇåĞ¡æŢř `islice()` è£ŤåŻđäÿĂäÿłåŘŕäżěçŤ§æĹŘæŇĞåőŽåĚČçt’ăçŽĎè£■äżčåŹÍïijŇåőČéĂŽè£ĞéĄ■åŐĘåçĎűåŘŐæL’■åijĂåğŃäÿĂäÿłäÿłçŽĎè£ŤåŻđåĚČçt’äïijŇåźűçŻt’åĹřåĹĞçL’ĞçżŞæİ§çt’ćåijŢä¡■ç¡õãĂĆ

è£ŹéĞŇèęĄçİĂéĞ■åijžèřČçŽĎäÿĂçĆźæŸý `islice()` äijŽæűĹèĂŮæŐL’äijăåĚěçŽĎè£■äżčåŹÍäÿ■çŽĎæŢřæ■õãĂĆ å£Ěéążè ĂČèŹŚåĹřè£■äżčåŹÍæŸýäÿ■åŘŕéĂĘçŽæL’Ăäżěåę ĆæđIJä¡äéIJĂèęĄäźŃåŘŐåĘ■æňąèő£éŮőè£Źäÿłè£■äżčåŹÍçŽĎèŕÏïijŇéĆčä¡ăåřšå¿ŮåĚĹåřĘåőČéĞ

6.8 4.8 èůşè£ĞåŘŕè£■äżčåŕźèśąçŽĎåijĂåğŃéČÍåĹĘ

éŮőécŸ

ä¡ăæČşéĄ■åŐĘäÿĂäÿłåŘŕè£■äżčåŕźèśąïijŇä¡ĘæŸŕåőČåijĂåğŃçŽĎæ§ŘäžŽåĚČçt’ää¡ăåźűäÿ■æĎ§åĚt’è

èğčåĘşæŰźæąĹ

itertools æĺąåİŮäÿ■æIJL'äÿĂäžŻåĞ¡æŢřåŔŕäżěåőŇæĹŘè£ŹäÿłäżżåŁąãĂĆ éęŰåĚĹäžŃçż■çŽĎæŸŕ itertools.dropwhile() åĞ¡æŢřãĂĆä¡£çŤĺæŮűïijŇä¡äçżŹåőČäijåéĂŠäÿĂäÿ åőČäijŽè£ŤåŻđäÿĂäÿłè£■äžčåŹĺåŕźèśąïijŇäÿćåijČåŐ§æIJL'åžŔåĹŮäÿ■çŻť'åĹřåĞ¡æŢřè£ŤåŻđFlaseäźŃåL'■ç

äÿžäžĘæijŤçd'žïijŇåĄĞåőŽä¡ăåIJĺèŕżåŔŰäÿĂäÿłåijĂåğŃéČĺåĹĘæŸŕåĞăèąŇæşĺéĞŁçŽĎæžŘæŰĞäżűã

```
>>> 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¡ăæČşèůşè£ĞåijĂåğŃéČĺåĹĘçŽĎæşĺéĞŁèąŇçŽĎèŕİïijŇåŔŕäżěè£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§žäžŐæăźæ■őæ§ŘäÿłæţŃèŕŢåĞ¡æŢřèůşè£ĞåijĂåğŃçŽĎåĚČçť'ăãĂĆ åęĆæđIJä¡ăåůšçżŔæŸŐçąőç§ěéĄŞäžĘèęĄèůşè£ĞçŽĎåĚČçť'ăçŽĎäÿłæŢřçŽĎèŕİïijŇéĆčäźĹåŔŕäżěä¡£çŤĺ itertools.islice() æİěäžčæŻ£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)

```
15
>>>
```

åIJÍè£Źäÿłä¿Ńå■Řäÿ■ïijŇ `islice()` åĞ¡æŢřæIJĂåŘŐéĆčäÿł `None` åŔĆæŢřæŇĞåőŽäžĘä¡ăèęĄèŐůåŔŰäżŐçňň3äÿłåĹřæIJĂåŘŐçŽĎæĽ'ĂæIJL'åĚČçţ'äïijŇ åęĆæđIJ `None` åŠŇ3çŽĎä¡■ç¡őåŕźěřČïijŇæĎŔæĂİåřśæŸŕäžĚäžĚèŐůåŔŰåĽ'■äÿĽ'äÿłåĚČçţ'ăæĄřæĄřçŽÿåŔ (è£Źäÿłèů§åĹĞçĽ'ĞçŽĎçŻÿåŔ■æŞ■ä¡IJ `[3:]` åŠŇ `[:3]` åŐ§çŘĘæŸŕäÿĂæăůçŽĎ)ãĂĆ

èőÍèőž

åĞ¡æŢř `dropwhile()` åŠŇ `islice()` åĚűåőđåřśæŸŕäÿd'äÿłåÿőåŁĽ'åĞ¡æŢřïijŇäÿžçŽĎåřśæŸŕéĄ£åÍ

```
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£ĞäÿĂäÿłåŔŕè£■äžčåŕźèśąçŽĎåijĂåğŃéČĺåĹĘèů§éĂŽåÿÿçŽĎè£Ğæżđ'æŸŕäÿ■åŘŃçŽĎãĂĆ æŕŤåęĆïijŇäÿŁè£řäžčçăĄçŽĎçňňäÿĂäÿłéČĺåĹĘåŔŕèČ¡äijŽè£ŹæăůéĞ■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£ŹæăůåĘŹçąőåőđåŔŕäžěèůşè£ĞåijĂåğŃéČĺåĹĘçŽĎæşĺéĞŁèąŇïijŇä¡ĘæŸŕåŘŇæăůäź§äijŽèůşè£ĞæŰ æ■ćåŘěèŕÍèőšïijŇæĹŚäžñçŽĎèğčåĘşæŰźæąĹæŸŕäžĚäžĚèůşè£ĞåijĂåğŃéČĺåĹĘæżąèűşæţŃèŕŢæÍąäžűçŽĎè

æIJĂåŘŐéIJĂèęĄçİĂéĞ■åijžèřČçŽĎäÿĂçĆźæŸŕïijŇæIJñèŁĆçŽĎæŰźæąĹéĂĆçŤĺäžŐæL'ĂæIJL'åŕŕè£ æŕŤåęĆçŤ§æĹŘåŹĺïijŇæŰĞäžűåŘŁåĚűçśžäijijçŽĎåŕźèśąãĂĆ

6.9 4.9 æŐŠåĹŮçżĎåŘĹçŽĎè£■äžč

éŮőécŸ

ä¡ăæČşè£■äžčéĄ■åŐĘäÿĂäÿłéŻĘåŘĹäÿ■åĚČçţ'ăçŽĎæL'ĂæIJL'åŕŕèČ¡çŽĎæŐŠåĹŮæĹŰçżĎåŘĹ

èğčåĘşæŰźæąĹ

itertoolsæĺąåİŮæŔŘä¿ŻäžĘäÿL'äÿłåĞ¡æŢřæİěèğčåĘşè£ŹçśżéŮőéćŸãĂĆ
åĚűäÿ■äÿĂäÿłæŸf `itertools.permutations()` ïijŇ
åőČæŐěåŘŮäÿĂäÿłéŻĘåŘĹåźűäžğçŢ§äÿĂäÿłåĚČçżĎåžŘåĹŮïijŇæŕŔäÿłåĚČçżĎçŢśéŻĘåŘĹäÿ■æL'ĂæIJL'
äź§åřśæŸŕèŕť'éĂŽè£ĞæL'ŞäźśéŻĘåŘĹäÿ■åĚČçť'ăæŐŠåĹŮéąžåžŘçŢ§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£åžęçŽĎæL'ĂæIJL'æŐŠåĹŮïijŇä¡ăåŔŕäžëäijăéĂŠäÿĂäÿłåŔŕéĂL'çŽ

```
>>> for p in permutations(items, 2):
...     print(p)
...
('a', 'b')
('a', 'c')
('b', 'a')
('b', 'c')
('c', 'a')
('c', 'b')
>>>
```

ä¡£çŢĺ `itertools.combinations()` åŔŕå¿ŮåĹřè¿ŞåĚěéŻĘåŘĹäÿ■åĚČçť'ăçŽĎæL'ĂæIJL'çŽĎçżĎ

```
>>> 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)

```
...
('a',)
('b',)
('c',)
>>>
```

åŕźäžŐ `combinations()` æÌèèőšïijŇåĚČçţ'äçŽĎéąžåžŔåůšçżŔäÿ■éĞ■èęĄäžĘãĂĆ äź§åřśæŸŕèŕţ'ïijŇçżĎåŘĹ `('a', 'b')` èů§ `('b', 'a')` åĚűåőđæŸŕäÿĂæăůçŽĎ(æIJĂçżĹåŔłäijŽè¿ŞåĞžåĚűäÿ■äÿĂäÿł)ãĂĆ

åIJÍèőąçőŮçżĎåŘĹçŽĎæŮűåĂŹïijŇäÿĂæŮęåĚČçţ'äèćńéĂĽ'åŔŰåřśäijŽäżŐåĂŹéĂĽ'äÿ■åĽ'ŤéŹđ'æŐĽ' èĂŇåĞ¡æŢř `itertools.combinations_with_replacement()` åĚĄèőÿåŘŇäÿĂäÿłåĚČçţ'äèćńéĂĽ'æŃĺ'åđ'Žæňąïij ŇæŕŤåęĆïijŽ

```
>>> 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£ŹäÿĂåřŔèŁĆæĹŚäżňåŘŚä¡ăåśŢçđ'žçŽĎäżĔäżĔæŸŕ `itertools` æÍąåİŮçŽĎäÿĂéČÍåĹĘåŁ§èČ¡ãĂĆ år¡çőąä¡ăäź§åŔŕäžèèĞłåůśæĽ'ŃåŁÍåőđçŐřæŐŠåĹŮçżĎåŘĹçőŮæşŢïijŇä å¡ŞæĹŚäżňçćřåĹřçIJŃäÿŁåŐżæIJĽ'äžŽåđ'■æİĆçŽĎè£■äžčéŮőécŸæŮűïijŇæIJĂåě¡åŔŕäžěåĚĹåŐżçIJŃçIJŃi åęĆæđIJè£ŹäÿłéŮőécŸå¿ĹæŹőéĄ■ïijŇéČčäźĹå¿ĹæIJĽ'åŔŕèČ¡äijŽåIJÍéĞŇéİćæĽ'¿åĹřèğčåĘşæŰźæąĹïijĄ

6.10 4.10 äžŔåĹŮäÿŁçţ'ćåijŢåĂijè£■äžč

éŮőécŸ

ä¡ăæČşåIJÍè£■äžčäÿĂäÿłäžŔåĹŮçŽĎåŘŇæŮűèů§èÿłæ■čåIJÍèćńåđ'ĎçŘĘçŽĎåĚČçţ'ăçţ'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§èąŇåŕůè¿ŞåĞž(èąŇåŕůäżŐ1åijĂåğŃ)ïijŇä¡ăåŔŕäžěäijäéĂŠäÿĂäÿłåijĂåğŃåŔĆæŢř

```
>>> 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¡ăéĄ■åŐĘæŰĞäžűæŮűæČşåIJÍéŤŹèŕŕæűĹæĄŕäÿ■ä¡£çŤÍèąŇåŕůåőŽä¡■æŮűåĂŹéÍ

```
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’ĂäżěïijŇåęĆæđIJä¡ăæČşåřĘäÿĂäÿłæŰĞäžűäÿ■åĞžçŐřçŽĎå■Ţèŕ■æŸăåřĎåĹřåőČåĞžçŐřçŽĎèąŇåŕůäÿ
enumerate() æİěåőŇæĹŘïijŽ

```
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¡ăåd’ĎçŘĘåőŇæŰĞäžűåŘŐæL’Şå■ř word_summary
ïijŇäijŽåŔŚçŐřåőČæŸŕäÿĂäÿłå■ŮåĚÿ(åĞĘçąőæİěèőšæŸŕäÿĂäÿł defaultdict
)ïijŇ åŕźäžŐæŕŔäÿłå■Ţèŕ■æIJL’äÿĂäÿł key ïijŇæŕŔäÿł key
åŕźåžŤçŽĎåĂijæŸŕäÿĂäÿłçŤśè£Źäÿłå■Ţèŕ■åĞžçŐřçŽĎèąŇåŕůçżĎæĹŘçŽĎåĹŮèąĺãĂĆ
åęĆæđIJæ§Řäÿłå■Ţèŕ■åIJÍäÿĂèąŇäÿ■åĞžçŐřè£Ğäÿd’æňąijŇéĆčäźĹè£ŹäÿłèąŇåŕůäź§äijŽåĞžçŐřäÿd’æňą

åŘŇæŮűäź§åŔŕäžĕä¡ĲäÿžæŰĞæĲňçŽĎäÿĂäÿłçőĂå■Ţçż§èőąãĂĆ

èőĺèőž

å¡Şä¡ăæČşéćĺåďŰåőŽäźĽäÿĂäÿłèőąæŢřåŘŸéĞŔçŽĎæŮűåĂŹïijŇä¡£çŤĺ enumerate() åĞ¡æŢřäijŽæŻť åŁăçőĂå■ŢãĂĆä¡ăåŔŕèČ¡äijŽåČŔäÿŃéİćè£Žæăůå EŹäżčçăĄïijŽ

```
lineno = 1
for line in f:
    # Process line
    ...
    lineno += 1
```

ä¡ĘæŸŕåęĆæđĲä¡£çŤĺ enumerate() åĞ¡æŢřæİĕäżčæŻ£åřśæŸ¿å¿ŮæŻť åŁăäijŸéŻĔäžĘïijŽ

```
for lineno, line in enumerate(f):
    # Process line
    ...
```

enumerate() åĞ¡æŢřè£ŤåŻđçŽĎæŸŕäÿĂäÿł enumerate åŕźèśąåőđä¿ŃïijŇ åőČæŸŕäÿĂäÿłè£■äżčåŹĺïijŇè£ŤåŻđè£đçż■çŽĎåŇĚåŘńäÿĂäÿłèőąæŢřåŠŇäÿĂäÿłåĂijçŽĎåĚČçżĎïijŇ åĚČçżĎäÿ■çŽĎåĂijéĂŽè£ĞåIJĺäijăåĚěåžŔåĹŮäÿŁèřČçŤĺ next() è£ŤåŻđãĂĆ

è£ŸæIJĽäÿĂçĆźåŔŕèČ¡åźűäÿ■å¿ĹéĞ■èęAïijŇä¡ĘæŸŕäź§åĂijå¿ŮæşĺæĎŔïijŇ æIJĽæŮűåĂŹå¡Şä¡ääIJĺäÿĂäÿłåůšçżŔèğčåŐŃåŘŐçŽĎåĚČçżĎåžŔåĹŮäÿŁä¡£çŤĺ enumerate() åĞ¡æŢřæŮűå¿Låőźæ ŸŞèřČåĚěéŽůéŸśãĂĆ ä¡ăå¿ŮåČŔäÿŃéİćæ■čçąőçŽĎæŰźåijŔè£Žæăůå EŹïijŽ

```
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

éŮőéćŸ

ä¡ăæČşåŘŇæŮűè£■äżčåď ŽäÿłåžŔåĹŮïijŇæŕŔæňąåĹĘåĹńäżŐäÿĂäÿłåžŔåĹŮäÿ■åŔŰäÿĂäÿłåĚČçť ăãĂĆ

èğčåĘşæŰźæąĹ

äÿžäžĘåŘŇæŮűè£■äżčåď ŽäÿłåžŔåĹŮïijŇä¡£çŤĺ zip() åĞ¡æŢřãĂĆæŕŤåęĆïijŽ

```
>>> 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£■äžčåŹĺïijŇåĚűäÿ■xæİěèĞłaïijŇyæİěèĞłbãĂĆ äÿĂæŮęåĚűäÿ■æ§Řäÿłåžŕ̊åĹŮåĹřåžŢçżŞår¿ïijŇè£■äž åŻăæ■ď'è£■äžčéŢ£åžęèů§åŔĆæŢřäÿ■æIJĂç§■åžŕ̊åĹŮéŢ£åžęäÿĂèĞť'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£ŸåŔŕäžěä¡£çŤÍ itertools.zip_longest() åĞ¡æŢřæİěäžčæŻ£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¡ăæČşæĹŘåŕžåď ĎçŘEæŢřæ■őçŽĎæŮűåĂŹ zip() åĞ¡æŢřæŸŕå¿ĹæIJL'çŤÍçŽĎãĂĆ æŕŤåęĆïijŇåĄĞèő¿ä¡ăåď't'åĹŮèąĺåŠŇäÿĂäÿłåĂijåĹŮèąĺïijŇåřśåČŔäÿŃéİ

```
headers = ['name', 'shares', 'price']
values = ['ACME', 100, 490.1]
```

ä¡£çŤÍzip()åŔŕäžěèől'ä¡ăářĘåőČäžňæL'ŞåŇĚåźűçŤ§æĹŘäÿĂäÿłå■ŮåĚÿïijŽ

```
s = dict(zip(headers,values))
```

æĹŰèĂĚä¡ăäź§åŔŕäžěåČŔäÿŃéİćè£ŹæăůäžğçŤ§è¿ŞåĞžïijŽ

```
for name, val in zip(headers, values):
    print(name, '=', val)
```

èŹ¡çĎűäÿ■åÿÿèğĄïijŇä¡ĘæŸŕ zip() åŔŕäžěæŐěåŔŮåď'ŽäžŐäÿd'äÿłçŽĎåžŔåĹŮçŽĎåŔĆæŢřãĂĆ è£ŹæŮűåĂŹæL'ĂçŤ§æĹŘçŽĎçżŞæđIJåĚČçżĎäÿ■åĚČçt'ääÿłæŢřèů§è¿ŞåĚěåžŔåĹŮäÿłæŢřäÿĂæăůãĂĆæŕŤ

```
>>> 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')
>>>
```

æIJĂåŘŐåijžèřČäÿĂçĆźåřśæŸŕïijŇ zip() äijŽåĹŻåžžäÿĂäÿłè£■äžčåŹĺæÍěä¡IJäÿžçżŞæđIJè£ŤåŻđãĂĆ åęĆæđIJä¡ăéIJĂèęĄåřĘçżŞåŕźçŽĎåĂijå■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■åŘŇéŻĘåŘĹäÿŁåĚČçt'ăçŽĎè£■äžč

éŮőécŸ

ä¡ăæČşåIJĺåď'ŽäÿłåŕźèśąæL'ğèąŇçŻÿåŘŇçŽĎæŞ■ä¡IJïijŇä¡ĘæŸŕè£ŹäžŻåŕźèśąåIJĺäÿ■åŘŇçŽĎåőźåŹĺä

èğčåĘşæŰźæąĹ

itertools.chain() æŰźæşŢåŔŕäžěçŤĺæİěçőĂåŇŰè£ŹäÿłäżžåŁąãĂĆ åőČæŐěåŔŮäÿĂäÿłåŔŕè£■äžčåŕźèśąåĹŮèąĺä¡IJäÿžè¿ŞåĚčijŇåźűè£ŤåŻđäÿĂäÿłè£■äžčåŹĺïijŇæIJL'æŢĹçŽĎ äÿžäžĘæijŤçd'žæÿĚæěŽïijŇèĂČèŹŚäÿŃéİćè£ŹäÿłäżŃå■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¡ăæČşåŕźäÿ■åŘŇçŽĎéŻĘåŘĹäÿ■æL'ĂæIJL'åĚČçt

```
# 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■èğčåĘşæŰźæąĹèęĄæŕŤåČŔäÿŃéİćè£Źæăůä¡£çŤĺäÿd'äÿłå■ŢçŃňçŽĎå¿łçŐŕæŻt'åŁääijŸéŻĚïijŇ

```
for item in active_items:
    # Process item
    ...

for item in inactive_items:
    # Process item
    ...
```

èőÍèőž

itertools.chain() æŐěåŔŮäÿĂäÿłæĹŰåd'Žäÿłåŕŕè£■äžčåŕźèśąä¡IJäÿžè¿ŞåĚěåŔĆæŢřãĂĆ çĎűåŘŐåĹŻåžžäÿĂäÿłè£■äžčåŹĺïijŇä¿İæňąè£đçż■çŽĎè£ŤåŻđæŕŔäÿłåŕŕè£■äžčåŕźèśąäÿ■çŽĎåĚČçt'äãĂĆ è£Źçğ■æŰźåijŔèęĄæŕŤåĚĹåřĘåžŘåĹŮåŘĹåźűåĘ■è£■äžčèęĄénŸæŢĹçŽĎåd'ŽãĂĆæŕŤåęĆïijŽ

```
# Inefficent
for x in a + b:
```

(continues on next page)

```
    ...

# Better
for x in chain(a, b):
    ...
```

çňňäÿĂçğ■æŰźæąĹäÿ■ïijŇ a + b æŞ■ä¡IJäijŽåĹŻåžžäÿĂäÿłåĚĺæŰřçŽĎåžŔåĹŮåźűèęĄæśĆaåŠŇbçŽchian() äÿ■äijŽæIJL'è£ŹäÿĂæ■ěïijŇæL'Ăäžěåę ĆæđIJè¿ŞåĚěåžŔåĹŮéİđåÿÿåď'ğçŽĎæŮűåĂŹäijŽå¿ĹçIJÅåźűäÿŤå¡ŞåŔŕè£■äžčåŕźèśąçśžåđŃäÿ■äÿĂæăůçŽĎæŮűåĂŹ chain() åŘŇæăůåŔŕäžěå¿Ĺåě¡çŽĎåůěä¡IJãĂĆ

6.13 4.13 åĹŻåžžæŢřæ■őåď'ĎçŘĘçőąéĄŞ

éŮőécŸ

ä¡ăæČşäžěæŢřæ■őçőąéĄŞ(çśžäijijUnixçőąéĄŞ)çŽĎæŰźåijŔè£■äžčåď'ĎçŘĘæŢřæ■õãĂĆæŕŤåęĆïijŇä¡ăæIJL'äÿłåď'ğéĞŔçŽĎæŢřæ■őéIJĂèęĄåď'ĎçŘĘïijŇä¡ĘæŸŕäÿ■èČ¡åřĘåőČäžñäÿĂæňąæĂğæŤ¿

èğčåĘşæŰźæąĹ

çŤ§æĹŘåŹĺåĞ¡æŢřæŸŕäÿĂäÿłåőđçŐřçőąéĄŞæIJžåĹűçŽĎåě¡åŁđæşŢãĂĆäÿžäžĘæijŤçď'žïijŇåĄĞåőŽä¡ăèęĄåď'ĎçŘĘäÿĂäÿłéİđåÿÿåď'ğçŽĎæŮěå£ŮæŰĞäžűçŻőå¡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¿æŕŔäÿłæŮěå£ŮæŰĞäžűåŇĚåŘńè£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 -
...
```

äÿžäžĘåď'ĎçŘĘè£ŹäžŽæŰĞäžűïijŇä¡ååŔŕäžěåőŽäźL'äÿĂäÿłçŤšåď'Žäÿłæl'ğèąŇçL'źåőŽäžžåŁąçŃñçńŇ

```
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¡ăåŔŕäžěå¿Ĺåőźæ ŸŞçŽĎåřĘè£ŹäžŻåĞ¡æŢřè£đèţůæİěåĹŻåžžäÿĂäÿłåď ĎçŘĘçőąéĄŞãĂĆ
æŕŤåęĆïijÑäÿžäžĘæ§ěæĽ¿åŇĚåŘńå■Ţèŕ■pythonçŽĎæĽĂæIJĽæŮěå£ŮèąŇïijÑä¡ăåŔŕäžěè£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)
```

åęĆædIJåřĘæİěçŽĎæŮűåĂŹä¡ăæČşæL'l'åśŢçőąéĄŞïijÑä¡ăçŤŽèĞşåŔŕäžěåIJÍçŤ§æĹŘåŹÍèąĺè¿¿åijŔäÿ■æŕŤåęĆïijÑäÿÑéÍćè£ŹäÿłçL'ĹæIJñèőąçőŮåĞžäijăè¿ŞçŽĎå■ÙèŁĆæŢřåźűèőąçőŮåĚűæĂżåŠÑãĂĆ

```
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

äžěçőąéĄŞæŰźåijŔåď'ĎçŘĘæŢřæ■őåŔŕäžěçŤÍæİěèğčåĘşåŘĎçśżåĚűäžŰéŮőéćŸïijÑåŇĚæŃňèğčædŘïi

äÿžäžĘçŘĘèğčäÿŁè£řäžčçăĄïijÑéĞ■çĆźæŸŕèęĄæŸŐçŹ¡ `yield` èŕ■åŘëä¡IJäÿžæŢřæ■őçŽĎçŤ§äžğèĂĚèĂŇ `for` å¿łçŐŕèŕ■åŘëä¡IJäÿžæŢřæ■őçŽĎæűĹèť'źèĂĚãĂĆ å¡Şè£ŹäžŻçŤ§æĹŘåŹÍèćńè£đåIJÍäÿĂèţůåŘŐïijÑærŔäÿł `yield` äijŽåřĘäÿĂäÿłå■ŢçŃňçŽĎæŢřæ■őåĚČçť'ääijăéĂŠçžŹè£■äžčåď'ĎçŘĘçőąéĄŞçŽĎäÿŃäÿĂéŸűæőţãĂĆ åIJÍä¿Ńå■ŘæIJĂåŘŐéČÍåĹĘïijÑ `sum()` åĞ¡æŢřæŸŕæIJĂçżĹçŽĎçÍŃåžŔél'śåŁÍèĂĚïijÑærŔæňąäžŐçŤ§æĹŘ

è£Źçğ■æŰźåijŔäÿĂäÿłéİđåÿÿåě¡çŽĎçL'źçĆźæŸŕærŔäÿłçŤ§æĹŘåŹÍåĞ¡æŢřå¿ĹåřŔåźűäÿŤéČ¡æŸŕçŃňç å¿Ĺåď'ŽæŮűåĂŹïijÑè£ŹäžŻåĞ¡æŢřåęĆædIJærŤè¿ČéĂŽçŤÍçŽĎèŕÍåŔŕäžěåIJÍåĚűäžŰåIJžæŹŕéĞ■åď'■ä¡£çŤÍ åźűäÿŤæIJĂçżĹåřĘè£ŹäžŻçżĎäžűçżĎåŘÍèţůæİěçŽĎäžčçăĄçIJŃäÿŁåŐżéİđåÿÿçőĂå■ŢïijÑäź§å¿Ĺåőźæ'ŸŞç

ä¡£çŤÍè£Źçğ■æŰźåijŔçŽĎåĘĚå■ŸæŢĹçŐĞäź§äÿ■å¿Ůäÿ■æŘŘãĂĆäÿŁè£řäžčçăĄåą■şä¿£æŸŕåIJÍäÿĂäÿ äžŃåőđäÿŁïijÑçŤśäžŐä¡£çŤÍäžĘè£■äžčæŰźåijŔåď'ĎçŘĘïijÑäžčçăĄè£ŘèąŃè£ĞçÍŃäÿ■åŔłéIJĂèęĄå¿ĹåřŔå

åIJÍèřČçŤÍ `gen_concatenate()` åĞ¡æŢřçŽĎæŮűåĂŹä¡ääŔŕèČ¡äijŽæIJL'äžŻäÿ■åď'łæŸŐçŹ¡ãĂĆ è£ŹäÿłåĞ¡æŢřçŽĎçŽőçŽĎæŸŕåřĘè¿ŞåĚěåžŘåĹŮæÑijæŐěæĹŘäÿĂäÿłå¿ĹéŢ£çŽĎèąŃåžŘåĹŮãĂĆ `itertools.chain()` åĞ¡æŢřåŘŇæăůæIJL'ćśżäijijçŽĎåŁ§èČ¡ïijÑä¡ĘæŸŕåőČéIJĂèęĄåřĘæL'ĂæIJL'åŔŕè åIJÍäÿŁéİćè£ŹäÿłäżÑå■ŘäÿïijÑä¡ääŔŕèČ¡äijŽåĘŹçśżäijijè£ŹæăůçŽĎèŕ■åŘě `lines = itertools.chain(*files)` ïijÑ è£ŹåřĘåŕijèĞť `gen_opener()` çŤ§æĹŘåŹÍèćńæŔŘåL'■åĚÍéČÍæűĹèť'źæŐL'ãĂĆ ä¡ĘçŤśäžŐ `gen_opener()` çŤ§æĹŘåŹÍæŕŘæňąçŤ§æĹŘäÿĂäÿłæL'ŞåijÄè£ĞçŽĎæŰĞäžűïijÑ ç■L'åĹřäÿŃäÿĂäÿłè£■äžčæ■ééłď'æŮűæŰĞäžűåřśåĚşéŮ■äžĘïijÑåŽăæ■d' `chain()` åIJÍè£ŹéĞŃäÿ■èČ¡è£ŹæăůäĄ¡£çŤÍãĂĆ äÿŁéİćçŽĎæŰźæąĹåŔŕäžěéĄ£åĚ■è£ŹçğæČĚåĘţãĂĆ

`gen_concatenate()` åĞ¡æŢřäÿ■åĞžçŐŕè£Ğ `yield from` èŕ■åŘëïijÑåőČåřĘ `yield` æŞ■ä¡IJäžčçŘĘåĹřçĹűçŤ§æĹŘåŹÍäÿŁåŐžãĂĆ èŕ■åŘě `yield from it` çőĂå■ŢçŽĎè£ŤåŽđçŤ§æĹŘåŹÍ `it` æL'Ääžğç Ť§çŽĎæL'ĂæIJL'åÄijãĂĆ åĚşäžŐè£ŹäÿłæĹŚäžňåIJÍ4.14åřŔèŁĆäijŽæIJL'æŻť'è£ŽäÿĂæ■ěçŽĎæŔŔè£řãĂĆ

æIJĂåŘŐè£ŸæIJL'äÿĂçĆźéIJĂèęĄæşĺæĎŔçŽĎæŸŕïijŇçőąéĄŞæŰźåijŔåźűäÿ■æŸŕäÿĞèČ¡çŽĎãĂĆ
æIJL'æŮűåĂŹä¡ăæČşçńŇå■şåď'ĎçŘĘæL'ĂæIJL'æŢřæ■őãĂĆ çĎűèĂŇïijŇå■şä¿£æŸŕè£ŹçğæČĚåĘţïijŇä¡£ç

David Beazley åIJĺäžŰçŽĎ Generator Tricks for Systems Programmers
æŢŹçÍŃäÿ■åŕźäžŐè£Źçğ■æŁĂæIJŕæIJL'éİđåÿÿæůśåĚěçŽĎèőšèğčãĂĆåŔŕäžěåŔĆèĂČè£ŹäÿłæŢŹçÍŃèŐůåŔ

6.14 4.14 åśŢåijĂåţŇåěŮçŽĎåžŔåĹŮ

éŮőécŸ

ä¡ăæČşåřĘäÿĂäÿłåď'ŽåśĆåţŇåěŮçŽĎåžŔåĹŮåśŢåijĂæĹŘäÿĂäÿłå■ŢåśĆåĹŮèąĺ

èğčåĘşæŰźæąĹ

åŔŕäžěåĘŹäÿĂäÿłåŇĚåŘń `yield from` èŕ■åŔěçŽĎéĂŠå¡ŠçŤ§æĹŘåŹĺæİěè¡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)
```

åIJĺäÿŁéİćäžčçăĄäÿ■ïijŇ `isinstance(x, Iterable)` æčĂæ§ěæ§ŘäÿłåĚČçť'ăæŸŕåŘęæŸŕåŔŕè£■äžčçŽĎãĂĆ åęĆæđIJæŸŕçŽĎèŕİïijŇ `yield from` åřśäijŽè£ŤåŽđæL'ĂæIJL'å■Řä¿ŃçÍŃçŽĎåĂijãĂĆæIJĂçżĹè£ŤåŽđçżŞæđIJåřśæŸŕäÿĂäÿłæšąæIJL'åţŇå

éćİåď'ŰçŽĎåŔĆæŢř `ignore_types` åŠŇæčĂæţŃèŕ■åŘě `isinstance(x, ignore_types)` çŤĺæİěåřĘå■Ůçňęäÿšå$ŇŇå■ŮèŁĆæŐŠéŹď'åIJĺåŔŕè£■äžčåŕźèśąåď'ŰïijŇéŸšæ■ćåřĘåőČä
è£ŹæăůçŽĎèŕİå■ŮçňęäÿšæŢřçżĎåřśèČ¡æIJĂçźĹè£ŤåŽđæĹŠäžňæL'ĂæIJ§æIJŻçŽĎçżŞæđIJäžĘãĂĆæŕŤåęĆ

```
>>> items = ['Dave', 'Paula', ['Thomas', 'Lewis']]
>>> for x in flatten(items):
...     print(x)
...
Dave
Paula
Thomas
Lewis
>>>
```

èőÍèőž

èŕ■åŘě `yield from` åIJÍä¡ăæČşåIJÍçŤ§æĹŘåŹÍäÿ■èřČçŤÍåĚűäžŰçŤ§æĹŘåŹÍä¡IJäÿžå■Řä¿ŃçÍŃçŽĎæåęĆæđIJä¡ääÿ■ä¡£çŤÍåőČçŽĎèŕÍïijŇéĆčäźĹåřśå£Ěéąžå ĘŹéćÍåď 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§èğĽæŻť åě¡ïijŇåźűäÿŤäź§ä¡£å¿ŮäżčçăĄæŻť çőĂæť Ąæÿ ĚçĹ¡ãĂĆ

äźŃåĽ■æŔŘåĹřçŽĎåŕźäžŐå■Ůçňęäÿšå ŠŇå■ŮèŁĆçŽĎéćÍåď ŰæčĂæ§ěæŸŕäÿžäžĘéŸšæ■ćåřĘåőČäżňå åęĆæđIJè£ŸæIJĽåĚűäžŰä¡ääÿ■æČşåśŢåijĂçŽĎçśżåđŃïijŇä£őæŤźåŔĆæŢř `ignore_types` å■şåŔŕãĂĆ

æIJĂåŘŐèęĄæşÍæĎŔçŽĎäÿĂçĆźæŸŕïijŇ `yield from` åIJÍæűĽåŔŁåĹřå§žäžŐå■ŔçÍŃåŠŇçŤ§æĹŘåŹÍçŽĎåźűåŔŞçijŰçÍŃäÿ■æĽőæijŤçİĂæŻť åŁäéĞ■èęĄçŽĎèğŠè åŔŕäžěåŔĆèĂČ12.12åřŔèŁĆæ§ěçIJŃåŔęåď ŰäÿĂäÿłä¿Ńå■ŘãĂĆ

6.15 4.15 éąžåžŔè£■äžčåŘĹåźűåŘŐçŽĎæŐŠåžŔè£■äžčårźèšą

éŮőécŸ

ä¡ăæIJĽäÿĂçşżåĹŮæŐŠåžŔåžŔåĹŮïijŇæČşåřĘåőČäżňåŘĹåźűåŘŐå¿ŮåĹřäÿĂäÿłæŐŠåžŔåžŔåĹŮåźűå

èğčåĘşæŰźæąĹ

`heapq.merge()` åĞ¡æŢřåŔŕäžěåÿőä¡äèğčåĘşè£ŹäÿłéŮőécŸãĂĆæŕŤåęĆïijŽ

```
>>> 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)

```
10
11
```

èőÍèőž

heapq.merge åŔŕè£■äžčçĽźæĂğæĎŔåŞşçİĂåőČäÿ■äijŽçńŃéľ ňèŕżåŔŰæĽĂæIJĽåžŔåĹŮãĂĆ è£Źåřśæ ĎŔåŞşçİĂä¡äåŔŕäžěåIJĺéİđåÿÿéŢ£çŽĎåžŔåĹŮäÿ■ä¡£çŤĺåőČïijŇèĂŇäÿ■äijŽæIJĽåď łåďğçŽĎåijĂé æŕŤåęĆïijŇäÿŃéİćæŸŕäÿĂäÿłä¿Ńå■ŘæİĕæijŤçď žåęĆä¡ŢåŘĹåźűäÿďäÿłæŐŠåžŔæŰĞäżűïijŽ

```
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)
```

æIJĽäÿĂçĆźèęĄåijžèřČçŽĎæŸŕ heapq.merge() éIJĂèęĄæĽĂæIJĽè¿ŞåĚěåžŔåĹŮå£Ěéążæ Ÿŕæ ŐŠ çĽźåĹńçŽĎïijŇåőČåźűäÿ■äijŽéćĎåĚĹèŕżåŔŰæĽĂæIJĽæŢřæ■őåĹřåăĘæăĹäÿ■æĹŰèĂĚéćĎåĚĹæŐŠåžŔïij åőČäżĚäżĚæŸŕæčĂæ§ěæĽĂæIJĽåžŔåĹŮçŽĎåijĂåğŃéČĺåĹĘåźűè£ŤåŽđæIJĂåřŔçŽĎéĆčäÿłïijŇè£Źäÿłè£Ç

6.16 4.16 è£■äžčåŹĺäžčæŻ£whileæŮăéŹŘå¿łçŐŕ

éŮőéćŸ

ä¡ääIJĺäžčçăĄäÿ■ä¡£çŤĺ while å¿łçŐŕæİěè£■äžčåď ĎçŘĘæŢřæ■őïijŇåŻăäÿžåőČéIJĂèęĄèřČçŤĺæ§Řäÿ èČ¡äÿ■èČ¡çŤĺè£■äžčåŹĺæİěéĞ■åĘŹè£Źäÿłå¿łçŐŕåŚćïij§

èğčåĘşæŰźæąĹ

äÿĂäÿłåÿÿèğĄçŽĎIOæŞ■ä¡IJçĺŃåžŔåŔŕèČ¡äijŽæČşäÿŃéİćè£Źæăůïij

```
CHUNKSIZE = 8192

def reader(s):
    while True:
        data = s.recv(CHUNKSIZE)
        if data == b'':
            break
        process_data(data)
```

è£Źçğ■äžčçăĄéĂŽåÿÿåŔŕäžěä¡£çŤĺ iter() æİěäžčæŻ£ïijŇåęĆäÿŃæĽĂçď žïijŽ

```
def reader2(s):
    for chunk in iter(lambda: s.recv(CHUNKSIZE), b''):
```

(continues on next page)

```
        pass
        # process_data(data)
```

åęĆæđIJä¡ăæĂĂçŰŚåőČåĹřåžŢèČ¡äÿ■èČ¡æ■čåÿÿåůěä¡IJïijŇåŔŕäžěèŕŢéłŇäÿŃäÿĂäÿłçőĂå■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'źæĂgæŸŕåőČæŐěåŔŮäÿĂäÿłåŔŕéĂL'çŽĎ `callable` åŕźèśąåŠŇäÿĂäÿłæăĞèőř(çżŞář¿)åĂijä¡IJäÿžè¿ŞåĚěåŔĆæŢřãĂĆ å¡Şäżěè£Źçğ■æŰźåijŔä¡£çŤÍçŽĎæŮűåĂŹïijŇåőČäijŽåĹŽåžžäÿĂäÿłè£■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¿ŃæİěèőšïijŇåęĆæđIJä¡ăæČşäżŐåěŮæŐěå■ŮæĹŰæŰĞäžűäÿ■äžěæŢřæ■őåİŮçŽĎæŰźåijŔèŕżåŘŰæŢřæ `read()` æĹŰ `recv()` ïijŇ åźűåIJÍåŘŐéİćçť'ğèů§äÿĂäÿłæŰĞäžűçżŞår¿æţŃèřŢæİěåĘşåőŽæŸŕåŘęçżĹæ■ćãĂ `iter()` èřČçŤÍåřśåŔŕäžěåřĘäÿď'èĂĚçżŞåŘĹètůæİěäžĘãĂĆ åĚűäÿ■ `lambda` åĞ¡æŢřåŔĆæŢřæŸŕäÿžäžĘåĹŽåžžäÿĂäÿłæŮăåŔĆçŽĎ `callable` åŕźèśąïijŇåźűäÿž `recv` æĹŰ `read()` æŰźæşŢæŔŘä¿ŻäžĘ `size` åŔĆæŢřãĂĆ

7 çňňäžŤçńäïijŽæŰĞäžűäÿŐlO

æL'ĂæIJL'çÍŃåžŔéČ¡èęĄåď'ĎçŘĘè¿ŞåĚěåŠŇè¿ŞåĞžãĂĆ è£ŹäÿĂçńäářĘæűţçŻŰåď'ĎçŘĘäÿ■åŘŇçśžåđŃçŽĎæŰĞäžűïijŇåŇĚæŃňæŰĞæIJňåŠŇäžŇè£ŻåĹűæŰĞäžűï åŕźæŰĞäžűåŘ■åŠŇçŽőå¡ŢçŽĎæŞ■ä¡IJäź§äijŽæűL'åŔŁåĹřãĂĆ

Contents:

7.1 5.1 èŕžåĘŹæŰĞælJňæŢřæ■ő

éŮőécŸ

ä¡ăéIJĂèęĄèŕžåĘŹåŘĎçğ■äÿ■åŘŇçijŰçăĄçŽĎæŰĞæIJňæŢřæ■őïijŇæŕŤåęĆASCIIïijŇUTF-8æĹŰUTF-16çijŰçăĄç■L'ãĂĆ

èğčåĘşæŰźæąĹ

ä¡£çŤĺåÿęæIJL' rt æĺąåijŔçŽĎ open() åĞ¡æŢřèŕžåŔŰæŰĞæIJňæŰĞäżűãĂĆåęĆäÿŇæL'Ăçd'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 æĺąåijŔçŽĎ open() åĞ¡æŢřïijŇ åęĆæđIJäźŇåL'■æŰĞäżűåĘĚåőźå■ŸåIJĺåĹŹæÿĚéŹd'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■æůżåŁăåĘĚåőźijŇä¡£çŤĺæĺąåijŔäÿž at çŽĎ open() åĞ¡æŢřãĂĆ

æŰĞäżűçŽĎèŕžåĘŹæŞ■ä¡IJéżŸèŏd'ä¡£çŤĺçşżçż§çijŰçăĄïijŇåŔŕäžěéĂŽè£ĞèřČçŤĺ sys.getdefaultencoding() æİěå¿ŮåĹřãĂĆ åIJĺåd'ğåd'ŽæŢřæIJžåŹĺäÿŁéİćéČ¡æŸŕutf-8çijŰçăĄãĂĆåęĆæđIJä¡ăåůšçżŔç§ěéĄŞä¡ăèęĄèŕžåĘŹçŽĎæŰĞæIJňæŸŕåĚűäżŰçijŰçăĄæŰźåijŔïijŇéĆčäźĹåŔŕäžěéĂŽè£ĞäijăéĂŠäÿĂäÿłåŔŕéĂL'çŽĎ encoding åŔĆæŢřçżŹopen()åĞ¡æŢřãĂĆåęĆäÿŇæL'Ăçd'žïijŽ

```
with open('somefile.txt', 'rt', encoding='latin-1') as f:
    ...
```

PythonæŤŕæŇĄéÍđåÿÿåď˙ŽçŽĎæŰĞæIJňçijŰçăĄãĂĆåĞääÿłåÿÿèğĄçŽĎçijŰçăĄæŸŕascii, latin-1, utf-8åŠŇutf-16ãĂĆ åIJĺwebåžŤçŤĺçÍŃåžŔäÿ■éĂŽåÿÿéČ¡ä¡£çŤĺçŽĎæŸŕUTF-8ãĂĆ asciiåŕźåžŤäžŐU+0000åĹřU+007FèŇČåŻť˙åĘĚçŽĎ7ä¡■å■ŮçňęãĂĆ latin-1æŸŕå■ŮèŁĆ0-255åĹřU+0000èĞşU+00FFèŇČåŻť˙åĘĚUnicodeå■Ůçňę çŽĎçŽť˙æŐěæŸäåřĎãĂĆ å¡ŞèŕžåŔŰäÿĂäÿłæIJłç§ěçijŰçăĄçŽĎæŰĞæIJňæŮűä¡£çŤĺlatin-1çijŰçăĄæřÿè£IJäÿ■äijŽäžğçŤ§èğčçăĄéŤŹèŕŕãĂĆ ä¡£çŤĺlatin-1çijŰçăĄèŕžåŔŰäÿĂäÿłæŰĞäžűçŽĎæŮůåĂŹäź§èőÿäÿ■èČ¡äžğçŤ§åőŇåĚĺæ■čçąőçŽĎæŰĞæIJňèğčçăĄæŢřä¡ĘæŸŕåőČäź§èČ¡äžŐäÿ■æŔŘåŔŰåĞžèűşåď˙§åď˙ŽçŽĎæIJL’çŤĺæŢřæ■őãĂĆåŘŇæŮűïijŇåęĆæđIJä¡ăäźŃåŘ

èőĺèőž

èŕžåĘŹæŰĞæIJňæŰĞäžűäÿĂèĹňæĺěèőšæŸŕæŕŤè¿ČçőĂå■ŢçŽĎãĂĆä¡ĘæŸŕäź§åĞäçĆźæŸŕéIJĂèęĄæşéęŰåĚĹïijŇåIJĺä¿Ńå■ŘçÍŃåžŔäÿ■çŽĎwithèŕ■åŔěçżŹèćńä¡£çŤĺåĹřçŽĎæŰĞäžűåĹŻåžžäžĘäÿĂäÿłäÿŁäÿŃæä¡Ę `with` æŐğåĹűåİŰçżŞæİ§æŰűïijŇæŰĞäžűäijŽèĞłåŁĺåĚşéŮ■ãĂĆä¡ääź§åŔŕäžěäÿ■ä¡£çŤĺ `with` èŕ■åŔěïijŇä¡ĘæŸŕè£ŹæŮűåĂŹä¡ăářšå£Ěéążèőřå¿ŮæĽŃåŁĺåĚşéŮ■æŰĞäžűïijŽ

```
f = open('somefile.txt', 'rt')
data = f.read()
f.close()
```

åŔęåď˙ŰäÿĂäÿłéŮőécŸæŸŕåĚşäžŐæ■ćèąŇçňęçŽĎèŕĘåĹńéŮőécŸïijŇåIJĺUnixåŠŇWindowsäÿ■æŸŕäÿ■ `\n` åŠŇ `\r\n`)ãĂĆ éżŸèőď˙æČĚåĘţäÿŃïijŇPythonäijŽäžěçż§äÿĂæĺąåijŔåď˙ĎçŘĘæ■ćèąŇçňęãĂĆ è£Źçğ■æĺąåijŔäÿŃïijŇåIJĺèŕžåŔŰæŰĞæIJňçŽĎæŮűåĂŹïijŇPythonåŔŕäžěèŕĘåĹńæĽ˙ĂæIJL’çŽĎæŹőéĂŽæ■ `\n` å■ŮçňęãĂĆ çśžäijijçŽĎïijŇåIJĺè¿ŞåĞžæŮűäijŽářĘæ■ćèąŇçňę `\n` è¡ňæ■ćäÿžçşżçż§éżŸèőď˙çŽĎæ■ćèąŇçňęãĂĆ åęĆæđIJä¡ăäÿ■åÿŇæIJŻè£ŹçğéżŸèőď˙çŽĎåď˙ĎçŘĘæŰźåijŔi `open()` åĞ¡æŢřäijääĚěåŔĆæŢř `newline=''` ïijŇářšåČŔäÿŃéÍćè£ŹæăůïijŽ

```
# Read with disabled newline translation
with open('somefile.txt', 'rt', newline='') as f:
    ...
```

äÿžäžĘèŕť˙æŸŐäÿď˙èĂĚäźŃéŮť˙çŽĎåůőåijĆïijŇäÿŃéÍćæĹŠåIJĺUnixæIJžåŹĺäÿŁéÍćèŕžåŔŰäÿĂäÿłWindows `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'
>>>
```

æIJĂåŘŐäÿĂäÿłéŮőécŸåŕśæŸŕæŰĞæIJňæŰĞäžűäÿ■åŔŕèČ¡åĞžçŐřçŽĎçijŰçăĄéŤŹèŕŕãĂĆ ä¡Ęä¡ăèŕžåŔŰæĹŰèĂĚåĘŹåĚěäÿĂäÿłæŰĞæIJňæŰĞäžűæŮűïijŇä¡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)
>>>
```

åęĆædIJåĞžçŐřè£ŹäÿłéŤŹèŕřïijŇéĂŽåÿÿèąÍçd'žä¡ăèŕżåŔŰæŰĞæIJňæŮűæŇĞåőŽçŽĎçijŰçăĄäÿ■æ■čç
ä¡ăæIJĂåě¡äžŤçżĘéŸĚèŕżèŕť æŸŐåźűçąőèőd'ä¡ăçŽĎæŰĞäžűçijŰçăĄæŸŕæ■čçąőçŽĎ(æŕŤåęĆä¡£çŤÍUTF-
8èĂŇäÿ■æŸŕLatin-1çijŰçăĄæĹŰåĚűäžŰ)ãĂĆ åęĆædIJçijŰçăĄéŤŹèŕřè£ŸæŸŕå■ŸåIJÍçŽĎèŕÍïijŇä¡ååŔŕäžěopen() åĞ¡æŢřäijăéĂŠäÿĂäÿłåŔŕéĂĽ'çŽĎ errors åŔĆæŢřæÍěåď ĎçŘĘè£ŹäžŻéŤŹèŕřãĂĆ
äÿŃéÍćæŸŕäÿĂäžŻåď 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!'
>>>
```

åęĆædIJä¡ăçżŔåÿÿä¡£çŤÍ errors åŔĆæŢřæÍěåď ĎçŘĘçijŰçăĄéŤŹèŕřïijŇåŔŕèČ¡äijŽèőľä¡ăçŽĎçŤ§æt
åŕźäžŐæŰĞæIJňåď ĎçŘĘçŽĎéęŰèęĄåŐ§åĹŹæŸŕçąőä£Ìä¡ăæĂżæŸŕä¡£çŤÍçŽĎæŸŕæ■čçąőçijŰçăĄãĂĆå¡Şa
8)ãĂĆ

7.2 5.2 æĽŞå■řè¿ŞåĞžèĞşæŰĞäžűäÿ■

éŮőécŸ

ä¡ăæČşåřĘ print() åĞ¡æŢřçŽĎè¿ŞåĞžéĞ■åőŽåŘŚåĹřäÿĂäÿłæŰĞäžűäÿ■åŐżãĂĆ

èğčåĘşæŰźæąĹ

åIJÍ print() åĞ¡æŢřäÿ■æŇĞåőŽ file åĚşéŤőå■ŮåŔĆæŢřïijŇåČŔäÿŃéÍćè£ŹæăůïijŽ

```
with open('d:/work/test.txt', 'wt') as f:
    print('Hello World!', file=f)
```

èőÍèőž

åĚşäžŐè¿ŞåĞžéĞ■åőŽåŘŚåĹřæŰĞäżűäÿ■åřśè£ŹäžŻäžĘãĂĆä¡ĘæŸŕæIJL'äÿĂçĆźèęĄæşĺæĎŔçŽĎåřśæåęĆæđIJæŰĞäżűæŸŕäžŇè£ŻåĹűæĺąåijŔçŽĎèŕİïijŇæL'Şå■řåřśäijŽåĞžéŤŹãĂĆ

7.3 5.3 ä¡£çŤĺåĚűäżŰåĹĘéŽŤçňęæĹŰèąŇçżĹæ■ćçňęæL'Şå■ř

éŮőécŸ

ä¡ăæČşä¡£çŤĺ `print()` åĞ¡æŢřè¿ŞåĞžæŢřæ■őïijŇä¡ĘæŸŕæČşæŤźåŘŸéżŸèőd'çŽĎåĹĘéŽŤçňęæĹŰè

èğčåĘşæŰźæąĹ

åŘŕäżěä¡£çŤĺåIJĺ `print()` åĞ¡æŢřäÿ■ä¡£çŤĺ `sep` åŠŇ `end` åĚşéŤőå■ŮåŘĆæŢřïijŇäżěä¡ăæČşèęĄçŽĎæŰźåijŔè¿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■ćèąŇãĂĆæŕŤåęĆïijŽ

```
>>> 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¡äæČşä¡£çŤĺéĐçl'žæăijåĹĘéŽŤçňęæİěè¿ŞåĞžæŢřæ■őçŽĎæŮűåĂŹïijŇçżŹ `print()` åĞ¡æŢřäijăéĂŠäÿĂäÿł `sep` åŘĆæŢřæŸŕæIJĂçőÅå■ŢçŽĎæŰźæąĹãĂĆæIJL'æŮűåĂŹä¡äăijŽçIJŃåĹřäÿĂäžŽçĺŃåžŘåŚŸäijŽä¡£çŤĺ `str.join()` æİěåőŇæĹŘåŘŇæăůçŽĎäžŃæČĚãĂĆæŕŤåęĆïijŽ

```
>>> print(','.join(('ACME','50','91.5')))
ACME,50,91.5
>>>
```

str.join() çŽĎéŮőécŸåIJÍäžŐåőČäžĚäżĚéĂĆçŤÍäžŐå■Ůçňęäÿšã ĂĆè£ŹæĎŔåŚşçİĂä¡ăéĂŽåÿÿéIJ

```
>>> 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¡ŞçĎůåŔŕäžěäÿ■çŤÍéĆčäźĹéžżçČęïijŇåŔłéIJĂèęĄåČŔäÿŇéİćè£ŹæăůåĘŹïijŽ

```
>>> print(*row, sep=',')
ACME,50,91.5
>>>
```

7.4 5.4 èŕžåĘŹå■ŮèŁĆæŢřæ■ő

éŮőécŸ

ä¡ăæČşèŕžåĘŹäžŇè£ŻåĹűæŰĞäżűïijŇæŕŤåęĆåŻ¿çL’ĞïijŇåčřé§şæŰĞäżűç■L’ç■L’ãĂĆ

èğčåĘşæŰźæąĹ

ä¡£çŤÍæÍąåijŔäÿž rb æĹŰ wb çŽĎ open() åĞ¡æŢřæİèèŕžåŔŰæĹŰåĘŹåĚěäžŇè£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')
```

åIJÍèŕžåŔŰäžŇè£ŻåĹűæŢřæ■őæŮűïijŇéIJĂèęĄæŇĞæŸŐçŽĎæŸŕæL’ĂæIJL’è£ŤåŻđçŽĎæŢřæ■őéČ¡æ
çśżäijijçŽĎïijŇåIJÍåĘŹåĚěçŽĎæŮűåĂŹïijŇå£Ěéążä£İèŕĄåŔĆæŢřæŸŕäžěå■ŮèŁĆå¡ćåijŔåŕźåd’ŰæŽt’éIJšæŢ

èőÍèőž

åIJÍèŕžåŔŰäžŇè£ŻåĹűæŢřæ■őçŽĎæŮűåĂŹïijŇå■ŮèŁĆå■Ůçňęäÿšå ŠŇæŰĞæIJňå■Ůçňęäÿšç ŽĎèŕ■äźL
çL’źåĹńéIJĂèęĄæşÍæĎŔçŽĎæŸŕïijŇçt’ćåijŢåŠŇè£■äžčåŁÍä¡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£ŻåĹűæĺąåijŔçŽĎæŰĞäżűäÿ■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¡çŽ

```
import array
nums = array.array('i', [1, 2, 3, 4])
with open('data.bin','wb') as f:
    f.write(nums)
```

è£ŹäÿłéĂĆçŤĺäžŐäżżä¡ŢåőđçŐřäžĘèćńçğřäźŃäÿžâĂİçijŞåĘšæŐěåŔčâĂİçŽĎåŕźèśąïijŇè£Źçğ■åŕźèśąäi
äžŇè£ŻåĹűæŢřæ■őçŽĎåĘŹåĚěåřśæŸŕè£ŹçśżæŞ■ä¡IJäźŃäÿĂãĂĆ

å¿Ĺåď’Žåŕźèśąè£ŸåĚĄèőÿéĂŽè£Ğä¡£çŤĺæŰĞäżűåŕźèśąçŽĎ `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■æŁĂæIJŕçŽĎæŮűåĂŹéIJĂèęĄæäijåď'ŰåřŔå£ČïijŇåŻääÿžåőČéĂŽåÿÿåĚůæIJL'åź
åŔŕäžěæ§ěçIJŃ5.9åřŔèŁĆäÿ■åŔęåď'ŰäÿĂäÿłèŕżåŔŰäžŇè£ŻåĹűæŢřæ■őåĹřåŔŕä£őæŤźçijŞåĘšåŇžçŽĎä¿Ńå

7.5 5.5 æŰĞäžűäÿ■å■ŸåIJĺæĽ■èČ¡åĘŹåĚě

éŮőécŸ

ä¡ăæČşåČŔäÿĂäÿłæŰĞäžűäÿ■åĘŹåĚěæŢřæ■őïijŇä¡ĘæŸŕåL'■æŔŘå£ĚéążæŸŕè£Źäÿłæ
äź§åřśæŸŕäÿ■åĚĄèőÿèęEçŻŰåůšå■ŸåIJĺçŽĎæŰĞäžűåĘĚåőźãĂĆ

èğčåĘşæŰźæąĹ

åŔŕäžěåIJĺ `open()` åĞ¡æŢřäÿ■ä¡£çŤĺ `x` æĺąåijŔæİěäžčæŻ£ `w`
æĺąåijŔçŽĎæŰźæşŢæİěèğčåĘşè£ŹäÿłéŮőécŸãĂĆærŤåęĆïijŽ

```
>>> 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'
>>>
```

åęĆædIJæŰĞäžűæŸŕäžŇè£ŻåĹűçŽĎïijŇä¡£çŤĺ `xb` æİěäžčæŻ£ `xt`

èőĺèőž

è£ŹäÿĂåřŔèŁĆæijŤçď'žäžĘåIJĺåĘŹæŰĞäžűæŮűéĂŽåÿÿäijŽéĄĞåĹřçŽĎäÿĂäÿłéŮőécŸçŽĎåőŇç¿Őèğ
äÿĂäÿłæŻ£äžčæŰźæąĹæŸŕåĚĹæţŃèŕŢè£ŹäÿłæŰĞäžűæŸŕåŘęå■ŸåIJĺïijŇåČŔäÿŃéĺcè£Źæăůïijž

```
>>> import os
>>> if not os.path.exists('somefile'):
...     with open('somefile', 'wt') as f:
```

(continues on next page)

```
...          f.write('Hello\n')
... else:
...     print('File already exists!')
...
File already exists!
>>>
```

æŸ¿èĂŇæŸŞèğĄïijŇä¡£çŤÍxæŰĞäżűæĺąåijŔæŻťåŁăçőÅå■ŢãĂĆèęĄæşĺæĎŔçŽĎæŸŕxæĺąåijŔæŸŕäÿ
`open()` åĞ¡æŢřçL'źæIJL'çŽĎæL'ľåśŢãĂĆ åIJĺPythonçŽĎæŮğçL'ĹæIJňæĹŰèĂĚæŸŕPythonåőđçŐřçŽĎåžŢ

7.6 5.6 å■Ůçňęäÿšç ŽĎl/OæŞ■ä¡lJ

éŮőécŸ

ä¡ăæČşä¡£çŤĺæŞ■ä¡IJçśżæŰĞäżűåŕźèśąçŽĎçÍŃåžŔæİěæŞ■ä¡IJæŰĞæIJňæĹŰäžŇè£ŻåĹűå■Ůçňęäÿšã

èğčåĘşæŰźæąĹ

ä¡£çŤÍ `io.StringIO()` åŠŇ `io.BytesIO()` çśżæİěåĹŻåžžçśżæŰĞäżűåŕźèśąæŞ■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¡çŤĺäžÕæŰĞæIJňãĂĆåęĆæđIJä¡ăèęĄæŞ■ä¡IJäžŇè£ŻåĹűæŢřæ■őïijŇèęĄä¡£çŤĺ
`io.BytesIO` çśżæİěäżčæŻ£ãĂĆærŤåęĆïijŽ

```
>>> s = io.BytesIO()
>>> s.write(b'binary data')
>>> s.getvalue()
b'binary data'
>>>
```

èőÍèőž

å¡Şä¡ăæČşæÍąæŃ§äÿĂäÿłæŹőéĂŽçŽĎæŰĞäžűçŽĎæŮűåĂŹ `StringIO` åŠŇ `BytesIO` çśżæŸŕå¿ĹæIJL'çŤÍçŽĎãĂĆ æŕŤåęĆïijŇåIJÍå■ŢåĚČæţŃèŕŢäÿ■ïijŇä¡ääŔŕäžěä¡£çŤÍ `StringIO` æİěåĹŽåžžäÿĂäÿłåŇĚåŘńætŃèŕŢæŢřæ■őçŽĎçśżæŰĞäžűåŕźèśąïijŇè£ŹäÿłåŕźèśąåŔŕäžěèćńäijăçżŹæ§ŘäÿłåŔĆæŢřäÿžæŹőéĂŽæŰĞäžűåŕźèśąçŽĎåĞ¡æŢřãĂĆ

éIJĂèęĄæşÍæĎŘçŽĎæŸŕïijŇ `StringIO` åŠŇ `BytesIO` åőđä¿ŃåźűæšąæIJL'æ■čçąőçŽĎæŢt'æŢřçśżåđŃçŽĎæŰĞäžűæŔŔè£řçňęãĂĆ åŽäæ■d'ïijŇåőČäžňäÿ■èČ¡åIJÍéĆčäžŻéIJĂèęĄä¡£çŤÍçIJ§åőđçŽĎçşżçż§çžğæŰĞäžűåęĆæŰĞäžűïijŇçőąéĄŞ

7.7 5.7 èŕżåĘŹåŐŃçijl'æŰĞäžű

éŮőécŸ

ä¡ăæČşèŕżåĘŹäÿĂäÿłgzipæĹŰbz2æăijåijŔçŽĎåŐŃçijl'æŰĞäžűãĂĆ

èğčåĘşæŰźæąĹ

`gzip` åŠŇ `bz2` æÍąåİŮåŔŕäžěå¿Ĺåőźæ ŸŞçŽĎåd'ĎçŘĘè£ŹäžŻæŰĞäžűãĂĆ äÿd'äÿłæÍąåİŮéČ¡äÿž `open()` åĞ¡æŢřæŔŘä¿ŻäžĘåŖęåd'ŰçŽĎåőđçŐřæİèèğčåEşè£ŹäÿłéŮőécŸãĂĆ æŕŤåęĆïijŇäÿžäžĘäžěæŰĞæIJňå¡ćåijŔèŕżåŔŰåŐŃçijl'æŰĞäžűïijŇåŔŕäžěè£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()
```

çśżäijijçŽĎïijŇäÿžäžĘåĘŹåĚěåŐŃçijl'æŢřæ■őïijŇåŔŕäžěè£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¡£çŤÍæŰĞæIJňæÍąåijŔåźűæL'ğèąŇUnicodeçŽĎçijŰçăĄ/èğčç çśżäijijçŽĎïijŇåęĆæđIJä¡ăæČşæŞ■ä¡IJäžŇè£ŻåĹűæŢřæ■őïijŇä¡£çŤÍ `rb` æĹŰèĂĚ `wb` æŰĞäžűæÍąåijŔå■şåŔŕãĂĆ

èőÍèőž

åď ğéČÍåĹĘæČĚåĘţäÿŃèŕżåĘŹåŐŃçijľæŢřæ■őéČ¡æŸŕå¿ĹçőĂå■ŢçŽĎãĂĆä¡ĘæŸŕèęĄæşĺæĎŔçŽĎæŸ
åęĆæđIJä¡ääÿ■æŇĞåőŽæĺąåijŔïijŇéĆčäźĹéžŸèőď çŽĎåřśæŸŕäžŇè£ŻåĹűæĺąåijŔïijŇåęĆæđIJè£ŹæŮűåĂŹç
`gzip.open()` åŠŇ `bz2.open()` æŐěåŘŮèů§åĘĚç¡őçŽĎ `open()` åĞ¡æŢřäÿĂæăůçŽĎåŔĆæŢřijŇ åŇĚæŃň `encoding`ïijŇ`errors`ïijŇ`newline` ç■L'ç■L'ãĂĆ

å¡ŞåĘŹåĚěåŐŃçijľæŢřæ■őæŮűïijŇåŔŕäžěä¡£çŤÍ `compresslevel` è£ŹäÿłåŔŕéĂL'çŽĎåĚşéŤőå■ŮåŔĆæŢřæÍěæŇĞåőŽäÿĂäÿłåŐŃçijľçžğåĹńãĂĆæŕŤåęĆïijŽ

```
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¿ĹářŠèćńç§ěéĄŞçŽĎçL'źæĂğïijŇ åőČäžňåŔŕäžěä¡IJçŤĺåIJĺäÿĂäÿłåůšå■ŸåIJĺázűäžěäžŇè£

```
import gzip
f = open('somefile.gz', 'rb')
with gzip.open(f, 'rt') as g:
    text = g.read()
```

è£ŹæăůåřśåĚĄèőÿ `gzip` åŠŇ `bz2` æĺąåİŮåŔŕäžěåůěä¡IJåIJĺèőÿåď ŽçśżæŰĞäžűåŕźèśąäÿŁïijŇæŕŤåęĆåě

7.8 5.8 åŻžåőŽåď ğåřŔèőřå¡ŢçŽĎæŰĞäžűè£■äžč

éŮőécŸ

ä¡äæČşåIJĺäÿĂäÿłåŻžåőŽéŢ£åžęèőřå¡ŢæĹŰèĂĚæŢřæ■őåİŮçŽĎéŻĘåŘĹäÿŁè£■äžčïijŇèĂŇäÿ■æŸŕåIJ

èğčåĘşæŰźæąĹ

éĂŽè£ĞäÿŃéİćè£ŹäÿłåřŔæŁĂåůğä¡£çŤÍ `iter` åŠŇ `functools.partial()` åĞ¡æŢřijŽ

```
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¿Ńå■Řäÿ■çŽĎ `records` åŕźèśąæŸŕäÿĂäÿłåŔŕè£■äžčåŕźèśąïijŇåőČäijŽäÿ■æŰ■çŽĎäžğçŤ§åŻž
èęĄæşĺæĎŔçŽĎæŸŕåęĆæđIJæĂżèőřå¡Ţåď ğåřŔäÿ■æŸŕåİŮåď ğåřŔçŽĎæŢt' æŢřåĂ■çŽĎèŕĪïijŇæIJĂåŘŐäÿĂ

èőÍèőž

`iter()` åĞ¡æŢřæIJL'äÿĂäÿłéšIJäÿžäžžç§ěçŽĎçL'źæĂğåřśæŸŕřijŇåęĆæđIJä¡ăçżŹåőČäijăéĂŠäÿĂäÿłåI
è£Źäÿłè£■äžčåŹÍäijŽäÿĂçŻť'èřČçŤÍäijăåĚěçŽĎåŔŕèřČçŤÍłåŕźèśąçŻť'åĽřåőČè£ŤåŻđæĂĞèőřåĂijäÿžæ■ćïijŇè£

åIJÍä¿Ńå■Řäÿ■ïijŇ `functools.partial` çŤÍæÌěåĹŻåžžäÿĂäÿłæŕŔæňąèćńèřČçŤÍæŮűäžŐæŰĞäžűä
æăĞèőřåĂij `b''` åřśæŸŕå¡ŞåĹřè¿¿æŰĞäžűçżŞåř¿æŮűçŽĎè£ŤåŻđåĂijãĂĆ

æIJĂåŘŐåĘ■æŔŘäÿĂçĆźïijŇäÿŁéÍćçŽĎä¿Ńå■Řäÿ■çŽĎæŰĞäžűæŮűäžěäžŇè£ŻåĹűæĺąåijŔæL'ŞåijĂç
åęĆæđIJæŸŕèŕźåŔŰåŻžåőŽåd'ğåřŔçŽĎèőřå¡ŢïijŇè£ŹéĂŽåÿÿæŸŕæIJĂæŹőéĄ■çŽĎæČĚåĘţãĂĆ
èĂŇåŕźäžŐæŰĞæIJñæŰĞäžűïijŇäÿĂèąŇäÿĂèąŇçŽĎèŕźåŔŰ(éżŸèőd'çŽĎè£■äžčèąŇäÿž)æŻť'æŹőéĄ■çĆźã

7.9 5.9 èŕźåŔŰäžŇè£ŻåĹűæŢřæ■őåĹřåŔŕåŔŸçijŞåĘšåŇžäÿ■

éŮőécŸ

ä¡ăæČşçŻť'æŐěèŕźåŔŰäžŇè£ŻåĹűæŢřæ■őåĹřäÿĂäÿłåŔŕåŔŸçijŞåĘšåŇžäÿ■ïijŇèĂŇäÿ■éIJĂèęĄåĄŽäż
æĹŰèĂĚä¡ăæČşåŐ§åIJřä£őæŤźæŢřæ■őåźűårĘåőČåĘŹåŽđåĹřäÿĂäÿłæŰĞäžűäÿ■åŐżãĂĆ

èğčåĘşæŰźæąĹ

äÿžäžĘèŕźåŔŰæŢřæ■őåĹřäÿĂäÿłåŔŕåŔŸæŢřçżĎäÿ■ïijŇä¡£çŤÍæŰĞäžűåŕźèśąçŽĎ
`readinto()` æŰźæşŢãĂĆærŤåęĆïijŽ

```
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
```

äÿŃéÍćæŸŕäÿĂäÿłæijŤçd'žè£ŹäÿłåĞ¡æŢřä¡£çŤÍæŰźæşŢçŽĎä¿Ńå■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)

```
11
>>>
```

èőÍèőž

æŰĞäżűåŕźèśąçŽĎ `readinto()` æŰźæşŢèČ¡èćńçŤÍæİěäÿžécĎåĚĹåĹĘéĚ■åĘĚå■ŸçŽĎæŢřçżĎåąńåĚ `array` æĺąåİŮæĹŰ `numpy` åžŞåĹŻåżžçŽĎæŢřçżĎãĂĆ åŠŇæŹőéĂŽ `read()` æŰźæşŢäÿ■åŘŇçŽĎæŸŕïijŇ `readinto()` åąńåĚĚåůšå■ŸåIJÍçŽĎçijŞåĘšåŇžèĂŇäÿ■æŸŕäÿžæŰřåŕźèśąéĞ åŻăæ■ď¨ïijŇä¡ăåŘŕäżěä¡£çŤĺåőČæİěéĄ£åĚ■åď¨ğéĞŘçŽĎåĘĚå■ŸåĹĘéĚ■æŞ■ä¡IJãĂĆ æŕŤåęĆïijŇåęĆæđIJä¡ăèŕżåŘŰäÿĂäÿłçŤśçŻÿåŘŇåď¨ğåŕŘçŽĎèőřå¡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¡IJïijŇçŤŽè

```
>>> 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¡ăå£ĚéążæčĂæ§ěåőČçŽĎè£ŤåŻđåĂijïijŇä

åęĆæđIJå■ŮèŁĆæŢřåŕŘäžŐçijŞåĘšåŇžåď¨ğåŕŘïijŇèąÍæŸŐæŢřæ■őèćńæĹłæŰ■æĹŰèĂĚèćńçăť¨åİŘäžĘ

æIJĂåŘŐïijŇçŢŹå£ČèğĆåŕ§åĚűäżŰåĞ¡æŢřåžŞåŠŇæÍąåİŮäÿ■åŠŇ `into` çŻÿåĚşçŽĎåĞ¡æŢř(æŕŤåęĆ `recv_into()` ïijŇ `pack_into()` ç■Ľ)ãĂĆ PythonçŽĎå¿Ĺåď¨ŻåĚűäžŰéČĺåĹĘåůšçżŘèČ¡æŤŕæŇĄçŻť¨æŐěçŽĎI/OæĹŰæŢřæ■őèő£éŮőæŞ■ä¡IJïijŇè£Źä

åĚşäžŐèğčæđŘäžŇè£ŻåĹűçżŞæđĎåŠŇ `memoryviews` ä¡£çŤÍæŰźæşŢçŽĎæŻť¨énŸçžğä¿Ňå■ŘïijŇèŕůåŘĆèĂČ6.12åŕŘèŁĆãĂĆ

7.10 5.10 åĘĚå■ŸæŸăåřĎçŽĎäžŇè£ŻåĹűæŰĞäżű

éŮőécŸ

ä¡ăæČşåĘĚå■ŸæŸăåřĎäÿĂäÿłäžŇè£ŻåĹűæŰĞäżűåĹřäÿĂäÿłåŔŕåŔŸå■ŮèŁĆæŢřçżĎäÿ■ïijŇçŽőçŽĎåŀ

èğčåĘşæŰźæąĹ

ä¡£çŤÍ mmap æĺąåİŮæİěåĘĚå■ŸæŸăåřĎæŰĞäżűãĂĆ äÿŇéİćæŸŕäÿĂäÿłåůěåĚůåĞ¡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¡ăéIJĂèęĄæIJL'äÿĂäÿłåůšåĹŻåžžåźűäÿŤåĘĚåőźäÿ■äÿžçľžçŽĎæŰĞäż äÿŇéİćæŸŕäÿĂäÿłä¿Ňå■ŘïijŇæŢŹä¡ăæĂŐæăůåĹİåğŇåĹŻåžžäÿĂäÿłæŰĞäżűåźűårĘåĚűåĘĚåőźæL'ľåĚĚåĹř

```
>>> 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 åŕźèšąåŘŇæăůäź§åŔŕäžëä¡IJäÿžäÿĂäÿłäÿŁäÿŃæŰĞçőąçŘĘåŹĺæİěä¡£çŤĺïj
è£ŹæŮűåĂŹåžŢåśÇçŽĎæŰĞäžűäijŽèćńèĞłåŁĺåĚşéÚ■ãĂĆæŕŤåęĆïijŽ

```
>>> with memory_map('data') as m:
...     print(len(m))
...     print(m[0:10])
...
1000000
b'Hello World'
>>> m.closed
True
>>>
```

éżŸèőď'æČĚåĘţäÿŃïijŇ memeory_map() åĞ¡æŢřæL'ŞåijĂçŽĎæŰĞäžűåŘŇæŮűæŤŕæŇĄèŕžåŠŇåĘŹ
äżżä¡ŢçŽĎä£őæŤźåĘĚåőźéČ¡äijŽåď'■åĹűåŻđåŐ§æİěçŽĎæŰĞäžűäÿ■ãĂĆ
åęĆæđIJéIJĂèęĄåŔłèŕżçŽĎèő£éŮőæĺąåijŔïijŇåŔŕäžěçżŹåŔĆæŢř access èţŃåĂijäÿž
mmap.ACCESS_READ ãĂĆæŕŤåęĆïijŽ

```
m = memory_map(filename, mmap.ACCESS_READ)
```

åęĆæđIJä¡ăæČşåIJĺæIJñåIJřä£őæŤźæŢřæ■őïijŇä¡ĘæŸŕåŔĹäÿ■æČşåřĘä£őæŤźåĘŹåŻđåĹřåŐ§ågŃæŰĞ
mmap.ACCESS_COPY ïijŽ

```
m = memory_map(filename, mmap.ACCESS_COPY)
```

èőĺèőž

äÿžäžĘéŽŔæIJžèő£éŮőæŰĞäžűçŽĎåĘĚåőźïijŇä¡£çŤĺ mmap
åřĘæŰĞäžűæŸăåřĎåĹřåĘĚå■ŸäÿæŸŕäÿĂäÿłéńŸæŢĹåŠŇäijŸéŻĚçŽĎæŰźæşŢãĂĆ
ä¿ŃåęĆïijŇä¡ăæŮăéIJĂæL'ŞåijĂäÿĂäÿłæŰĞäžűåźűæL'ğèąŇåď'ğéĞŔçŽĎ seek() ïijŇ
read() ïijŇ write() èřČçŤĺïijŇ åŔłéIJĂèęĄçőĂå■ŢçŽĎæŸăåřĎæŰĞäžűåźűä¡£çŤĺåĹĞçL'ĞæŞ■ä¡IJèő£é

äÿĂèĹñæİěèőšïijŇ mmap() æL'ĂæŽť'éIJšçŽĎåĘĚå■ŸçIJŃäÿŁåŐżåřśæŸŕäÿĂäÿłäžŇè£ŻåĹűæŢřçżĎåŕź
ä¡ĘæŸŕïijŇä¡ăåŔŕäžěä¡£çŤĺäÿĂäÿłåĘĚå■ŸèğĘåŻ¿æİěèğčæđŘåĚűäÿ■çŽĎæŢřæ■őãĂĆæŕŤåęĆïijŽ

```
>>> 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
>>>
```

éIJĂèęĄåijžèřČçŽĎäÿĂçĆźæŸŕïijŇåĘĚå■ŸæŸăåřĎäÿĂäÿłæŰĞäžűåźűäÿ■äijŽåŕijèĞť'æŢť'äÿłæŰĞäžűé
äź§åřśæŸŕèŕť'ïijŇæŰĞäžűåźűæšąæIJL'èćńåď'■åĹűåĹřåĘĚå■ŸçijŞå■ŸæĹŰæŢřçżĎäÿ■ãĂĆçŻÿåŔ■ïijŇæŞ■ä¡
å¡Şä¡ăèő£éŮőæŰĞäžűçŽĎäÿ■åŘŇåŇžå§§æŮűïijŇè£ŹäžŻåŇžå§§çŽĎåĘĚåőźæL'■æăźæ■őéIJĂèęĄèćńèŕžåŔ
èĂŇéĆčäžŻäžŐæšąèćńèő£éŮőåĹřçŽĎéČĺåĹĘè£ŸæŸŕçŢŹåIJĺçčĄçŻŸäÿŁãĂĆæL'ĂæIJL'è£ŹäžŻè£ĞçĺŃæŸ

åęĆæđIJåď ŽäÿłPythonèğčéĞŁåŹĺåĘĚå■ŸæŸäåřĎåŘŇäÿĂäÿłæŰĞäżűïijŇå¿ŮåĹřçŽĐ mmap åŕźèśąèČ¡åď§èćńçŤĺæİěåIJĺèğčéĞŁåŹĺçŻť æŐěäžď æ■ćæŢřæ■őãĂĆ äź§åřśæŸŕèŕť ïijŇæĽĂæIJĽèğčéĞŁåŹĺéČ¡èČ¡åŘŇæŮűèŕżåĘŹæŢřæ■őïijŇåźűäÿŤåĚűäÿ■äÿĂäÿłèğčéĞŁåŹĺ å¿ĹæŸŐæŸ¿ïijŇè£ŹéĞŇéIJĂèęĄèĂČèŹŚåŘŇæ■ĕçŽĐéŮőéćŸãĂĆä¡ĘæŸŕè£Źçğ■æŰźæşŢæIJĽæŮűåĂŹåř

è£ŹäÿĂåřŔèŁĆäÿ■åĞ¡æŢřår¡éĞŔåĘŹå¿Ůå¿ĹéĂŽçŤĺïijŇåŘŇæŮűéĂĆçŤĺäžŐUnixåŠŇWindowsåźşåŔ èęĄæşĺæĎŔçŽĐæŸŕä¡£çŤĺ mmap() åĞ¡æŢřæŮűäijŽåIJĺåžŢåśĆæIJĽäÿĂäžŻåźşåŔřçŽĐåůőåijĆæĂğãĂĆ åŘęåď ŰïijŇè£ŸæIJĽäÿĂäžŻéĂĽéąźåŔŕäżěçŤĺæİěåĹŻåżžåŇ£åŘ■çŽĐåĘĚå■ŸæŸäåřĎåŇžå§§ãĂĆ åęĆæđIJä¡ăåŕźè£ŹäÿłæĎ§åĚť èűčïijŇçąőä£İä¡äžŤçżĘçăŤèŕżäžĘPythonæŰĞæąčäÿ■ è£ŹæŰźéİćçŽĐåĘĚåőź ãĂĆ

7.11 5.11 æŰĞäżűèůŕå¿ĎåŘ■çŽĐæŞ■ä¡IJ

éŮőéćŸ

ä¡ăéIJĂèęĄä¡£çŤĺèůŕå¿ĎåŘ■æİěèŐůåŔŰæŰĞäżűåŘ■ïijŇçŻőå¡ŢåŘ■ïijŇçżİåŕźèůŕå¿Ďç■L'ç■L'ãĂĆ

èğčåĘşæŰźæąĹ

ä¡£çŤĺ os.path æĺąåİŮäÿ■çŽĐåĞ¡æŢřæİěæŞ■ä¡IJèůŕå¿ĎåŘ■ãĂĆ äÿŃéİćæŸŕäÿĂäÿłäžď äžŠäijŔä¿Ňå■ŘæİěæijŤçď žäÿĂäžŻåĚşéŤőçŽĐçL'źæĂğïijŽ

```
>>> 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

åŕźäžŐäżżä¡ŢçŽĎæŰĞäžűåŘ■çŽĎæŞ■ä¡IJïijŇä¡äéČ¡åžŤèŕěä¡£çŤÍ `os.path` æĺąåİŮïijŇèĂŇäÿ■æŸŕä¡£çŤÍæăĞåĞĘå■ŮçňęäÿšæŞ■ä¡IJæİěæđĎéĂäèĞłåůśçŽĎäżčçăĄãĂĆçL'źåĹńæŸŕäÿžäžĘåŔŕçğżæď'■æĂğèĂČèŹŚçŽĎæŮűåĂŹæŻť'åžŤåęĆæ■ď'ïijŇ åŻääÿž `os.path` æĺąåİŮç§ěéĄŞUnixåŠŇWindowsçşżçż§äźŃéŮť'çŽĎåůőåijĆåźűäÿŤèČ¡åď'§åŔŕéİăåIJŕåď'ĎçŘĘçśżäijij `Data/data.csv` åŠŇ `Data\data.csv` è£Źæăůç ŽĎæŰĞäžűåŘ■ãĂĆ åĚűæňąïijŇä¡äçIJ§çŽĎäÿ■åžŤèŕěæţłèť'źæŮűéŮť'åŐżéĞ■åď'■éĂäè¡őå■ŘãĂĆéĂŽåÿÿæIJĂåě¡æŸŕçŻť'æŐëä¡£

èęĄæşĺæĎŔçŽĎæŸŕ `os.path` è£ŸæIJL'æŻť'åď'ŽçŽĎåŁ§èČ¡åIJÍè£ŹéĞŇåźűæšąæIJL'åĹŮäÿ¿åĞžæİěãåŔŕäžěæ§ěéŸĚåőŸæŰźæŰĞæąčæİèèŐůåŘŰæŻť'åď'Žäÿ ŐæŰĞäžűæţŃèŕŢïijŇçňęåŘůéŞ¿æŐęç■L'çŻÿåĚşçŽ

7.12 5.12 æţŃèŕŢæŰĞäžűæŸŕåŘęå■ŸåIJÍ

éŮőécŸ

ä¡ăæČşæţŃèŕŢäÿĂäÿłæŰĞäžűæĹŰçŻőå¡ŢæŸŕåŘęå■ŸåIJÍãĂĆ

èğčåĘşæŰźæąĹ

ä¡£çŤÍ `os.path` æĺąåİŮæİěæţŃèŕŢäÿĂäÿłæŰĞäžűæĹŰçŻőå¡ŢæŸŕåŘęå■ŸåIJÍãĂĆærŤåęĆïijŽ

```
>>> import os
>>> os.path.exists('/etc/passwd')
True
>>> os.path.exists('/tmp/spam')
False
>>>
```

ä¡ăè£ŸèČ¡è£ŻäÿĂæ■ěæţŃèŕŢè£ŹäÿłæŰĞäžűæŮűäżĂäźĹçśżåđŃçŽĎãĂĆ åIJÍäÿŃéİćè£ŹäžŻæţŃèŕŢäÿ■ïijŇåęĆæđIJæţŃèŕŢçŽĎæŰĞäžűäÿ■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'
>>>
```

åęĆædIJä¡ăè£ŸæČşèŐůåŔŰåĚČæŢřæ■ő(æŕŤåęĆæŰĞäžűåď ğářŔæĹŰèĂĚæŸŕä£őæŤźæŮěæIJ§)ïijŇäź
os.path æĺąåİŮæİěèğčåEşïijŽ

```
>>> 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£ŻèąŇæŰĞäžűæţŃèŕŢæŸŕå¿ĹçőĂå■ŢçŽĎãĂĆ
åIJĺåĘŹè£ŹäžŻèĎŽæIJňæŮűïijŇåŔŕèČ¡åŤŕäÿĂéIJĂäèęĄæşĺæĎŔçŽĎåřśæŸŕä¡ăéIJĂäèęĄèĂČèŹŚæŰĞäžűæİČ

```
>>> 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 èŐůåŔŰæŰĞäžűåď źäÿ■çŽĎæŰĞäžűåĹŮèąĺ

éŮőécŸ

ä¡ăæČşèŐůåŔŰæŰĞäžűçşżçż§äÿ■æ§ŘäÿłçŻőå¡ŢäÿŃçŽĎæĽĂæIJĽæŰĞäžűåĹŮèąĺãĂĆ

èğčåEşæŰźæąĹ

ä¡£çŤÍ os.listdir() åĞ¡æŢřæİěèŐůåŔŰæ§ŘäÿłçŻőå¡Ţäÿ■çŽĎæŰĞäžűåĹŮèąĺïijŽ

```
import os
names = os.listdir('somedir')
```

çżŞædIJäijŽè£ŤåŻđçŻőå¡Ţäÿ■æĽĂæIJĽæŰĞäžűåĹŮèąĺïijŇåŇĚæŃňæĽĂæIJĽæŰĞäžűïijŇå■ŘçŻőå¡Ť
åęĆædIJä¡ăéIJĂäèęĄéĂŽè£Ğæ§Řçğ■æŰźåijŔè£Ğæżď æŢřæ■őïijŇåŔŕäžěèĂČèŹŚçżŞåŘĹ
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£Ğæżď'äÿĂäÿłçŻőå¡ŢçŽĎåĘĚåőźäź§æŸŕå¿ĹæIJĽçŤĺçŽĎãĂĆæŕŤåęĆïijŽ

```
pyfiles = [name for name in os.listdir('somedir')
           if name.endswith('.py')]
```

åŕźäžŐæŰĞäżűåŘ■çŽĎåŇźéĚ■ïijŇä¡ăåŔŕèČ¡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')]
```

èőĺèőž

èŐůåŔŰçŽőå¡Ţäÿ■çŽĎåĹŮèąĺæŸŕå¿Ĺåőźæ ŸŞçŽĎïijŇä¡ĘæŸŕåĚűè£ŤåŻđçżŞæđIJåŔłæŸŕçŽőå¡Ţäÿ■åč
åęĆæđIJä¡äè£ŸæČşèŐůåŔŰåĚűäžŰçŽĎåĚČä£ąæĄfïijŇæŕŤåęĆæŰĞäžűåď'ğåřŔïijŇä£őæŤźæŮűéŮť'ç■Ľç■
ä¡ăæĹŰèőÿè£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)

```
# 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£Ÿ æIJĽäÿĂçĆźèęĄæşĺæĎŔçŽĎåřśæŸŕïijŇæIJĽæŮűåĂŹåIJĺåď ĎçŘĘæŰĞäżűåŘ■çijŰçăĄé éĂŽåÿÿæİěèő šïijŇåĞ¡æŢř os.listdir() è£ŤåŽđçŽĎåőđä¡ŞåĹŮèąĺäijŽæăźæ■őçşżçż§éżŸèőď çŽĎæŰĞä ä¡ĘæŸŕæIJĽæŮűåĂŹäź§äijŽçćřåĹřäÿĂäžŻäÿ■èČ¡æ■čåÿÿèğčçăĄçŽĎæŰĞäżűåŘ■ãĂĆ åĚşäžŐæŰĞäżűåŘ■çŽĎåď ĎçŘĘéŮőéćŸïijŇåIJĺ5.14åŠŇ5.15årŔèŁĆæIJĽæŻť èŕęçżĘçŽĎèőšèğčãĂĆ

7.14 5.14 å£¡çŢěæŰĞäżűåŘ■çijŰçăĄ

éŮőéćŸ

ä¡ăæČşä¡£çŤĺåŐ§åğŃæŰĞäżűåŘ■æĽğèąŇæŰĞäżűçŽĎI/OæŞ■ä¡IJïijŇäź§åřśæŸŕèŕť æŰĞäżűåŘ■åźűæ

èğčåĘşæŰźæąĹ

éżŸèőď æČĚåĘţäÿŃïijŇæĽĂæIJĽçŽĎæŰĞäżűåŘ■éČ¡äijŽæăźæ■ő sys.getfilesystemencoding() è£ŤåŽđçŽĎæŰĞæIJňçijŰçăĄæİěçijŰçăĄæĹŰèğčçăĄãĂĆæŕŤåęĆïijŽ

```
>>> sys.getfilesystemencoding()
'utf-8'
>>>
```

åęĆæđIJåŽăäÿžæ§Řçğ■åŐ§åŽää¡ăæČşå£¡çŢěè£Źçğ■çijŰçăĄïijŇåŔŕäżěä¡£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)

```
Spicy!
>>>
```

æ■čåęĆä¡ăæĽĂèğĄïijŇåIJÍæIJĂåŘŐäÿd'äÿłæŞ■ä¡IJäÿ■ïijŇå¡Şä¡ăçżŹæŰĞäžűçŻÿåĚşåĞ¡æŢřåęĆ
`open()` åŠŇ `os.listdir()` äijăéĂŠå■ÚèŁĆå■ŮçňęäÿšæŮűïijŇæŰĞäžűåŘ■çŽĎåd'ĎçŘĘæŰźåijŔäijŽçĺ

èőĺèőž

éĂŽåÿÿæİěèőšïijŇä¡ääÿ■éIJĂèęĄæŃĚå£ČæŰĞäžűåŘ■çŽĎçijŰçăĄåŠŇèğčçăĄïijŇæŹőéĂŽçŽĎæŰĞäž
ä¡ĘæŸŕïijŇæIJL'äžŻæŞ■ä¡IJçşżż§åĚĄèőÿçŢĺæĹůéĂŽè£ĞåĄűçĎűæĹŰæĄűæĎŔæŰźåijŔåŐżåĹŻåżžåŘ■å■
è£ŹäžŻæŰĞäžűåŘ■åŔŕèČ¡äijŽçěđçğŸåIJřäÿ■æŰ■éĆčäžŻéIJĂèęĄåd'ĎçŘĘåd'ğéĞŔæŰĞäžűçŽĎPythonçĺŃ

èŕżåŔŰçŽőå¡ŢåźűéĂŽè£ĞåŐ§åğŃæIJłèğčçăĄæŰźåijŔåd'ĎçŘĘæŰĞäžűåŘ■åŔŕäžěæIJL'æŢĹçŽĎéĄ£å
ař¡çőąè£ŹæăůäijŽåÿęæİěäÿĂåőŽçŽĎçijŰçĺŃéŽ¿åžęãĂĆ

åĚşäžŐæĽŞå■řäÿ■åŔŕèğčçăĄçŽĎæŰĞäžűåŘ■ïijŇèŕůåŔĆèĂČ5.15åřŔèŁĆãĂĆ

7.15 5.15 æĽŞå■řäÿ■åŘĹæşŢçŽĎæŰĞäžűåŘ■

éŮőécŸ

ä¡ăçŽĎçĺŃåžŔèŐůåŔŰäžĘäÿĂäÿłçŻőå¡Ţäÿ■çŽĎæŰĞäžűåŘ■åĹŮèąĺïijŇä¡ĘæŸŕå¡ŞåőČèŕŢçİĂåŐżæĽŞ
åĞžçŐřäžĘ `UnicodeEncodeError` åijĆåÿÿåŠŇäÿĂæİąåěĞæĂłçŽĎæűĹæĄŕâĂŤâĂŤ
`surrogates not allowed` ãĂĆ

èğčåĘşæŰźæąĹ

å¡ŞæĽŞå■řæIJłç§ěçŽĎæŰĞäžűåŘ■æŮűïijŇä¡£çŤĺäÿŃéİćçŽĎæŰźæşŢåŔŕäžěéĄ£åĚ■è£ŹæăůçŽĎéŤŹè

```
def bad_filename(filename):
    return repr(filename)[1:-1]

try:
    print(filename)
except UnicodeEncodeError:
    print(bad_filename(filename))
```

èőĺèőž

è£ŹäÿĂåřŔèŁĆèőĺèőžçŽĎæŸŕåIJĺçijŰåĘŹå£Ěéąžåd'ĎçŘĘæŰĞäžűçşżż§çŽĎçĺŃåžŔæŮűäÿĂäÿłäÿ■åd
éżŸèőd'æČĚåĘţäÿŃïijŇPythonåĄĞåőŽæĽĂæIJL'æŰĞäžűåŘ■éČ¡åůšçżŔæăźæ■ő
`sys.getfilesystemencoding()` çŽĎåĂijçijŰçăĄè£ĞäžĘãĂĆ
ä¡ĘæŸŕïijŇæIJL'äÿĂäžŻæŰĞäžűçşżż§åźűæšąæIJL'åijžåĹűèęĄæśĆè£ŹæăůåĄŽïijŇåŻăæ■d'åĚĄèőÿåĹŻåžža
è£Źçğ■æČĚåĘţäÿ■åd'łåÿÿèğĄïijŇä¡ĘæŸŕæĂżäijŽæIJL'äžŻçŤĺæĹůåĘŠéŹľè£ŹæăůåĄŽæĹŰèĂĚæŸŕæŮăæĽ

åŕŕèČ¡æŸŕåIJĺäÿĂäÿłæIJL’çijžéŹůçŽĎäžčçăĄäÿ■çżŹ open()
åĞ¡æŢřäijăéĂŠäžĘäÿĂäÿłäÿ■åŘĹèğĎèŇČçŽĎæŰĞäžűåŘ■)ãĂĆ

å¡ŞæĽ’ğèąŇçśżäijij os.listdir() è£ŹæăůçŽĎåĞ¡æŢřæŮűïijŇè£ŹäžŻäÿ■åŘĹèğĎèŇČçŽĎæŰĞäžű
äÿĂæŰźéİćïijŇåőČäÿ■èČ¡äżĚäżĚåŔłæŸŕäÿćåijČè£ŹäžŻäÿ■åŘĹæăijçŽĎåŘ■å■ŮãĆćèĂŇåŔęäÿĂæŰźéİćïij
Pythonåŕźè£ŹäÿłéŮőéćŸçŽĎèğčåĘşæŰźæąĹæŸŕäžŐæŰĞäžűåŘ■äÿ■èŐůåŔŰæIJłèğčçăĄçŽĎå■ŮèŁĆåĂijæ
\xhh åźűåřĘåőČæŸăåřĎæĹŘUnicodeå■Ůçňę \udchh èąĺçď žçŽĎæL’ĂèřŞçŽĎâĂİäžčçŘĘçijŰçăĄâĂİãĂĆ
äÿŃéİćäÿĂäÿłä¿Ńå■ŘæijŤçď žäžĘå¡ŞäÿĂäÿłäÿ■åŘĹæăijçŽőå¡ŢåĹŮèąĺäÿ■åŘńæIJL’äÿĂäÿłæŰĞäžűåŘ■äÿžb
1èĂŇäÿ■æŸŕUTF-8çijŰçăĄ)æŮűçŽĎæăůå■ŘïijŽ

```
>>> import os
>>> files = os.listdir('.')
>>> files
['spam.py', 'b\udce4d.txt', 'foo.txt']
>>>
```

åęĆædIJä¡ăæIJL’äžčçăĄéIJĂèęĄæŞ■ä¡IJæŰĞäžűåŘ■æĹŰèĂĚåřĘæŰĞäžűåŘ■äijăéĂŠçžŹ
open() è£ŹæăůçŽĎåĞ¡æŢřijŇäÿĂåĹĞéČ¡èČ¡æ■čåÿÿåůëä¡IJãĂĆ
åŘłæIJL’å¡Şä¡ăæČşèęĄè¿ŞåĞžæŰĞäžűåŘ■æŮűæL’■äijŽçćřåĹřäžŻéžžçČę(æŕŤåęĆæL’Şå■řè¿ŞåĞžåĹřåśŔåźŢ
çL’źåĹńçŽĎïijŇå¡Şä¡ăæČşæL’Şå■řäÿŁéİćçŽĎæŰĞäžűåŘ■åĹŮèąĺæŮűïijŇä¡ăçŽĎçĺŇäžŔřśäijŽåť’ľæžČïijŽ

```
>>> 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
>>>
```

çĺŇåžŔåť’ľæžČçŽĎåŐ§åŻăářśæŸŕå■Ůçňę \udce4 æŸŕäÿĂäÿłéİđæşŢçŽĎUni-
codeå■Ůçňęă ĂĆ åőČåĚűăđđæŸŕäÿĂäÿłèćńçğřäÿžäżčçŘĘå■ŮçňęåŕźçŽĎåŔŇå■ŮçňęçżĎåŘĹçŽĎåŘŐå■ŁéČ
çŤśäžŐçijžåřŚäžĘåĽ■å■ŁéČĺåĹĘïijŇåŻăæ■ď’åőČæŸŕäÿłéİđæşŢçŽĎUnicodeãĂĆ
æĽĂäżěïijŇåŤŕäÿĂèČ¡æĹŘåŁ§è¿ŞåĞžçŽĎæŰźæşŢåřśæŸŕå¡ŞéĄĞåĹřäÿ■åŘĹæşŢæŰĞäžűåŘ■æŮűéĞĞåŔU
æŕŤåęĆåŔŕäžěåřĘäÿŁè£řäżčçăĄä£őæŤźåęĆäÿŃïijŽ

```
>>> 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¡őåŔŰåĘşäžŐä¡äèĞłåůśăĂĆ
åŘęåď’ŰäÿĂäÿłéĂL’æŃľåřśæŸŕéĂŽè£Ğæ§ŘçğæŰźåijŔéĞ■æŰřçijŰçăĄïijŇçď’žä¿ŃåęĆäÿŃïijŽ

```
def bad_filename(filename):
    temp = filename.encode(sys.getfilesystemencoding(), errors=
↪'surrogateescape')
    return temp.decode('latin-1')
```

èŕŚèĂĚæşÍ:

```
surrogateescape:
è£ŹçğæŸŕPythonåIJÍçżİåd'ğéČÍåĹĘéİćåŘŚOSçŽĎAPIäÿæL'Ăä¡£çŤÍçŽĎéŤŹèŕŕåd'ĎçŘĘåŹÍiiĵ
åőČèČ¡äżěäÿĂçğäijŸéŻĚçŽĎæŰźåijŔåd'ĎçŘĘçŤśæŞä¡IJçşżçż$æŔŘä¿ŻçŽĎæŢřæőçŽĎçijŰçăĄé
åIJÍèğčçăĄåĞžéŤŹæŮűäijŽåřĘåĞžéŤŹå Ůè£Ćå ŸåĆÍåĹřäÿĂäÿłå¿ĹåřŚèćńä¡£çŤÍåĹřçŽĎUnicode
åIJÍçijŰçăĄæŮűåřĘéĆčäžżéŽŘèŮŔåĂijåŔĹè£ŸåŐ$åŻđåŐ$åĚĹèğčçăĄåd'śèt'ěçŽĎå Ůè£ĆåžŔåĹŮ
åőČäÿ äżĚåŕźäžŐOS␣
↪APIéİđåÿÿæIJL'çŤÍïijÑäź$èČ¡å¿Ĺåőźæ Ÿ$çŽĎåd'ĎçŘĘåĚűäżŰæČĚåĘţäÿŃçŽĎçijŰçăĄéŤŹèŕŕã
```

ä¡£çŤÍè£ŹäÿłçL'ĹæIJñäžğçŤ§çŽĎè¿ŞåĞžåęĆäÿŃïijŽ

```
>>> for name in files:
...     try:
...         print(name)
...     except UnicodeEncodeError:
...         print(bad_filename(name))
...
spam.py
bÃd'd.txt
foo.txt
>>>
```

è£ŹäÿĂåřŔèŁĆäÿżécŸåŔŕèČ¡äijŽèćńåď'ğéČÍåĹĘèŕżèĂĚæL'Ăå£çŢěãĂĆä¡ĘæŸŕåęĆæđIJä¡ăåIJÍçijŰåĘ
åřśå£Ěéążå¿ŮèĂČèŹŚåĹře£ŹäÿłãĂĆåŘęåĹŹä¡ăåŔŕèČ¡äijŽåIJÍæ§Ŕäÿłå ŚÍæIJńèćńåŔńåĹřåŁđåĚñåőď'åŐżèřČ

7.16 5.16 åćđåŁăæĹŰæŤźåŔŸåůšæL'ŞåijÄæŰĞäżűçŽĎçijŰçăĄ

éŮőécŸ

ä¡ăæČşåIJÍäÿ åĚşéŮ äÿĂäÿłåůšæL'ŞåijĂçŽĎæŰĞäżűåL'æŔŘäÿŃåćđåŁăæĹŰæŤźåŔŸåőČçŽĎUnicoc

èğčåĘşæŰźæąĹ

åęĆæđIJä¡ăæČşçżŹäÿĂäÿłäżěäžŇè£ŻåĹűæĺąåijŔæL'ŞåijĂçŽĎæŰĞäżűæůżåŁăUnicodeçijŰçăĄ/èğčçăĄ
åŔŕäżěä¡£çŤÍ `io.TextIOWrapper()` åŕźèśąåŇĚèčĚåőČãĂĆæŕŤåęĆïijŽ

```
import urllib.request
import io

u = urllib.request.urlopen('http://www.python.org')
```

(continues on next page)

```
f = io.TextIOWrapper(u, encoding='utf-8')
text = f.read()
```

åęĆædIJä¡ăæČşä£őæŤźäÿĂäÿłåůšçżŕæĽŞåijĂçŽĎæŰĞæIJňæĺąåijŔçŽĎæŰĞäżűçŽĎçijŰçăĄæŰźåijŔ
`detach()` æŰźæşŢçğżéŹďæŐĽåůšå■ŸåIJĺçŽĎæŰĞæIJňçijŰçăĄåśĆïijŇ
åźűä¡£çŤĺæŰřçŽĎçijŰçăĄæŰźåijŔäžčæŽ£ãĂĆäÿŃéĺćæŸŕäÿĂäÿłåIJÍ `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£Źæăůå ĄŽåŔŕèČ¡äijŽäÿ■æŰ■ä¡ăçŽĎçżĹçńŕïijŇè£ŹéĞŇäžĚäžĚæŸŕäÿžäžĘæijŢçd'žèĂŇåůšãĂĆ

èőÍèőž

I/Oçşżçż§çŤśäÿĂçşżåĹŮçŽĎåśĆæňąædĎåżžèĂŇæĹŘãĂĆä¡ääŔŕäžěèŕŢçİĂè£ŘèąŇäÿŃéĺćè£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'>
>>>
```

åIJÍè£Źäÿłä¿Ńå■Räÿ■ïijŇ`io.TextIOWrapper` æŸŕäÿĂäÿłçijŰçăĄåŠŇèğčçăĄU-
nicodeçŽĎæŰĞæIJňåď ĎçŘĘäśĆïijŇ `io.BufferedWriter`
æŸŕäÿĂäÿłåď ĎçŘĘäžŇè£ŽåĹűæŢŕæ■őçŽĎåÿęçijŞåĘšçŽĎI/OåśĆïijŇ `io.FileIO`
æŸŕäÿĂäÿłèąĺçď žæŞ■ä¡IJçşżçż§åžŢåśĆæŰĞäžűæŔŔè£ŕçňęçŽĎåŐ§ågŃæŰĞäžűãĂĆ
åćđåŁăæĹŰæŤźåŔŸæŰĞæIJňçijŰçăĄäijŽæűL'åŔŁåćđåŁăæĹŰæŤźåŔŸæIJĂäÿŁéĺćçŽĎ
`io.TextIOWrapper` åśĆãĂĆ

äÿĂèĹňæÍěèőšïijŇåČŔäÿŁéĺćä¿Ńå■Řè£ŹæăůéĂŽè£ĞèŐ£éŮőåśđæĂğåĂijæÍěçŻť'æŐěæŞ■ä¡IJäÿ■åŘŇç
ä¿ŃåęĆïijŇåęĆædIJä¡ăèŕŢçİĂä¡£çŤĺäÿŃéĺćè£ŹæăůçŽĎæŁĂæIJŕæŤźåŔŸçijŰçăĄçIJŃçIJŃäijŽåŔŚçŤ§äžĂä

```
>>> 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)

```
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
ValueError: I/O operation on closed file.
>>>
```

çżŞæđIJåĞžéŤŹäžĘïijŇåŻääÿžfçŽĎåŐ§ågŃåĂijåůšçżŔèćńçăť åİŔäžĘåźűåĚşéŮ■äžĘåžŢåśĆçŽĎæŰĞäž

detach() æŰźæşŢäijŽæŰ■åijĂæŰĞäžűçŽĎæIJĂéąűåśĆåźűè£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¡ăářśåŔŕäžěçżŹè£ŤåŻđçżŞæđIJæůżåŁääÿĂäÿłæŰřçŽĎæIJĂéąűå

```
>>> f = io.TextIOWrapper(b, encoding='latin-1')
>>> f
<_io.TextIOWrapper name='sample.txt' encoding='latin-1'>
>>>
```

ář¡çőąåůšçżŔåŘŚä¡ăæijŤçď žäžĘæŤźåŔŸçijŰçăĄçŽĎæŰźæşŢïijŇ
ä¡ĘæŸŕä¡ăè£ŸåŔŕäžěåĽľçŤĺè£Źçğ■æŁĂæIJŕæİěæŤźåŔŸæŰĞäžűèąŇåď ĎçŘĘãĂĄéŤŹèŕŕæIJžåĹűäžěåŔŁæ

```
>>> sys.stdout = io.TextIOWrapper(sys.stdout.detach(), encoding=
↪'ascii',
...                                errors='xmlcharrefreplace')
>>> print('Jalape\u00f1o')
Jalape&#241;o
>>>
```

æşĺæĎŔäÿŇæIJĂåŘŐè¿ŞåĞžäÿ■çŽĎéİđASCIIå■Ůçňę Ãś æŸŕåęĆä¡Ţècń ñ åŔŰäžčçŽĎãĂĆ

7.17 5.17 ářĘå■ŮèŁĆåĘŹåĚěæŰĞæIJňæŰĞäžű

éŮőécŸ

ä¡ăæČşåIJĺæŰĞæIJňæĺąåijŔæĽŞåijĂçŽĎæŰĞäžűäÿ■åĘŹåĚěåŐ§ågŃçŽĎå■ŮèŁĆæŢřæ■õãĂĆ

èğčåĘşæŰźæąĹ

åřĘå■ŮèŁĆæŢřæ■őçŻť'æŐěåĘŹåĚěæŰĞäžűçŽĎçijŞåĘšåŇžå■şåŔŕïijŇä¿ŇåęĆïijŽ

```
>>> 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£ĞèŕżåŔŰæŰĞæIJňæŰĞäžűçŽĎ `buffer` åšđæĂğæĮěèŕżåŔŰäžŇè£ŻåĹűæŢřæ■őãĂĆ

èőĺèőž

I/Oçşżçż§äžěåšĆçžğçżŞæđĎçŽĎå¡ćåijŔæđĎåżžèĂŇæĹŘãĂĆ
æŰĞæIJňæŰĞäžűæŸŕéĂŽè£ĞåIJĺäÿĂäÿłæŃěæIJL'çijŞåĘšçŽĎäžŇè£ŻåĹűæĺąåijŔæŰĞäžűäÿŁåćđåŁääÿĂäÿ
`buffer` åšđæĂğæŇĞåŘŠåŕźåžŤçŽĎåžŢåšĆæŰĞäžűãĂĆåęĆæđIJä¡ăçŻť'æŐèèő£éŮőåőĆçŽĎèŕĮåřšäijŽçżŢè

æIJňåřŔèŁĆä¿Ňå■ŘåšŢçd'žçŽĎ `sys.stdout` åŔŕèČ¡çIJŃètůæĮěæIJL'çĆźçL'źæőŁãĂĆ
éžŸèőd'æČĚåĘţäÿŃïijŇ`sys.stdout` æĂżæŸŕäžěæŰĞæIJňæĺąåijŔæL'Şåij ĂçŽĎãĂĆ
ä¡ĘæŸŕåęĆæđIJä¡ăåIJĺåĘŹäÿĂäÿłéIJĂèęĄæL'Şå■řäžŇè£ŻåĹűæŢřæ■őåĹŕæăĞåĞĘè¿ŞåĞžçŽĎèĎŽæIJňçŽĎ

7.18 5.18 åřĘæŰĞäžűæŔŔè£řçňęåŇĚèčĚæĹŘæŰĞäžűåŕźèšą

éŮőécŸ

ä¡ăæIJL'äÿĂäÿłåŕźåžŤäžŐæŞ■ä¡IJçşżçż§äÿŁäÿĂäÿłåůšæL'ŞåijĂçŽĎI/OéĂŽéĄŞ(æŕŤåęĆæŰĞäžűãĂĄç
ä¡ăæČşåřĘåőČåŇĚèčĚæĹŘäÿĂäÿłæŻť'énŸåšĆçŽĎPythonæŰĞäžűåŕźèšąãĂĆ

èğčåĘşæŰźæąĹ

äÿĂäÿłæŰĞäžűæŔŔè£řçňęåŠŇäÿĂäÿłæL'ŞåijĂçŽĎæŹőéĂŽæŰĞäžűæŸŕäÿ■äÿĂæăůçŽĎãĂĆ
æŰĞäžűæŔŔè£řçňęäžĚäžĚæŸŕäÿĂäÿłçŤśæŞ■ä¡IJçşżçż§æŇĞåőŽçŽĎæŢť'æŢřijŇçŤĺæĚæŇĞäžčæ§Řäÿłçşż
åęĆæđIJä¡ăçćřåůgæIJL'è£ŹäźĹäÿĂäÿłæŰĞäžűæŔŔè£řçňęïijŇä¡ăåŔŕäžěéĂŽè£Ğä¡£çŤĺ
`open()` åĞ¡æŢřæĚěåřĘåĚűåŇĚèčĚäÿžäÿĂäÿłPythonçŽĎæŰĞäžűåŕźèšąãĂĆ
ä¡ăäžĚäžĚåŔłéIJĂèęĄä¡£çŤĺè£ŹäÿłæŢť'æŢřåĂijçŽĎæŰĞäžűæŔŔè£řçňęä¡IJäÿžçňňäÿĂäÿłåŔĆæŢřæĚěäžčæŽ

```
# Open a low-level file descriptor
import os
fd = os.open('somefile.txt', os.O_WRONLY | os.O_CREAT)
```

(continues on next page)

```
# Turn into a proper file
f = open(fd, 'wt')
f.write('hello world\n')
f.close()
```

å¡ŞénŸåśĆçŽĎæŰĞäžűåŕźèśąèćńåĚşéŮ■æĹŰèĂĚçăť åİŔçŽĎæŮűåĂŹïijŇåžŢåśĆçŽĎæŰĞäžűæŔŕè£ì åęĆæđIJè£Źäÿłåźűäÿ■æŸŕä¡ăæČşèęĄçŽĎçżŞæđIJïijŇä¡ăåŔŕäžěçżŹ `open()` åĞ¡æŢřäijăéĂŠäÿĂäÿłåŔŕéĂĽ'çŽĎ `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£Źçğ■åŇĚèčĚæŰĞäžűæŔŕè£řçňęçŽĎæŁĂæIJŕåŔŕäžěå¿ĹæŰźä¿£çŽĎåřĘäÿĂä åęĆçőąéĄŞãĂĄåěŮæŐěå■Ůç■Ľ˜ãĂĆäÿ¿ä¿ŃæİèèőšïijŇäÿŃéİćæŸŕäÿĂäÿłæŞ■ä¡IJçőąéĄŞçŽĎä¿Ńå■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)
```

éIJÄèęĄéĞ■çĆźåijžèřČçŽĎäÿĂçĆźæŸŕïijŇäÿŁéİćçŽĎä¿Ńå■ŘäżĚäżĚæŸŕäÿžäžĘæijŤçđ'žåĘĚç¡őçŽĎ `open()` åĞ¡æŢřçŽĎäÿĂäÿłçĽźæĂğïijŇäźűäÿŤäź§åŔłéĂĆçŤĺäžŐå§žäžŐUnixçŽĎçşżçż§ãĂĆ

åęĆæđIJä¡ăæČşåřĘäÿĂäÿłçśżæŰĞäžűæŐěåŔčä¡IJçŤĺåIJĺäÿĂäÿłåěŮæŐěå■Ůåźűåÿ
makefile() æŰźæşŢãĂĆ ä¡ĘæŸŕåęĆæđIJäÿ■èĂČèŹŚåŔŕçğżæd'■æĂğçŽĎèŕĺïijŇéĆčäÿŁéİćçŽĎèğčåĘşæ
makefile() æĂğèČ¡æŻt'åě¡äÿĂçĆźãĂĆ

ä¡ääź§åŔŕäžěä¡£çŤĺè£ŹçğżæŁĂæIJŕæİěædĎéĂääÿĂäÿłåĹńåŘ■ïijŇåĚĄèőÿäžěäÿ■åŘŇäžŐçňňäÿĂæňąa
ä¿ŇåęĆïijŇäÿŇéİćæijŢçd'žåęĆä¡ŢåĹŻåžžäÿĂäÿłæŰĞäžűåŕźèśąïijŇåőČåĚĄèőÿä¡äè¿ŞåĞžäžŇè£ŻåĹűæŢřæ

```
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()
```

åř¡çőąåŔŕäžěåřĘäÿĂäÿłåůšå■ŸåIJĺçŽĎæŰĞäžűæŔŔè£řçňęåŇĚèčĚæĹŘäÿĂäÿłæ■čåÿÿçŽĎæŰĞäžűåŕźè
ä¡ĘæŸŕèęĄæşĺæĎŔçŽĎæŸŕåźűäÿ■æŸŕæL'ĂæIJL'çŽĎæŰĞäžűæĺąåijŔéČ¡èćńæŤŕæŇĄïijŇåźűäÿŤæ§ŘäžŻçš
(çL'źåĹńæŸŕæűL'åŘŁåĹřéŤŹèŕŕåď'ĎçŘĘãĂĄæŰĞäžűçżŞář¿æİąäžűç■L'ç■L'çŽĎæŮűåĂŹ)ãĂĆ
åIJĺäÿ■åŘŇçŽĎæŞ■ä¡IJçşżçż§äÿŁè£ŹçğżèąŇäÿžäź§œŸŕäÿ■äÿĂæăůïijŇçL'źåĹńçŽĎïijŇäÿŁéİćçŽĎä¿Ňå■Ře
æĹŠèŕť'äžĘè£ŹäźĹåď'ŻïijŇæĎŔæĂİåřśæŸŕèől'ä¡ääĚåĹĘæţŃèŕŢèĞłåůśçŽĎåőđçŐŕäžčçăĄïijŇçąőä£İåőČèč

7.19 5.19 åĹŻåžžäÿť'æŮűæŰĞäžűåŠŇæŰĞäžűåď'ź

éŮőécŸ

ä¡ăéIJĂèęĄåIJĺçĺŃåžŔæL'ğèąŇæŮűåĹŻåžžäÿĂäÿłäÿť'æŮűæŰĞäžűæĹŰçŻőå¡ŢïijŇåźűåÿŇæIJŻä¡£çŤĺå

èğčåĘşæŰźæąĹ

tempfile æĺąåİŮäÿ■æIJL'å¿Ĺåď'ŽçŽĎåĞ¡æŢřåŕäžěåőŇæĹŘè£ŹäžžåŁąãĂĆ
äÿžäžĘåĹŻåžžäÿĂäÿłåŇ£åŘ■çŽĎäÿť'æŮűæŰĞäžűïijŇåŔŕäžěä¡£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
```

æĹŰèĂĚïijŇåęĆæđIJä¡ăåŰIJæňćïijŇä¡ăè£ŸåŔŕäžěåČŔè£Źæăůä¡£çŤĺäÿť'æŮűæŰĞäžűïijŽ

```
f = TemporaryFile('w+t')
# Use the temporary file
...
f.close()
# File is destroyed
```

TemporaryFile() çŽĎçňňäÿĂäÿłåŔĆæŢřæŸŕæŰĞäžűæĺąåijŔïijŇéĂŽåÿÿæİěèőšæŰĞæIJňæĺąåijŔä
w+t ïijŇäžŇè£ŻåĹűæĺąåijŔä¡£çŤĺ w+b ãĂĆ è£ŹäÿłæĺąåijŔåŘŇæŮűæŤŕæŇĄèŕžåŠŇåĘŹæŞ■ä¡IJïijŇåIJĺè£Ź
TemporaryFile() åŔęåď Úè£ŸæŤŕæŇĄèů§åĘĚç¡őçŽĎ open()
åĞ¡æŢřäÿĂæăůçŽĎåŔĆæŢřãĂĆæŕŤåęĆïijŽ

```
with TemporaryFile('w+t', encoding='utf-8', errors='ignore') as f:
    ...
```

åIJĺåď ğåď ŽæŢřUnixçşżçż§äÿŁïijŇéĂŽè£Ğ TemporaryFile()
åĹŻåżžçŽĎæŰĞäžűéČ¡æŸŕåŇ£åŘ■çŽĎïijŇçŤŽèĞşè£đçŻőå¡ŢéČ¡æšąæIJL’ãĂĆ
åęĆæđIJä¡ăæČşæL’Şçăť è£ŹäÿłéŹŔåĹűïijŇåŔŕäżěä¡£çŤĺ NamedTemporaryFile()
æİěäžčæŻ£ãĂĆæŕŤåęĆïijŽ

```
from tempfile import NamedTemporaryFile

with NamedTemporaryFile('w+t') as f:
    print('filename is:', f.name)
    ...

# File automatically destroyed
```

è£ŹéĞŇïijŇèćńæL’ŞåijĂæŰĞäžűçŽĎ f.name åšđæĂğåŇĚåŘńäžĘèŕěäÿť æŮűæŰĞäžűçŽĎæŰĞäžűåŘ
å¡Şä¡ăéIJĂèęĄåřĘæŰĞäžűåŘ■äijăéĂŠçżŹåĚűäžŰäžčçĂąæİěæL’ŞåijĂè£ŹäÿłæŰĞäžűçŽĎæŰűåĂŹïijŇè£Źä
åŠŇ TemporaryFile() äÿĂæăůïijŇçżŞæđIJæŰĞäžűåĚşéŮ■æŮűäijŽèćńèĞłåŁĺåĹăéŹđ’æŐL’ãĂĆ
åęĆæđIJä¡ăäÿ■æČşè£ŹäźĹåĄŽïijŇåŔŕäžěäijăéĂŠäÿĂäÿłåĚşéŤőå■ŮåŔĆæŢř
delete=False å■şåŔŕãĂĆæŕŤåęĆïijŽ

```
with NamedTemporaryFile('w+t', delete=False) as f:
    print('filename is:', f.name)
    ...
```

äÿžäžĘåĹŻåżžäÿĂäÿłäÿť æŮűçŻőå¡ŢïijŇåŔŕäžěä¡£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 åžŤèřěæŸřåď ĎçŘĘäÿť æŮűæŰĞäžűçŽőå¡ŢçŽĎæIJĂçőĂå■ŢçŽĎæŰź åIJÍäÿĂäÿłæŻťä¡ŐçŽĎçžğåĹńïijŇä¡ååŔŕäžěä¡£çŤÍ mkstemp() åŠŇ mkdtemp() æÌěåĹŻåžžäÿť æŮűæŰĞäžűåŠŇçŽőå¡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¿ŇåęĆïijŇåĞ¡æŢř mkstemp() äżĚäžĚåřśè£ŤåŻđäÿĂäÿłåŐ§ågŇçŽĎOSæŰĞäžűæŔŔè£řçňęïijŇä¡ãéIJĂèę åŘŇæăůä¡ãè£ŸéIJĂèęĄèĞłåůśæÿĚçŘĘè£ŹäžŻæŰĞäžűãĂĆ

éĂŽåÿÿæÌěèőšïijŇäÿť æŮűæŰĞäžűåIJÍçşżçż§éżŸèőď çŽĎä¡■ç¡őèćńåĹŻåžžïijŇæŕŤåęĆ /var/tmp æĹŰçśżäijijçŽĎåIJřæŰźãĂĆ äÿžäžĘèŐůåŔŰçIJ§åőđçŽĎä¡■ç¡őïijŇåŔŕäžěä¡£çŤÍ tempfile.gettempdir() åĞ¡æŢřãĂĆæŕŤåęĆïijŽ

```
>>> tempfile.gettempdir()
'/var/folders/7W/7WZl5sfZEF0pljrEB1UMWE+++TI/-Tmp-'
>>>
```

æĽĂæIJL'åŠŇäÿť æŮűæŰĞäžűçŻÿåĚşçŽĎåĞ¡æŢřéČ¡åĚĄèőÿä¡ăéĂŽè£Ğä¡£çŤÍåĚşéŤőå■ŮåŔĆæŢř prefix ãĂĄsuffix åŠŇ dir æÌěèĞłåőŽäźĽçŽőå¡ŢäžěåŔŁåŚ¡åŘ■èğĎåĹŹãĂĆæŕŤåęĆïijŽ

```
>>> f = NamedTemporaryFile(prefix='mytemp', suffix='.txt', dir='/tmp
↪')
>>> f.name
'/tmp/mytemp8ee899.txt'
>>>
```

æIJĂåŘŐè£ŸæIJL'äÿĂçĆźïijŇår¡åŔŕèČ¡äžěæIJĂåőĽ'åĚÍçŽĎæŰźåijŔä¡£çŤÍ tempfile æÍąåİŮæÌěåĹŻåžžäÿť æŮűæŰĞäžűãĂĆ åŇĚæŃňäžĚçżŹå¡ŞåĽ'■çŤÍæĹůæŐĹæİČèő£éŮőäžěåŔŁåIJÍæŰĞäžű èęĄæşÍæĎŔçŽĎæŸŕäÿ■åŘŇçŽĎåźşåŔŕåŔŕèČ¡äijŽäÿ■äÿĂæăůãĂĆåŽăæ■ď'ä¡ăæIJĂåě¡éŸĚèŕż åőŸæŰźæŰĞæąč æÌěäžĘèğčæŻť åď ŽçŽĎçżĘèŁĆãĂĆ

7.20 5.20 äÿŐäÿšèąŇçńŕåŔčçŽĎæŢřæ■őéĂŽä£ą

éŮőécŸ

ä¡ăæČşéĂŽè£ĞäÿšèąŇçńŕåŔčèŕżåĘŹæŢřæ■őïijŇåĚÿåđŃåIJžæŹŕåřśæŸŕåŠŇäÿĂäžŽçąňäžűěő¿åď ĞæĽ'Ş

èğčåĘşæŰźæąĹ

åř¡çőąä¡ääŔŕäžěéĂŽè£Ğä¡£çŤÍPythonåĘĚç¡őçŽĎI/OæÍąåİŮæİěåőŇæĹŘè£ŹäÿłäżżåŁąïijŇä¡Ęåŕźäž
pySerialåŇĚ ãĂĆ è£ŹäÿłåŇĚçŽĎä¡£çŤÍéİđåÿÿçőĂå■ŢïijŇåĚĹåőL'èčĚpySerialïijŇä¡£çŤÍçśžäijijäÿŃéİćè£Źæ

```
import serial
ser = serial.Serial('/dev/tty.usbmodem641', # Device name varies
                    baudrate=9600,
                    bytesize=8,
                    parity='N',
                    stopbits=1)
```

èő¿åd'ĞåŘ■åŕźäžŐäÿ■åŘŇçŽĎèő¿åd'ĞåŠŇæŞ■ä¡IJçşżçż§æŸŕäÿ■äÿĂæăůçŽĎãĂĆ
æŕŤåęĆïijŇåIJÍWindowsçşżçż§äÿŁïijŇä¡ääŔŕäžěä¡£çŤÍ0, 1ç■L'èąÍçd'žçŽĎäÿĂäÿłèő¿åd'ĞæİěæL'ŞåijĂéĂŽä
äÿĂæŮęçńŕåŔčæL'ŞåijĂïijŇéĆčåŕšåŔŕäžěä¡£çŤÍ `read()`ïijŇ`readline()` åŠŇ `write()`
åĞ¡æŢřèŕżåĘŹæŢřæ■őäžĘãĂĆä¿ŃåęĆïijŽ

```
ser.write(b'G1 X50 Y50\r\n')
resp = ser.readline()
```

åd'ğåd'ŽæŢřæČĚåĘţäÿŃïijŇçőĂå■ŢçŽĎäÿšåŔčéĂŽä£ąäžŐæ■d'åŔŸå¿Ůå■ĄåĹĘçőĂå■ŢãĂĆ

èőÍèőž

åř¡çőąèąÍéİćäÿŁçIJŃèţůæİěå¿ĹçőĂå■ŢïijŇåĚűåődäÿšåŔčéĂŽä£ąæIJL'æŮűåĂŹäź§æŸŕæŇžéžżçČęçŽĎ
æŐÍè■Řä¡ää¡£çŤÍçňňäÿL'æŰźåŇĚåęĆ `pySerial` çŽĎäÿĂäÿłåŐ§åŽăæŸŕåőČæŔŘä¿ŻäžĘåŕźéńŸçžğçL'źæĂ
(æŕŤåęĆèűĚæŮűïijŇæŐğåĹűæţĄïijŇçijŞåĘšåŇžåĹůæŰřïijŇæŔąæL'Ńå■Řèőőç■L'ç■L')ãĂĆäÿ¿äÿłä¿Ńå■Řïij
`RTS-CTS` æŔąæL'Ńå■Řèőőïij Ň ä¡ääŔłéIJĂèęĄçžŹ `Serial()` äijåéĂŠäÿĂäÿł
`rtscts=True` çŽĎåŔĆæŢřå■şåŔŕãĂĆ åĚűåőŸæŰźæŰĞæąčéİđåÿÿåőŇåŰĎïijŇåŽăæ■d'æĹŚåIJÍè£ŹéĞŇa

æŮűåĹżèőřä¡ŔæL'ĂæIJL'æűL'åŔŁåĹřäÿšåŔčçŽĎI/OéČ¡æŸŕäžŇè£ŽåĹűæÍąåijŔçŽĎãĂĆåŽăæ■d'ïijŇçą
(æĹŰæIJL'æŮűåĂŹæL'ğèąŇæŰĞæIJñçŽĎçijŰçăĄ/èğčçăĄæŞ■ä¡IJ)ãĂĆ
åŔęåd'Űå¡Şä¡äéIJĂèęĄåĹŽåžžäžŇè£ŽåĹűçijŰçăĄçŽĎæŇĞäžd'æĹŰæŢřæ■őåŇĚçŽĎæŮűåĂŹïijŇ`struct`
æÍąåİŮäź§æŸŕéİđåÿÿæIJL'çŤÍçŽĎãĂĆ

7.21 5.21 åžŔåĹŮåŇŰPythonåŕźèśą

éŮőécŸ

ä¡äéIJĂèęĄåřĘäÿĂäÿłPythonåŕźèśąåžŔåĹŮåŇŰäÿžäÿĂäÿłå■ŮèŁĆæţĄïijŇäžëä¿£åřĘåőČä£Íå■ŸåĹřäÿĂ

èğčåĘşæŰźæąĹ

åŕźäžŐåžŔåĹŮåŇŰæIJĂæŹőéĄ■çŽĎåĄŽæşŢåŕśæŸŕä¡£çŤÍ `pickle`
æÍąåİŮãĂĆäÿžäžĘåřĘäÿĂäÿłåŕźèśąä£Íå■ŸåĹřäÿĂäÿłæŰĞäžűäÿ■ïijŇåŔŕäžěè£ŹæăůåĄŽïijŽ

```
import pickle

data = ... # Some Python object
f = open('somefile', 'wb')
pickle.dump(data, f)
```

äÿžäžĘåřĘäÿĂäÿłåŕźèśąè¡ňåĆĺäÿžäÿĂäÿłå■ŮçňęäÿšïijŇåŔŕäżěä¡£çŤĺ `pickle.dumps()` ïijŽ

```
s = pickle.dumps(data)
```

äÿžäžĘäżŐå■ŮèŁĆæţĄäÿ■æĄćåd'■äÿĂäÿłåŕźèśąïijŇä¡£çŤĺ `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)
```

èőĺèőž

åŕźäžŐåd'ğåd'ŽæŢřåžŤçŤĺçĺŃåžŔæİěèőšïijŇdump() åŠŇ load() åĞ¡æŢřçŽĎä¡£çŤĺåršæŸŕä¡ăæIJL'æŢĹä¡£çŤĺ `pickle` æĺąåİŮæĽ'ĂéIJĂçŽĎåĚíéČĺäžĘãĂĆ åőČåŔŕéĂĆçŤĺäžŐçżİåd'ğéČĺåĹĘPythonæŢřæ■őçśžåđŃåŠŇçŤĺæĹůèĞłåőŽäźĽ'çśżçŽĎåŕźèśąåőđä¿ŃãĂĆ åęĆæđIJä¡ăçćřåĹřæ§Řäÿłåžşåŕźäżěèől'ä¡ăåIJĺæŢřæ■őåžŞäÿ■ä‡İå■Ÿ/æĄćåd'■PythonåŕźèśąæĹŰèĂĚæŸŕéĂŽ éĆčäźĹå¿ĹæIJL'åŔŕèČ¡è£ŹäÿłåžŞçŽĎåžŢåśĆårśä¡£çŤĺäžĘ `pickle` æĺąåİŮãĂĆ

`pickle` æŸŕäÿĂçğ■PythonçL'źæIJL'çŽĎèĞłæŔŔè£řçŽĎæŢřæ■őçijŰçăĄãĂĆ éĂŽè£ĞèĞłæŔŔè£řïijŇèćńåžŔåĹŮåŇŰåŘŐçŽĎæŢřæ■őåŇĚåŘńæŕŔäÿłåŕźèśąåijĂåğŃåŠŇçżŞæİ§äžěåŔŁåć åŻăæ■d'ïijŇä¡ăæŮăéIJĂæŃĚå£Čåŕźèśąèőřå¡ŢçŽĎåőŽäźL'ïijŇåőČæĂżæŸŕèČ¡åůë¡IJãĂĆ äÿ¿äÿłä¿Ńå■ŘïijŇåęĆæđIJèęĄåd'ĎçŘĘåd'ŽäÿłåŕźèśąïijŇä¡ääŔŕäżěè£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¡åžŔåĹŮåŇŰåĞ¡æŢřïijŇçśżïijŇè£ŸæIJL’æŐěåŔčïijŇä¡ĘæŸŕçżŞæđIJæŢřæ■őäżĚäżĚářĘåőČä

```
>>> import math
>>> import pickle.
>>> pickle.dumps(math.cos)
b'\x80\x03cmath\ncos\nq\x00.'
>>>
```

å¡ŞæŢřæ■őåŔ■åžŔåĹŮåŇŰåŻđæİěçŽĎæŮűåĂŹïijŇäijŽåĚĹåĄĞåőŽæL’ĂæIJL’çŽĎæžŘæŢřæ■őæŮűåŀ
æĺąåİŮãĂĄçśżåŠŇåĞ¡æŢřäijŽèĞłåŁĺæŇL’éIJĂåŕijåĚèè£ŻæİěãĂĆåŕźäžŐPythonæŢřæ■őèćńäÿ■åŘŇæIJžåŹĺ
æŢřæ■őçŽĎä£Ía■ŸåŔŕèČ¡äijŽæIJL’éŮőécŸïijŇåŻääÿžæL’ĂæIJL’çŽĎæIJžåŹĺéČ¡å£Ëéążèő£éŮőåŘŇäÿĂäÿł

æşÍ

```
å■ČäÿĞäÿ■èęĄåŕźäÿ■ä£ąäżżçŽĎæŢřæ■őä¡£çŤÍpickle.load()ãĂĆ
pickleåIJÍåŁăè¡¡æŮűæIJL’äÿĂäÿłåL’ŕä¡IJçŤÍåřśæŸŕåőČäijŽèĞłåŁÍåŁăè¡¡çŻÿåžŤæÍąåİŮåź
ä¡ĘæŸŕæ§ŘäÿłåİŕäžžåęĆæđIJç§ĚéĄ§pickleçŽĎåůëä¡IJåŐ§çŘĘïijŇ
äżŰåřśåŔŕäżěåĹżåżžäÿĂäÿłæĄűæĎŔçŽĎæŢřæ■őåŕijèĞt’PythonæL’ğèąŇéŽŔæĎŔæŇĞåőŽçŽĎçşżçż
åŽăæ■d’ïijŇäÿĂåőŽèęĄä£ÌèŕĄpickleåŘłåIJÍçŻÿäžŠäźŃéŮt’åŔŕäżěèőd’èŕĄåŕźæŰźçŽĎèğčæđŘ
```

æIJL’äžŻçśżåđŃçŽĎåŕźèśąæŸŕäÿ■èČ¡èćńåžŔåĹŮåŇŰçŽĎãĂĆè£ŹäžŻéĂŽåÿÿæŸŕéĆčäžŻä¿ÌèţŰåđ’Űé
æŕŤåęĆæL’ŞåijĂçŽĎæŰĞäżűïijŇç¡ŚçżIJè£đæŐěïijŇçž£çÍŃïijŇè£ŻçÍŃïijŇæăĹåÿğç■L’ç■L’ãĂĆ
çŤÍæĹůèĞłåőŽäźL’çśżåŔŕäżěéĂŽè£ĞæŔŘä¿Ż __getstate__()
åŠŇ __setstate__() æŰźæşŢæİěçżŢè£Ğè£ŹäžŻéŹŘåĹűãĂĆ
åęĆæđIJåőŽäźL’äžĘè£Źäÿd’äÿłæŰźæşŢïijŇpickle.dump()
åřśäijŽèřČçŤÍ __getstate__() èŐůåŔŰåžŔåĹŮåŇŰçŽĎåŕźèśąãĂĆ
çśżäijijçŽĎïijŇ__setstate__() åIJÍåŔ■åžŔåĹŮåŇŰæŮűèćńèřČçŤÍãĂĆäÿžäžĘæijŢçd’žè£Źäÿłåůëä¡IJåĆ
äÿŇéİćæŸŕäÿĂäÿłåIJÍåĘĚéČÍåőŽäźL’äžĘäÿĂäÿłçž£çÍŃä¡Ęäż■çĎűåŔŕäżěåžŔåĹŮåŇŰåŠŇåŔ■åžŔåĹŮåŇŰç

```
# 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£Řèą Ñäÿ Ńéİćç ŽĎåžŔåĹŮåŇŰèŕŢéłÑäžčçăĄïijŽ

```
>>> 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■èŕŢéłÑäÿŃïijŽ

```
>>> f = open('cstate.p', 'rb')
>>> pickle.load(f)
countdown.Countdown object at 0x10069e2d0>
T-minus 19
T-minus 18
...
```

ä¡ăåŔŕäžěçIJŃåĹřçž£çÍŃåŔĹåěĞè£źèĹńçŽĎéĞ■çŤ§äžĘïijÑäžŐä¡ăçńñäÿĂæňąåžŔåĹŮåŇŰåőČçŽĎåIJ

pickle åŕźäžŐåd'ğåđŃçŽĎæŢřæ■őçżŞæđĎæŕŤåęĆä¡£çŤÍ array æĹŰ numpy æĺąåİŮåĹŻåžžçŽĎäžŇè£ŻåĹűæŢřçżĎæŢĹçŐĞåźűäÿ■æŸŕäÿĂäÿłéńŸæŢĹçŽĎçijŰçăĄæŰźåijŔãĂĆ åęĆæđIJä¡ăéIJĂèęĄçğżåŁĺåd'ğéĞŔçŽĎæŢřçżĎæŢřæ■őïijÑä¡ăæIJĂåě¡æŸŕåĚĹåIJĺäÿĂäÿłæŰĞäźűäÿ■åřĘåĚ (éIJĂèęĄçńñäÿL'æŰźåžŞçŽĎæŤŕæŇĄ)ãĂĆ

çŤśäžŐ pickle æŸŕPythonçL'źæIJL'çŽĎåźűäÿŤéŹĎçİĂåIJĺæžŘçăĄäÿŁïijŇæL'ĂæIJL'åęĆæđIJéIJĂèę ä¿ŃåęĆïijŇåęĆæđIJæžŘçăĄåŔŸåŁĺäžĘïijÑä¡ăæL'ĂæIJL'çŽĎå■ŸåĆĺæŢřæ■őåŔŕèČ¡äijŽèćńçăt'åİŔåźűäÿŤåŔ åİęçŹ¡æİèèőšïijŇåŕźäžŐåIJĺæŢřæ■őåžŞåŠŇå■ŸæąćæŰĞäźűäÿ■å■ŸåĆĺæŢřæ■őæŮűïijÑä¡ăæIJĂåě¡ä¡£çŤĺæŻ è£ŹäžŽçijŰçăĄæäijåijŔæŻt'æăĞåĞĘïijŇåŔŕäžèèćńäÿ■åŘŇçŽĎèŕ■èĺĂæŤŕæŇĄïijŇåźűäÿŤäź§èČ¡å¿Ĺåě¡çŽĎ

æIJĂåŘŐäÿĂçĆźèęĄæşĺæĐŘçŽĎæŸŕ pickle æIJL'åd'ğéĞŔçŽĎéĚ■ç¡őéĂL'éąźåŠŇäÿĂäžŽæčŸæL'Ń åŕźäžŐæIJĂåÿÿèğĄçŽĎä¡£çŤĺåIJžæŹŕïijÑä¡ăäÿ■éIJĂèęĄåŐżæŃĚå£Čè£ŹäÿłïijÑä¡ĘæŸŕåęĆæđIJä¡ăèęĄåIJĺ æIJĂåě¡åŐżæ§ěéŸĚäÿĂäÿŃ åőŸæŰźæŰĞæąč ãĂĆ

8 çńñåĚ■çńăïijŽæŢřæ■őçijŰçăĄåŠŇåd'ĎçŘĘ

è£ŹäÿĂçńăäÿżèęĄèőĺèőžä¡£çŤĺPythonåd'ĎçŘĘåŘĎçğ■äÿ■åŘŇæŰźåijŔçijŰçăĄçŽĎæŢřæ■őïijŇærŤåę åŠŇæŢřæ■őçżŞæđĎéĆčäÿĂçńăäÿ■åŘŇçŽĎæŸŕïijŇè£Źçńăäÿ■äijŽèőĺèőžçL'źæőŁçŽĎçőŮæşŢéŮőécŸïijŇè

Contents:

8.1 6.1 èŕžåĘŹCSVæŢřæ■ő

éŮőécŸ

ä¡ăæČşèŕžåĘŹäÿĂäÿłCSVæăijåijŔçŽĎæŰĞäžűãĂĆ

èğčåĘşæŰźæąĹ

åŕźäžŐåď˘ğåď˘ŽæŢřçŽĎCSVæăijåijŔçŽĎæŢřæ■őèŕžåĘŹéŮőécŸïijŇéČ¡åŔŕäžěä¡£çŤÍ csv åžŞãĂĆ ä¿ŇåęĆïijŽåĄĞèő¿ä¡ăåIJÍäÿĂäÿłåŘ■åŔństocks.csvæŰĞäžűäÿ■æIJL˘äÿĂäžŻèĆąçĕÍåÿĆåIJžæŢï

```
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
```

äÿŇéIćåŘŚä¡ăåśŢçď˘žåęĆä¡ŢåřĘè£ŹäžŻæŢřæ■őèŕžåŔŰäÿžäÿĂäÿłåĚČçżĎçŽĎåžŔåĹŮïijŽ

```
import csv
with open('stocks.csv') as f:
    f_csv = csv.reader(f)
    headers = next(f_csv)
    for row in f_csv:
        # Process row
        ...
```

åIJÍäÿŁéIćçŽĎäžčçăĄäÿ■ïijŇ `row` äijŽæŸŕäÿĂäÿłåĹŮèąÍãĂĆåŽăæ■d˘ïijŇäÿžäžĘè©£éŮőæ§Řäÿłå■Ůæő `row[0]` è©£éŮőSymbolïijŇ `row[4]` è©£éŮőChangeãĂĆ

çŤśäžŐè£Źçğ■äÿŇæăĞèő£éŮőéĂŽåÿÿäijŽåijŢètţæůůæůĘïijŇä¡ăåŔŕäžěèĂČèŹŚä¡£çŤÍłåŚ¡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¡£çŤÍłåĹŮåŘ■åęĆ `row.Symbol` åŠŇ `row.Change` äžčæŽ£äÿŇæăĞèő£éŮőãĂĆ éIJĂèęĄæşÍæĎŔçŽĎæŸŕè£ŹäÿłåŔłæIJL˘åIJÍłåĹŮåŘ■æŸŕåŘĹæşŢçŽĎPythonæă ä¡ăåŔŕèČ¡éIJĂèęĄä£őæŤźäÿŇåŐ§ågŇçŽĎåĹŮåŘ■(åęĆårĘéIđæăĞèŕEçňęå■ŮçňęæŽ£æ■ćæĹŘäÿŇåĹŠçž£äź

åŔęåď˘ŰäÿĂäÿłéĂL˘æŃl˘åřśæŸŕårĘæŢřæ■őèŕžåŔŰåĹřäÿĂäÿłå■ŮåĚÿåžŔåĹŮäÿ■åŐžãĂĆåŔŕäžěè£Źæă

```
import csv
with open('stocks.csv') as f:
    f_csv = csv.DictReader(f)
    for row in f_csv:
        # process row
        ...
```

åIJÍè£ŹäÿłçL'ĹæIJñäÿ■ïijÑä¡ăåŔŕäżěä¡£çŤĺåĹŮåŘ■åŐżèő£éŮőæŕŔäÿĂèąÑçŽĎæŢřæ■őäžĘãĂĆæŕŤåęĆæĹŰèĂĚ `row['Change']`

äÿžäžĘåĘŹåĚěCSVæŢřæ■őïijÑä¡ăäż■çĎűåŔŕäżěä¡£çŤĺcsvæĺąåİŮïijÑäÿ■è£Ğè£ŹæŮűåĂŹåĚĹåĹŻåżžäÿ `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)
```

åęĆædIJä¡ăæIJL'äÿĂäÿłå■ŮåĚÿåžŔåĹŮçŽĎæŢřæ■őïijÑåŔŕäżěåČŔè£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)
```

èőÍèőž

ä¡ăåžŤèŕěæĂżæŸŕäijŸåĚĹéĂL'æŃl'csvæĺąåİŮåĹĘåL'šæĹŰèğčædŘCSVæŢřæ■õăĂĆä¿ŃåęĆïijÑä¡ăåŔŕ

```
with open('stocks.csv') as f:
for line in f:
    row = line.split(',')
    # process row
    ...
```

ä¡£çŤÍè£Źçğ■æŰźåijŔçŽĎäÿĂäÿłçijžçĆźåřśæŸŕä¡ăäż■çĎűéIJĂèęĄåŐžåď ĎçŘĘäÿĂäžŻæčŸæĽŃçŽĎç
æŕŤåęĆïijŇåęĆæđIJæ§ŘäžŻå■Ůæőţå Äijèćńåij ŢåŔůåŇĚåŻť ïijŇä¡ăäÿ■å¿Ůäÿ■åŐžéŹď è£ŹäžŻåijŢåŔůãĂĆ
åŘęåď ŰïijŇåęĆæđIJäÿĂäÿłèćńåijŢåŔůåŇĚåŻť çŽĎå■Ůæőţçćřåůğå ŘńæIJL äÿĂäÿłéĂŮåŔůïijŇéĆčäźĹçÍŃåž

éżŸèőď æČĚåĘţäÿŃïijŇcsv åžŞåŔŕèŕĘåĹńMicrosoft Ex-
celæĽĂä¡£çŤÍçŽĎCSVçijŰçăĄèğĎåĹŹãĂĆ è£ŹæĹŰèőÿäźş§æŸŕæIJĂåÿÿèğĄçŽĎå¡ćåijŔïijŇåźűäÿŤäź§äijŽ
çĎűèĂŇïijŇåęĆæđIJä¡ăæ§ěçIJŃcsvçŽĎæŰĞæąčïijŇåřśäijŽåŔŚçŐřæIJL å¿Ĺåď Žçğ■æŰźæşŢåřĘåőČåžŤçŤÍ
ä¿ŃåęĆïijŇåęĆæđIJä¡ăæČşèŕžåŔŰäžětabåĹĘåĽšçŽĎæŢŕæ■őïijŇåŔŕäžěè£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¡ăæ■čåIJÍèŕžåŔŰCSVæŢŕæ■őåźűåřĘåőČäžňè¡ňæ■ćäÿžåŚ¡åŘ■åĚČçżĎïijŇéIJĂèęĄæşĺæĎŔåŕź
ä¿ŃåęĆïijŇäÿĂäÿłCSVæăijåijŔæŰĞäžűæIJL äÿĂäÿłåŇĚåŘńéÍđæşŢæăĞèŕĘçňęçŽĎåĹŮåď ť èąŇïijŇçśžäijijž

```
StreetÂăAddress,Num-Premises,Latitude,Longitude 5412ÂăNÂăCLARK,10,
↪41.980262,-87.668452
```

è£ŹæăůæIJĂçżĹäijŽåŕijèĞť åIJÍåĹŻåžžäÿĂäÿłåŚ¡åŘ■åĚČçżĎæŮűäžğçŤ§äÿĂäÿł
ValueError åijĆåÿÿèĂŇåď śèť ěãĂĆ äÿžäžĘèğčåĘşè£ŹéŮőécŸïijŇä¡ăåŔŕèČ¡äÿ■å¿Ůäÿ■åĚĹåŐžä£őæ■čå
ä¿ŃåęĆïijŇåŘŕäžěåČŔäÿŃéÍćè£ŹæăůåIJÍéÍđæşŢæăĞèŕĘçňęäÿŁä¡£çŤÍäÿĂäÿłæ■čåĹŹèąÍè¿¿åijŔæŻ£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£ŸæIJL éĞ■èęĄçŽĎäÿĂçĆźéIJĂèęĄåijžèřČçŽĎæŸŕïijŇcsväžğçŤ§çŽĎæŢŕæ■őéČ¡æŸŕå■Ůçňęäÿšçśžå
åęĆæđIJä¡ăéIJĂèęĄåĄŽè£ŹæăůçŽĎçśžåđŃè¡ňæ■ćïijŇä¡ăå£Ěéążè ĞłåůśæĽŃåŁĺåŐžåőđçŐřăĂĆ
äÿŃéÍćæŸŕäÿĂäÿłåIJÍCSVæŢŕæ■őäÿŁæĽğèąŇåĚűäžŰçśžåđŃè¡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 ŰïijŇäÿŃéÍćæŸŕäÿĂäÿłè¡ňæ■ćå■ŮåĚÿäÿ■çĽźőŽå■ŮæőţçŽĎä¿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)
```

éĂŽåÿÿæÌèèőšïijŇä¡ăåŔŕèČ¡åźűäÿ■æČşè£ĞåďŽåŐżèĂČèŹŚè£ŹäžŽè¡ňæ■ćéŮőécŸãĂĆ
åIJĺåőđéŹĚæČĚåĘţäÿ■ïijŇCSVæŰĞäžűéČ¡æĹŰåďŽæĹŰåřŚæIJL'äžŽçijžåď'śçŽĎæŢřæ■őïijŇèćńçăt'åİŔçŽĎ
åŽäæ■d'ïijŇéŹd'éİđä¡ăçŽĎæŢřæ■őçąőåőđæIJL'ä£ÌéŽIJæŸŕåĞĘçąőæŮăèŕŕçŽĎïijŇåŘęåĹŹä¡åå£ĚéążèĂČèŹ

æIJĂåŘŐïijŇåęĆæđIJä¡ăèŕżåŔŰCSVæŢřæ■őçŽĎçŻőçŽĎæŸŕåĄŽæŢřæ■őåĹĘæđŘåŠŇçż§èőąçŽĎèŕİïij
ä¡ăåŔŕèČ¡éIJĂèęĄçIJŃäÿĂçIJŃ `Pandas` åŇĚãĂĆPandas
åŇĚåŘńäžĘäÿĂäÿłéİđåÿÿæŰźä¿£çŽĎåĞ¡æŢřåŔń `pandas.read_csv()`
ïijŇ åőČåŔŕäžěåŁăè¡¡CSVæŢřæ■őåĹřäÿĂäÿł `DataFrame` åŕźèśąäÿ■åŐżãĂĆ
çĎűåŘŐåĹl'çŤÍè£Źäÿłåŕźèśąä¡ăåřśåŔŕäžěçŤ§æĹŘåŘĎçğ■å¡ćåijŔçŽĎçż§èőąãĂĄè£ĞæżđæŢřæ■őäžěåŘŁæI
åIJĺ6.13åřŔèŁĆäÿ■äijŽæIJL'è£ŹæăůäÿĂäÿłä¿Ńå■ŘãĂĆ

8.2 6.2 èŕżåĘŹJSONæŢřæ■ő

éŮőécŸ

ä¡ăæČşèŕżåĘŹJSON(JavaScript Object Notation)çijŰçăĄæäijåijŔçŽĎæŢřæ■õãĂĆ

èğčåĘşæŰźæąĹ

`json` æĺąåİŮæŔŘä¿ŻäžĘäÿĂçğ■å¿ĹçőĂå■ŢçŽĎæŰźåijŔæİěçijŰçăĄåŠŇèğčçăĄJSONæŢřæ■õãĂĆ
åĚűäÿ■äÿđ'äÿłäÿżèęĄçŽĎåĞ¡æŢřæŸŕ `json.dumps()` åŠŇ `json.loads()`
ïijŇ èęĄæŕŤåĚűäžŰåžŔåĹŮåŇŰåĞ¡æŢřåžŞåęCpickleçŽĎæŐěåŘčåřŚå¿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■őçżŞæđÌ

```
data = json.loads(json_str)
```

åęĆæđIJä¡ăèęĄåď ĎçŘĘçŽĎæŸŕæŰĞäżűèĂŇäÿ■æŸŕå■Ůçňęäÿšïij Ňä¡ăåŔŕäżĕä¡£çŤÍ json.dump() åŠŇ json.load() æĬěçijŰçăĄåŠŇèğčçăĄJSONæŢřæ■őãĂĆä¿Ňåę Ćïij Ž

```
# 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)
```

èőÍèőž

JSONçijŰçăĄæŤŕæŇĄçŽĎå§žæIJňæŢřæ■őçśżåđŃäÿž None ïijŇ bool ïijŇ int ïijŇ float åŠŇ str ïijŇ äżĕåŔŁåŇĚåŘńè£ŹäžŻçśżåđŃæŢřæ■őçŽĎlistsïijŇtuplesåŠŇdictionariesãĂĆ åŕźäžŐdictionariesïijŇkeyséIJĂèęĄæŸŕå■ŮçňęäÿšçśżåđŃ(å■ŮåĚÿäÿ■äżżä¡Ţéİđå■ŮçňęäÿšçśżåđŃçŽĎkeyåĪ äÿžäžĘéĄţå¿łJSONèğĎèŇČïijŇä¡ăåžŤèŕěåŔłçijŰçăĄPythonçŽĎlistsåŠŇdictionariesãĂĆ èĂŇäÿŤïijŇåIJÍwebåžŤçŤĺçĺŃåžŔäÿ■ïijŇéąűåśĆåŕźèśąèćńçijŰçăĄäÿžäÿĂäÿłå■ŮåĚÿæŸŕäÿĂäÿłæăĞåĞĘåĄ

JSONçijŰçăĄçŽĎæăijåijŔåŕźäžŐPythonèŕ■æşŢèĂŇåůšåĞăäźŐæŸŕåőŇåĚĺäÿĂæăůçŽĎïijŇéŹď 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■ŢçŽĬ çĽźåĹńæŸŕå¡ŞæŢřæ■őçŽĎåţŇåěŮçŞæđĎåśĆæňąå¿Łæůśæĺ ŰèĂĚåŇĚåŘńåď ğéĞŘçŽĎå■Ůæőţæ ŰűãĂĆ äÿžäžĘèğčåĘşè£Źäÿłéŏőéć Ÿïij ŇåŔŕäżĕèĂČèŹŚä¡£çŤÍpprintæĺąåİŮçŽĎ pprint() åĞ¡æŢřæĬěäżčæŽ£æŹőéĂŽçŽĎ print() åĞ¡æŢřãĂĆ åőČäijŽæŇĽçĚğkeyçŽĎå■Ůæŕ■éąžåžŔåźűäżĕäÿĂçğ■æŽ ť åŁăç¿Őèğ ĆçŽĎæŰźåijŔè¿ŞåĞžãĂĆ äÿŃéİćæŸŕäÿĂäÿłæijŤçď žåęĆä¡ŢæijĆäžőçŽĎæĽŞå■řè¿ŞåĞžTwitteräÿŁæŘIJçť ćçżŞæđIJçŽĎä¿Ńå■Řïij Ž

```
>>> 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)

```
{'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'}
>>>
```

äÿĂèĹňæİěèőšïijŇJSONèğčçăĄäijŽæăźæ■őæŔŘä¿ŻçŽĎæŢřæ■őåĹŻåžždictsæĹŰlistsãĂĆåęĆæđIJä¡ăæČşèęĄåĹŻåžžåĚűäžŰçśžåđŃçŽĎåŕźèśąïijŇåŔŕäžěçżŹ `json.loads()` äijăéĂŠobject_pairs_hookæĹŰobject_hookåŔĆæŢřãĂĆä¿ŃåęĆïijŇäÿŃéİćæŸŕæijŤçđ'žåęĆä¡Ţèğčçă ĄJSONæŢřæ■őåźűåIJĺäÿĂäÿłOrderedDictäÿ■ä£ĮçŢŹåĚűéąžåžŔ

```
>>> 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)])
>>>
```

äÿŃéİćæŸŕåęĆä¡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)

```
>>> data.shares
50
>>> data.price
490.1
>>>
```

æIJĂåŘŐäÿĂäÿłä¿Ńå■Řäÿ■ïijŇJSONèğčçăĄåŘŐçŽĎå■ŮåĚÿä¡IJäÿžäÿĂäÿłå■Ţäÿłå
ŔĆæŢřäijăéĂŠçżŹ__init__() ãĂĆ çĎűåŘŐïijŇä¡ăåřšåŔŕäżěéŽŔå£ČæĽĂæňšçŽĎä¡£çŤĺåőČäžĘïijŇærŤåęĆä¡IJäÿžäÿĂäÿłå

åIJÍçijŰçăĄJSONçŽĎæŮűåĂŹïijŇè£ŸæIJL'äÿĂäžŻéĂL'éąźå¿ĹæIJL'çŤĺãĂĆ
åęĆæđIJä¡ăæČşèŐůå¿Ůæij Ćäžőç ŽĎæăijåijŔåŇŰå■Ůçňęäÿšå ŘŐè¿ŞåĞžïijŇåŔŕäżěä¡£çŤÍ
json.dumps() çŽĎindentåŔĆæŢřãĂĆ åőČäijŽä¡£å¿Ůè¿ŞåĞžåŠÑpprint()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■æŸŕJSONåŔŕåžŔåĹŮåŇŰçŽĎãĂĆä¿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
>>>
```

åęĆædIJä¡ăæČşåžŔåĹŮåŇŰåŕźèśąåőđä¿ŃïijŇä¡ăåŔŕäżěæŔŘä¿ŻäÿĂäÿłåĞ¡æŢřïijŇåőČçŽĎè¿ŞåĚěæŸŕ

```
def serialize_instance(obj):
    d = { '__classname__' : type(obj).__name__ }
    d.update(vars(obj))
    return d
```

åęĆædIJä¡ăæČşåŔ■è£ĞæİěèŐůåŔŰè£Źäÿłåőđä¿ŃïijŇåŔŕäżěè£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£ŸæIJL'å¿Ĺåď'ŽåĚűäžŰéĂL'éąźæİěæŐğåĹűæŻť'ä¡ŐçžğåĹńçŽĎæŢřå■ŮãĂĄçL'źæőŁåĂi
åŔŕäżěåŔĆèĂČåőŸæŰźæŰĞæąčèŐůåŔŰæŻť'åď'ŽçżĘèŁĆãĂĆ

8.3 6.3 èğčædŘçőÅå■ŢçŽĎXMLæŢřæ■ő

éŮőécŸ

ä¡ăæČşäżŐäÿĂäÿłçőÅå■ŢçŽĎXMLæŰĞæąčäÿ■æŔŘåŔŰæŢřæ■őãĂĆ

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'èűčçŽĎäžŇãĂĆ

èőÍèőž

åIJÍå¿Ĺåď ŽåžŤçŤÍçÍŃåžŔäÿ■åď ĎçŘĘXMLçijŰçăĄæăijåijŔçŽĎæŢřæ■őæŸŕå¿Ĺåÿÿè ğĄçŽĎãĂĆ äÿ■äžĚåŻääÿžXMLåIJÍInternetäÿŁéÍćåůšçżŔèćńáź£æşŽåžŤçŤÍäžÓæŢřæ■őäžď æ■ćïijŇ åŘŇæŮűåőČäź§æŸŕäÿĂçğ■å■ŸåĆÍåžŤçŤÍçÍŃåžŔæŢřæ■őçŽĎåÿÿçŤÍæăijåijŔ(æŕŤåęĆå■Ůåď ĎçŘĘïijŇé§§ä æŐěäÿŃæÍěçŽĎèőÍèőžäijŽåĚĹåĄĞåőŽèŕżèĂĚåůšçżŔåŕźXMLå§žçąĂæŕŤè¿ČçĘ§æĆL’äžĘãĂĆ

åIJÍå¿Ĺåď ŽæČĚåĘţäÿŃïijŇå¡Şä¡£çŤÍXMLæÍěäžĚäžĚå■ŸåĆÍæŢřæ■őçŽĎæŮűåĂŹïijŇåŕźåžŤçŽĎæŰĞ ä¿ŃåęĆïijŇäÿŁéÍćä¿Ńå■Řäÿ■çŽĎRSSèőćéŸĚæžŘçśżäijijäžŐäÿŃéÍćçŽĎæäijåijŔïijŽ

```
<?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 äÿłXMLæŰĞæąčåźűåřEåĚűè¡ňæ■ćæĹŘ çĎűåŘŐïijŇä¡ăåřśèČ¡ä¡£çŤÍ `find()` ãĂĄ`iterfind()` åŠŇ `findtext()` ç■Ľ æŰźæşŢæİěæŘIJçť ćçĽźåőŽçŽĎXMLåĚČçť ääžĘãĂĆ è£ŹäžŻåĞ¡æŢřçŽĎåŔĆæŢřåřśæŸŕæ§ŘäÿłæŇĞåőŽçŽĎæăĞç■¿åŘ■ïijŇä¿Ňåę Ć `channel/item` æĹŰ `title` ãĂĆ

æŕŔæňąæŇĞåőŽæ§Řäÿłæ ăĞç■¿æŮűïijŇä¡ăéIJĂèęĄéĄ■åŐĘæŢť äÿłæŰĞæąčçżŞæđĎãĂĆæŕŔæňąæŘIJ åŘŇæăůïijŇæŕŔæňąæŞ■ä¡IJæĽ ĂæŇĞåőŽçŽĎæăĞç■¿åŘ■äź§æŸŕçŢůåğŃåĚČçť äçŽĎçŻÿåŕźèůŕå¿ĎãĂĆ ä¿ŇåęĆïijŇæĽğèąŇ `doc.iterfind('channel/item')` æİěæŘIJçť ćæĽ ĂæIJĽåIJĺ `channel` åĚČçť ääÿŃéİćçŽĎ `item` åĚČçť äãĂĆ `doc` äžčèąĺæŰĞæąčçŽĎæIJĂéąűåśĆ(äź§åřśæŸŕçňňäÿĂçžğçŽĎ `rss` åĚČçť ä)ãĂĆ çĎűåŘŐæŐěäÿŃæİěçŽĎèřČçŤÍ `item.findtext()` äijŽäżŐåůšæĽ¿åĹřçŽĎ `item` åĚČçť ää¡■ç¡őåijĂåğŃæŘIJçť ćãĂĆ

`ElementTree` æĺąåİŮäÿ■çŽĎæŕŔäÿłåĚČçť äæIJĽäÿĂäžŻéĞ■èęĄçŽĎåśđæĂğåŠŇæŰźæşŢïijŇåIJĺèğ `tag` åśđæĂğåŇĚåŘńäžĘæăĞç■¿çŽĎåŘ■å■ŮïijŇ`text` åśđæĂğåŇĚåŘńäžĘåĚĚéČÍçŽĎæŰĞæIJňïijŇèĂŇ `get()` æŰźæşŢèČ¡èŐůåŔŰåśđæĂğåĂijãĂĆä¿ŇåęĆïijŽ

```
>>> 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')
>>>
```

æIJĽäÿĂçĆźèęĄåijžèřČçŽĎæŸŕ `xml.etree.ElementTree` åźűäÿ■æŸŕXML-Lèğčæđ ŘçŽĎåŤřäÿĂæŰźæşŢãĂĆ åŕźäžŐæŻť éńŸçžğçŽĎåžŤçŤÍçĺŃåžŔïijŇä¡ăéIJĂèęĄèĂČèŹŚä¡£çŤÍ `lxml` ãĂĆ åőČä¡£çŤÍäžĘåŠŇElementTreeåŘŇæăůçŽĎçijŰçĺŃæŐěåŔčïijŇåŻăæ■ď äÿŁéİćçŽĎä¿Ňå■ŘåŘŇ ä¡ăåŔłéIJĂèęĄåřĘåĹŽåijĂåğŃçŽĎimportèŕ■åŔěæ■ćæĹŘ `from lxml.etree import parse` åřśèąŇäžĘãĂĆ `lxml` åőŇåĚÍéĄţå¿łXMLæăĞåĞĘïijŇåźűäÿŤéĂ§åžęäź§éİđåÿÿå£ńïijŇåŘŇæŰűè£Ÿæ

8.4 6.4 åćđéĞŔåijŘèğčæđŘåď ğåđŃXMLæŰĞäżű

éŮőéćŸ

ä¡ăæČşä¡£çŤÍåř¡åŔŕèČ¡åřŚçŽĎåĘĚå■ŸäżŐäÿĂäÿłèűĚåď ğçŽĎXMLæŰĞæąčäÿ■æŔŘåŔŰæŢřæ■őãĂĆ

èğčåĘşæŰźæąĹ

äżżä¡ŢæŮűåĂŹåŔłèęĄä¡ăéĄĞåĹřåćđéĞŔåijŘçŽĎæŢřæ■őåď ĎçŘĘæŮűïijŇçňňäÿĂæŮűéŮť åřśåžŤèŕěa äÿŃéİćæŸŕäÿĂäÿłå¿ĹçőĂå■ŢçŽĎåĞ¡æŢřïijŇåŔłä¡£çŤÍå¿ĹåřŚçŽĎåĘĚå■ŸåřśèČ¡åćđéĞŔåijŘçŽĎåď ĎçŘĘ

```
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¡æŢřïijŇä¡äéIJĂèęĄåĚĹæIJL’äÿĂäÿłåď ğåđŃçŽĎXMLæŰĞäżűãĂĆ éĂŽåÿÿä¡ăåŘŕäžěåIJĺæŤ£åžIJç¡ŚçńŹæĹŰåĚňåĚśæŢřæ■ős¡ŚçńŹäÿŁæL’¿åĹřè£Źæăůç ŽĎæŰĞäżűãĂĆ ä¿ŇåęĆïijŇä¡ăåŘŕäžěäÿŃè¡¡XMLæăijåijŘçŽĎèŁİåŁăåŞěå§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)

```
            <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¡ăæČşåĘŹäýĂäýłèĎŽæIJňæİěæŇL’çĚğåİŚæť’ijæŁěåŚŁæŢřéĞŔæŐŠåĹŮéĆőçijŰåŔůçăĄãĂĆä

```
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£ŹäýłèĎŽæIJňåŤřäýĂçŽĎéŮőécŸæŸřåőČäijŽåĚĹářĘæŢť’äýłXMLæŰĞäžűåŁăè¡¡åĹřåĘĚå■Ÿäý■çĎűã
åIJĺæĹŚçŽĎæIJžåŹĺäýŁïijŇäýžäžĘè£ŘèąŇè£ŹäýłçĺŇåžŔéIJĂèęĄçŤĺåĹř450MBåůęåŔşçŽĎåĘĚå■Ÿçľ’žéŮť’ã
åęĆæđIJä¡£çŤĺåęĆäýŇäžčçăĄïijŇçĺŇåžŔåŔłéIJĂèęĄä£őæŤźäýĂçĆźçĆźïijŽ

```
from collections import Counter

```

(continues on next 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£Źäÿłç L'ŁæIJňçŽĎäžčçăĄè£ŘèąŇæŮűåŔłéIJĂèęĄ7MBçŽĎåĘĚå■Ÿ–åď ğåď ğèŁĆçžę

èőĺèőž

è£ŹäÿĂèŁĆçŽĎæŁĂæIJŕäijŽä¿ÌèţŰ `ElementTree` æĺąåİŮäÿ■çŽĎäÿď äÿłæăÿå£ČåŁ§èČ¡ãĂĆ çňňäÿĂïijŇ`iterparse()` æŰźæşŢåĚĄèőÿåŕźXMLæŰĞæąčè£ŻèąŇåćđéĞŔæŞ■ä¡IJãĂĆ ä¡£çŤĺæŮűïijŇä¡ăéIJĂèęĄæŔŘä¿ŻæŰĞäżűåŘ■åŠŇäÿĂäÿłåŇĚåŘńäÿŇéİćäÿĂçğ■æĹŰåď Žçğ■çśžåđŇçŽĎä `start` , `end`, `start-ns` åŠŇ `end-ns` ãĂĆ çŤś `iterparse()` åĹŻåżžçŽĎè£■äžčåŹĺäijŽäžğçŤ§å¡ćåęĆ `(event, elem)` çŽĎåĚČçżĎïijŇ åĚűäÿ■ `event` æŸŕäÿŁè£řäžŇäżűåĹŮèąĺäÿ■çŽĎæ§ŘäÿĂäÿłïijŇèĂŇ `elem` æŸŕçŻÿåžŢçŽĎXM-LåĚČçť ăăĂĆä¿ŃåęĆïijŽ

```
>>> 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` äžŃäžűåIJÍæ§ŘäÿłåĚČçť ăçňňäÿĂæňąèćńåĹŻåżžåźűäÿŤè£ŸæšąæIJL'èćńæŔŠåĚěåĚűäžŰæŢřæ■ èĂŇ `end` äžŃäžűåIJÍæ§ŘäÿłåĚČçť ăåůšçżŔåőŇæĹŘæŮűèćńåĹŻåżžãĂĆ åř¡çőąæšąæIJL'åIJÍä¿Ńå■Řäÿ■æijŤçď žïijŇ `start-ns` åŠŇ `end-ns` äžŃäžűèćńçŤÍæİěåď ĎçŘĘXMLæŰĞæąčåŚ¡åŘ■çľ žéŮť çŽĎåčřæŸŐãĂĆ

è£ŹæIJňèŁĆä¿Ńå■Řäÿ■ïijŇ `start` åŠŇ `end` äžŃäžűèćńçŤÍæİěçőąçŘĘåĚČçť ăåŠŇæăĞç■¿æăĹãĂĆ æăĹäžčèąĺäžĘæŰĞæąčèćńèğčæđŘæŮűçŽĎåśĆæňąçżŞæđĎïijŇ è£ŸèćńçŤĺæİěåĹď æŰ■æ§ŘäÿłåĚČçť ăæŸŕåŘęåŇźéĚ■äijäçżŹåĞ¡æŢř `parse_and_remove()` çŽĎèůŕå¿ĎãĂĆ åęĆæđIJåŇźéĚ■ïijŇåŕśåĹľçŤÍ `yield` èŕ■åŔěåŘŚèřČçŤÍèĂĚè£ŤåŻđè£ŹäÿłåĚČçť ăãĂĆ

åIJÍ `yield` äźŃåŘŐçŽĎäÿŇéİćè£Źäÿłèŕ■åŔěæĽ■æŸŕä¡£å¿ŮçÍŇåžŔå■ăçŤÍæđĄåřŚåĘĚå■ŸçŽĎElemer

```
elem_stack[-2].remove(elem)
```

è£Źäýłèŕ■åŘĕä¡£å¿ŮäźŃåĽ■çŤś yield äžğçŤ§çŽĎåĔČçť'äžŐåőČçŽĎçĹűèĿĆçĆźäý■åĹäéŹď'æŐĽ'ã
åĄĞèő¿åůšçżŔæšąæIJĽ'åĚűåőČçŽĎåIJřæŰźåijŢçŤĺè£Źäýłå ĚČçť'äžĘïijŇéĆčäźĹè£ŹäýłåĚČçť'äåřśèćńéŤĂæ

åŕźèĿĆçĆźçŽĎè£■äżčåijŔèğčæđŘåŠŇåĽäéŹď'çŽĎæIJĂçżĹæŢĽæđIJåřśæŸŕäýĂäýłåIJĺæŰĞæąčäýŁéń
æŰĞæąčæăŚçżŞæđĎäžŐåğŃèĞłçżĹæšąèćńåőŃæŢť'çŽĎåĹŻåżžè£ĞãĂĆåř¡çő¡åęĆæ■ď'ïijŇè£ŸæŸŕèČ¡éĂŽ

è£Źçğ■æŰźæąĽçŽĎäýżèęĄçijžéŹůåřśæŸŕåőČçŽĎè£Řèą Ňæ ĂğèČ¡äžĘãĂĆ
æĹŠèĞłåůśæţŃèŕŢçŽĎçżŞæđIJæŸŕïijŇèŕżåŔŰæŢť'äýłæŰĞæąčåĹřåĘĚå■Ÿäý■çŽĎçĽĹæIJňçŽĎè£Řèą ŇéĂ
ä¡ĘæŸŕåőČå■ť'ä¡£çŤĺäžĘèűĚè£ĞåŘŐèĂĚ60åĂ■çŽĎåĘĚå■ŸãĂĆ
åŽăæ■ď'ïijŇåęĆæđIJä¡ăæŻť'åĚşå£ČåĘĚå■Ÿä¡£çŤĺéĞŔçŽĎèŕİïijŇéĆčäźĹåćđéĞŔåijŔçŽĎçĽĹæIJňåőŇèČIJ

8.5 6.5 åřĘå■ŮåĚÿè¡ňæ■ćäÿžXML

éŮőéćŸ

ä¡ăæČşä¡£çŤĺäýĂäýłPythonå■ŮåĚÿå■ŸåĆĺæŢřæ■őïijŇåźűåřĘåőČè¡ňæ■ćæĹŘXMLæăijåijŔãĂĆ

èğčåĘşæŰźæąĹ

åř¡çő¡ xml.etree.ElementTree åžŞéĂŽåÿÿçŤĺæİĕåĄŽèğčæđŘåůěä¡IJïijŇåĔűåőđåőČäź§åŔŕäžĕåĹ
ä¿Ńåę Ćïij ŇèĂČèŹŚåęĆäýŃè£Źäýłå Ğ¡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
```

äýŃéİćæŸŕäýĂäýłä¡£çŤĺä¿Ńå■ŘïijŽ

```
>>> s = { 'name': 'GOOG', 'shares': 100, 'price':490.1 }
>>> e = dict_to_xml('stock', s)
>>> e
<Element 'stock' at 0x1004b64c8>
>>>
```

è¡ňæ■ćçżŞæđIJæŸŕäýĂäýł Element åőđä¿ŃãĂĆåŕźäžŐI/OæŞ■ä¡IJïijŇä¡£çŤĺ xml.
etree.ElementTree äý■çŽĎ tostring() åĞ¡æŢřå¿Ĺåőźæ ŸŞåřśèČ¡åřĘåőČè¡ňæ■ćæĹŘäýĂäýłå■ŮèĿ

```
>>> from xml.etree.ElementTree import tostring
>>> tostring(e)
b'<stock><price>490.1</price><shares>100</shares><name>GOOG</name></
 ↪stock>'
>>>
```

åęĆædIJä¡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>'
>>>
```

åęĆædIJä¡ăè£ŸæČşä£ÍæŇĄåĚČçť'ăçŽĎéąžåžŔïijŇåŔŕäżěèĂČèŹŚædĎéĂăäÿĂäÿł `OrderedDict` æÍěäżčæŻ£äÿĂäÿłæŹőéĂŽçŽĎå■ŮåĚÿãĂĆèŕůåŔĆèĂČ1.7åřŔèŁĆãĂĆ

èőÍèőž

å¡ŞåĹŻåżžXMLçŽĎæŮűåĂŹïijŇä¡ăèćńéŹŘåĹűåŔłèČ¡ædĎéĂăå■ŮçňęäÿšçśżåđŇçŽĎåĂijãĂĆä¿ŇåęĆï

```
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)
```

éŮőécŸæŸŕåęĆædIJä¡ăæL'ŇåŁÍçŽĎåŐżædĎéĂăçŽĎæŮűåĂŹåŔŕèČ¡äijŽçćřåĹřäÿĂäžŻéžżçČęãĂĆä¿Ň

```
>>> 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>'
>>>
```

æşÍæĎŔåĹřçÍŇåžŔçŽĎåŘŐéÍćéĆčäÿłä¿Ňå■Řäÿ■ïijŇå■Ůçňę âĂŸ<âĂŹ åŠŇ âĂŸ>âĂŹ èćńæŻ£æ■ćæĹŘäžĘ `<` åŠŇ `>`

äÿŃéÌćäżĚä¿ŻåŔĆèĂČïijŇåęĆæđIJä¡ăéIJĂèęĄæĽŃåŁÍåŐżè¡ňæ■ćè£ŹäžŻå■Ůçňę¨ijŇåŔŕäžěä¡£çŤÍ xml.sax.saxutils äÿ■çŽĎ escape() åŠŇ unescape() åĞ¡æŢřãĂĆä¿ŃåęĆïijŽ

```
>>> from xml.sax.saxutils import escape, unescape
>>> escape('<spam>')
'&lt;spam&gt;'
>>> unescape(_)
'<spam>'
>>>
```

éŹď¨äžĘèČ¡åĹŻåżžæ■čçąőçŽĎè¿ŞåĞžåď´Űïij Ňè£ŸæIJĽåŔęåď´ŰäÿĂäÿłåŐ§åŻăæŐÍè■Řä¡ääĹŻåżž Element åőđä¿ŃèĂŇäÿ■æŸŕå■Ůçňęäÿšïij Ň éĆčářśæŸŕä¡£çŤĺå■ŮçňęäÿšçżĎåŘĹæđĎéĂääÿĂäÿłæŻť åď´ğçèĂŇ Element åőđä¿ŃåŔŕäžěäÿ■çŤÍèĂČèŹŚèğčæđŘXMLæŰĞæIJńçŽĎæČĚåĘţäÿŃéĂŽè£Ğåď´Žçğ■æŰźåiäź§ářśæŸŕèŕť´ïijŇä¡ääŔŕäžěåIJÍäÿĂäÿłéńŸçžğæŢřæ■őçżŞæđĎäÿŁåőŇæĹŘä¡ăæĽ´ĂæIJĽçŽĎæŞ■ä¡IJïijŇåźű

8.6 6.6 èğčæđŘåŠŇä£őæŤźXML

éŮőécŸ

ä¡ăæČşèŕžåŔŰäÿĂäÿłXMLæŰĞæąčïijŇåŕźåőČæIJĂäÿĂäžŻä£őæŤźïijŇçĎűåŘŐåřĘçżŞæđIJåĘŹåŻđXM

èğčåĘşæŰźæąĹ

ä¡£çŤÍ xml.etree.ElementTree æÍąåİŮåŔŕäžěå¿ĹåőźæŸŞçŽĎåď´ĎçŘĘè£ŹäžŻäżžåŁąãĂĆ çňňäÿĂæ■ěæŸŕäžěéĂŽåÿÿçŽĎæŰźåijŔæİèèğčæđŘè£ŹäÿłæŰĞæąčãĂĆä¿ŃåęĆïijŇåĄĞèő¿ä¡ăæIJĽäÿĂäÿłå pred.xml çŽĎæŰĞæąčïijŇçśżäijijäÿŃéÍćè£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)

```
        <v>1867</v>
        <rn>22</rn>
    </pre>
</stop>
```

äýŃéÍćæŸŕäýĂäýłåĹľçŤÍ ElementTree æÍěèŕżåŔŰè£ŹäýłæŰĞæąčåźűåŕźåőČåĄŽäýĂäžŽä£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 ĎçŘEçżŞæđIJæŸŕäýĂäýłåČŔäýŃéÍćè£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§çŽĎïijŇä¡ĘæŸŕä¡ăå£ĚéążçĽćèőřçŽĎæŸŕæĽĂæł
åřĘåőČä¡IJäÿžäÿĂäÿłåĹŮèąÍæİěåď ĎçŘĘãĂĆä¿ŃåęĆïijŇåęĆæđIJä¡ăåĹăéŹď æ§Räÿłåĕčçť ăïijŇéĂŽè£ĞèřČ
`remove()` æŰźæşŢäżŐåőČçŽĎçŻť æŐěçĹűèŁĆçĆźäÿ■åĹăéŹď ãĂĆ
åęĆæđIJä¡ăæŔŠåĚěæĹŰåćđåŁăæŰřçŽĎåĕČçť ăïijŇä¡ăåŘŇæăůä¡£çŤÍçĹűèŁĆçĆźåĕČçť ăçŽĎ
`insert()` åŠŇ `append()` æŰźæşŢãĂĆ è£ŸèČ¡åŕźåĕČçť ăä¡£çŤÍçť ćåijŢåŠŇåĹĞçĽĞæŞ■ä¡IJïijŇæŕŤåęĆ
`element[i]` æĹŰ `element[i:j]`

åęĆæđIJä¡ăéIJĂèęĄåĹŻåżžæŰřçŽĎåĕČçť ăïijŇåŕäžěä¡£çŤÍæIJňèŁĆæŰźæąĹäÿ■æijŤçď žçŽĎ
`Element` çśżãĂĆæĹŠäžňåIJĺ6.5åřŔèŁĆåůšçżŔèŕęçżĘèőÍèőžè£ĞäžĘãĂĆ

8.7 6.7 åĹľçŤĺåŚ¡åŘ■çľžéŮťèğčæđŘXMLæŰĞæąč

éŮőécŸ

ä¡ăæČşèğčæđŘæ§ŘäÿłXMLæŰĞæąčïijŇæŰĞæąčäÿ■ä¡£çŤÍäžĘXMLåŚ¡åŘ■çľžéŮťãĂĆ

èğčåĘşæŰźæąĹ

èĂČèŹŠäÿŃéİćè£Źäÿłä¡£çŤÍäžĘåŚ¡åŘ■çľžéŮť çŽĎæŰĞæąčïijŽ

```
<?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)

```
>>> 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¡ăåŔŕäžěéĂŽè£ĞářĘåŚ¡åŘ■çľžéŮť'åď'ĎçŘĘéĂżè¿ŚåŇĚèčĚäÿžäÿĂäÿłåůěåĚůçśżæİěçőĂåŇŰè£Źäÿłèŧ

```
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£ĞäÿŇéİćçŽĎæŰźåijŔä¡£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'
>>>
```

èőÍèőž

èğčæđŘåŘńæIJL'åŚ¡åŘ■çľžéŮť'çŽĎXMLæŰĞæąčäijŽærŤè¿ČçźĄçŘŘãĂĆ äÿŁéİćçŽĎ `XMLNamespaces` äžĔäžĚæŸŕåĚĄèőÿä¡ăä¡£çŤĺçijľçŢěåŘ■äžčæŻ£åőŇæŢť'çŽĎURIåřĘåĚűåŔŸå¿Ůçĺ■å¿őç

å¿ĹäÿŀåźÿçŽĎæŸŕïijŇåIJĺå§žæIJňçŽĎ `ElementTree` èğčæđŘäÿ■æšąæIJL'äżżä¡ŢéĂŤå¿ĎèŐůåŔŰåŚ¡åŘ■çľžéŮť'çŽĎä£qæĄŕãĂĆ ä¡ĘæŸŕïijŇåęĆæđIJä¡ăä¡£çŤĺ `iterparse()` åĞ¡æŢřçŽĎèŕİåřšåŔŕäžěèŐůåŔŰæŻť'åď'ŽåĚşäžŐåŚ¡åŘ■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)

```
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¡£çŤĺåĹřåĔűäżŰéńŸçžğ
åžžèőőä¡ăæIJĂåě¡æŸŕä¡£çŤÍ lxml åĞ¡æŢřåžŞæÌěäžčæŻ£ ElementTree ãĂĆ
ä¿ŃåęĆïijŇlxml åŕźåĹľçŤÍDTDéłŇèŕĄæŰĞæąčãĂĄæŻť'åě¡çŽĎXPathæŤŕæŇĄåŠŇäÿĂäžŻåĔűäżŰéńŸçžğ
è£ŹäÿĂåřŔèŁĆåĔűåőđåŔłæŸŕæŢŹä¡ääęĆä¡Ţèőľ'XMLèğčæđŘçĺ■å¿őçőĂå■ŢäÿĂçĆźãĂĆ

8.8 6.8 äÿŐåĚşçşžåđŃæŢřæ■őåžŞçŽĎäžď'äžŠ

éŮőécŸ

ä¡ăæČşåIJĺåĚşçşžåđŃæŢřæ■őåžŞäÿ■æ§ěèŕćãĂĄåćđåŁăæĹŰåĹäéŹď'èőřå¡ŢãĂĆ

èğčåĘşæŰźæąĹ

Pythonäÿ■èąĺçď'žåď'ŽèąŇæŢřæ■őçŽĎæăĞåĞĘæŰźåijŔæŸŕäÿĂäÿłçŤśåĚČçżĎæđĎæĹŘçŽĎåžŔåĹŮãĂ

```
stocks = [
    ('GOOG', 100, 490.1),
    ('AAPL', 50, 545.75),
    ('FB', 150, 7.45),
    ('HPQ', 75, 33.2),
]
```

ä¿Ìæ■őPEP249ïijŇéĂŽè£Ğè£Źçğ■å¡ćåijŔæŔŘä¿ŻæŢřæ■őïijŇ
åŔŕäžèå¿ĹåőźæŸ§çŽĎä¡£çŤÍPythonæăĞåĞĘæŢřæ■őåžŞAPIåŠŇåĚşçşžåđŃæŢřæ■őåžŞè£ŻèąŇäžď'äžŠãĂĆ
æĽĂæIJĽæŢřæ■őåžŞäÿŁçŽĎæŞ■ä¡IJéČ¡éĂŽè£ĞSQLæ§ěèŕćèŕ■åŘěæÌěåőŇæĹŘãĂĆæŕŔäÿĂèąŇè¿ŞåĚěè¿Ş

äÿžäžĘæijŤçď'žèŕť'æŸŐïijŇä¡ăåŔŕäžěä¡£çŤÍPythonæăĞåĞĘåžŞäÿ■çŽĎ sqlite3
æÍąåİŮãĂĆ åęĆæđIJä¡ää¡£çŤÍçŽĎæŸŕäÿĂäÿłäÿ■åŘŇçŽĎæŢřæ■őåžŞ(æŕŤåęĆMySqlãĂĄPostgresqlæĹŰèĂ
è£Ÿå¿ŮåőĽèčĚçŻÿåžŤçŽĎçňňäÿĽ'æŰźæÍąåİŮæÌěæŘŘä¿ŻæŤŕæŇĄãĂĆ
äÿ■è£ĞçŻÿåžŤçŽĎçijŰçÍŃæŐěåŘčåĞääźŐéČ¡æŸŕäÿĂæăůçŽĎïijŇéŹď'äžĘäÿĂçĆźçĆźçżĘå¿őåůőåĹńåď'Ůã

çňňäÿĂæ■ěæŸŕè£đæŐěåĹřæŢřæ■őåžŞãĂĆéĂŽåÿÿä¡ăèęĄæĽ'ğèąŇ connect()
åĞ¡æŢřijŇ çżŹåőČæŔŘä¿ŻäÿĂäžŻæŢřæ■őåžŞåŘ■ãĂĄäÿżæIJžãĂĄçŤÍæĹůåŘ■ãĂĄåŕĘçăĄåŠŇåĚűäżŰå£Ě

```
>>> import sqlite3
>>> db = sqlite3.connect('database.db')
>>>
```

äÿžäžĘåď ĎçŘĘæŢřæ■őïijŇäÿŃäÿĂæ■ěä¡ăéIJĂèęĄåĹŻåžžäÿĂäÿłæÿÿæăĞãĂĆ äÿĂæŮęä¡ăæIJL’äžĘæÿÿæăĞïijŇéĆčäźĹä¡ăåřšåŘŕäžěæL’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■æŔŠåĚěåď Žæİąèőŕå¡ŢïijŇä¡£çŤĺçśżäijijäÿŃéİćè£Źæăůç ŽĎèŕ■åŘěïijŽ

```
>>> c.executemany('insert into portfolio values (?,?,?)', stocks)
<sqlite3.Cursor object at 0x10067a730>
>>> db.commit()
>>>
```

äÿžäžĘæL’ğèąŇæ§Řäÿłæ§ěèŕćïijŇä¡£çŤĺåČŔäÿŃéİćè£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)
>>>
```

åęĆædIJä¡ăæČşæŐěåŔŮçŤĺæĹůè¿ŞåĚěä¡IJäÿžåŔĆæŢřæİěæL’ğèąŇæ§ěèŕćæŞ■ä¡IJïijŇå£Ěéążçąőä£İä¡ă

```
>>> 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)
>>>
```

èőÍèőž

åIJÍæŕŤè¿Čä¡ŐçŽĎçžğåĹńäÿŁåŠŇæŢřæ■őåžŞäžď äžŠæŸréİđåÿÿçőĂå■ŢçŽĎãĂĆ ä¡ăåŘłéIJĂæŔŘä¿ŻSQLèŕ■åŘěåźűěřČçŤĺçŻÿåžŤçŽĎæĺąåİŮåřšåŘŕäžěæŻťæŰřæĹŰæŔŘåŘŰæŢřæ■őäžĘãĂ èŹ¡èŕťåęĆæ■ď ïijŇè£ŸæŸŕæIJL’äÿĂäžŻæŕŤè¿ČæčŸæL’ŃçŽĎçżĘèŁĆéŮőécŸéIJĂèęĄä¡ăéĂŘäÿłåĹŮåĞžåĆ

äÿĂäÿłéŽ¿çĆźæŸŕæŢřæ■őåžŞäÿ■çŽĎæŢřæ■őåŠŇPythonçśžåđŃçŻt'æŐěçŽĎæŸăåřĎãĂĆ
åŕźäžŐæŮěæIJ§çśžåđŃïijŇéĂŽåÿÿåŔŕäžěä¡£çŤÍ datetime æĺąåİŮäÿ■çŽĎ datetime
åőđä¿ŃïijŇ æĹŰèĂĚåŔŕèČ¡æŸŕ time æĺąåİŮäÿ■çŽĎçşżçż§æŮűéŮt'æĹşãĂĆ
åŕźäžŐæŢřå■ŮçśžåđŃïijŇçL'źåĹńæŸŕä¡£çŤĺåĹřåřŔæŢřçŽĎéĞŚèđ■æŢřæ■őïijŇåŔŕäžěçŤÍ
decimal æĺąåİŮäÿ■çŽĎ Decimal åőđä¿Ńæİěèąĺçd'žãĂĆ
äÿ■åźÿçŽĎæŸŕïijŇåŕźäžŐäÿ■åŘŇçŽĎæŢřæ■őåžŞèĂŇèÍĂåĚůä¡ŞæŸăåřĎèğĎåĹŹæŸŕäÿ■äÿĂæăůçŽĎïijŇä¡

åŔęåd'ŰäÿĂäÿłæŻt'åŁăåd'■æİĆçŽĎéŮőécŸåřśæŸŕSQLèŕ■åŔěå■ŮçňęäÿšçŽĎædĎéĂããĂĆ
ä¡ăå■ČäÿĞäÿ■èęĄä¡£çŤÍPythonå■ŮçňęäÿšæăijåijŔåŇŰæŞ■ä¡IJçňę(åęĆ%)æĹŰèĂĚ
.format() æŰźæŞŢæİěåĹŻåżžè£ŹæăůçŽĎå■Ůçňęäÿšã ĂĆ
åęĆædIJäijăéĂŠçżŹè£ŹäžŻæăijåijŔåŇŰæŞ■ä¡IJçňęçŽĎåĂijæİěèĞłäžŐçŤÍæĹůçŽĎè¿ŞåĚěïijŇéĆčäźĹä¡ăçŽ
http://xkcd.com/327)ãĂĆ æ§èèŕćèŕ■åŔěäÿ■çŽĎéĂŽéĚ■çňę ?
æŇĞçd'žåŘŐåŔŕæŢřæ■őåžŞä¡£çŤĺåőČèĞłåůśçŽĎå■ŮçňęäÿšæŻ£æ■ćæIJžåĹűïijŇè£ŹæăůæŻt'åŁăçŽĎåőL'å

äÿ■åźÿçŽĎæŸŕïijŇäÿ■åŘŇçŽĎæŢřæ■őåžŞåŘŐåŔŕåŕźäžŐéĂŽéĚ■çňęçŽĎä¡£çŤÍæŸŕäÿ■äÿĂæăůçŽĎã
? æĹŰ %s ïijŇ è£ŸæIJL'åĚűäžŰäÿĂäžŻä¡£çŤÍäžĘäÿ■åŘŇçŽĎçňęåŘůïijŇærŤåęĆ:0æĹŰ:1æİěæŇĞçd'žåŔĆ
åŘŇæăůçŽĎïijŇä¡ăè£ŸæŸŕå¿ŮåŐżåŔĆèĂČä¡ăä¡£çŤÍçŽĎæŢřæ■őåžŞæĺąåİŮçŻýåžŢçŽĎæŰĞæąčãĂĆ
äÿĂäÿłæŢřæ■őåžŞæĺąåİŮçŽĎ paramstyle åśđæĂğåŇĚåŘńäžĘåŔĆæŢřåijŢçŤÍéčŐæäijçŽĎä£ąæĄřãĂĆ

åŕźäžŐçőĂå■ŢçŽĎæŢřæ■őåžŞæŢřæ■őçŽĎèŕżåĘŹéŮőécŸïijŇä¡£çŤÍæŢřæ■őåžŞAPIéĂŽåÿÿéİđåÿÿçőĂ
åęĆædIJä¡ăèęĄåd'ĎçŘĘæŻt'åŁăåd'■æİĆçŽĎéŮőécŸïijŇåžžèőőä¡ăä¡£çŤÍæŻt'åŁäéńŸçžğçŽĎæŐěåŔčïijŇæí
çśžäijij SQLAlchemy è£ŹæăůçŽĎåžŞåĚĄèőÿä¡ăä¡£çŤÍPythonçśžæİěèąĺçd'žäÿĂäÿłæŢřæ■őåžŞèąĺïijŇ
åźűäÿŤèČ¡åIJÍéŽŘèŮŘåžŢåśĆSQLçŽĎæČĚåĘţäÿŃåődçŐřåŘĎçğ■æŢřæ■őåžŞçŽĎæŞ■ä¡IJãĂĆ

8.9 6.9 çijŰçăĄåŠŇèğčçăĄå■ĄåĚ■è£ŻåĹűæŢř

éŮőécŸ

ä¡ăæČşåřĘäÿĂäÿłå■ĄåĚ■è£ŻåĹűå■Ůçňęäÿšèğčçă ĄæĹŘäÿĂäÿłå■ŮèŁĆå■ŮçňęäÿšæĹŰèĂĚårĘäÿĂäÿłå

èğčåĘşæŰźæąĹ

åęĆædIJä¡ăåŔłæŸŕçőĂå■ŢçŽĎèğčçăĄæĹŰçijŰçăĄäÿĂäÿłå■ĄåĚ■è£ŻåĹűçŽĎåŐ§ågŃå■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'
>>>
```

çśžäijijçŽĎåŁ§èČ¡åŘŇæăůåŔŕäžěåIJÍ base64 æĺąåİŮäÿ■æL'¿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£ŻåĹűæŸřå¿ĹçőĂå■Ţäÿ
Łéİćäÿďçğ■æŁĂæIJŕçŽĎäÿżèęĄäÿ■åŘŇåIJÍäžŐåď ğářŔåĘŹçŽĎåď ĎçŘĘãĂĆ åĞ¡æŢř `base64.b16decode()` åŠŇ `base64.b16encode()` åŔłèČ¡æŞ■ä¡IJåď ğåĘŹå¡ćåijŔçŽĎå■ĄåĚ■è£ŻåĹűå■Ůæí■ïijŇ èĂŇ `binascii` æÍąåİŮäÿ■çŽĎåĞ¡æŢřåď ğářŔåĘŹéČ¡èČ¡åď ĎçŘĘãĂĆ

è£ŸæIJL'äÿĂçĆźéIJĂèęĄæşÍæĎŔçŽĎæŸŕçijŰçăĄåĞ¡æŢřæL'Ăäžğç Ť§çŽĎè¿ŞåĞžæĂżæŸŕäÿĂäÿłå■Ů
åęĆæđIJæČşåijžåĹűäžěUnicodeå¡ćåijŔè¿ŞåĞžïijŇä¡äéIJĂèęĄåćđåŁääÿĂäÿłéćİåď ŰçŽĎçŢŇéİćæ■ěéłď ãĂĆã

```
>>> h = base64.b16encode(s)
>>> print(h)
b'68656C6C6F'
>>> print(h.decode('ascii'))
68656C6C6F
>>>
```

åIJÍèğčçăĄå■ĄåĚ■è£ŻåĹűæŢřæŮűïijŇåĞ¡æŢř `b16decode()` åŠŇ `a2b_hex()` åŔŕäžěæŐěåŔŮå■ŮèŁĆæĹŰunicodeå■Ůçňęäÿšã ĂĆ ä¡ĘæŸŕïijŇunicodeå■Ůçňęäÿšå£Ěéążäž ĚäžĚåŔłåŇĚåŘńASCIIçijŰçăĄçŽĎå■ĄåĚ■è£ŻåĹűæŢřã ĂĆ

8.10 6.10 çijŰçăĄèğčçăĄBase64æŢřæ■ő

éŮőécŸ

ä¡ăéIJĂèęĄä¡£çŤÍBase64æăijåijŔèğčçăĄæĹŰçijŰçăĄäžŇè£ŻåĹűæŢřæ■õã ĂĆ

èğčåĘşæŰźæąĹ

`base64` æÍąåİŮäÿ■æIJL'äÿď äÿłåĞ¡æŢř `b64encode()` and `b64decode()` åŔŕäžěåÿőä¡ăèğčåĘşè£ŹäÿłéŮőécŸãĂĆä¿ŃåęĆ;

```
>>> # Some byte data
>>> s = b'hello'
>>> import base64
```

(continues on next page)

```
>>> # Encode as Base64
>>> a = base64.b64encode(s)
>>> a
b'aGVsbG8='

>>> # Decode from Base64
>>> base64.b64decode(a)
b'hello'
>>>
```

èőÍèőž

Base64çijŰçăĄäžĚäžĚçŤÍäžŐéÍćåŘŚå■ŮèŁĆçŽĎæŢřæ■őæŕŤåęĆå■ŮèŁĆå■Ůçňęäÿšå
æ■ď'åď'ŰïijŇçijŰçăĄåď'ĎçŘĘçŽĎè¿ŞåĞžçżŞæđIJæĂżæŸŕäÿĂäÿłå■ŮèŁĆå■Ůçňęäÿšã
åęĆæđIJä¡ăæČşæůůåŘĹä¡£çŤÍBase64çijŰçăĄçŽĎæŢřæ■őåŠŇUnicodeæŰĞæIJňïijŇä¡ää£Ëéążæůżå

```
>>> a = base64.b64encode(s).decode('ascii')
>>> a
'aGVsbG8='
>>>
```

å¡Şèğčçă Ą Base64çŽĎæŮűåĂŹïijŇå■ŮèŁĆå■Ůçňęäÿšå
ä¡ĘæŸŕïijŇUnicodeå■Ůçňęäÿšå

8.11 6.11 èŕżåĘŹäžŇè£ŻåĹűæŢřçżĎæŢřæ■ő

éŮőécŸ

ä¡ăæČşèŕżåĘŹäÿĂäÿłäžŇè£ŻåĹűæŢřçżĎçŽĎçżŞæđĎåŇŰæŢřæ■őåĹřPythonåĚČçżĎäÿ■ãĂĆ

èğčåĘşæŰźæąĹ

åŘŕäżěä¡£çŤÍ struct æÍąåİŮåď'ĎçŘĘäžŇè£ŻåĹűæŢřæ■őãĂĆ
äÿŇéÍćæŸŕäÿĂæőţçď'žä¿ŃäžčçăĄåřĘäÿĂäÿłPythonåĚČçżĎåĹŮèąĺåĘŹåĚěäÿĂäÿłäžŇè£ŻåĹűæŰĞäžűïijŇå
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)

```
# 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¿Ĺåd'Žçğ■æŰźæşŢæİèèŕžåŔŰè£ŹäÿłæŰĞäžűåźűè£ŤåŻđäÿĂäÿłåĚČçżĎåĹŮèąĺãĂĆ éęŰåĚĹïijŇåęĆæđIJä¡ăæĽŞçőŮäžěåİŮçŽĎå¡ćåijŔåćđéĞŔèŕžåŔŰæŰĞäžűïijŇä¡ååŔŕäżěè£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'äÿłæŰĞäžűäÿĂæňąæĂğèŕžåŔŰåĹřäÿĂäÿłå■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
        ...
```

äÿd'çğ■æČĚåĘţäÿŇçŽĎçżŞæđIJéČ¡æŸŕäÿĂäÿłåŔŕè£ŤåŻđçŤĺæİèåĹŻåżžèŕěæŰĞäžűçŽĎåŐ§ågŇåĚČçż

èőĺèőž

åŕźäžŐéIJĂèęĄçijŰçăĄåŠŇèğčçăĄäžŇè£ŻåĹűæŢřæ■őçŽĎçĺŃåžŔèĂŇèĺĂïijŇéĂŽåÿÿäijŽä¡£çŤĺ `struct` æĺąåİŮãĂĆ äÿžäžĘåčřæŸŐäÿĂäÿłæŰřçŽĎçżŞæđĎä¡ŞïijŇå Ŕłéİ JĂèęĄåČŔè£ŹæăůåĹŻåżžäÿĂäÿł

Struct åőđä¿Ńå■şåŔřïijŽ

```
# Little endian 32-bit integer, two double precision floats
record_struct = Struct('<idd')
```

çžŞæđĎä¡ŞéĂŽåÿÿäijŽä¡£çŤĺäÿĂäžŻçžŞæđĎçăĄåĂiji, d, fç■L' [åŔĆèĂČ PythonæŰĞæąč]ãĂĆ è£ŹäžŻäžčçăĄåĹĘåĹńäžčèąĺæ§Řäÿłç L'źåőŽçŽĎäžŇè£ŻåĹűæŢřæ■őçśžåđŃåęĆ32ä¡■a çňňäÿĂäÿłå■Ůçňę < æŇĞåőŽäžĘå■ŮèŁĆéąžåžŔãĂĆåIJĺè£Źäÿłä¿Ńå■Řäÿ■ïijŇåőČèąĺçď'žâĂİä¡Őä¡■åIJĺåL'■ æŽť'æŤźè£Źäÿłå■Ůçňęäÿž > èąĺçď'žéńŸä¡■åIJĺåL'■ïijŇæĹŰèĂĚæŸŕ ! èąĺçď'žç¡ŚçžIJå■ŮèŁĆéąžåžŔãĂĆ

äžğçŤ§çŽĎ Struct åőđä¿ŃæIJL'å¿Ĺåď'Žåśđæ Ăğå ŠŇæŰźæşŢçŤĺæİěæŞ■ä¡IJçŽÿåžŤçśžåđŃçŽĎçžŞæđ size åśđæĂğåŇĚåŘńäžĘçžŞæđĎçŽĎå■ŮèŁĆæŢřïijŇè£ŹåIJĺI/OæŞ■ä¡IJæŮűéİđåÿÿæIJL'çŤĺãĂĆ pack() åŠŇ unpack() æŰźæşŢèćńçŤĺæİěæL'ŞåŇĚåŠŇèğčåŇĚæŢřæ■őãĂĆærŤåęĆïijŽ

```
>>> 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\x08@
 ↪'
>>> record_struct.unpack(_)
(1, 2.0, 3.0)
>>>
```

æIJL'æŮűåĂŹä¡ăè£ŸäijŽçIJŃåĹř pack() åŠŇ unpack() æŞ■ä¡IJäžĕæĺąåİŮçžğåĹńåĞ¡æŢřèćńèřČçŤĺïijŇçśžäijijäÿŃéİćè£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\x08@
 ↪'
>>> struct.unpack('<idd', _)
(1, 2.0, 3.0)
>>>
```

è£ŹæăůåŔřäžĕåůĕä¡IJïijŇä¡ĘæŸřæĎ§èğL'æšąæIJL'åőđä¿ŃæŰźæşŢéĆčäźĹäijŸéŻĚïijŇçL'źåĹńæŸřåIJĺà éĂŽè£ĞåĹŻåžžäÿĂäÿł Struct åőđä¿ŃïijŇæăijåijŔäžčçăĄåŔłäijŽæŇĞåőŽäÿĂæňąåźűäÿŤæL'ĂæIJL'çŽĎæ è£ŹæăůäÿĂæİĕäžčçăĄçžť'æŁď'åŕśåŔŸå¿ŮæŽť'åŁăçőÅå■ŢäžĘ(åŻääÿžä¡ăåŔłéIJĂèęĄæŤźåŔŸäÿĂåď'ĎäžčçăĄ

èŕżåŔŰäžŇè£ŻåĹűçžŞæđĎçŽĎäžčçăĄèęĄçŤĺåĹřäÿĂäžŻéİđåÿÿæIJL'èűčèĂŇäijŸç¿ŐçŽĎçijŰçĺŃæŁĂ åIJĺåĞ¡æŢřãĂĂread_records äÿ■ïijŇiter() èćńçŤĺæİěåĹŻåžžäÿĂäÿłè£ŤåŻđåŽžåőŽåď'ğåřŔæŢřæ■őå è£ŹäÿłèŁ■äžčåŹĺäijŽäÿ■æŰ■çŽĎèřČçŤĺäÿĂäÿłçŤĺæĹůæŔŘä¿ŽçŽĎåŔŕèřČçŤĺåŕźèśą(æŕŤåęĆ lambda: f.read(record_struct.size))ïijŇ çŽť'åĹřåőČè£ŤåŻđäÿĂäÿłçL'źæőŁçŽĎåĂij(åęĆbâ

```
>>> f = open('data.b', 'rb')
>>> chunks = iter(lambda: f.read(20), b'')
>>> chunks
```

(continues on next 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£■äžčåŕźèśąçŽĎäÿĂäÿłåŐ§åŻăæŸŕåőČèČ¡åĚĄèőÿä¡£çŤĺäÿĂäÿ
åęĆæđIJä¡ääÿ■ä¡£çŤĺè£Źçğ■æŁĂæIJŕïijŇéĆčäźĹäžčçăĄåŔŕèČ¡äijŽåČŔäÿŇéİćè£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’ğåđŇäžŇè£ŻåĹűæŢřçżĎäÿ■æŔŘåŔŰäžŇè
åŻääÿžåőČäÿ■äijŽäžğçŤ§äżżä¡ŢçŽĎäÿť’æŮűåŕźèśąæĹŰèĂĚè£ŻèąŇåĘĚå■Ÿåď’■åĹűæŞ■ä¡IJãĂĆ
ä¡ăåŘłéIJĂèęĄçżŹåőČäÿĂäÿłå■ŮèŁĆå■Ůçňęäÿš(æĹŰæŢřçżĎ)åŠŇäÿĂäÿłå■ŮèŁĆåĄŔçğžéĞŔïijŇåőČäijŽă

åęĆæđIJä¡ăä¡£çŤĺ `unpack()` æİěäžčæŻ£ `unpack_from()` ïijŇ
ä¡ăéIJĂèęĄä£őæŤźäžčçăĄæİěæđĎéĂăåď’ğéĞŔçŽĎåřŔçŽĎåĹĞçL’Ğäžěåŕŀè£ŻèąŇåĄŔçğžéĞŔçŽĎèőąçőŮ

```
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’äžĘäžčçăĄçIJŃäÿŁåŐżå¿Ĺåď’■æİĆåď’ŰïijŇè£Ÿå¿ŮåĄŽå¿Ĺåď’Žéćİåď’ŰçŽĎåůěä¡
åď’■åĹűæŢřæ■őäżěåŘŁæđĎéĂăåřŔçŽĎåĹĞçL’Ğåŕźèśąã ĂĆ åęĆæđIJä¡ăåĞĘåď’ĞäżŐèŕżåŔŰåĹřçŽĎäÿĂäÿłå
äijŽèąĺçŐřçŽĎæŻť’åĞžèL’šãĂĆ

åIJĺèğčåŇĚçŽĎæŮűåĂŹïijŇ`collections` æĺąåİŮäÿ■çŽĎåŚ¡åŘ■åĚČçżĎåŕźèśąæĹŰèőÿæŸŕä¡ăæČşè
åőČåŔŕäžěèőľ’ä¡ăçżŹè£ŤåŻđåĚČçżĎèő¿ç¡őåšđæĂğåŘ■çğřãĂĆä¿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)
```

åęĆædIJä¡ăçŽĎçÍŃåžŔéIJĂèęĄåď'ĎçŘĘåď'ğéĞŔçŽĎäžŇè£ŻåĹűæŢřæ■őïijŇä¡ăæIJĂåě¡ä¡£çŤÍ
numpy æÍąåİŮãĂĆ ä¿ŃåęĆïijŇä¡ăåŔŕäżěårĘäÿĂäÿłäžŇè£ŻåĹűæŢřæ■őèŕżåŔŰåĹřäÿĂäÿłçżŞædĎåŇŰæŢřçż

```
>>> 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)
>>>
```

æIJĂåŘŐæŔŘäÿĂçĆźïijŇåęĆædIJä¡ăéIJĂèęĄäżŐåůšç§ěçŽĎæŰĞäżűæăijåijŔ(åęĆåŻ¿çL'ĞæăijåijŔïijŇ
åĚĹæčĂæ§ěçIJŃçIJŃPythonæŸŕäÿ■æŸŕåůšçżŔæŔŘä¿ŻäžEçŐŕå■ŸçŽĎæĺąåİŮãĂĆåŻăäÿžäÿ■åĹřäÿĞäÿ■å¿

8.12 6.12 èŕżåŔŰåţŇåěŮåŠŇåŔŕåŔŸéŢ£äžŇè£ŻåĹűæŢřæ■ő

éŮőécŸ

ä¡ăéIJĂèęĄèŕżåŔŰåŇĚåŘńåţŇåěŮæĹŰèĂĚåŔŕåŔŸéŢ£èőŕå¡ŢéŻęåŘĹçŽĎåď■æİĆäžŇè£ŻåĹűæăijåijŔ

èğčåĘşæŰźæąĹ

struct æÍąåİŮåŔŕèćńçŤÍæİěçijŰçăĄ/èğčçăĄåĞäáźŐæL'ĂæIJL'çśżåđŃçŽĎäžŇè£ŻåĹűçŽĎæŢřæ■őçż
æİěèąÍçď'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) ],
]
```

çŐřåIJÍåĄĞèő¿è£ŹäÿłæŢřæ■őèćńçijŰçăĄåĹřäÿĂäÿłäżěäÿŇåĹŮåď't'éČÍåijĂåğŃçŽĎäžŇè£ŻåĹűæŰĞäż

```
+------+--------+-----------------------------------+
|Byte  | Type   |  Description                      |
+======+========+===================================+
|0     | int    |  æŰĞäżűäżčçăĄïijĹ0x1234ïijŇåřŔçńŕïijL'          |
+------+--------+-----------------------------------+
|4     | double |  x çŽĎæIJĂåřŔåĂijïijĹåřŔçńŕïijL'                |
+------+--------+-----------------------------------+
|12    | double |  y çŽĎæIJĂåřŔåĂijïijĹåřŔçńŕïijL'                |
+------+--------+-----------------------------------+
|20    | double |  x çŽĎæIJĂåd'ğåĂijïijĹåřŔçńŕïijL'                 |
```

(continues on next page)

```
+------+--------+-------------------------------------+
|28    | double |  y çŽĎæIJĂåd’ğåĂijïijĺåřŕĢńŕïijL’                |
+------+--------+-------------------------------------+
|36    | int    |  äÿL’èğŠå¡ćæŢřéĞŕïijĺåřŕĢńŕïijL’                |
+------+--------+-------------------------------------+
```

çţ’ğèů§çİĂåď’t’éČĺæŸŕäÿĂçşżåĹŮçŽĎåď’Žè¿źå¡ćèőřå¡ŢïijŇçijŰçăĄæăijåijŔåęĆäÿŃïijŽ

```
+------+--------+-------------------------------------------+
|Byte  | Type   |  Description                              |
+======+========+===========================================+
|0     | int    |  èőřå¡ŢéŢ£åžęïijĺNå■ŮèŁĆïijL’                 ⌋
 ↪   |
+------+--------+-------------------------------------------+
|4-N   | Points |  (X,Y) åİŘæăĞïijŇäžěæţőçĆźæŢřèąÍçd’ž        ⌋
 ↪    |
+------+--------+-------------------------------------------+
```

äÿžäžĘåĘŹè£ŹæăůçŽĎæŰĞäżűïijŇä¡ăåŔŕäžěä¡£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))
```

åřĘæŢřæ■őèŕżåŔŰåŻđæİěçŽĎæŮűåĂŹïijŇåŔŕäžěåĹľçŤĺåĞ¡æŢř `struct.unpack()` ïijŇäžčçăĄå¿ĹçŻÿäijijïijŇå§žæIJňåřśæŸŕäÿŁéİćåĘŹæŞ■ä¡IJçŽĎéĂĘåžŔãĂĆåęĆäÿŃïijŽ

```
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)

```
            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
```

åř¡çőąè£ŹäÿłäžčçăĄåŔŕäžěåůěä¡IJïijŇä¡ĘæŸŕéĞŇéİćæůûæİĆäžĘå¿Ĺåď Žèŕžå ŔŰãĂĄèğčåŇĚæŢřæ■őçž
éĆčæIJłåĚ■äź§åď łçźĄæİĆäžĘçĆźãĂĆåŻäæ■ď å¿ĹæŸ¿çĎűåžŤèŕěæIJL’åŖęäÿĂçğ■èğčåĘşæŰźæşŢåŔŕäžěçő

åIJĺæIJňářŔèŁĆæŐěäÿŇæİěçŽĎéČĺåĹĘïijŇæĹŠäijŽéĂŘæ■ěæijŤçď žäÿĂäÿłæŻť åŁääijŸçğĂçŽĎèğčæć
çŻőæăĞæŸŕåŔŕäžěçżŹçÍŇåžŔåŠŸæŔŘä¿ŻäÿĂäÿłéńŸçžğçŽĎæŰĞäžűæăijåijŔåŇŰæŰźæşŢïijŇåźűçőĂåŇŰ
æIJňářŔèŁĆæŐěäÿŇæİěçŽĎéČĺåĹĘäžčçăĄåžŤèŕěæŸŕæŢť æIJňäźęäÿ■æIJĂåď■æİĆæIJĂéńŸçžğçŽĎä¿Ňå■
äÿĂåőŽèęĄäżŤçżĘçŽĎéŸĚèŕżæĹŠäžňçŽĎèőĺèőžéČĺåĹĘïijŇåŖęåď ŰäźšèęĄåŔĆèĂČäÿŇåĚűäžŰçńäèŁĆåĘ

éęŰåĚĹïijŇå¡Şèŕżå ŔŰå■ŮèŁĆæŢřæ■őçŽĎæŮűåĂŹïijŇéĂŽåÿÿåIJĺæŰĞäžűåijĂåğŇéČĺåĹĘäijŽåŇĚåŘ
åř¡çőąstructæĺąåİŮåŔŕäžěèğčåŇĚè£ŹäžŻæŢřæ■őåĹřäÿĂäÿłåĚČçżĎäÿ■åŐžïijŇåŖęåď ŰäÿĂçğ■èąĺçď žè£Źçğ
åřšåČŔäÿŇéİćè£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£ŹéĞŇæĹŠäžňä¡£çŤĺäžĘäÿĂäÿłæŔŘè£řåŹĺæİěèąĺçď žærŔäÿłçżŞædĎå■ŮæőţïijŇærŔäÿłæŔŘè£řåŹĺåŇ
å■ŸåĆĺåIJĺåĘĚéČĺçŽĎåĘĚå■ŸçijŞåĘšäÿ■ãĂĆåIJĺ `__get__()` æŰźæşŢäÿ■ïijŇstruct.
unpack_from() åĞ¡æŢřèćńçŤĺæİěäžŐçijŞåĘšäÿ■èğčåŇĚäÿĂäÿłåĂijïijŇçIJĄåŐžäžĘéćİåď ŰçŽĎåĹĘçL’Č

Structure çśžåřśæŸŕäÿĂäÿłå§žçąĂçśżïijŇæŐěåŔŮå■ŮèŁĆæŢřæ■őåźűå■ŸåĆĺåIJĺåĘĚéČĺçŽĎåĘĚå■
StructField æŔŘè£řåŹĺä¡£çŤĺãĂĆ è£ŹéĞŇä¡£çŤĺäžĘ memoryview()

ïijŇæĹŚäžňäijŽåIJĺåŘŐéĺćèŕęçżĘèőšèğčåőČæŸŕçŤĺæİěåźšåŸŻçŽĎãĂĆ

ä¡£çŤĺè£Źäÿłäżčçă Ąïij Ňä¡ăçŐřåIJĺåřśèČ¡åőŽäźĽäÿĂäÿłénŸåśĆæňąçŽĎçżŞæđĎåŕźèśąæİěèąĺçd'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■ŘåĹl'çŤĺè£ŹäÿłçśżæİěèŕżåŘŰäźŃåL'■æĹŚäžňåĘŹåĚěçŽĎåd'Žè¿źå¡ćæŢřæ■őçŽĎåd't'

```
>>> 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¿ĹæIJL'èűčïijŇäÿ■è£ĞèŁ£Źçğ■æŰźåijŔè£ŸæŸŕæIJL'äÿĂäžŽçČęäžžçŽĎåIJřæŰźãĂĆéęŰåĚĹïijŇ
ä¡ĘæŸŕè£Źäÿłäżčçă Ąè£ŸæŸŕæIJL'çĆźèĞČèĆ£ïijŇè£ŸéIJĂèęĄä¡£çŤĺèĂĚæŇĞåőŽå¿Ĺåd'ŽåžŢåśĆçŽĎçżĘ
`StructField`ïijŇæŇĞåőŽåĄŔçğžéĞŔç■L')ãĂĆ åŔęåd'ŰïijŇè£ŤåŻđçŽĎçżŞæđIJçśżåŘŇæăůçąőåőđäÿĂä

äżżä¡ŢæŮűåĂŹåŔłèęĄä¡äéĄĞåĹřäžĘåČŔè£Źæăůåę Ůä¡ŹçŽĎçśżåőŽäźĽïijŇä¡ăåžŤèŕěèĂČèŹŚäÿŃä¡£ç
åĚČçśżæIJL'äÿĂäÿłçL'źæĂğåŕśæŸŕåőČèČ¡åd'§èćńçŤĺæİěåąńåĚĚèőÿåd'Žä¡ŐåśĆçŽĎåőđçŐŕçżĘèŁĆïijŇäżŐ
äÿŃéİćæĹŚæİěäÿ¿äÿłä¿Ńå■ŘïijŇä¡£çŤĺåĚČçśżçĺ■å¿őæŤźéĂăäÿŃæĹŚäžňçŽĎ `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)

```
            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¡£çŤĺæŰřçŽĎ `Structure` çśżïijŇä¡ăåŔŕäžěåČŔäýŃéİćè£ŹæăůåőŽäźĽäýĂäýłçżŞæđĎïijŽ

```
class PolyHeader(Structure):
    _fields_ = [
        ('<i', 'file_code'),
        ('d', 'min_x'),
        ('d', 'min_y'),
        ('d', 'max_x'),
        ('d', 'max_y'),
        ('i', 'num_polys')
    ]
```

æ■čąęĆä¡ăæĽĂèğĄïijŇè£ŹæăůåĘŹåřśçőĂå■Ţåď'ŽäžĘãĂĆæĹŚäžňæůżåŁăçŽĎçśżæŰźæşŢ `from_file()` èőľ'æĹŚäžňåIJĺäý■éIJĂèęĄç§ěéĄŞäżżä¡ŢæŢřæ■őçŽĎåď'ğářŔåŠŇçżŞæđĎçŽĎæČĚåĘţäýŃå

```
>>> 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
>>>
```

äýĂæŮęä¡ăåijĂåğŃä¡£çŤĺäžĘåĚČçśżïijŇä¡ăářśåŔŕäžěèőľ'åőČåŔŸå¿ŮæŻť'åŁăæŹžèČ¡ãĂCä¿ŃåęĆïijŇå äýŃéİćæŸŕářźåĽ■éİćåĚČçśżçŽĎäýĂäýłářŔçŽĎæŤźè£ŻïijŇæŔŘä¿ŻäžĘäýĂäýłæŰřçŽĎè¿ĚåŁľ'æŔŔè£řåŹĺa

```
class NestedStruct:
    '''
```

(continues on next 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 æŔŕè£řåŹÍèćńçŤÍæİěåŔäåŁäåŔęåď Űäÿ Ääÿłåő ŽäźĽåIJÍæ§ åőČéĂŽè£ĞåřĘåŐ§åğŃåĘĚå■ŸçijŞåĘšè£ŻèąŇåĽĞçĽĞæŞ■ä¡IJåŘŐåőđä¿ŃåŇŰçżŹåőŽçŽĎçżŞæđĎçśžåđŇ æĽĂäżěè£Źçğ■åĽĞçĽĞæŞ■ä¡IJäÿ■äijŽåijŢåŔŚäżżä¡ŢçŽĎéćİåď ŰçŽĎåĘĚå■Ÿåď ■åĽűãĂĆçŻÿåŔ■ïijŇåőČ åŔęåď ÚïijŇäÿžäžĘéŸšæ■ćéĞ■åď ■åőđä¿ŃåŇŰïijŇéĂŽè£Ğä¡£çŤĺåŠŇ8.10åřŔèŁĆåŘŇæăůçŽĎæŁĂæIJŕïijŇ

ä¡£çŤÍè£ŹäÿłæŰřçŽĎä£őæ■čçĽĹïijŇä¡ăářśåŔŕäžěåČŔäÿŃéİćè£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'äžžæČŁèőűçŽĎæŸŕřijŇåőČäź§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
>>>
```

åĹřçŽőåL'■äÿžæ■ćïijŇäÿĂäÿłåď'ĎçŘĘåőŽéŢ£èőřå¡ŢçŽĎæąĘæđűåůšçżŔåĘŹåě¡äžĘãĂĆä¡ĘæŸŕåęĆæđI
æŕŤåęĆïijŇåď'Žè¿źå¡ćæŰĞäžűåŇĚåŘńåŔŸéŢ£çŽĎéČĺåĹĘãĂĆ

äÿĂçğ■æŰźæąĹæŸŕåĘŹäÿĂäÿłçśżæİěèąÍçď'žå■ŮèŁĆæŢřæ■őïijŇåŘŇæŮűåĘŹäÿĂäÿłåůëåĚůåĞ¡æŢřæ

```
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)

```
    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()` æŸŕäýĂäýłåůěåĚůŕijŇçŤĺæĺěäžŐäýĂäýłæŰĞäžűäý■èŕżåè£ŹäźŕŧæŸŕå¿Ĺåď'ŽæŰĞäžűæăijåijŔåýÿçŤĺçŽĎæŰźåijŔãĂĆä¡IJäýžè¿ŞåĚěïijŇåőČæŐěåŔŮäýĂäýłåŇĚåŘńåŕéĂĽçŽĎ `includes_size` åŔĆæŢřæŇĞåőŽäžĘå■ŮèŁĆæŢřæŸŕåŘęåŇĚåŘńåď'ť éČĺåď'ğářŔãĂĆäýŃéÍćæŸŕäýĂäýłä¿Ńå■ŘæŢŹä¡ăæĂŐæăůä¡£çŤĺäžŐåď'Žè¿źå¡ćæŰĞäžűäý■èŕżåŔŰå■ŢçŃňçŽĎåď'Žè¿źå¡ća

```
>>> 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>]
>>>
```

åŔŕäžěçIJŃåĞžïijŇ`SizedRecord`åőđä¿ŃçŽĎåĘĚåőźè£ŸæšąæIJL'èćńèğčæđŘåĞžæĺěãĂĆ åŔŕäžěä¡£çŤĺ `iter_as()` æŰźæşŢæĺěè¿¿åĹřçŽőçŽĎïijŇè£ŹäýłæŰźæşŢæŐěåŮäýĂäýłçżŞæđĎæăijåijŔål `Structure` çśżä¡IJäýžè¿ŞåĚěãĂĆ è£ŹæăůåŘåŔŕäžěå¿ĹçĄţæťżçŽĎåŐżèğčæđŘæŢřæ■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)

```
(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
>>>
```

åřĘæĽĂæIJL'è£ŹäžŻçżŞåŘĹèţůæİěïijŇäýŃéİćæŸŕäýĂäýł `read_polys()` åĞ¡æŢřçŽĎåŔęåď'ŰäýĂäýłä£őæ■čçĽ'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)

```
    return polys
```

èőÍèőž

è£ŹäÿĂèŁĆåŘŚä¡ăåśŢçd'žäžĘèőÿåd'ŽéńŸçžğçŽĎçijŰçÍŃæŁĂæIJŕïijŇåŇĚæŃňæŔŔè£řåŹÍïijŇåżűè£§ë
çĎűèĂŇïijŇåőČäžňéČ¦äÿžäžĘåŘŇäÿĂäÿłçĽźåőŽçŽĎçŽőæăĞæIJ■åŁąãĂĆ

äÿŁéÍćçŽĎåőđçŐřçŽĎäÿĂäÿłäÿżèęĄçĽźå¿ĄæŸřåőČæŸřå§žäžŐæĞŠèğčåŇĚçŽĎæĂÍæČşãĂĆå¡ŞäÿĂä
`Structure` åőđä¿ŃèćńåĹŻåżžæŮűïijŇ `__init__()` äžĚäžĚåŔłæŸřåĹŻåżžäÿĂäÿłå■ŮèŁĆæŢřæ■őçŽĎå
çĽźåĹńçŽĎïijŇè£ŹæŮűåĂŹåźűæšąæIJĽäżżä¡ŢçŽĎèğčåŇĚæĹŰèĂĚåĚűäžŰäÿŐçżŞæđĎçŻÿåĚşçŽĎæŞ■ä¡
è£ŹæăůåĄŽçŽĎäÿĂäÿłåŁÍæIJżæŸřä¡ăåŔřèČ¡äžĚäžĚåŔłåŕźäÿĂäÿłå■ŮèŁĆèőřå¡ŢçŽĎæ§ŘäÿĂářŔéČĺåĹĘæ

äÿžäžĘåőđçŐřæĞŠèğčåŇĚåŠŇæL'ŞåŇĚïijŇéIJĂèęĄä¡£çŤÍ
`StructField` æŔŔè£řåŹÍçśżãĂĆ çŤÍæĹůåIJÍ `_fields_`
äÿ■åĹŮåĞžæÍěçŽĎæŕŔäÿłåśđæĂğéČ¡äijŻèćńè¡ňåŇŰæĹŘäÿĂäÿł `StructField`
æŔŔè£řåŹÍïijŇ åőČářĘçŻÿåĚşçżŞæđĎæäijåijŔçăĄåŠŇåĄŔçğżåĂijä£Íå■ŸåĹřå■ŸåĆÍçijŞå■Ÿäÿ■ãĆåĚČçś
`StructureMeta` åIJĺåď'ŽäÿłçżŞæđĎçśżèćńåőŽäźĽæŮűèĞłåŁĺåĹŻåżžäžĘè£ŹäžŽæŔŔè£řåŹÍãĂĆ
æĹŚäžňä¡£çŤÍåĚČçśżçŽĎäÿĂäÿłäÿżèęĄåŐ§åŽăæŸřåőČä¡£å¿ŮçŤÍæĹůéÍđåÿÿæŰźä¿£çŽĎéĂŽè£ĞäÿĂäÿłér

`StructureMeta` çŽĎäÿĂäÿłå¿Ĺå¿őåęŹçŽĎåIJřæŰźářśæŸřåőČäijŽåŻžåőŽå■ŮèŁĆæŢřæ■őéąžäžŔãĂ
äź§ářśæŸřèřťïijŇåęĆæđIJäżżæĎŔçŽĎåśđæĂğæŇĞåőŽäžĘäÿĂäÿłå■ŮèŁĆéąžäžŔ(<èąÍçď'žä¡Őä¡■äijŸåĚĹ
æĹŰèĂĚ >èąÍçď'žéńŸä¡■äijŸåĚĹ)ïijŇ éĆčåŘŐéÍćæĽĂæIJĽå■ŮæőţçŽĎéąžäžŔéČ¡äžèè£ŹäÿłéąžäžŔäÿžåĞE
æŕŤåęĆïijŇä¡ăåŔřèČ¡æIJĽäÿĂäžŻæŕŤè¿Čåď'■æÍĆçŽĎçżŞæđĎïijŇářśåČŔäÿŇéÍćè£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') ]
```

äźŇåL'■æĹŚäžňæŔŘåĹřè£ĞïijŇmemoryview() çŽĎä¡£çŤÍåŔŕäžěåÿőåŁl'æĹŚäžňéĄ£åĚ■åĘĚå■ŸçŽĎ
å¡ŞçżŞæđĎå■ŸåIJÍåţŇåěŮçŽĎæŮűåĂŹïijŇmemoryviews åŔŕäžěåŔäåŁäåŘŇäÿĂåĘĚå■ŸåŇžå§§äÿŁåőŽä
è£ŹäÿłçL'źæĂğæŕŤè¿Čå¿őåęŹïijŇä¡ĘæŸřåőČåĚşæşÍçŽĎæŸřåĘĚå■ŸègĘåŻ¿äÿŐæŹőéĂŽå■ŮèŁĆæŢřçżĎçŽ
åęĆæđIJä¡ååIJÍäÿĂäÿłå■ŮèŁĆå■ŮçňęäÿšæĹŰå■ŮèŁĆæŢřçżĎäÿŁæL'ğèąŇåĹĞçL'ĞæŞ■ä¡IJïijŇä¡ăéĂŽåÿÿä
èĂŇåĘĚå■ŸègĘåŻ¿åĹĞçL'Ğäÿ■æŸřè£ŹæăůçŽĎïijŇåőČäžĚäžĚæŸřåIJÍåůšå■ŸåIJÍçŽĎåĘĚå■ŸäÿŁéÍćåŘäă

è£ŸæIJL'å¿Ĺåď'ŽçŻÿåĚşçŽĎçńäèŁĆåŔŕäžěåÿőåŁl'æĹŚäžňæL'l'åśŢè£ŹéĞŇèőÍèőžçŽĎæŰźæąĹãĂĆ
åŔĆèĂČ8.13åřŔèŁĆä¡£çŤÍæŔŔè£řåŹÍæđĎåžžäÿĂäÿłçśżåđŇçşżçż§ãĂĆ
8.10åřŔèŁĆæIJL'æŻť'åď'ŽåĚşäžŐåžűè£§èőąçőŮåśđæĂğåĂijçŽĎèőÍèőžïijŇåźűäÿŤèů§NestedStructæŔŔè£řå
9.19åřŔèŁĆæIJL'äÿĂäÿłä¡£çŤÍåĚČçśżæÍěåĹÍåğŇåŇŰçśżæĹŘåŚŸçŽĎä¿Ńå■ŘïijŇåŠŇ

StructureMeta çśżéİđåÿÿçŻÿäijijãĂĆ PythonçŽĎ ctypes
æžŘçăĄåŘŇæăuäź§å¿ĹæIJL'èűčïijŇåőČæŔŘä¿ŻäžĘåŕźåőŽäźL'æŢřæ■őçżŞæđĎãĂĄæŢřæ■őçżŞæđĎåţŇåěŮ

8.13 6.13 æŢřæ■őçŽĎçť'ŕåŁääÿŐçż§èőąæŞ■ä¡IJ

éŮőécŸ

ä¡äéIJĂèęĄåd'ĎçŘĘäÿĂäÿłå¿Ĺåd'ğçŽĎæŢřæ■őéŻĘåźűéIJĂèęĄèőąçőŮæŢřæ■őæĂżåŠŇæĹŰåĚűäžŰçż

èğčåĘşæŰźæąĹ

åŕźäžŐäżżä¡ŢæűL'åŘŁåĹřçż§èőąãĂĄæŮűéŮť'åžŘåĹŮäżěåŘŁåĚűäžŰçŻÿåĚşæŁĂæIJŕçŽĎæŢřæ■őåĹĘ
PandasåžŞ ãĂĆ

äÿžäžĘèől'ä¡ääĚĹä¡ŞéłŇäÿŃïijŇäÿŃéİćæŸŕäÿĂäÿłä¡£çŤÍPandasæİěåĹĘædŘèŁİåŁääŞěå§ŐåÿĆçŽĎ
èĂĄéijääŠŇåŢőé¡£çśżåŁÍçL'l'æŢřæ■őåžŞ çŽĎä¿Ńå■ŘãĂĆ åIJÍæĹŠåĘŹè£ŹçŕĞæŰĞçńăçŽĎæŮűåĂŹïijŇè£Z

```
>>> 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)

```
>>> 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)

```
>>> 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£ŹéĞŇäÿ■åŔŕèČ¡äżŃ
ä¡ĘæŸŕåŔłèęĄä¡ăéIJĂèęĄåŐżåĹĘæđŘåď'ğåđŃæŢřæ■őéŽĘåŘĹãĂĄåŕźæŢřæ■őåĹĘçżĎãĂĄèőąçőŮåŘĎçğ■

9 çňňäÿČçńäïijŽåĞ¡æŢř

ä¡£çŤÍ def èŕ■åŔěåőŽäźL'åĞ¡æŢřæŸŕæL'ĂæIJL'çÍŃåžŔçŽĎå§žçąĂãĂĆ
æIJñçńäçŽĎçŻőæăĞæŸŕèőšèğčäÿĂäžŻæŻť'åŁăéńŸçžğåŠŇäÿ■åÿÿèğĄçŽĎåĞ¡æŢřåőŽäźL'äÿŐä¡£çŤÍæĺąåijŔ
æűL'åŘŁåĹřçŽĎåĘĚåőźåŇĚæŃňéżŸèőď'åŘĆæŢřãĂĄäżżæĎŘæŢřéĞŘåŘĆæŢřãĂĄåijžåĹűåĚşéŤőå■ŮåŘĆ
åŔęåď'ŰïijŇäÿĂäžŻéńŸçžğçŽĎæŐğåĹűæţĄåŠŇåĹl'çŤĺåŻđèřČåĞ¡æŢřäijäéĂŠæŢřæ■őçŽĎæŁĂæIJŕåIJÍè£Ź

Contents:

9.1 7.1 åŔŕæŐěåŔŮäżżæĎŘæŢřéĞŘåŘĆæŢřçŽĎåĞ¡æŢř

éŮőécŸ

ä¡ăæČşæđĎéĂääÿĂäÿłåŔŕæŐěåŔŮäżżæĎŘæŢřéĞŘåŘĆæŢřçŽĎåĞ¡æŢřãĂĆ

èğčåĘşæŰźæąĹ

äÿžäžĘèČ¡èől'äÿĂäÿłåĞ¡æŢřæŐěåŔŮäżżæĎŘæŢřéĞŘçŽĎä¡■ç¡őåŘĆæŢřïijŇåŔŕäžěä¡£çŤĺäÿĂäÿł*åŘĆ

```
def avg(first, *rest):
    return (first + sum(rest)) / (1 + len(rest))
```

(continues on next page)

(çż■äÿŁéąţ)

```
# Sample use
avg(1, 2) # 1.5
avg(1, 2, 3, 4) # 2.5
```

åIJÍè£ŹäÿłäżÑå■Řäÿ■ïijŇrestæŸŕçŤśæL'ĂæIJL'åĚűäžŰä¡■ç¡őåŔĆæŢřçżĎæĹŘçŽĎåĚČçżĎãĂĆçĎűåŘ

äÿžäžĘæŐěåŔŮäżżæĎŔæŢřéĞŘçŽĎåĚşéŤőå■ŮåŔĆæŢřïijŇä¡£çŤÍäÿĂäÿłäžě**åijĂåď'çŽĎåŔĆæŢřã

```
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£ŹéĞŇïijŇattrsæŸŕäÿĂäÿłåŇĚåŘńæL'ĂæIJL'èćńäijäåĚěè£ŻæİęçŽĎåĚşéŤőå■ŮåŔĆæŢřçŽĎå■ŮåĚ

åęĆæđIJä¡ăè£ŸåÿŇæIJŻæ§ŘäÿłåĞ¡æŢřèČ¡åŘŇæŮűæŐěåŔŮäżżæĎŔæŢřéĞŘçŽĎä¡■ç¡őåŔĆæŢřåŠŇål

```
def anyargs(*args, **kwargs):
    print(args) # A tuple
    print(kwargs) # A dict
```

ä¡£çŤÍè£ŹäÿłåĞ¡æŢřæŮűïijŇæL'ĂæIJL'ä¡■ç¡őåŔĆæŢřäijŽèćńæŤ¿åĹřargsåĚČçżĎäÿ■ïijŇæL'ĂæIJL'åĚş

èőÍèőž

äÿĂäÿł*åŔĆæŢřåŔłèČ¡åĞžçŐřåIJÍåĞ¡æŢřåőŽäźL'äÿ■æIJĂåŘŐäÿĂäÿłä¡■ç¡őåŔĆæŢřåŘŐéİćïijŇèĂŇ
**åŔĆæŢřåŔłèČ¡åĞžçŐřåIJÍæIJĂåŘŐäÿĂäÿłåŔĆæŢřãĆ æIJL'äÿĂçĆźèęĄæşlæĎŔçŽĎæŸŕïijŇåIJÍ*åŔĆæ

```
def a(x, *args, y):
    pass

def b(x, *args, y, **kwargs):
    pass
```

è£Źçğ■åŔĆæŢřåřśæŸŕæĹŚäžňæL'Ăèŕť'çŽĎåijžåĹűåĚşéŤőå■ŮåŔĆæŢřïijŇåIJÍåŘŐéİć7.2åřŔèŁĆè£Ÿäij

9.2 7.2 åŔłæŐěåŔŮåĚşéŤőå■ŮåŔĆæŢřçŽĎåĞ¡æŢř

éŮőécŸ

ä¡ăåÿŇæIJŻåĞ¡æŢřçŽĎæ§ŘäžŻåŔĆæŢřåijžåĹűä¡£çŤĺåĚşéŤőå■ŮåŔĆæŢřäijăéĂŠ

èğčåĘşæŰźæąĹ

ářĘåijžåĹűåĚşéŤőå■ŮåŔĆæŢřæŤ¿åĹřæ§Řäÿł*åŔĆæŢřæĹŰèĂĚå■Ţäÿł*åŘŐéİćåřśèČ¡è¿¿åĹřè£Źçğ■æ

```
def recv(maxsize, *, block):
    'Receives a message'
    pass

recv(1024, True) # TypeError
recv(1024, block=True) # Ok
```

åĹľçŤĺè£Źçğ■æŁĂæIJřijŇæĹŚäžňè£ŸèČ¡åIJĺæŐěåŔŮäżżæĎŔåd'Žäÿłä¡■ç¡őåŔĆæŢřçŽĎåĞ¡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
```

èőÍèőž

å¿Ĺåd'ŽæČĚåĘţäÿŃïijŇä¡£çŤĺåijžåĹűåĚşéŤőå■ŮåŔĆæŢřäijŽæŕŤä¡£çŤĺä¡■ç¡őåŔĆæŢřèąĺæĎŔæŻť'åŁă
ä¿ŇåęĆïijŇèĂČèŹŠäÿŇåęĆäÿŇäÿĂäÿłåĞ¡æŢřèřČçŤĺïijŽ

```
msg = recv(1024, False)
```

åęĆædIJèřČçŤĺèĂĚårźrecvåĞ¡æŢřåźűäÿ■æŸŕå¿ĹçĘ§æĆL'ïijŇéĆčäžŰèĆŕåőŽäÿ■æŸŐçŹ¡éĆčäÿłFalseå
ä¡ĘæŸŕïijŇåęĆædIJäžčçăĄåŔŸæĹŘäÿŃéİćè£Źæăůå■ŘçŽĎèřĺåřśæÿĚæěŽåd'ŽäžĘïijŽ

```
msg = recv(1024, block=False)
```

åŘęåd'ŰïijŇä¡£çŤĺåijžåĹűåĚşéŤőå■ŮåŔĆæŢřäź§äijŽæŕŤä¡£çŤĺ**kwargsåŔĆæŢřæŻť'åě¡ïijŇåŻääÿžåIJ

```
>>> help(recv)
Help on function recv in module __main__:
recv(maxsize, *, block)
    Receives a message
```

åijžåĹűåĚşéŤőå■ŮåŔĆæŢřåIJĺäÿĂäžŻæŻťéńŸçžğåIJžåŘĹåŘŇæăůäź§å¿ĹæIJL’çŤĺãĂĆ
ä¿ŇåęĆïijŇåőČäžňåŔŕäžěèćńçŤĺæĺěåIJĺä¡£çŤĺ*argsåŠŇ**kwargsåŔĆæŢřä¡IJäÿžè¿ŞåĚěçŽĎåĞ¡æŢřäÿ■æŔŠ

9.3 7.3 çżŹåĞ¡æŢřåŔĆæŢřåćđåŁääĚČä£ąæĄŕ

éŮőécŸ

ä¡ääĘŹåě¡äžĘäÿĂäÿłåĞ¡æŢřijŇçĎűåŘŐæČşäÿžè£ŹäÿłåĞ¡æŢřçŽĎåŔĆæŢřåćđåŁääÿĂäžŻécİåď’ŰçŽĎ

èğčåĘşæŰźæąĹ

ä¡£çŤĺåĞ¡æŢřåŔĆæŢřæşĺèğčæŸŕäÿĂäÿłå¿Ĺåě¡çŽĎåŁđæşŢïijŇåőČèČ¡æŔŘçd’žçĺŇåžŔåŚŸåžŤèŕěæĂŐ
ä¿ŇåęĆïijŇäÿŇéİćæIJL’äÿĂäÿłècńæşĺèğčäžĘçŽĎåĞ¡æŢřijŽ

```
def add(x:int, y:int) -> int:
    return x + y
```

pythonèğčéĞŁåŹĺäÿ■äijŽåŕźè£ŹäžŻæşĺèğčæůžåŁääżżä¡ŢçŽĎèŕ■äźL’ãĂĆåőČäžňäÿ■äijŽècńçśžådŇæčĂ
çĎűèĂŇïijŇåŕźäžŐéĆčäžŻéŸĚèŕżæžŘçăĄçŽĎäžżæİěèőšåřšå¿ĹæIJL’åÿőåŁľåŢęãĂĆçńňäÿL’æŰźåůěåĚůåŠŇ

```
>>> help(add)
Help on function add in module __main__:
add(x: int, y: int) -> int
>>>
```

åř¡çőąä¡ääŔŕäžěä¡£çŤĺäżżæĎŘçśžådŇçŽĎåŕźèšąçżŹåĞ¡æŢřæůžåŁääşĺèğč(ä¿ŇåęĆæŢřå■ŮïijŇå■Ůçňęä

èőĺèőž

åĞ¡æŢřæşĺèğčåŔłå■ŸåĆĺåIJĺåĞ¡æŢřçŽĎ __annotations__
åšđæĂğäÿ■ãĂĆä¿ŇåęĆïijŽ

```
>>> add.__annotations__
{'y': <class 'int'>, 'return': <class 'int'>, 'x': <class 'int'>}
```

åř¡çőąæşĺèğčçŽĎä¡£çŤĺæŰźæşŢåŔŕèČ¡æIJL’å¿Ĺåď’Žçğ■ïijŇä¡ĘæŸŕåőČäžňçŽĎäÿżèęĄçŤĺéĂŤè£ŸæŸŕ
åŻääÿžpythonåźűæšąæIJL’çśžådŇåčřæŸŐïijŇéĂŽåÿÿæİěèőšäżĚäżĚéĂŽè£ĞéŸĚèŕżæžŘçăĄå¿ĹéŽ¿ç§ěéĄŞå
è£ŹæŮűåĂŹä¡£çŤĺæşĺèğčåřśèČ¡çżŹçĺŇåžŔåŚŸæŻťåď’ŽçŽĎæŔŘçd’žïijŇèőľäžŰäžňåŘŕäžěæ■čçąőçŽĎä¡£ç

åŔĆèĂČ9.20åřŘèŁĆçŽĎäÿĂäÿłæŻťåŁäéńŸçžğçŽĎä¿Ňå■ŘïijŇæijŤçd’žäžĘåęĆä¡ŢåĹľçŤĺæşĺèğčæİěåč

9.4 7.4 è£ŤåŻđåď’Žäÿłå Äijç ŽĎåĞ¡æŢř

éŮőécŸ

ä¡ääÿŇæIJŻæđĎéĂääÿĂäÿłåŘŕäžěè£ŤåŻđåď’Žäÿłå ÄijçŽĎåĞ¡æŢř

èğčåĘşæŰźæąĹ

äÿžäžĘèČ¡è£ŤåŻđåd'ŽäÿłåĂijïijŇåĞ¡æŢřçŻt'æŐěreturnäÿĂäÿłåĚČçżĎářśèąŇäžĘãĂĆä¿ŃåęĆïijŽ

```
>>> def myfun():
... return 1, 2, 3
...
>>> a, b, c = myfun()
>>> a
1
>>> b
2
>>> c
3
```

èőÍèőž

åř¡çőąmyfun()çIJŃäÿŁåŐżè£ŤåŻđäžĘåd'ŽäÿłåĂijïijŇåőđéŹĚäÿŁæŸŕåĚĹåĹŻåżžäžĘäÿĂäÿłåĚČçżĎçĎu
è£Źäÿłèí■æşŢçIJŃäÿŁåŐżæŕŤè¿ČåěĞæĂłïijŇåőđéŹĚäÿŁæĹŚäžňä¡£çŤĺçŽĎæŸŕéĂŮåŔůæİěçŤ§æĹŘäÿĂäÿ

```
>>> a = (1, 2) # With parentheses
>>> a
(1, 2)
>>> b = 1, 2 # Without parentheses
>>> b
(1, 2)
>>>
```

å¡ŞæĹŚäžňèřČçŤĺè£ŤåŻđäÿĂäÿłåĚČçżĎçŽĎåĞ¡æŢřçŽĎæŮűåĂŹ
ïijŇéĂŽåÿÿæĹŚäžňäijŽåřĘçżŞæđIJèţŃåĂijçżŹåd'ŽäÿłåŔŸéĞŔïijŇåřśåČŔäÿŁéİćçŽĎéĆčæăůãĂĆ
åĚűåőđè£ŹåřśæŸŕ1.1åřŔèŁĆäÿ■æĹŚäžňæĽĂèŕt'çŽĎåĚČçżĎèğčåŇĚãĂĆè£ŤåŻđçżŞæđIJäź§åŔŕäžěèţŃåĂij
è£ŹæŮűåĂŹè£ŹäÿłåŔŸéĞŔåĂijåřśæŸŕåĞ¡æŢřè£ŤåŻđçŽĎéĆčäÿłåĚČçżĎæIJňèžńäžĘïijŽ

```
>>> x = myfun()
>>> x
(1, 2, 3)
>>>
```

9.5 7.5 åőŽäźĽælJĽéżŸèőd'åŔĆæŢřçŽĎåĞ¡æŢř

éŮőécŸ

ä¡ăæČşåőŽäźĽäÿĂäÿłåĞ¡æŢřæĹŰèĂĚæŰźæşŢïijŇåőČçŽĎäÿĂäÿłæĹŰåd'ŽäÿłåŔĆæŢřæŸŕåŔŕéĂĽ'çŽ

èğčåĘşæŰźæąĹ

åőŽäźĽäÿĂäÿłæIJL'åŔŕéĂL'åŔĆæŢřçŽĎåĞ¡æŢřæŸŕéİđåÿÿçőĂå■ŢçŽĎïijŇçŻt'æŐěåIJĺåĞ¡æŢřåőŽäźĽ

```
def spam(a, b=42):
    print(a, b)

spam(1) # Ok. a=1, b=42
spam(1, 2) # Ok. a=1, b=2
```

åęĆædIJéżŸèőd'åŔĆæŢřæŸŕäÿĂäÿłåŔŕä£őæŤźçŽĎåőźåŹĺæŕŤåęĆäÿĂäÿłåĹŮèąĺãĂĄéŻĘåŘĹæĹŰèĂĚ

```
# Using a list as a default value
def spam(a, b=None):
    if b is None:
        b = []
    ...
```

åęĆædIJä¡ăåźűäÿ■æČşæŔŘä¿ŻäÿĂäÿłéżŸèőd'åĂijïijŇèĂŇæŸŕæČşäżĚäżĚæţŃèŕŢäÿŃæ§ŘäÿłéżŸèőd'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
>>>
```

äżŤçżĘèğĆår§åŔŕäżěåŘŚçŐřåĹřäijăéĂŠäÿĂäÿłNoneåĂijåŠŇäÿ■äijăåĂijäÿd'çğ■æČĚåĘţæŸŕæIJL'åůőåĹ

èőÍèőž

åőŽäźĽåÿęéżŸèőd'åĂijåŔĆæŢřçŽĎåĞ¡æŢřæŸŕå¿ĹçőĂå■ŢçŽĎïijŇä¡EçżÍäÿ■äżĚäżĚåŔłæŸŕè£ŹäÿłïijŇ

éęŰåĚĹïijŇéżŸèőd'åŔĆæŢřçŽĎåĂijäżĚäżĚåIJĺåĞ¡æŢřåőŽäźĽçŽĎæŮűåĂŹèţŃåĂijäÿĂæñąãĂĆèŕŢçĺ

```
>>> x = 42
>>> def spam(a, b=x):
...     print(a, b)
...
>>> spam(1)
1 42
>>> x = 23 # Has no effect
```

(continues on next page)

```
>>> spam(1)
1 42
>>>
```

æşĺæĎŔåĹřå¡ŞæĹŚäžňæŤźåŔŸxçŽĎåĂijçŽĎæŮűåĂŹåŕźéżŸèŏď'åŔĆæŢřåĂijåźűæšąæIJL'å¡śåŞ■ïijŇè

åĚűæňąïijŇéżŸèŏď'åŔĆæŢřçŽĎåĂijåžŤèŕěæŸŕäÿ■åŔŕåŔŸçŽĎåŕźèśąïijŇæŕŤåęĆNoneãĂĄTrueãĂĄFal
çĽźåĹńçŽĎïijŇå■ČäÿĞäÿ■èęĄåČŔäÿŃéİćè£Źæăůå EŹäžčçăĄïijŽ

```
def spam(a, b=[]): # NO!
    ...
```

åęĆæđIJä¡ăè£ŹäźĹåĄŽäžĘïijŇå¡ŞéżŸèŏď'åĂijåIJĺåĚűäżŮåIJřæŰźèćńä£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■æČĚåĘţçŽĎåŔŞçŤ§ïijŇæIJĂå
çĎůåŘŐåIJĺåĞ¡æŢřéĞŇéİćæčĂæ§ěåőČïijŇåL'■éİćçŽĎä¿Ńå■ŘåŕśæŸŕè£Źæăůå ĄŽçŽĎãĂĆ

åIJĺæţŃèŕŢNoneåĂijæŮűä¡£çŤĺ is æŞ■ä¡IJçňęæŸŕå¿ĹéĞ■èęĄçŽĎïijŇäź§æŸŕè£Źçğ■æŰźæąĹçŽĎåĚş
æIJL'æŮűåĂŹåđ'ğåőűäijŽçŁŕäÿŃäÿŃéİćè£ŹæăůçŽĎéŤŹèŕŕïijŽ

```
def spam(a, b=None):
    if not b: # NO! Use 'b is None' instead
        b = []
    ...
```

è£ŹäźĹåĘŹçŽĎéŮőécŸåIJĺäžŐår¡çőąNoneåĂijçąőåőđæŸŕècńå¡ŞæĹŘFalseïijŇ
ä¡ĘæŸŕè£ŸæIJL'åĚűäżŰçŽĎårźèśą(æŕŤåęĆéŢ£åžęäÿž0çŽĎå■Ůçňęäÿšă ĂĄåĹŮèąĺãĂĄåĚČçżĎãĂĄå■ŮåĚÿ
åŽăæ■đ'ïijŇäÿŁéİćçŽĎäžčçăĄäijŽèŕŕåŕĘäÿĂäžŽåĚűäżŰè¿ŞåĚěäź§å¡ŞæĹŘæŸŕæšąæIJL'è¿ŞåĚěãĆćæŕŤåęĆ

```
>>> 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¡æŢřéIJĂèęĄæţŃèŕŢæ§Řäÿłå
è£ŹæŮűåĂŹéIJĂèęĄåřŘå£ČçŽĎæŸŕä¡ääÿ■èČ¡çŤĺæ§ŘäÿłézŸèőď åĂijæŕŤåęĆNoneãĂĄ
0æĹŰèĂĚFalseåĂijæİěæţŃèŕŢçŤĺæĹůæŘŘä¿ŻçŽĎåĂij(åŽăäÿžè£ŹäžŻåĂijéČ¡æŸŕåŘĹæşŢçŽĎåĂijïijŇæŸ
åŽăæ■ď ïijŇä¡ăéIJĂèęĄåĚűäžŰçŽĎèğčåĘşæŰźæąĹãĂĆ

äÿžäžĘèğčåĘşè£ŹäÿłéŮőéćŸïijŇä¡ääåŘŕäżěåĹŻåżžäÿĂäÿłçŃňäÿĂæŮăäžŇçŽĎçğĄæIJĽåŕźèśąåőđä¿Ńïij
åIJÍåĞ¡æŢřéĞŃéİćïijŇä¡ääåŘŕäżěéĂŽè£ĞæčĂæ§ěèćńäijăéĂŠåŘĆæŢřåĂijèů§è£Źäÿłåőđä¿ŃæŸŕåŘęäÿĂæăůæ
è£ŹéĞŃçŽĎæĂİèůŕæŸŕçŤĺæĹůäÿ■åŘŕèČ¡åŐżäijăéĂŠè£Źäÿł_no_valueåőđä¿Ńä¡IJäÿžè¿ŞåĚěãĂĆ
åŽăæ■ď ïijŇè£ŹéĞŃéĂŽè£ĞæčĂæ§ěè£ŹäÿłåĂijåřśèČ¡çąőåőŽæ§ŘäÿłåŘĆæŢřæŸŕåŘęèćńäijăéĂŠè£ŹæİěäžĘ

è£ŹéĞŃåŕź `object()` çŽĎä¡£çŤĺçIJŃäÿŁåŐżæIJĽçĆźäÿ■åď łåÿÿèğĄãĂĆ`object`
æŸŕpythonäÿ■æĽĂæIJĽçśżçŽĎå§žçśżãĂĆ ä¡ääåŘŕäżěåĹŻåżž `object`
çśżçŽĎåőđä¿ŃïijŇä¡ĘæŸŕè£ŹäžŻåőđä¿ŃæšąäżĂäźĹåőđéŹĚçŤĺå'ĎïijŇåŽăäÿžåőČåźűæšąæIJĽäżżä¡ŢæIJĽ
äź§æšąæIJĽäżżä¡Ţåőđä¿ŃæŢřæ■ő(åŽăäÿžåőČæšąæIJĽäżżä¡ŢçŽĎåőđä¿Ńå■ŮåĚÿïijŇä¡ăçŤŽèĞşéČ¡äÿ■èČ¡
ä¡ääåŤřäÿĂèČ¡åĄŽçŽĎåřśæŸŕæţŃèŕŢåŘŇäÿĂæĂğãĂĆè£ŹäÿłåĹŽåě¡çňęåŘĹæĹŚçŽĎèęĄæśĆïijŇåŽăäÿžæĹ

9.6 7.6 åőŽäźĽåŇ£åŘ■æĹŰåĘĚèĄŤåĞ¡æŢř

éŮőéćŸ

ä¡ăæČşäÿž `sort()` æŞ■ä¡IJåĹŻåżžäÿĂäÿłå¿Ĺç§■çŽĎåŽđèřČåĞ¡æŢřïijŇä¡ĘåŘĹäÿ■æČşçŤĺ
`def` åŐżåĘŹäÿĂäÿłå■ŢèąŇåĞ¡æŢřïijŇ èĂŇæŸŕäÿŇæIJŻéĂŽè£Ğæ§Řäÿłå£ńæ■ůæŰźåijŔäżěåĘĚèĄŤæŰźåij

èğčåĘşæŰźæąĹ

å¡ŞäÿĂäžŻåĞ¡æŢřå¿ĹçőĂå■ŢïijŇäżĚäżĚåŔłæŸŕèőąçőŮäÿĂäÿłèąĺè¿¿åijŔçŽĎåĂijçŽĎæŮűåĂŹïijŇåřś

```
>>> add = lambda x, y: x + y
>>> add(2,3)
5
>>> add('hello', 'world')
'helloworld'
>>>
```

è£ŹéĞŃä¡£çŤĺçŽĎlambdaèąĺè¿¿åijŔèů§äÿŃéİćçŽĎæŢĹæđIJæŸŕäÿĂæăůçŽĎïijŽ

```
>>> def add(x, y):
...     return x + y
...
>>> add(2,3)
5
>>>
```

lambdaèąĺè¿¿åijŔåĚÿåđŃçŽĎä¡£çŤĺåIJžæŹŕæŸŕæŐŠåžŔæĹŰæŢřæ■őreduceç■ĽïijŽ

```
>>> names = ['David Beazley', 'Brian Jones',
...          'Raymond Hettinger', 'Ned Batchelder']
>>> sorted(names, key=lambda name: name.split()[-1].lower())
```

(continues on next page)

```
['Ned Batchelder', 'David Beazley', 'Raymond Hettinger', 'Brian␣
↪Jones']
>>>
```

èőÍèőž

åř¡çőąlambdaèąĺè¿¿åijŔåĚĄèőÿä¡ăåőŽäźL’çőĂå■ŢåĞ¡æŢřïijŇä¡ĘæŸŕåőČçŽĎä¡£çŤÍæŸŕæIJL’éŹŘåĹűç
ä¡ăåŘłèČ¡æŇĞåőŽå■Ţäÿłèąĺè¿¿åijŔïijŇåőČçŽĎåĂijåřśæŸŕæIJĂåŘŐçŽĎè£ŤåŽđåĂijãĂĆäź§åřśæŸŕèŕť’äÿ■è
åŇĚæŃňåď’Žäÿłèŕ■åŘěãĂĄæİąäžűèąĺè¿¿åijŔãĂĄè£■äžčäžěåŘĽåijĆåÿÿåď’ĎçŘĘç■L’ç■L’ãĂĆ

ä¡ăåŘŕäžěäÿ■ä¡£çŤÍlambdaèąĺè¿¿åijŔåřśèČ¡çijŰåĘŹåď’ğéČĺåĹĘpythonäžčçăĄãĂĆ
ä¡ĘæŸŕïijŇå¡ŞæIJL’äžžçijŰåĘŹåď’ğéĞŔèőąçőŮèąĺè¿¿åijŔåĂijçŽĎç§■åřŔåĞ¡æŢřæĹŰèĂĚéIJĂèęĄçŤÍæĹůa
ä¡ăåřśäijŽçIJŃåĹřlambdaèąĺè¿¿åijŔçŽĎèžńå¡śäžĘãĂĆ

9.7 7.7 åŇ£åŘ■åĞ¡æŢřæ■ŢèŐůåŔŸéĞŔåĂij

éŮőécŸ

ä¡ăçŤÍlambdaåőŽäźL’äžĘäÿĂäÿłåŇ£åŘ■åĞ¡æŢřïijŇåźűæČşåIJÍåőŽäźL’æŮűæ■ŢèŐůåĹřæ§ŘäžŽåŔŸéĞ

èğčåĘşæŰźæąĹ

åĚĹçIJŃäÿŃäÿŃéİćäžčçăĄçŽĎæŢĹæđIJïijŽ

```
>>> x = 10
>>> a = lambda y: x + y
>>> x = 20
>>> b = lambda y: x + y
>>>
```

çŐřåIJÍæĹŚéŮőä¡ăïijŇa(10)åŠŇb(10)è£ŤåŽđçŽĎçżŞæđIJæŸŕäžĂäźĹïij§åęĆæđIJä¡àèőd’äÿžçżŞæđIJæŸ

```
>>> a(10)
30
>>> b(10)
30
>>>
```

è£ŹåĚűäÿ■çŽĎåěěåęŹåIJÍäžŐlambdaèąĺè¿¿åijŔäÿ■çŽĎxæŸŕäÿĂäÿłèĞłçŤšåŔŸéĞŔïijŇ
åIJÍè£ŘèąŇæŮűçżŚåőŽåĂijïijŇèĂŇäÿ■æŸŕåőŽäźL’æŮűåršçżŚåőŽïijŇè£Źèů§åĞ¡æŢřçŽĎéżŸèőd’åĂijåŔĆæ
åŽăæ■d’ïijŇåIJÍèřČçŤÍè£Źäÿłlambdaèąĺè¿¿åijŔçŽĎæŮűåĂŹïijŇxçŽĎåĂijæŸŕæL’ğèąŇæŮűçŽĎåĂijãĂĆä¿Ň

```
>>> x = 15
>>> a(10)
25
```

(continues on next page)

```
>>> x = 3
>>> a(10)
13
>>>
```

åęĆæđIJä¡ăæČşèőľæ§ŘäÿłåŇ£åŘ■åĞ¡æŢřåIJĺåőŽäźĽæŮűåřśæ■ŢèŐůåĹřåĂijïijŇåŕřäžěářĘéĆčäÿłåŕĆ

```
>>> x = 10
>>> a = lambda y, x=x: x + y
>>> x = 20
>>> b = lambda y, x=x: x + y
>>> a(10)
20
>>> b(10)
30
>>>
```

èőĺèőž

åIJĺè£ŹéĞŇåĹŮåĞžæİěçŽĎéŮőécŸæŸŕæŰřæĽŃå¿Ĺåőźæ ŸŞçŁŕçŽĎéŤŹèŕŕïijŇæIJĽäžŻæŰřæĽŃåŕřè
ærŤåęĆïijŇéĂŽè£ĞåIJĺäÿĂäÿłå¿łçŐřæĹŰåĹŮèąĺæŐĺåŕijäÿ■åĹŻåžžäÿĂäÿłlambdaèąĺè¿¿åijŕåĹŮèąĺïijŇåźűa

```
>>> 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£■äžčçŽĎæIJĂåŘŐäÿĂäÿłåĂijãĂĆçŐřåIJĺæĹŚă

```
>>> 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¡£çŤĺåĞ¡æŢřéżŸèőď åĂijåŔĆæŢřå¡ćåijŔïijŇlambdaåĞ¡æŢřåIJĺåőŽäźĽæŮűåřśèČ¡çżŚåőŽåĹřå

9.8 7.8 åĞŔåřŚåŔŕèřČçŤĺåŕźèśąçŽĎåŔĆæŢřäÿłæŢř

éŮőéćŸ

ä¡ăæIJL'äÿĂäÿłèćńåĚűäżŰpythonäżčçăĄä¡£çŤĺçŽĎcallableåŕźèśąïijŇåŔŕèČ¡æŸŕäÿĂäÿłåŻđèřČåĞ¡æŢřä¡ĘæŸŕåőČçŽĎåŔĆæŢřåd'łåď'ŽäžĘïijŇåŕijèĞt'èřČçŤĺæŮűåĞžéŤŹãĂĆ

èğčåĘşæŰźæąĹ

åęĆæđIJéIJĂèęĄåĞŔåřŚæ§ŘäÿłåĞ¡æŢřçŽĎåŔĆæŢřäÿłæŢřïijŇä¡ăåŔŕäżěä¡£çŤĺ `functools.partial()` ãĂĆ `partial()` åĞ¡æŢřåĚĄèőÿä¡ăçżŹäÿĂäÿłæĹŰåd'ŽäÿłåŔĆæŢřèő¿ç¡őåŻžäÿžäžĘæijŤçd'žæÿĚæěŽïijŇåĄĞèő¿ä¡ăæIJL'äÿŃéİćè£ŹæăůçŽĎåĞ¡æŢřïijŽ

```
def spam(a, b, c, d):
    print(a, b, c, d)
```

çŐřåIJĺæĹŚäżňä¡£çŤĺ `partial()` åĞ¡æŢřæİěåŻžåőŽæ§ŘäžŻåŔĆæŢřåĂijïijŽ

```
>>> 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()` åŻžåőŽæ§ŘäžŻåŔĆæŢřåźűè£ŤåŻđäÿĂäÿłæŰřçŽĎcallableåŕźèśąãĂĆè£Źã çĎűåŘŐèů§äźŃåĽ■åůšçżŔèţŃåĂijè£ĞçŽĎåŔĆæŢřåŘĹåźűèţůæİěïijŇæIJĂåŘŐåřĘæĽĂæIJL'åŔĆæŢřäijäéĂ

èőĺèőž

æIJňèŁĆèęĄèğčåĘşçŽĎéŮőéćŸæŸŕèőľ'åŐ§æIJňäÿ■åĚijåőźçŽĎäżčçăĄåŔŕäżěäÿĂèţůåůěä¡IJãĂĆäÿŃéİ

çňňäÿĂäÿłä¿Ńå■ŘæŸŕïijŇåĄĞèő¿ä¡ăæIJL'äÿĂäÿłçĆźçŽĎåĹŮèąĺæİěèąĺçd'ž(x,y)åİŘæăĞåĚĈçżĎãĂĆ ä¡ăåŔŕäżěä¡£çŤĺäÿŃéİćçŽĎåĞ¡æŢřæİěèőąçőŮäÿd'çĆźäźŃéŮt'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¡ăæČşäžěæ§Řäył̨çĆźäÿžå§žçĆźïijŇæăźæ■őçĆźåŠŇå§žçĆźäźŃéŮťçŽĎèůİçężæİěæŐŠå åĹŮèąÍçŽĎ `sort()` æŰźæşŢæŐěåŔŮäÿĂäÿłåĚşéŤőå■ŮåŔĆæŢřæİěèĞłåőŽäźĽæŐŠåžŔéĂżè¿ŚïijŇ ä¡ĘæŸŕåőČåŔłèČ¡æŐěåŔŮäÿĂäÿłå■ŢäÿłåŔĆæŢřçŽĎåĞ¡æŢř(distance()å¿ĹæŸŐæŸ¿æŸŕäÿ■çňęåŘĹæİąäżűçŐřåIJÍæĹŚäżňåŔŕäžěéĂŽè£Ğä¡£çŤÍ `partial()` æİěèğčåĘşè£ŹäÿłéŮőécŸïijŽ

```
>>> pt = (4, 3)
>>> points.sort(key=partial(distance,pt))
>>> points
[(3, 4), (1, 2), (5, 6), (7, 8)]
>>>
```

æŻť'è£ŻäÿĂæ■ěïijŇ`partial()` éĂŽåÿÿèćńçŤÍæİěå¿őèřČåĚűäžŮåžŞåĞ¡æŢřæĽĂä¡£çŤÍçŽĎåŻđèřČå ä¿ŃåęĆïijŇäÿŃéİćæŸŕäÿĂæőţäžčçăĄïijŇä¡£çŤÍ `multiprocessing` æİěåijĆæ■čèőąçőŮäÿĂäÿłçżŞæđIJåĂijïijŇ çĎűåŘŐè£ŹäÿłåĂijèćńäijåéĂŠçżŹäÿĂäÿłæŐěåŘŮäÿĂäÿłresultåŹ

```
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()` æŔŘä¿ŻåŻđèřČåĞ¡æŢřæŮűïijŇéĂŽè£Ğä¡£çŤÍ `partial()` äijåéĂŠéćİåď'ŰçŽĎ `logging` åŔĆæŢřãĂĆ èĂŇ `multiprocessing` åŕźè£ŹäžŻäÿĂæŮăæĽĂç§ěâĂŤâĂŤåőČäżĚäżĚåŘłæŸŕä¡£çŤÍå■ŢäÿłåĂijæİěèřČçŤÍåŻđèřČåĞ¡æŢřãĂĆ

ä¡IJäÿžäÿĂäÿłçśžäijijçŽĎä¿Ńå■ŘïijŇèĂČèŹŚäÿŃçijŰåĘŹç¡ŚçżIJæIJ■åŁąåŹÍçŽĎéŮőécŸïijŇ`sockets` æÍąåİŮèőĺ'åőČåŘŸå¿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¡ăæČşçżŹEchoHandlerâćđåŁääÿĂäÿłåŔŕäžěæŐěåŔŮåĚűäžŰéĚ■ç¡őéĂL’éąźçŽĎ `__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£őæŤźåŘŐïijŇæĹŚäžňåřšäÿ■éIJĂèęĄæŸ¿åijŔåIJřåIJÍTCPServerçśžäÿ■æůžåŁääL’■çijĂäžĘãĂĆä¡ĘæŸŕä¡ăåĘ■æňąè£ŘèąŇçÍŃåžŔåŘŐäijŽæŁěçśžäijijäÿŃéİćçŽĎéŤŹèŕŕïijŽ

```
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
↪'
```

åĹİçIJŃèţůæİěåě¡åČŔå¿ĹéŽ¿ä£őæ■čè£ŹäÿłéŤŹèŕŕïijŇéŹď äžĘä£őæŤź `socketserver` æĺąåİŮæžŘäžčçăĄæĹŰèĂĚä¡£çŤĺæ§ŘäžŽåěĞæĂłçŽĎæŰźæşŢäźŃåď ŰãĂĆä¡ĘæŸŕïijŇåęĆæđIJä¡£çŤĺ `partial()` åŕśèČ¡å¿Ĺè¡żæİ¿çŽĎèğčåĘşâĂŤâĂŤçžŹåőČäijăéĂŠ `ack` åŔĆæŢřçŽĎåÄijæİěåĹİåğŃåŇŰå■şåŔŕïijŇåęĆäÿŃïijŽ

```
from functools import partial
serv = TCPServer(('', 15000), partial(EchoHandler, ack=b'RECEIVED:
↪'))
serv.serve_forever()
```

åIJĺè£Źäÿłä¿Ńå■Řäÿ■ïijŇ`__init__()` æŰźæşŢäÿ■çŽĎack-åŔĆæŢřåčřæŸŐæŰźåijŔçIJŃäÿŁåŐżå¿ĹæIJL’èűčïijŇåĚűåőđåŕśæŸŕåčřæŸŐackäÿžäÿÄäÿłåijžåĹűåĚşéŤőå■Ůå ĚşäžŐåijžåĹűåĚşéŤőå■ŮåŔĆæŢřéŰőécŸæĹŚäžňåIJĺ7.2åŕŔèŁĆæĹŚäžňåůšçžŔèőĺèőžè£ĞäžĘïijŇèŕźèĂĚåŔ

å¿Ĺåď ŽæŰűåĂŹ `partial()` èČ¡åőđçŐřçŽĎæŢĹæđIJïijŇlambdaèąĺè¿¿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¡£çŤĺ `partial()` åŔŕäžěæŻť'åŁăçŻť'èğĆçŽĎèąĺè¿¿ä¡äçŽĎæĎŔåŻ¿(çżŹæ§ŘäžŽåŔĆæŢřécĎ

9.9 7.9 åřĘå■ŢæŰźæşŢçŽĎçśżè¡ňæ■ćäÿžåĞ¡æŢř

éŮőécŸ

ä¡ăæIJL'äÿĂäÿłéŹđ' `__init__()` æŰźæşŢåđ'ŰåŔłåőŽäźL'äžĘäÿĂäÿłæŰźæşŢçŽĎçśżãĂĆäÿžäžĘçőĂ

èğčåĘşæŰźæąĹ

åđ'ğåđ'ŽæŢřæČĚåĘţäÿŃïijŇåŔŕäżěä¡£çŤĺéŮ■åŇĚæİěåřĘå■ŢäÿłæŰźæşŢçŽĎçśżè¡ňæ■ćæĹŘåĞ¡æŢřãĂ
äÿ¿äÿłä¿Ńå■ŘïijŇäÿŃéİćçđ'žä¿Ńäÿ■çŽĎçśżåĚĄèőÿä¡£çŤĺèĂĚæăźæ■őæ§Řäÿłæĺąæİ£æŰźæąĹæİěèŐůåŔŰå

```
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£ŹäÿłçśżåŕŕäżěèćńäÿĂäÿłæŻť'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)

```
for line in yahoo(names='IBM,AAPL,FB', fields='sl1c1v'):
    print(line.decode('utf-8'))
```

èőÍèőž

åď ğéČÍåĹĘæČĚåĘţäÿŃïijÑä¡ăæŃěæIJL'äÿĂäÿłå■ŢæŰźæşŢçśżçŽĎåŐ§åŻăæŸŕéIJĂèęĄå■ŸåĆÍæ§ŘäžæŕŤåęĆïijÑåőŽäźL'UrlTemplateçśżçŽĎåŤŕäÿĂçŻőçŽĎåřśæŸŕåĚĹåIJÍæ§Řäÿłå IJřæŰźå■ŸåĆÍæÍąæİ£åĂijïijÑ

ä¡£çŤÍäÿĂäÿłåĘĚéČÍåĞ¡æŢřæĹŰèĂĚéŮ■åŇĚçŽĎæŰźæąĹéĂŽåÿÿäijŽæŻť äijŸéŽĚäÿĂäžŽãĂĆçőĂå■ åŘłäÿ■è£ĞåIJÍåĞ¡æŢřåĘĚéČÍåÿęäÿŁäžĘäÿĂäÿłéćÍåď ŰçŽĎåŔŸéĞŘçŐŕåćČãĂĆéŮ■åŇĚåĚşéŤőçĽźçĆźåřśåŻăæ■d'ïijÑåIJÍæĹŚäžñçŽĎèğčåĘşæŰźæąĹäÿ■ïijÑopener() åĞ¡æŢřèőřä¡ŔäžĘ template åŘĆæŢřçŽĎåĂijïijÑåźűåIJÍæŐěäÿŃæİěçŽĎèřČçŤÍäÿ■ä¡£çŤÍåőČãĂĆ

äżżä¡ŢæŮűåĂŹåŘłèęĄä¡ăçćřåĹřéIJĂèęĄçżŹæ§Řäÿłå Ğ¡æŢřåćđåŁăéćÍåď ŰçŽĎçŁűæĂĄä£ąæĄŕçŽĎéŮ çŻÿæŕŤåřĘä¡ăçŽĎåĞ¡æŢřè¡ňæ■ćæĹŘäÿĂäÿłçśżèĂŇèÍĂïijŇéŮ■åŇĚéĂŽåÿÿæŸŕäÿĂçğ■æŻť åŁăçőĂæť ĄåŠ

9.10 7.10 åÿęéćÍåď ŰçŁűæĂĄä£ąæĄŕçŽĎåŻđèřČåĞ¡æŢř

éŮőéćŸ

ä¡ăçŽĎäżčçăĄäÿ■éIJĂèęĄä¿ÍèţŰåĹřåŻđèřČåĞ¡æŢřçŽĎä¡£çŤÍ(æŕŤåęĆäžŇäżűåď ĎçŘĘåŹÍãĂĄç■L'å¿Ě åźűäÿŤä¡ăè£ŸéIJĂèęĄèől'åŻđèřČåĞ¡æŢřæŇěæIJL'éćÍåď ŰçŽĎçŁűæĂĄåĂijïijÑäžěä¿£åIJÍåőČçŽĎåĘĚéČÍä

èğčåĘşæŰźæąĹ

è£ŹäÿĂåřŔèŁĆäÿżèęĄèőÍèőžçŽĎæŸŕéĆčäžŻåĞžçŐřåIJÍå¿Ĺåď ŽåĞ¡æŢřåžŞåŠŇæąĘæđűäÿ■çŽĎåŻđèřČ äÿžäžĘæijŤçď žäÿŐæţŃèŕŢïijÑæĹŚäžňåĚĹåőŽäźL'åęĆäÿŃäÿĂäÿłéIJĂèęĄèřČçŤÍåŻđèřČåĞ¡æŢřçŽĎåĞ¡æŢ

```
def apply_async(func, args, *, callback):
    # Compute the result
    result = func(*args)

    # Invoke the callback with the result
    callback(result)
```

åőđéŹĚäÿŁïijÑè£Źæőţäžčçă ĄåŘŕäžěåĄŽäżżä¡ŢæŻť énŸçžğçŽĎåď ĎçŘĘïijÑåŇĚæŃňçž£çÍŃãĂĄè£Žçİ æĹŚäžňäžĚäžĚåŘłéIJĂèęĄåĚşæşĺåŻđèřČåĞ¡æŢřçŽĎèřČçŤÍãĂĆäÿŃéİćæŸŕäÿĂäÿłæijŤçď žæĂŐæăůä¡£çŤÍä

```
>>> 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)

```
Got: 5
>>> apply_async(add, ('hello', 'world'), callback=print_result)
Got: helloworld
>>>
```

æşÍæĎŔåĹř `print_result()` åĞ¡æŢřäžĚäžĚåŔłæŐěåŔŮäýĂäýłåŔĆæŢř `result` ãĂĆäý■èČ¡åĘ■äijååĚěåĚűäžŰä£ąæĄŕãĂĆ èĂŇå¡Şä¡ăæČşèőľ åŻđěřČåĞ¡æŢřèő£éŮőåĚűäžŰåŔŸéĞŔæĹŰè

äýžäžĘèőľ åŻđěřČåĞ¡æŢřèő£éŮőåd' ŰéČĺä£ąæĄŕïijŇäýĂçğ■æŰźæşŢæŸŕä¡£çŤĺäýĂäýłçżŚáőŽæŰźæşŢ æŕŤåęĆïijŇäýŇéÍćè£ŹäýłçśżäijŽä£Íå■ŸäýĂäýłåĘĚéČĺåžŔåĹŮåŔůïijŇæŕŔæňąæŐěæŤűåĹřäýĂäýł `result` çŽĎæŮűåĂŹåžŔåĹŮåŔůåŁă1ïijŽ

```
class ResultHandler:

    def __init__(self):
        self.sequence = 0

    def handler(self, result):
        self.sequence += 1
        print('[{}] Got: {}'.format(self.sequence, result))
```

ä¡£çŤĺè£ŹäýłçśżçŽĎæŮűåĂŹïijŇä¡ăåĚĹåĹŻåžžäýĂäýłçśżçŽĎåőđä¿ŃïijŇçĎűåŘŐçŤĺáőČçŽĎ `handler()` çżŚáőŽæŰźæşŢæÍěåĄŽäýžåŻđěřČåĞ¡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¡IJäýžçśżçŽĎæŽ£äžčïijŇåŕŕäžěä¡£çŤĺäýĂäýłéŮ■åŇĚæ■ŢèŐůçŁűæĂĄåĂijïijŇ

```
def make_handler():
    sequence = 0
    def handler(result):
        nonlocal sequence
        sequence += 1
        print('[{}] Got: {}'.format(sequence, result))
    return handler
```

äýŃéÍćæŸŕä¡£çŤĺéŮ■åŇĚæŰźåijŔçŽĎäýĂäýłä¿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'åŔęåď'ŰäÿĂäÿłæŻť'éńŸçžğçŽĎæŰźæşŢïijŇåŔŕäžëä¡£çŤĺå■ŔçĺŃæİěåőŇæĹŘåŘŇæăůçŽĎäžŇ

```
def make_handler():
    sequence = 0
    while True:
        result = yield
        sequence += 1
        print('[{}] Got: {}'.format(sequence, result))
```

åŕźäžŐå■ŔçĺŃïijŇä¡ăéIJĂèęĄä¡£çŤĺåőĆçŽĎ send() æŰźæşŢä¡IJäÿžåŻđěřČåĞ¡æŢřïijŇåęĆäÿŃæĽ'Ăç

```
>>> 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
>>>
```

èőĺèőž

å§žäžŐåŻđěřČåĞ¡æŢřçŽĎè¡ŕäžűéĂŽåÿÿéČ¡æIJL'åŔŕèČ¡åŔŸå¿Ůéİđåÿÿåď'■æİĆãĂĆäÿĂéČĺåĹĘåŐ§åŻ
åŻäæ■d'ïijŇèŕůæśĆæL'ğèąŇåŠŇåď'ĎçŘĘçżŞæđIJäźŃéŮt'çŽĎæL'ğèąŇçŐŕåćČåőđéŹĚäÿŁåůšçżŔäÿćåď'śäžĘ
éĆčä¡ăářšå£ĚéążåŐżèğčåĘşåęĆä¡Ţä£İå■ŸåŠŇæĄćåď'■çŻÿåĚşçŽĎçŁűæĂĄä£ąæĄŕäžĘãĂĆ

èĞşářŚæIJL'äÿd'çğ■äÿżèęĄæŰźåijŔæİěæ■ŢèŐůåŠŇä£İå■ŸçŁűæĂĄä£ąæĄŕïijŇä¡ăåŔŕäžěåIJĺäÿĂäÿłářź
äÿd'çğ■æŰźåijŔçŻÿæŕŤïijŇéŮ■åŇĚæĹŰèőÿæŸŕæŻť'åŁăè¡żéĞŔçžğåŠŇèĞłçĎűäÿĂçĆźïijŇåŻääÿžåőČäżňå
åőČäžňè£ŸèČ¡èĞłåŁĺæ■ŢèŐůæL'ĂæIJL'èćńä¡£çŤĺåĹŕçŽĎåŔŸéĞŔãĆĆåŻäæ■d'ïijŇä¡ăæŮăéIJĂåŐżæŃĚå£

åęĆædIJä¡£çŤĺéŮ■åŇĚïijŇä¡ăéIJĂèęĄæşĺæĎŔåŕźéĆčäžŻåŔŕä£őæŤźåŔŸéĞŔçŽĎæŞ■ä¡IJãĂĆåIJĺäÿŁé
nonlocal åčřæŸÖèŕ■åŔęŢĺæĺěæŇĞçd'żæŐěäÿŃæĺěçŽĎåŔŸéĞŔäijŽåIJĺåŻđěřČåĞ¡æŢřäÿ■èćńä£őæŤźã

èĂŇä¡£çŤĺäÿĂäÿłå■ŔçĺŃæĺěä¡IJäÿžäÿĂäÿłåŻđěřČåĞ¡æŢřåŕśæŻť'æIJL'èűčäžĘïijŇåőČèů§éŮ■åŇĚæŰźä
æ§Řçğ■æĎŔäźL'äÿŁæĺěèőšïijŇåőČæŸ¿å¿ŮæŻť'åŁăçőĂæť'ĄïijŇåŻääÿžæĂżåĚśåŕśäÿĂäÿłåĞ¡æŢřèĂŇåůšãĂ
åźűäÿŤïijŇä¡ăåŔŕäžěå¿ĹèĞłçŤśçŽĎä£őæŤźåŔŸéĞŔèĂŇæŮăéIJĂåŐżä¡£çŤĺ nonlocal
åčřæŸÖãĂĆ è£Źçğ■æŰźåijŔåŤŕäÿĂçijžçĆźåŕśæŸŕçŻÿåŕźäžŐåĚűäžŰPythonæŁĂæIJŕèĂŇèĺĂæĹŰèőÿæŕŤè¿
åŔęåď'Ůè£ŸæIJL'äÿĂäžŻæŕŤè¿ČéŽ¿æĞĆçŽĎéČĺåĹĘïijŇæŕŤåęĆä¡£çŤĺäźŃåL'■éIJĂèęĄèŕĆçŤĺ
next() ïijŇåőđéŹĚä¡£çŤĺæŮűè£Źäÿłæ■ěéłď'å¿ĹåőźæŸşèćńå£Ÿèőřã ĂĆ
år¡çőąåęĆæ■d'ïijŇå■ŔçĺŃè£ŸæIJL'åĚűäžŰçŤĺåď'ĎïijŇæŕŤåęĆä¡IJäÿžäÿĂäÿłåĘĚèĄŤåŻđěřČåĞ¡æŢřçŽĎåőŻ

åęĆædIJä¡ăäżĚäżĚåŘłéIJĂèęĄçżŹåŻđěřČåĞ¡æŢřäijăéĂŠéćİåď'ŰçŽĎåĂijçŽĎèŕİïijŇè£ŸæIJL'äÿĂçğ■ä
partial() çŽĎæŰźåijŔäźš§å¿ĹæIJL'çŤĺãĂĆ åIJĺæšąæIJL'ä¡£çŤĺ partial()
çŽĎæŮűåĂŹïijŇä¡ăåŔŕèČ¡çżŘåÿÿçIJŃåĹřäÿŃéİćè£Źçğ■ä¡£çŤĺlambdaèąĺè¿¿åijŔçŽĎåd'■æİĆäżčçăĄïijŽ

```
>>> apply_async(add, (2, 3), callback=lambda r: handler(r, seq))
[1] Got: 5
>>>
```

åŔŕäžěåŔĆèĂČ7.8åřŘèŁĆçŽĎåĞääÿłçd'žä¿ŃïijŇæŢŹä¡ăåęĆä¡Ţä¡£çŤĺ partial()
æİěæŻť'æŤźåŔĆæŢřç■¿åŘ■æİęčőĂåŇŰäÿŁè£řäžčçăĄãĂĆ

9.11 7.11 åĘĚèĄŤåŻđèřČåĞ¡æŢř

éŮőécŸ

å¡Şä¡ăçijŰåĘŹä¡£çŤĺåŻđèřČåĞ¡æŢřçŽĎäžčçăĄçŽĎæŮűåĂŹïijŇæŃĚå£Čå¿Ĺåď˙ŽářŕåĞ¡æŢřçŽĎæĽ'ľ'ä¡ăåÿŇæIJŻæĽ¿åĹřæ§ŘäÿłæŰźæşŢæĺěőľ'äžčçăĄçIJŃäÿŁåŐżæŻť˙åČŕæŸŕäÿĂäÿłæŹőéĂŽçŽĎæĽ'ğèąŇåžĚ

èğčåĘşæŰźæąĹ

éĂŽè£Ğä¡£çŤĺçŤ§æĹŘåŹĺåŠŇå■ŕçĺŃåŕŕäžěä¡£å¿ŮåŻđèřČåĞ¡æŢřåĘĚèĄŤåIJĺæ§ŘäÿłåĞ¡æŢřäÿ■ãĂĆäÿžäžĘæijŤçd'žèŕť˙æŸŐïijŇåĄĞèő¿ä¡ăæIJL'åęĆäÿŃæĽ'Ăçd'žçŽĎäÿĂäÿłæĽ'ğèąŇæ§Řçğ■èőąçőŮäżżåŁąçĎű

```
def apply_async(func, args, *, callback):
    # Compute the result
    result = func(*args)

    # Invoke the callback with the result
    callback(result)
```

æŐěäÿŃæĺěèőľ'æĹŚäžňçIJŃäÿĂäÿŃäÿŃéĺćçŽĎäžčçăĄïijŇåőČåŇĚåŘŕäžĘäÿĂäÿł 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£Źäÿd'äÿłäžčçăĄçL'Ğæőţå ĚĄèőÿä¡ăä¡£çŤĺ yield èŕ■åŘěåĘĚèĄŤåŻđèřČæ■ěéłd'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')
```

åęĆædIJä¡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■åŔěåď ŰïijŇåĚűäžŰåIJřæŰźåźűæšąæIJL’åĞžçŐřäżżä¡ŢçŽĎåŻđèřČåĞ¡æŢř(åĚűåőđæŸŕåIJĺåŘŐåŔřåőŽäź

èőÍèőž

æIJňářŔèŁĆäijŽåőđåőđåIJĺåIJÍçŽĎæţŃèřŢä¡ăåĚşäžŐåŻđèřČåĞ¡æŢřãĂĄçŤ§æĹŘåŹĺåŠŇæŐğåĹűæţĄç

éęŰåĚĹïijŇåIJÍéIJÄèęĄä¡£çŤĺåĹřåŻđèřČçŽĎäžčçãĄäÿ■ïijŇåĚşéŤőçĆźåIJÍäžŐå¡ŞåL’■èőąçőŮåůëä¡IJäi
å¡Şèőąçő Ůé Ğ■åŘræŮűïijŇåŻđèřČåĞ¡æŢřèćńèřČçŤÍæIěçžğçż■åď ĎçŘĘçżŞæđIJãĂĆ`apply_async()`
åĞ¡æŢřæijŤçď žäžĘæL’ğèąŇåŻđèřČçŽĎåőđéŹĚéĂżè¿Śïij Ň år¡çőąåőđéŹĚæČĚåĘţäÿ■åőČåŘŕèČ¡äijŽæŻť’åŁă

èőąçőŮçŽĎæŽĆåĄIJäÿŐéĞ■åŘræĂIèůŕèů§çŤ§æĹŘåŹĺåĞ¡æŢřçŽĎæL’ğèąŇæĺąåđŃäÿ■èřŃèĂŇåŘĹãĂ
åĚůä¡ŞæIěèőšïijŇ`yield` æŞ■ä¡IJäijŽä¡£äÿĂäÿłçŤ§æĹŘåŹĺåĞ¡æŢřäžğçŤ§äÿĂäÿłåĂijåźűæŽĆåĄIJãĂĆ
æŐěäÿŃæIěèřČçŤĺçŤ§æĹŘåŹĺçŽĎ `__next__()` æĹŰ `send()`
æŰźæşŢåŔĹäijŽèőľ’åőČäžŐæŽĆåĄIJåď ĎçżğçżæL’ğèąŇãĂĆ

æăźæ■őè£ŹäÿłæĂIèůŕïijŇè£ŹäÿĂåřŔèŁĆçŽĎæăÿå£Čåřšå IJÍ `inline_async()`
èčĚéěřåŹĺåĞ¡æŢřäÿ■äžĘãĂĆ åĚşéŤőçĆźåřšæŸŕïijŇèčĚéěřåŹĺäijŽéĂŘæ■ěéĄ■åŐEçŤ§æĹŘåŹĺåĞ¡æŢřçŽĎæ
`yield` èŕ■åŔěijŇæŕŘäÿĂæňąäÿĂäÿłãĂĆ äÿžäžĘè£ŹæăůåĄŽïijŇåĹŽåijĂåğŃçŽĎæŮűåĂŹåĹŻåžžäžĘäÿĂä
`result` éŸ§åĹŮåźűåŘŠéĞŇéIćæŤ¿åĚěäÿĂäÿł `None` åĂijãĂĆ

çĎűåŘŐåijĂåğŃäÿĂäÿłå¿łçŐŕæŞ■ä¡IJïijŇäżŐéŸ§åĹŮäÿ■åŔŰåĞžçżŞæđIJåĂijåźűåŔŚéĂĄçżŹçŢ§æĹŘåŹĺï
`yield` èŕ■åŔěïijŇ åIJĺè£ŹéĞŇäÿĂäÿł `Async` çŽĎåőđä¿Ńèćńæ Őěå ŘŮåĹřãĂĆçĎűåŘŐå¿łçŐŕåijĂåğŃæčĂæ
`apply_async()` ãĂĆ çĎűèĂŇïijŇè£Źäÿłèőąçő ŮæIJL'äÿłæIJĂèŕąåijĆéČĺåĹĘæŸŕåőČåźűæšąæIJL'ä¡£çŢĺäÿ
`put()` æŰźæşŢæİěåŻđèřČãĂĆ

è£ŹæŮűåĂŹïijŇæŸŕæŮűåĂŹèŕęçżĘèğčéĞŁäÿŃåĹřåžŢåŔŚçŢ§äžĘäżĂäźĹäžĘãĂĆäÿżå¿łçŐŕçńŃå■şè£
`get()` æŞ■ä¡IJãĂĆ åęĆæđIJæŢřæ■őå■ŸåIJĺïijŇåőČäÿĂåőŽæŸŕ `put()`
åŽđèřČå■ŸæŢ¿çŽĎçżŞæđIJãĂĆåęĆæđIJæšąæIJL'æŢřæ■őïijŇéĆčäźĹåĚĹæŽĆåĄIJæŞ■ä¡IJåźűç■L'å¿ĚçżŞæ
è£ŹäÿłåĚůä¡ŞæĂŐæăůåőđçŐŕæŸŕçŢś `apply_async()` åĞ¡æŢřæİěåĘşåőŽçŽĎãĂĆ
åęĆæđIJä¡ääÿ■çŻÿä£ąäijŽæIJL'è£ŹäźĹçěđåěĞçŽĎäžŃæČĚïijŇä¡ääŕŕäżěä¡£çŢĺ
`multiprocessing` åžŞæİèèŕŢäÿĂäÿŃïijŇ 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¡ääijŽåŔŚçŐřè£Źäÿłçij§çŽĎåřśæŸŕè£ŹæăůçŽĎïijŇä¡ĘæŸŕèęĄèğčéĞŁæÿĚæěŽåĚůä¡ŞçŽĺ

åřĘåd'■æİĆçŽĎæŐğåĹűæţĄéŽŘèŮŔåĹřçŢ§æĹŘåŹĺåĞ¡æŢřèČŇåŘŐçŽĎä¿Ńå■ŘåIJĺæăĞåĞĘåžŞåŠŇç
æŕŤåęĆïijŇåIJĺ `contextlib` äÿ■çŽĎ `@contextmanager`
èčĚéěřåŹĺä¡£çŢĺäžĘäÿĂäÿłäżd'äžžèť źèğčçŽĎæŁĂåůğïijŇ éĂŽè£ĞäÿĂäÿł `yield`
èŕ■åŔěåřĘè£ŻåĚěåŠŇçężåijĂäÿŁäÿŃæŰĞçőąçŘĘåŹĺçšŸåŘĹåIJĺäÿĂèţůãĂĆ
åŘęåd'ŰéİđåÿÿæţĄèąŇçŽĎ `Twisted` åŇĚäÿ■äź§åŇĚåŘńäžĘéİđåÿÿçśżäijijçŽĎåĘĚèĄŤåŽđèřČãĂĆ

9.12 7.12 èő£éŮőéŮ■åŇĚäÿ■åőŽäźĽçŽĎåŔŸéĞŔ

éŮőécŸ

ä¡ăæČşèęĄæĽl'åśŢåĞ¡æŢřäÿ■çŽĎæ§Řäÿłéŭ■åŇĚïijŇåĚĄèőÿåőČèČ¡èő£éŮőåŠŇä£őæŤźåĞ¡æŢřçŽĎå

èğčåĘşæŰźæąĹ

éĂŽåÿÿæİěèőšïijŇéŮ■åŇĚçŽĎåĘĚéČĺåŔŸéĞŔåŕźäžŐåd'ŰçŢŇæİěèőšæŸŕåőŇåĚĺéŽŘèŮŔçŽĎãĂĆ
ä¡ĘæŸŕïijŇä¡ääŔŕäžěéĂŽè£ĞçijŰåĘŹèő£éŮőåĞ¡æŢřåźűåřĘåĚüä¡IJäÿžåĞ¡æŢřåśđæĂğçżŠåőŽåĹŕéŭ■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)

```
    def set_n(value):
        nonlocal n
        n = value

    # Attach as function attributes
    func.get_n = get_n
    func.set_n = set_n
    return func
```

äýŃéÌćæŸŕä¡£çŤÍçŽĎä¿Ńå■Ř:

```
>>> f = sample()
>>> f()
n= 0
>>> f.set_n(10)
>>> f()
n= 10
>>> f.get_n()
10
>>>
```

èőÍèőž

äýžäžĘèŕť'æŸŐæýĚæěŽåőČåęCä¡Ţåůëä¡IJçŽĎïijŇæIJL'äýd'çĆźéIJĂèęĄèğčéĞŁäýĂäýŃãĂĆéęŰåĚĹïij
åčřæŸŐåŘŕäżěèől'æĹŚäžňçijŰåĘŹåĞ¡æŢřæİěä£őæŤźåĘĚéČĺåŔŸéĞŔçŽĎåĂijãĂĆ
åĚűæňąïijŇåĞ¡æŢřåšđæĂğåĚĄèőýæĹŚäžňçŤĺäýĂçğ■å¿ĹçőĂå■ŢçŽĎæŰźåijŔářĘèő£éŮőæŰźæşŢçżŚåőŽåĹ

è£ŸåŘŕäžěè£ŻäýĂæ■ěçŽĎæL'l'åśŢïijŇèől'éŮ■åŇĚæĺąæŃ§çśżçŽĎåőđä¿ŃãĂĆä¡äèęĄåĄŽçŽĎäżĚäżĚa

```
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)

```
        items.append(item)

    def pop():
        return items.pop()

    def __len__():
        return len(items)

    return ClosureInstance()
```

äýŃéÍćæŸŕäýĂäýłäžď'äžŠåijŔäijŽèŕÍæÍěæijŤçď'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£ŹäýłäžčçăĄè£ŘèąŇèţůæÍěäijŽæŕŤäýĂäýłæŹőéĂŽçŽĎçśžåőŽäźL'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¡ăäijŽå¿ŮåĹřçśžäijijåęĆäýŃçŽĎçżŞæđIJïijŽ

```
>>> from timeit import timeit
>>> # Test involving closures
>>> s = Stack()
>>> timeit('s.push(1);s.pop()', 'from __main__ import s')
```

(continues on next 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£Řèą ŇèţůæİěèęĄå£ńåď˙ğæęĆ8%ïijŇåď˙ğéČĺåĹĘåŐ§åŻãæŸéŮ■åŇĚæŻť˙å£ńæŸŕåŽääÿžäÿ■äijŽæűL˙åŔŁåĽřécİåď˙ŰçŽĎselfåŔŸéĞŔãĂĆ

Raymond Hettingerářźäž Őè£ŹäÿłéŮőécŸèő¿èöąåĞžäžĘæŻť˙åŁäéŽ¿äžěçŘĘèğčçŽĎæŤźè£ŻæŰźæąĹãĂ
èĂŇäÿŤåőČåŔłæŸŕçIJ§åődçśżçŽĎäÿĂäÿłåěĞæĂłçŽĎæŻ£æ■ćèĂŇåůšïijŇä¿ŃåęĆïijŇçśżçŽĎäÿżèęĄçĽźæĂ
åźűäÿŤä¡äèęĄåĄŽäÿĂäžŻåĚűäžŰçŽĎåůěä¡IJæĽ■èČ¡èőľäÿĂäžŻçĽźæőŁæŰźæşŢçŤ§æŢĹ(ærŤåęĆäÿŁéİć
`ClosureInstance` äÿ■éĞ■åĘŹè£ĞçŽĎ `__len__()` åődçŐřãĂĆ)

æIJĂåŘŐïijŇä¡äåŔŕèČ¡è£ŸäijŽèőľåĚűäžŰéŸĚèŕża¡ääžčçăĄçŽĎäžžæĎ§åĽřçŰŚæČŚïijŇäÿžäžĂäźĹåőČ
(å¡ŞçĎűïijŇäżŰäżňäź§æČşç§ěéĄŞäÿžäżĂäźĹåőČè£ŘèąŇèţůæİěäijŽæŻť˙å£ń)ãĂĆår¡çőąåęĆæ■d˙ïijŇè£Źårźä

æĂżä¡ŞäÿŁèőšïijŇåIJĺéĚ■ç¡őçŽĎæŮűåĂŹçżŹéŮ■åŇĚæůżåŁäæŰźæşŢäijŽæIJL˙æŻť˙åď˙ŽçŽĎåődçŤĺåŁ
ærŤåęĆä¡ăéIJĂèęĄéĞ■ç¡őåĘĚéČĺçŁűæĂĄãĂĄåĹůæŰřçijŞåĘšåŇžãĂĄæÿĚéŹď˙çijŞå■ŸæĹŰåĚűäžŰçŽĎåŕ

10 çňňåĚńçńäïijŽçśżäÿŐårźèšą

æIJňçńääÿżèęĄåĚşæşĺçĆźçŽĎæŸŕåŠŇçśżåőŽäźL˙æIJL˙åĚşçŽĎåÿÿèğĄçijŰçĺŃæĺąåđŃãĂĆåŇĚæŃňèőľ
çśżårĄèčĚæŁĂæIJŕãĂĄçżğæL˙£ãĂĄåĘĚå■ŸçőąçŗĕäżěåŔŁæIJL˙çŤĺçŽĎèő¿èőąæĺąåijŔãĂĆ

Contents:

10.1 8.1 æŤźåŔŸårźèšąçŽĎå■ŮçňęäÿšæŸ¿çď˙ž

éŮőécŸ

ä¡äæČşæŤźåŔŸårźèšąåődä¿ŃçŽĎæL˙Şå■řæĹŰæŸ¿çď˙žè¿ŞåĞžïijŇèőľåőČäžňæŻť˙åĚůåŔŕèŕżæĂğãĂĆ

èğčåĘşæŰźæąĹ

èęĄæŤźåŔŸäÿĂäÿłåődä¿ŃçŽĎå■Ůçňęäÿšèąĺçď˙žïijŇåŔŕéĞ■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)

```
        return 'Pair({0.x!r}, {0.y!r})'.format(self)

    def __str__(self):
        return '({0.x!s}, {0.y!s})'.format(self)
```

__repr__() æŰźæşŢè£ŤåŽđäÿĂäÿłåőđä¿ŃçŽĎäżčçăĄèąĺçď'žå¡ćåijŔïijŇéĂŽåÿÿçŤĺæİéé
åĘĚç¡őçŽĎ repr() åĞ¡æŢřè£ŤåŽđè£ŹäÿłåŰçňęäÿšïijŇèů§æĹŚäżňä¡£çŤĺäžďäžŠåijŔèğčé
__str__() æŰźæşŢåřĘåőđä¿Ńè¡ňæ■ćäÿžäÿĂäÿłåŰçňęäÿšïijŇä¡£çŤĺ str() æĹŰ
print() åĞ¡æŢřäijŽè¿ŞåĞžè£ŹäÿłåŰçňęäÿšãĂĆæŕŤæçĆïijŽ

```
>>> p = Pair(3, 4)
>>> p
Pair(3, 4) # __repr__() output
>>> print(p)
(3, 4) # __str__() output
>>>
```

æĹŚäżňåĲÍè£ŹéĞŇè£ŸæijŤçď'žäžĘåĲÍæăijåijŔåŇŰçŽĎæŮűåĂŹæĂŐæăůä¡£çŤĺäÿ■åŘŃçŽĎåŰçňę
çĽ'źåĹńæİěèőšïijŇ !r æäijåijŔåŇŰäżčçăĄæŇĞæŸŐè¿ŞåĞžä¡£çŤĺ
__repr__() æİěäżčæŻ£éżŸèőď'çŽĎ __str__() ãĂĆ
ä¡ăåĚŕäžěçŤĺåĽ■éİćçŽĎçśżæİěèŕŢçİĂæţŃèŕŢäÿŃïijŽ

```
>>> 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¡çŽĎäźăæČŕïijŇåŻäÿžåőČèČ¡çőĂåŇŰ
ä¿Ńåę ĆïijŇåęĆæđIJäżĚäżĚåŔłæŸŕæL'Şå■řè¿ŞåĞžæĹŰæŮěå£Ůè¿ŞåĞžæ§Řäÿłåőđä¿ŃïijŇéĆčäźĹçĺŃåžŔåŚ

__repr__() çŤ§æĹŘçŽĎæŰĞæIJňå■Ůçňęäÿšæăĺå ĞĘåĄŽæşŢæŸŕéIJĂèęĄèőľ
eval(repr(x)) == x äÿžçIJ§ãĂĆåęĆæđIJåőđåIJĺäÿ■èČ¡è£Źæăůå■ŘåĄŽïijŇåžŤèŕěåĹŻåżžäÿĂäÿłæIJL
< åŠŇ > æŃňèţůæİěãĂĆæŕŤæçĆïijŽ

```
>>> f = open('file.dat')
>>> f
<_io.TextIOWrapper name='file.dat' mode='r' encoding='UTF-8'>
>>>
```

åęĆæđIJ __str__() æšąæIJL'èćńåőŽäźĽïijŇéĆčäźĹåřśäijŽä¡£çŤĺ __repr__()
æİěäżčæŻ£è¿ŞåĞžãĂĆ

äÿŁéİćçŽĎ format() æŰźæşŢçŽĎä¡£çŤĺçIJŃäÿŁåŐżå¿ĹæIJL'èűčïijŇæäijåijŔåŇŰäżčçăĄ
{0.x} åŕźåžŤçŽĎæŸŕçňň1äÿłåŔĆæŢřçŽĎxåśđæĂğãĂĆ åŽăæ■ď'ïijŇåIJĺäÿŃéİćçŽĎåĞ¡æŢřäÿ■ïijŇ0åőđéŹ
self æIJňèžńïijŽ

```
def __repr__(self):
    return 'Pair({0.x!r}, {0.y!r})'.format(self)
```

ä¡IJäÿžè£Źçğ■åőđçŐřçŽĎäÿĂäÿłæŻ£äżčïijŇä¡ääź§åŔŕäżěä¡£çŤÍ %
æŞ■ä¡IJçňęïijŇåřśåČŔäÿŇéİćè£ŹæăůïijŽ

```
def __repr__(self):
    return 'Pair(%r, %r)' % (self.x, self.y)
```

10.2 8.2 èĞłåőŽäźĽå■Ůçňęäÿšç ŽĎæăijåijŔåŇŰ

éŮőécŸ

ä¡ăæČşéĂŽè£Ğ `format()` åĞ¡æŢřåŠŇå■Ůçňęäÿšæ Úźæş Ţä¡£å¿ŮäÿĂäÿłåŕźèśąèČ¡æŤŕæŇĄèĞłåőŽäźI

èğčåĘşæŰźæąĹ

äÿžäžĘèĞłåőŽäźĽå■Ůçňęäÿšç ŽĎæăijåijŔåŇŰïijŇæĹŚäżňéIJĂèęĄåIJÍçśżäÿŁéİćåőŽäźĽ
`__format__()` æŰźæş ŢãĂĆä¿ŇåęĆïijŽ

```
_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¿ŇåŔŕäžěæŤŕæŇĄæăijåijŔåŇŰæŞ■ä¡IJäžĘïijŇåęĆåŘŇäÿŇéİćè£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)

```
'The date is 2012-12-21'
>>> 'The date is {:mdy}'.format(d)
'The date is 12/21/2012'
>>>
```

èőĺèőž

__format__() æŰźæşŢçżŹPythonçŽĎå■ŮçňęäÿšæäijåijŔåŇŰåŁ§èČ¡æŔŘä¿ŻäžĘäÿĂäÿłéŠľå■ŘãĂ
è£ŹéĞŇéIJĂèęĄçIĂéĞ■åijžèřČçŽĎæŸŕæäijåijŔåŇŰäžčçăĄçŽĎèğčæđŘåůěä¡IJåőŇåĚĺçŤśçźèĞłåůśåĘşåőŽ
ä¿ŃåęĆïijŇåŔĆèĂČäÿŃéIćæIěèĞł datetime æĺąåİŮäÿ■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¡őçśžåđŃçŽĎæäijåijŔåŇŰæIJL'äÿĂäžŻæăĞåĞĘçŽĎçžęåőŽãĂĆ
åŘŕäżěåŔĆèĂČ stringæĺąåİŮæŰĞæąč èŕť'æŸŐãĂĆ

10.3 8.3 èőľåŕźèśąæŤŕæŇĄäÿŁäÿŃæŰĞçőąçŘĘå■Ŕèőő

éŮőécŸ

ä¡ăæČşèőľä¡ăçŽĎåŕźèśąæŤŕæŇĄäÿŁäÿŃæŰĞçőąçŘĘå■Ŕèőő(withèŕ■åŘě)ãĂĆ

èğčåĘşæŰźæąĹ

äÿžäžĘèőľäÿĂäÿłåŕźèśąåĚijåőź with èŕ■åŘěïijŇä¡ăéIJĂèęĄåőđçŐř __enter__()
åŠŇ __exit__() æŰźæşŢãĂĆ ä¿ŃåęĆïijŇèĂČèŹŚåęĆäÿŃçŽĎäÿĂäÿłçśžïijŇåőČèČ¡äÿžæĹŚäžňåĹŻåžžäÿ

```
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)

```
    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£Źäÿłçśżç ŽĎåĚşéŤőçĽ'źçĆźåIJĺäžŐåőČèąĺçđ'žäžĘäÿĂäÿłç¡ŚçżIJè£đæŐěïijŇä¡ĘæŸŕåĹİåğŃåŃŰçŽĎa
è£đæŐěçŽĎåżžçńŃåŠŇåĚşéŮ■æŸŕä¡£çŤĺ `with` èŕ■åŔěèĞłåŁĺåőŇæĹŘçŽĎïijŇä¿ŇåęĆïijŽ

```
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
```

èőĺèőž

çijŰåĘŹäÿŁäÿŃæŰĞçőąçŘĘåŹĺçŽĎäÿżèęĄåŐ§çŘĘæŸŕä¡ăçŽĎäżčçăĄäijŽæŤ¿åĹř `with` èŕ■åŔěåİŮäÿ■æĽ'ğèąŇãĂĆ å¡ŞåĞžçŐř `with` èŕ■åŔěçŽĎæŮűåĂŹïijŇåŕźèśąçŽĎ `__enter__()` æŰźæşŢèćńèğęåŔŚïijŇ åőČè£ŤåŽđçŽĎåĂij(åęĆæđIJæIJL'çŽĎèŕİ)äijŽèćńèţŃåĂijçżŹ `as` åčřæŸŐçŽĎåŔŸéĞŔãĂĆçĎűåŔŐïijŇ`with` èŕ■åŔěåİŮéĞŇéİćçŽĎäżčçăĄåijĂåğŃæĽ'ğèąŇãĂĆ æIJĂåŘŐïijŇ`__exit__()` æŰźæşŢèćńèğęåŔŚè£ŻèąŇæÿĚçŘĘåůěä¡IJãĂĆ

äÿ■çőą `with` äżčçăĄåİŮäÿ■åŔŚçŤ§äžĂäźĹïijŇäÿŁéİćçŽĎæŐğåĹűæţĄéČ¡äijŽæĽ'ğèąŇåőŇïijŇåŕśçőŮ
äžŃåőđäÿŁïijŇ`__exit__()` æŰźæşŢçŽĎçňňäÿĽäÿłåŔĆæŢřåŇĚåŘńäžĘåijĆåÿÿçśżåđŃãĂĄåijĆåÿÿåĂijåŠ
`__exit__()` æŰźæşŢèČ¡èĞłåůśåĘşåőŽæĂŐæăůåĹľçŤĺè£Źäÿłåij Çåÿÿä£ąæĄŕïijŇæĹŰèĂĚå£¡çŢěåőČåźű
åęĆæđIJ `__exit__()` è£ŤåŽđ `True` ïijŇéĆčäźĹåijĆåÿÿäijŽèćńæÿĚçľ'žïijŇåřśåě¡åČŔäżĂäźĹéČ¡æšąåŔŚçŤ
`with` èŕ■åŔěåŘŐéİćçŽĎçĺŃåžŔçż■åIJĺæ■čåÿÿæĽ'ğèąŇãĂĆ

è£ŸæIJL'äÿĂäÿłçżĘèŁĆéŮőéćŸåřśæŸŕ `LazyConnection` çśżæŸŕåŘęåĚĄèőÿåď'Žäÿł `with` èŕ■åŔěæİěåţŇåěŮä¡£çŤĺè£đæŐěãĂĆ å¿ĹæŸ¿çĎűïijŇäÿŁéİćçŽĎåőŽäźL'äÿ■äÿĂæňąåŔłèČ¡åĚĄèőÿäÿĂäÿłsocketè£đæŐěïijŇåęĆæđIJæ■čåIJĺä¡£çŤ
`with` èŕ■åŔěïijŇ åřśäijŽäžğçŤ§äÿĂäÿłåijĆåÿÿäžĘãĂĆäÿ■è£Gä¡ääåŔŕäžěåČŔäÿŃéİćè£Źæăůå£őæŤźäÿŃéİ

```
from socket import socket, AF_INET, SOCK_STREAM

class LazyConnection:
```

(continues on next 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
```

åIJÍçňňäžŇäýłçL'ĹæIJňäý■ïijŇLazyConnection çśżåŔŕäžěèćńçIJŃåĄŽæŸŕæ§Řäýłè£đæŐěåůěåŐĆã
æŕŔæňą __enter__() æŰźæşŢæL'ğèąŇçŽĎæŮűåĂŹïijŇåőČåd'■åĹűåĹŽåžžäýĂäýłæŰřçŽĎè£đæŐěåźűåì
__exit__() æŰźæşŢçőĂå■ŢçŽĎäżŐæăĹäÿ■åijźåĞžæIJĂåŘŐäýĂäýłè£đæŐěåźűåĚşéŮ■åőČãĂĆ
è£ŹéĞŇçĺ■å¿őæIJL'çĆźéŽ¿çŘĘèğčïijŇäý■è£ĞåőČèČ¡åĚĄèőýåţŇåěŮä¡£çŤÍ with
èŕ■åŘěåĹŽåžžåd'Žäýłè£đæŐěïijŇåřšåęĆäýŁéİćæijŤçd'žçŽĎéĆčæăůãĂĆ

åIJĺéIJĂèęĄçőąçŘĘäÿĂäžŻèţĎæžŘæŕŤåęĆæŰĞäžűãĂĄç¡ŚçżIJè£đæŐěåŠŇéŤĄçŽĎçijŰçĺŃçŐŕåćČäÿ■
è£ŹäžŻèţĎæžŘçŽĎäÿĂäýłäýżèęĄçL'źå¿ĄæŸŕåőČäżňå£ĚéążèćńæL'ŃåŁĺçŽĎåĚşéŮ■æĹŰéĞŁæŤ¿æİěçąőä£
ä¿ŃåęĆïijŇåęĆæđIJä¡ăèŕůæśĆäžĘäÿĂäýłéŤĄïijŇéĆčäźĹä¡ăå£Ěéążçąőä£İäźŃåŘŐéĞŁæŤ¿äžĘåőČïijŇåŘęåĹ
éĂŽè£ĞåőđçŐř __enter__() åŠŇ __exit__() æŰźæşŢåźűä¡£çŤÍ with
èŕ■åŘěåŕäžěå¿ĹåőźæŸŞçŽĎéĄ£åĚ■è£ŹäžŻéŮőécŸïijŇ åŻääýž __exit__()
æŰźæşŢåŕäžěèőľä¡ăæŮăéIJĂæŃĚå£Čè£ŹäžŻäžĘãĂĆ

åIJĺ contextmanager æĺąåİŮäÿ■æIJL'äÿĂäýłæăĞåĞĘçŽĎäÿŁäÿŃæŰĞçőąçŘĘæŰźæąĹæĺąæİ£ïijŇå
åŘŇæŮűåIJĺ12.6åřŔèŁĆäÿ■è£ŸæIJL'äÿĂäýłåŕźæIJňèŁĆçd'žä¿ŃçĺŃåžŔçŽĎçž£çĺŃåőL'åĚĺçŽĎä£őæŤźçL'Ĺ

10.4 8.4 åĹŻåžžåd'ğéĞŔåŕźèśąæŮűèŁĆçIJĄåĘĚå■ŸæŰźæşŢ

éŮőécŸ

ä¡ăçŽĎçĺŃåžŔèęĄåĹŻåžžåd'ğéĞŔ(åŘŕèČ¡äýŁçŹ¿äÿĞ)çŽĎåŕźèśąïijŇåŕijèĞt'å■ăçŤĺå¿Ĺåd'ğçŽĎåĘĚå■Ÿ

èğčåĘşæŰźææąĹ

åŕźäžŐäÿżèęĄæŸŕçŤĺæİěå¡ŞæĹŘçőĂå■ŢçŽĎæŢřæ■őçżŞæđĎçŽĎçśżèĂŃèĺĂïijŇä¡ăåŔŕäżěéĂŽè£Ğçżž
__slots__ åśđæĂğæİěæđĄåď ğçŽĎåĞŔåřŚåőđä¿ŃæĽ Ăå■ăçŽĎåĘĚå■ŸãĂĆæŕŤåęĆïijŽ

```python
class Date:
    __slots__ = ['year', 'month', 'day']
    def __init__(self, year, month, day):
        self.year = year
        self.month = month
        self.day = day
```

å¡Şä¡ăåőŽäźĽ __slots__ åŘŐïijŇPythonåřśäijŽäÿžåőđä¿Ńä¡£çŤĺäÿĂçğ■æŻť åŁăçť ğåĞŚçŽĎåĘĚéĆ
åőđä¿ŃéĂŽè£ĞäÿĂäÿłå¿ĹåřŔçŽĎåŻžåőŽåď ğåřŔçŽĎæŢřçżĎæİěæđĐåżžïijŇèĂŇäÿ■æŸŕäÿžæŕRäÿłåőđä¿Ń
åIJÍ __slots__ äÿ■åĹŮåĞžçŽĎåśđæĂğåŘ■åIJÍåĘĚéĆĺèćńæŸăåřĐåĹřè£Źäÿłæ̧ŢřçżĎçŽĎæŇĞåőŽåřŘæăĆ
ä¡£çŤĺslotsäÿĂäÿłäÿ■åě¡çŽĎåIJřæŰźåřśæŸŕæĹŚäżňäÿ■èČ¡åĘ■çžŹåőđä¿ŃæůżåłăæŰřçŽĎåśđæĂğäžĘïijŇåĹ
__slots__ äÿ■åőŽäźĽçŽĎéĆčäžŻåśđæĂğåŘ■ãĂĆ

èőĺèőž

ä¡£çŤĺslotsåŘŐèŁĆçIJAçŽĎåĘĚå■ŸäijŽèů§å■ŸåĆĺåśđæĂğçŽĎæŢřéĞŔåŠŇçśżåđŃæIJL'åĚşãĂĆ
äÿ■è£ĞïijŇäÿĂèĹňæİěèőšïijŇä¡£çŤĺåĹřçŽĎåĘĚå■ŸæĂżéĞŔåŠŇåřĘæŢřæ■őå■ŸåĆĺåIJĺäÿĂäÿłåĚČçżĎäÿ■å
äÿžäžĘçżŹä¡ăäÿĂäÿłçŻť èğĆèőď èŕĘïijŇåĄĞèő¿ä¡ääÿ■ä¡£çŤĺslotsçŻť æŐěå■ŸåĆĺäÿĂäÿłDateåőđä¿ŃïijŇ
åIJÍ64ä¡■çŽĎPythonäÿŁéİćèęĄå■ăçŤĺ428å■ŮèŁĆïijŇèĂŇåęĆæđIJä¡£çŤĺäžĘslotsïijŇåĘĚå■Ÿå■ăçŤĺäÿŃéŹ
åęĆæđIJçĺŃåžŔäÿ■éIJĂèęĄåŘŇæŮűåĹŻåżžåď ğéĞŔçŽĎæŮěæIJ§åőđä¿ŃïijŇéĆčäźĹè£Źäÿłåřśèč¡æđĄåď ğ

åř¡çőąslotsçIJŃäÿŁåŐżæŸŕäÿĂäÿłå¿ĹæIJL'çŤĺçŽĎçĽźæĂğïijŇå¿Ĺåď ŽæŮűåĂŹä¡äè£ŸæŸŕå¿ÚåĞŔåř
PythonçŽĎå¿Ĺåď ŽçĽźæĂğéČ¡ä¿İèţŰäžŐæŹőéĂŽçŽĎå§žäžŐå■ŮåĚÿçŽĎåőđçŐřãĂĆ
åŘęåď ŰïijŇåőŽäźĽäžĘslotsåŘŐçŽĎçśżäÿ■åĘ■æŤřæŇĄäÿĂäžŻæŹőéĂŽçśżçĽźæĂğäžĘïijŇæŕŤåęĆåď Žçżğ
åď ğåď ŽæŢřæČĚåĘţäÿŃïijŇä¡ăåžŤèŕěåŔłåIJĺéĆčäžŻçżŔåÿÿèćńä¡£çŤĺåĹřçŽĎçŤĺä¡IJæŢřæ■őçżŞæđĎçŽĎçś
(æŕŤåęĆåIJĺçĺŃåžŔäÿ■éIJĂèęĄåĹŻåżžæ§ŘäÿłçśżçŽ̌ĎåĞăçŹ¿äÿĞäÿłåőđä¿Ńåŕźèśą)ãĂĆ

åĚşäžŐ __slots__ çŽĎäÿĂäÿłåÿÿèğĄèŕŕåŇžæŸŕåőČåŔŕäżěä¡IJäÿžäÿĂäÿłåřĄèčĚåůěåĚůæİéé
åř¡çőąä¡£çŤĺslotsåŔŕäżěè¿¿åĹřè£ŹæăůçŽĎçŻőçŽĎïijŇä¡ĘæŸŕè£Źäÿłåźűäÿ■æŸŕåőČçŽĎåĹİèąůãĂĆ
__slots__ æŻť åď ŽçŽĎæŸŕçŤĺæİěä¡IJäÿžäÿĂäÿłåĘĚå■ŸäijŸåŇŰåůěåĚůãĂĆ

10.5 8.5 åIJÍçśżäÿ■åřĄèčĚåśđæĂğåŘ■

éŮőéćŸ

ä¡ăæČşåřĄèčĚçśżçŽĎåőđä¿ŃäÿŁéİćçŽĎâĂIJçğĄæIJL'âĂİæŢřæ■őïijŇä¡ĘæŸŕPythonèŕ■èĺĂåźűæšąæIJI

èğčåĘşæŰźææąĹ

PythonçĺŃåžŔåŠŸäÿ■åŐżä¿İèţŰèŕ■èĺĂçĽźæĂğåŐżåřĄèčĚæŢřæ■őïijŇèĂŇæŸŕéĂŽè£ĞéĄţå¿łäÿĂåőŽ
çňňäÿĂäÿłçžęåőŽæŸŕäżżä¡Ţäžěå■ŢäÿŃåĹŠçž£_åijĂåď ť çŽĎåŘ■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
```

Pythonåźűäÿ■äijŽçIJ§çŽĎéŸżæ■ćåĹńäžžèő£éŮőåĘĚéČĺåŘ■çğřãĂĆä¡ĘæŸŕåęĆæđIJä¡ãè£ŹäźĹåĄŽèĆŕ
åŘŇæŮűè£Ÿèę Ąæşĺæ ĎŔåĹřïijŇä¡£çŤĺäÿŇåĹŠçž£åijĂåď'ťçŽĎçžęåőŽåŘŇæăůéĂĆçŤĺäžŐæĺąåİŮåŘ■åŠŇæ
ä¿ŇåęĆïijŇåęĆæđIJä¡ăçIJŇåĹřæ§ŘäÿłæĺąåİŮåŘ■äžěå■ŢäÿŇåĹŠçž£åijĂåď'ť(æŕŤåęĆ_socket)ïijŇéĆčåőČåřš
çśżäijijçŽĎïijŇæĺąåİŮçžğåĹńåĞ¡æŢřæŕŤåęĆ `sys._getframe()`
åIJĺä¡£çŤĺçŽĎæŮűåĂŹåřšå¿ŮåŁääĂ■åřŔå£ČäžĘãĂĆ

ä¡ãè£ŸåŔŕèČ¡äijŽéĄĞåĹřåIJĺçśżåőŽäźL'äÿ■ä¡£çŤĺäÿď äÿłäÿŇåĹŠçž£(__)åijĂåď'ť çŽĎåŚ¡åŘ■ãĂĆæŕŤå

```
class B:
    def __init__(self):
        self.__private = 0

    def __private_method(self):
        pass

    def public_method(self):
        pass
        self.__private_method()
```

ä¡£çŤĺåŔŇäÿŇåĹŠçž£åijĂåğŇäijŽåŕijèĞť'èő£éŮőåŘ■çğřåŔŸæĹŘåĚűäżŰå¡ćåijŔãĂĆ
æŕŤåęĆïijŇåIJĺåL'■éİćçŽĎçśżBäÿ■ïijŇçğĄæIJL'åśđæĂğäijŽècńåĹĘåĹńéĞ■åŚ¡åŘ■äÿž
`_B__private` åŠŇ `_B__private_method` ãĂĆ è£ŹæŮűåĂŹä¡ăåŔŕèČ¡äijŽéŮőè£Źæăůé Ğ■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£ŹäÿłèůçĹűçśżBäÿ■çŽĎåŘ■çğřæŸŕåőŇåĚĺäÿ■åŘŇçŽĎãĂĆ

èőÍèőž

äÿŁéÌćæŔŘåĹřæIJL’äÿd’çğ■äÿ■åŘŇçŽĎçijŰçăAçžęåőŽ(å■ŢäÿŃåĹŠçž£åŠŇåŔŇäÿŃåĹŠçž£)æİěåŚ¡åŘ
åď’gåď’ŽæŢřèĂŇèÍĂïijŇä¡ăåžŤèŕěèőĺ’ä¡ăçŽĎéİđåĚňåĚśåŘ■çğřäžěå■ŢäÿŃåĹŠçž£åijĂåď’t’ãĂĆä¡ĘæŸŕïijŇåę
åźűäÿŤæIJL’äžŻåĘĚéČĺåśđæĂğåžŤèŕěåIJĺå■Řçśżäÿ■éŽŘèŮŔèţůæİěïijŇéĆčäźĹæL’■èĂČèŹŚä¡£çŤĺåŔŇäÿŃ

è£ŸæIJL’äÿĂçĆźèęĄæşĺæĎŔçŽĎæŸŕïijŇæIJL’æŮűåĂŹä¡ăåőŽäźL’çŽĎäÿĂäÿłåŔŸéĞŔåŠŇæ§Řäÿłä£İç

```
lambda_ = 2.0 # Trailing _ to avoid clash with lambda keyword
```

è£ŹéĞŇæĹŚäžňåźűäÿ■ä¡£çŤĺå■ŢäÿŃåĹŠçž£åL’■çijĂçŽĎåŐ§åŻăæŸŕåőČéĄ£åĚ■èŕŕèğčåőČçŽĎä¡£çŤĺ
(åęĆä¡£çŤĺå■ŢäÿŃåĹŠçž£åL’■çijĂçŽĎçŻőçŽĎæŸŕäÿžäžĘéŸšæ■ćåŚ¡åŘ■åĘšçłĄèĂŇäÿ■æŸŕæŇĞæŸŐè£Źă
éĂŽè£Ğä¡£çŤĺå■ŢäÿŃåĹŠçž£åŘŐçijĂåŔŕäžěèğčåĘşè£ŹäÿłéŮőécŸãĂĆ

10.6 8.6 åĹŻåžžåŔŕçőąçŘĘçŽĎåśđæĂğ

éŮőécŸ

ä¡ăæČşçżŹæ§Řäÿłåőđä¿ŃattributeåćđåŀăéŹd’èő£éŮőäÿŐä£őæŤźäźŃåď’ŰçŽĎåĚűäžŰåď’ĎçŘĘéĂżè¿Śi

èğčåĘşæŰźæąĹ

èĞłåőŽäźL’æ§ŘäÿłåśđæĂğçŽĎäÿĂçğ■çőĂå■ŢæŰźæşŢæŸŕåřĘåőČåőŽäźL’äÿžäÿĂäÿłpropertyãĂĆ
ä¿ŃåęĆïijŇäÿŃéİćçŽĎäžčçăĄåőŽäźL’äžĘäÿĂäÿłpropertyïijŇåćđåŀăåŕźäÿĂäÿłåśđæĂğçőĂå■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£ŕäžčçăĄäÿ■æIJL’äÿL’äÿłçŻÿåĚşèĄŢçŽĎæŰźæşŢïijŇè£ŹäÿL’äÿłæŰźæşŢçŽĎåŘ■å■ŮéČ¡å£Ěéąžäÿ
çňňäÿĂäÿłæŰźæşŢæŸŕäÿĂäÿł `getter` åĞ¡æŢŕïijŇåőČä¡£å¿Ů `first_name`

æĹŘäÿžäÿĂäÿłåśđæĂğãĂĆ åĚűäżŰäÿď'äÿłæŰźæşŢçżŹ `first_name` åśđæĂğæůżåŁääžĘ `setter` åŠŇ `deleter` åĞ¡æŢřãĂĆ éIJĂèęĄåijžèřČçŽĎđæŸŕåŔłæIJL'åIJÍ `first_name` åśđæĂğèćńåĹŻåżžåŘŐïijŇ åŘŐéİćçŽĎäÿď'äÿłèčĚéěřåŹÍ `@first_name.setter` åŠŇ `@first_name.deleter` æL'■èČ¡èćńåőŽäźL'ãĂĆ

propertyçŽĎäÿĂäÿłåĚşéŤőçL'źå¿ĄæŸŕåőČçIJŃäÿŁåŐżèů§æŹőéĂŽçŽĎattributeæšąäżĂäźĹäÿď'æăůïijŇ ä¡ĘæŸŕèő£éŮőåőČçŽĎđæŮűåĂŹäijŽèĞłåŁÍèğęåŔŚ `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
>>>
```

åIJÍåőđçŐřäÿĂäÿłpropertyçŽĎæŮűåĂŹïijŇåžŢåśĆæŢřæ■ő(åęĆæđIJæIJL'çŽĎèŕİ)äż■çĎűéIJĂèęĄå■ŸåĆ åŻăæ■d'ïijŇåIJÍgetåŠŇsetæŰźæşŢäÿ■ïijŇä¡ääijŽçIJŃåĹřářź `_first_name` åśđæĂğçŽĎæŞ■ä¡IJïijŇè£Źäź§æŸŕåőđéŹĚæŢřæ■őä£İå■ŸçŽĎåIJřæŰźãĂĆ åŔęåd'ŰïijŇä¡ăåŔŕèČ¡è£ŸäijŽéŮőäÿžäżĂäźĹ `__init__()` æŰźæşŢäÿ■èő¿ç¡őäžĘ `self.first_name` èĂŇäÿ■æŸŕ `self._first_name` ãĂĆ åIJÍè£Źäÿłä¿Ńå■Řäÿ■ïijŇæĹŠäżňåĹŻåżžäÿĂäÿłpropertyçŽĎçŽőçŽĎåřśæŸŕåIJÍèő¿ç¡őattributeçŽĎæŮűåĂŹ åŻăæ■d'ïijŇä¡ăåŔŕèČ¡æČşåIJÍåĹİåğŃåŇŰçŽĎæŮűåĂŹäź§è£ŻèąŇè£Źçğ■çśžåđŃæčĂæ§ěãĂĆéĂŽè£Ğèő¿ç¡ `self.first_name` ïijŇèĞłåŁÍèřČçŤÍ `setter` æŰźæşŢïijŇ è£ŹäÿłæŰźæşŢéĞŇéİćäijŽè£ŻèąŇåŔĆæŢřçŽĎæčĂæ§ěïijŇåŘęåĹŹåřśæŸŕçŻť'æŐèèő£éŮő `self._first_name` äžĘãĂĆ

è£ŸèČ¡åIJÍåůšå■ŸåIJÍçŽĎgetåŠŇsetæŰźæşŢå§žçąĂäÿŁåőŽäźL'propertyãĂĆä¿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)

```
    # 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

äýĂäýłpropertyåśđæĂğåĚűåőđåřśæŸŕäýĂçşżåĹŮçŻýåĚşçżŚåőŽæŰźæşŢçŽĎéŽĘåŘĹãĂĆåęĆæđIJä¡ăå
åřśäijŽåŘŚçŐřpropertyæIJňèžńçŽĎfgetãĂĄfsetåŠŇfdelåśđæĂğåřśæŸŕçśžéĞŇéİćçŽĎđæŹőéĂŽæŰźæşŢãĂĆ

```
>>> 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¡ăäý■äijŽçŻť æŐěåŔŰèřČçŤĺfgetæĹŰèĂĚfsetïijŇåőČäžňäijŽåIJĺèő£éŮőpropertyçŽ

åŘłæIJL’å¡Şä¡ăçąőåőđéIJĂèęĄåŕźattributeæL’ğèąŇåĚűäżŰéćİåď’ŰçŽĎđæŞ■ä¡IJçŽĎđæŮűåĂŹæL’■åžŤèřě
æIJL’æŮűåĂŹäýĂäžŽäżŐåĚűäżŰçijŰçÍŃèŕ■èÍĂ(æŕŤåęĆJava)è£ĞæİěçŽĎçÍŃåžŔåŚŸæĂżèőď’äÿžæL’ĂæIJL
æL’ĂäżěäžŰäžňèőď’äÿžäżčçăĄåžŤèřěåČŔäýŃéİćè£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
```

äý■èęĄåĘŹè£Źçğ■æšąæIJL’åĄŽäżżä¡ŢåĚűäžŰéćİåď’ŰæŞ■ä¡IJçŽĎpropertyãĂĆ
éęŰåĚĹïijŇåőČäijŽèőľ’ä¡ăçŽĎäżčçăĄåŔŸå¿Ůå¿ĹèĞČèĆ£ïijŇåźűäýŤè£ŸäijŽè£ůæČŚéŸĚèŕżèĂĚãĂĆ
åĚűæňąïijŇåőČè£ŸäijŽèőľ’ä¡ăçŽĎçÍŃåžŔè£ŘèąŇètůæİěåŔŸæĚćå¿Ĺåď’ŽãĂĆ
æIJĂåŘŐïijŇè£ŹæăůçŽĎèő¿èőąåźűæšąæIJL’åÿęæİěäżżä¡ŢçŽĎåě¡åď’ĎãĂĆ
çL’źåĹńæŸŕå¡Şä¡ăäżěåŘŐæČşçżŹæŹőéĂŽattributeèè£éŮőæůżåŁăéćİåď’ŰçŽĎåď’ĎçŘĘéĂżè¿ŚçŽĎæŮűåĂŹ
ä¡ăåŔŕäżěåřĘåőČåŔŸæĹŘäýĂäýłpropertyèĂŇæŮăéIJĂæŤźåŔŸåŐ§æİěçŽĎäžčçăĄãĂĆ
åŽääÿžèő£éŮőattributeçŽĎäžčçăĄè£ŸæŸŕä£ÍæŇĄåŐ§æăůãĂĆ

Propertiesè£ŸæŸŕäýĂçğ■åőŽäźL’åŁĺæĂĄèőąçőŮattributeçŽĎæŰźæşŢãĂĆ
è£Źçğ■çśžåđŃçŽĎattributesåźűäý■äijŽècńåőđéŹĚçŽĎå■ŸåĆĺïijŇèĂŇæŸŕåIJĺéIJĂèęĄçŽĎæŮűåĂŹèőąçőŮ

```
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£ŹéĞŇïijŇæĹŚäžňéĂŽè£Ğä¡£çŤÍpropertiesïijŇåřĘæĽ˙ĂæIJL˙çŽĎèő£éŮőæŐěåŔčå¡ćåijŔçż§äÿĂè
åŕźå■Łå¿ĎãĂĄçŻť˙å¿ĎãĂĄåŚĺéŢ£åŠŇéİćçğŕçŽĎèő£éŮőéČ¡æŸŕéĂŽè£ĞåśđæĂğèő£éŮőïijŇåřśèů§èő£éŮőç
åęĆæđIJäÿ■è£Źæăůå˛ŽçŽĎèŕİïijŇéĆčäźĹåřśèęĄåIJÍäžčçăĄäÿ■æůůåŘĹä¡£çŤÍçŐÅå■ŢåśđæĂğèő£éŮőåŠŇæ
äÿŃéİćæŸŕä¡£çŤÍçŽĎåőđä¿ŃïijŽ

```
>>> c = Circle(4.0)
>>> c.radius
4.0
>>> c.area  # Notice lack of ()
50.26548245743669
>>> c.perimeter  # Notice lack of ()
25.132741228718345
>>>
```

år¡çőąpropertiesåŔŕäžěåőđçŐřäijŸéŻĚçŽĎçijŰçÍŃæŐěåŔčïijŇä¡ĘæIJL˙äžŻæŮűåĂŹä¡ăè£ŸæŸŕäijŽæČ

```
>>> p = Person('Guido')
>>> p.get_first_name()
'Guido'
>>> p.set_first_name('Larry')
>>>
```

è£Źçğ■æČĚåĘţçŽĎåĞžçŐřéĂŽåÿÿæŸŕåŻääÿžPythonäžčçăĄèćńéŻĘæĹŘåĹřäÿĂäÿłåď˙ğåđŃå§žçąĂáźş
ä¿ŃåęĆïijŇæIJL˙åŔŕèČ¡æŸŕäÿĂäÿłPythonçśžåĞĘåď˙ĞåŁäåĚěåĹřäÿĂäÿłå§žäžŐè£IJçÍŃè£ĞçÍŃèřČçŤÍçŽĎåć
è£Źçğ■æČĚåĘţäÿŃïijŇçŻť˙æŐěä¡£çŤÍget/setæŰźæşŢ(æŹőéĂŽæŰźæşŢèřČçŤÍ)èĂŇäÿ■æŸŕpropertyæĹŰèőÿ

æIJĂåŘŐäÿĂçĆźïijŇäÿ■èęĄåČŔäÿŃéİćè£ŹæăůåĘŹæIJL˙åď˙ğéĞŔéĞ■åď˙■äžčçăĄçŽĎpropertyåőŽäźL˙ï

```
class Person:
    def __init__(self, first_name, last_name):
        self.first_name = first_name
        self.last_name = last_name
```

(continues on next 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
åŔŕäžěåŔĆèĂČ8.9åŠŇ9.21åřŔèŁĆçŽĎåĘĚåőźãĂĆ

10.7 8.7 èřČçŤÍçĹűçśżæŰźæşŢ

éŮőéćŸ

ä¡ăæČşåIJÍå■Řçśżäÿ■èřČçŤÍçĹűçśżçŽĎæ§ŘäÿłåůšçżŔèćńèęĘçŻŰçŽĎæŰźæşŢãĂĆ

èğčåĘşæŰźæąĹ

äÿžäžĘèřČçŤÍçĹűçśż(èűĚçśż)çŽĎäÿĂäÿłæŰźæşŢïijŇåŔŕäžěä¡£çŤÍ `super()` åĞ¡æŢřïijŇærŤåęĆïijŽ

```
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'Űäÿ ÄäÿłåÿÿèğĄçŤĺæşŢåĞžçŐřåIJĺèęEçŻŰPythonçL'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)
```

åIJĺäÿŁéİćäžčçăĄäÿ■ïijŇ__setattr__() çŽĎåőđçŐřåŇĚåŔńäÿÄäÿłåŔ■å■ŮæčÄæ§ěãĂĆ åęĆæđIJæ§ŘäÿłåśđæĂğåŔ■äżěäÿŇåĹŠçž£(_)åijĂåď't'ïijŇåřśéĂŽè£Ğ super() èřČçŤĺåŐ§åğŇçŽĎ __setattr__() ïijŇ åŘęåĹŹçŽĎèŕĺåřśåğŤæt'¿çżŹåĘĚéČĺçŽĎäžčçŘĘåŕźèšą self._obj åŐżåď'ĎçŘĘãĂĆ è£ŹçIJŃäÿŁåŐżæIJL'çĆźæĎŔæĂİïijŇåŻääÿžåřśçőŮæšąæIJL'æŸ¿åijŔçŽĎæ super() äż■çĎűåŔŕäžěæIJL'æŢĹçŽĎåůěä¡IJãĂĆ

èőĺèőž

åőđéŹĚäÿŁïijŇåď'ğåőűåŕźäžŐåIJĺPythonäÿ■åęĆä¡Ţæ■čçąőä¡£çŤĺ super() åĞ¡æŢřæŹőéA■ç§ëäźŃçŤŽåřŠãĂĆ ä¡ăæIJL'æŮűåĂŹäijŽçIJŃåĹřåČŔäÿŃéİćè£Źæăůç Żt'æŐěěřČçŤĺçĹűçśżç

```
class Base:
    def __init__(self):
        print('Base.__init__')

class A(Base):
    def __init__(self):
        Base.__init__(self)
        print('A.__init__')
```

åř¡çőąåŕźäžŐåď'ğéČĺåĹĘäžčçăĄèĂŇèĺĂè£ŹäźĹåĄŽæšąäżĂäźĹéŮőécŸïijŇä¡ĘæŸŕåIJĺæŻt'åď'■æİĆçŽĎ æŕŤåęĆïijŇèĂČèŹŠåęĆäÿŃçŽĎæČĚåĘţïijŽ

```
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£ŹæőţäżčçăĄåřśäijŽåŔŚçŐř `Base.__init__()` èćńèřČçŤĺäÿď æňąïijŇåęĆäÿŃæĽĂçď žïijŽ

```
>>> c = C()
Base.__init__
A.__init__
Base.__init__
B.__init__
C.__init__
>>>
```

åŔŕèČ¡äÿď æňąèřČçŤĺ `Base.__init__()` æšąäżĂäźĹåİŔåď ĎïijŇä¡ĘæIJL’æŮűåĂŹå■t’äÿ■æŸŕãĂĆ åŔęäÿĂæŰźéİćïijŇåĄĞèő¿ä¡ăåIJĺäżčçăĄäÿ■æ■ćæĹŘä¡£çŤĺ `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)

```
    def __init__(self):
        super().__init__()  # Only one call to super() here
        print('C.__init__')
```

è£ŘèąŇè£ŹäÿłæŰřçĽŁæIJňåŘŐïijŇä¡ăäijŽåŔŚçŐřæŕŔäÿł `__init__()` æŰźæşŢåŔłäijŽèćńèřČçŤĺäÿĂæňąäžĘïijŽ

```
>>> c = C()
Base.__init__
B.__init__
A.__init__
C.__init__
>>>
```

äÿžäžĘåijĎæÿĚåőČçŽĎåŐ§çŘĘïijŇæĹŚäžňéIJĂèęĄèŁśçĆźæŮűéŮť'èğčéĞŁäÿŃPythonæŸŕåęĆä¡Ţåőđ
åŕźäžŐä¡ăåőŽäźĽ'çŽĎæŕŔäÿĂäÿłçśžïijŇPythonäijŽèőąçőŮåĞžäÿĂäÿłæĽ'ĂèřŞçŽĎæŰźæşŢèğčæđŘéąžåžŔ(M
è£ŹäÿłMROåĹŮèąĺåřśæŸŕäÿĂäÿłçőĂå■ŢçŽĎæĽ'ĂæIJL'å§žçśžçŽĎçž£æĂğéąžåžŔèąĺãĂĆä¿ŃåęĆïijŽ

```
>>> C.__mro__
(<class '__main__.C'>, <class '__main__.A'>, <class '__main__.B'>,
<class '__main__.Base'>, <class 'object'>)
>>>
```

äÿžäžĘåőđçŐřçžğæĽ'£ïijŇPythonäijŽåIJÍMROåĹŮèąĺäÿŁäžŐåůęåĹřåŘşåijĂåğŃæ§ěæĽ'¿å§žçśžïijŇçŻť'

èĂŇè£ŹäÿłMROåĹŮèąĺçŽĎæđĎéĂăæŸŕéĂŽè£ĞäÿĂäÿłC3çž£æĂğåŇŰçőŮæşŢæİěåőđçŐřçŽĎãĂĆ
æĹŚäžňäÿ■åŐżæůśçľ'ûè£ŹäÿłçőŮæşŢçŽĎæŢřå■ęåŐ§çŘĘïijŇåőČåőđéŹĚäÿŁåřśæŸŕåŘĹåźűæĽ'ĂæIJL'çĹűçş

- å■ŘçśżäijŽåĚĹäžŐçĹűçśżèćńæčĂæ§ě
- åd'ŽäÿłçĺűçśżäijŽæăźæ■őåőČäžňåIJĺåĹŮèąĺäÿ■çŽĎéąžåžŔèćńæčĂæ§ě
- åęĆæđIJåŕźäÿŇäÿĂäÿłçśżå■ŸåIJĺäÿd'äÿłåŘĹæşŢçŽĎéĂL'æŇl'ïijŇéĂL'æŇl'çňňäÿĂäÿłçĹűçśż

èĂĄåőđèŕť'ïijŇä¡ăæL'ĂèęAç§ěéĄŞçŽĎåřśæŸŕMROåĹŮèąĺäÿ■çŽĎçśżéąžåžŔäijŽèől'ä¡ăåőŽäźL'çŽĎäžz

å¡Şä¡ăä¡£çŤĺ `super()` åĞ¡æŢřæŮűïijŇPythonäijŽåIJÍMROåĹŮèąĺäÿŁçžğçż■æŘIJçţ'ćäÿŇäÿĂäÿłçśżãĂ
åŘłèęĄæŕŔäÿłéĞ■åőŽäźĽ'çŽĎæŰźæşŢçż§äÿĂä¡£çŤĺ `super()`
åźűåŘłèŕČçŤĺåőČäÿĂæňąïijŇ éĆčäźĹæŐğåĹűæţĄæIJĂçžĹäijŽéĄ■åŐĘåőŇæŢť'äÿłM-
ROåĹŮèąĺïijŇæŕŔäÿłæŰźæşŢäź§åŘłäijŽèćńèřČçŤĺäÿĂæňąãĂĆ
è£Źäź§æŸŕäÿžäžĂäźĹåIJĺçňňäžŇäÿłä¿Ńå■Řäÿ■ä¡ääÿ■äijŽèřČçŤĺäÿd'æňą `Base.__init__()` çŽĎåŐ§åŽãĂĆ

`super()` æIJL'äÿłäžd'äžžåŘČæČŁçŽĎåIJřæŰźæŸŕåőČåźűäÿ■äÿĂåőŽåŐżæ§ěæĽ'¿æ§ŘäÿłçśżåIJÍMRO
ä¡ăçŤŽèĞşåŘŕäžěåIJĺäÿĂäÿłæšąæIJL'çŻť'æŐěçĹűçśżçŽĎçśżäÿ■ä¡£çŤĺåőČãĂĆä¿ŃåęĆïijŇèĂČèŹŠåęĆäÿŃè

```
class A:
    def spam(self):
        print('A.spam')
        super().spam()
```

åęĆæđIJä¡ăèŕŢçİĂçŻť'æŐěä¡£çŤĺè£ŹäÿłçśżåřšäijŽåĞžéŤŹïijŽ

```
>>> 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¡ää¡£çŤĺåď'ŽçżğæL'£çŽĎèŕİçIJŃçIJŃäijŽåŔŠçŤ§äžĂäźĹïijŽ

```
>>> class B:
...     def spam(self):
...         print('B.spam')
...
>>> class C(A,B):
...     pass
...
>>> c = C()
>>> c.spam()
A.spam
B.spam
>>>
```

ä¡ääŔŕäžěçIJŃåĹřåIJĺçśżAäÿ■ä¡£çŤĺ `super().spam()` åőđéŹĚäÿŁèřČçŤĺçŽĎæŸŕèů§çśżAæŕńæŮäåĚşçşżçŽĎçśżBäÿ■çŽĎ `spam()` æŰźæşŢãĂĆ è£ŹäÿłçŤĺçśżCçŽĎMROåĹŮèąĺářśåŔŕäžěåőŇåĚĺèğčéĞŁæÿĚæěŽäžĘïijŽ

```
>>> C.__mro__
(<class '__main__.C'>, <class '__main__.A'>, <class '__main__.B'>,
<class 'object'>)
>>>
```

åIJĺåőŽäźL'æůůåĚěçśżçŽĎæŮűåĂŹè£Źæäůä¡£çŤĺ `super()` æŸŕå¿ĹæŹőéĄ■çŽĎãĂĆåŔŕäžěåŔĆèĂČ8.13åŠŇ8.18ářŔèŁĆãĂĆ

çĎűèĂŇïijŇçŤśäžŐ `super()` åŔŕèČ¡äijŽèřČçŤĺäÿ■æŸŕä¡ăæČşèęĄçŽĎæŰźæşŢïijŇä¡ăåžŤèŕěéĄţå¿łäÿ éęŰåĚĹïijŇçąőä£ÍåIJĺçžğæL'£ä¡Şçşżäÿ■æL'ĂæIJL'çŻÿåŘŇåŘ■å■ŮçŽĎæŰźæşŢæŃěæIJL'åŔŕåĚijåőźçŽĎåŔ è£Źæäůåŕäžěçąőä£Í `super()` èřČçŤĺäÿĂäÿłéİđçŻť'æŐěçĹűçśżæŰźæşŢæŮűäÿ■äijŽåĞžéŤŹãĂĆ åĚűæňąïijŇæIJĂåě¡çąőä£ÍæIJĂéąűåśĆçŽĎçśżæŔŘä¿ŻäžĘè£ŹäÿłæŰźæşŢçŽĎåőđçŐřïijŇè£ŹæäůçŽĎèŕİåIJĺ

åIJĺPythonçď¿åŇžäÿ■åŕźäžŐ `super()` çŽĎä¡£çŤĺæIJL'æŮűåĂŹäijŽåijŢæİěäÿĂäžŻäžL'èőőãĂĆ åř¡çőąåęĆæ■ď'ïijŇåęĆæđIJäÿĂåĹĞéąžåĹl'çŽĎèŕİïijŇä¡ăåžŤèŕěåIJĺä¡ăæIJĂæŰřäžčçăĄäÿ■ä¡£çŤĺåőĆãĂĆ Raymond Hettingeräÿžæ■ď'åĘŹäžĘäÿĂçŕĞéİđåÿÿåě¡çŽĎæŰĞçńă âĂIJPythonâĂŹs `super()` Considered Super!âĂİ ïijŇ éĂŽè£Ğåď'ğéĞŔçŽĎä¿Ńå■ŘåŘŚæĹŚäžňèğčéĞŁäžĘäÿžäžĂäźĹ `super()` æŸŕæđĄåě¡çŽĎãĂĆ

10.8 8.8 å■Řçśżäÿ■æĽľåśŢproperty

éŮőécŸ

åIJĺå■Řçśżäÿ■ïijŇä¡ăæČşèęĄæĽľåśŢåőŽäźĽåIJĺçĹűçśżäÿ■çŽĎpropertyçŽĎåŁ§èČ¡ãĂĆ

èğčåĘşæŰźæąĹ

èĂČèŹŚåęĆäÿŃçŽĎäžčçăĄïijŇåőČåőŽäźĽäžĘäÿĂäÿłpropertyïijŽ

```
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")
```

äÿŃéİćæŸŕäÿĂäÿłçd'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¡ăäžĚäžĚåŔłæČşæL'l'åśŢpropertyçŽĎæ§ŘäÿĂäÿłæŰźæşŢïijŇéĆčäźĹåŔŕäžěåČŔäÿŃéİćè£Źæ

```
class SubPerson(Person):
    @Person.name.getter
    def name(self):
        print('Getting name')
        return super().name
```

æĹŰèĂĚïijŇä¡ăåŔłæČşä£őæŤzsetteræŰźæşŢïijŇářśè£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'åśŢäÿĂäÿłpropertyåŔŕèČ¡äijŽåijŢèţůå¿Ĺåd'Žäÿ■æŸŞår§ègL'çŽĎéŮőécŸïijŇ åŻăäÿžäÿĂäÿłpropertyåĚűåðđæŸŕ getterãĂĄsetter åŠŇ deleter æŰźæşŢçŽĎéŻĘåŘĹïijŇèĂŇäÿ■æŸŕå■ŢäÿłæŰźæşŢãĂĆ åŻăæ■d'ïijŇå¡Şä¡ăæL'l'åśŢäÿĂäÿłpropertyçŽĎæŮűåĂŹïijŇä¡ăéIJĂèęĄåĚĹçąőåőŽä¡ăæŸŕåŘęèęĄéĞ■æŰřåő

åIJÍçňňäÿĂäÿłä¿Ńå■Řäÿ■ïijŇæL'ĂæIJL'çŽĎpropertyæŰźæşŢéČ¡èćńéĞ■æŰřåőŽäźL'ãĂĆ åIJÍæŕŔäÿĂäÿłæŰźæşŢäÿ■ïijŇä¡£çŤÍäžĘ super() æİěèřČçŤÍçĹűçśżçŽĎåőđçŐřãĂĆ åIJÍ setter åĞ¡æŢřäÿ■ä¡£çŤÍ super(SubPerson, SubPerson).name.__set__(self, value) çŽĎèŕ■åŘěæŸŕæšąæIJL'éŤŹçŽĎãĂĆ äÿžäźĘåğŤæL'ŸçżŹäźŃåL'■åőŽäźL'çŽĎsetteræŰźæşŢïijŇéIJĂèęĄåřĘæŐğåĹűæİČäijäéĂŠçżŹäźŃåL'■åőŽäź __set__() æŰźæşŢãĂĆ äÿ■è£ĞïijŇèŐůåŘŰè£ŹäÿłæŰźæşŢçŽĎåŤŕäÿĂéĂŤå¿ĎæŸŕä¡£çŤÍçśżåŘŸéĞŔèĂ è£Źäź§æŸŕäÿžäźĂäźĹæĹŚäžňèęĄä¡£çŤÍ super(SubPerson, SubPerson) çŽĎåŐ§åŻăãĂĆ

åęĆædIJä¡ăåŔłæČşéĞ■åőŽäźL’åĚűäÿ■äÿĂäÿłæŰźæşŢïijŇéĆčåŔłä¡£çŤÍ @property æIJňèžńæŸŕäÿ■åď§çŽĎãĂĆæŕŤåęĆïijŇäÿŃéİćçŽĎäžčçăĄåřśæŮăæşŢåůëä¡IJïijŽ

```
class SubPerson(Person):
    @property  # Doesn't work
    def name(self):
        print('Getting name')
        return super().name
```

åęĆædIJä¡ăèŕŢçİĂè£ŘèąŇäijŽåŔŚçŐřsetteråĞ¡æŢřæŢt’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¡ăåžŤèŕëåČŔäźŃåL’■èŕt’è£ĞçŽĎéĆčæăůä£őæŤźäžčçăĄïijŽ

```
class SubPerson(Person):
    @Person.name.getter
    def name(self):
        print('Getting name')
        return super().name
```

è£ŹäźĹåĘŹåŘŐïijŇpropertyäźŃåL’■åůšçżŔåőŽäźL’è£ĞçŽĎæŰźæşŢäijŽèćńåď■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£ŹäÿłçL’źåĹńçŽĎèğčåĘşæŰźæąĹäÿ■ïijŇæĹŚäžňæšąåŁđæşŢä¡£çŤÍæŻt’åŁăéĂŽçŤÍçŽĎæŰźåijŔåČ Person çśžåŘ■ãĂĆ åęĆædIJä¡ääÿ■ç§ěéĄŞåĹřåžŢæŸŕåŞläÿłå§žçśžåőŽäźL’äžĘpropertyïijŇ éĆčä¡ăåŔłèČ¡éĂŽè£ĞéĞ■æŰřåőŽäźL’æL’ĂæIJL’propertyåźűä¡£çŤÍ super() æİěårĘæŐğåĹűæİČäijăéĂŠçżŹåL’■éİćçŽĎåőđçŐřãĂĆ

åĂijå¿ŮæşÍæĎŔçŽĎæŸŕäÿŁéİćæijŢçd’žçŽĎçňňäÿĂçğ■æŁĂæIJŕè£ŸåŔŕäžěèćńçŤÍæİěæL’l’åśŢäÿĂäÿłæ

```
# 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¿Ůæş́æĎŘçŽĎæŸŕïijŇèŕżåĹřè£ŹéĞŇæŮűïijŇä¡ååžŤèŕěäijŽåŔŞçŐřå■Řçśżå ŇŰ setter åŠŇ deleter æŰźæşŢåĚűåőđæŸŕå¿ĹçőĂå■ŢçŽĎãĂĆ è£ŹéĞŇæijŤçd'žçŽĎèğčåĘşæŰźæąĹåŘŇæăůéĂĆçŤĺïijŇä¡ĘæŸŕåIJÍ PythonçŽĎissueéąţéÍć æŁěåŚŁçŽĎäÿĂäÿłbugïijŇæĹŰèőÿäijŽä¡£å¿ŮåřĘæÍěçŽĎPythonçL'ĹæIJñäÿ■åĞžçŐřäÿĂäÿłæŻt'åŁăçőÄæt'

10.9 8.9 åĹŻåżžæŰřçŽĎçśżæĹŰåőđä¿ŃåśđæĂğ

éŮőécŸ

ä¡ăæČşåĹŻåżžäÿĂäÿłæŰřçŽĎæŇěæIJL'äÿĂäžŻécÍåd'ŰåŁ§èČ¡çŽĎåőđä¿ŃåśđæĂğçśżåđŇïijŇæŕŤåęĆç

èğčåĘşæŰźæąĹ

åęĆædIJä¡ăæČşåĹŻåżžäÿĂäÿłåĚĺæŰřçŽĎåőđä¿Ńåśđæ Ăğïij ÑåŔŕäžěéĂŽè£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£řåŹĺářśæŸŕäÿĂäÿłåőđçŐřäžĘäÿL'äÿłæăÿå£ČçŽĎåśđæĂğèő£éŮőæŞ■ä¡IJ(get, set, delete)çŽĎçśżïijŇ åĹĘåĹńäÿž __get__() ãĂĄ__set__() åŠŇ __delete__() è£ŹäÿL'äÿłçL'źæőŁçŽĎæŰźæşŢãĂĆ è£ŹäžŽæŰźæşŢæŐěåŔŮäÿĂäÿłåőđä¿Ńä¡IJäÿžè¿ŞåĚëijŇäźŃåŘŐçŽÿåžŤçŽĎæŞ■ä¡IJåőđä¿ŃåžŢåśĆçŽĎå■

äÿžäžĘä¡£çŤĺäÿĂäÿłæŔŔè£řåŹĺïijŇéIJĂářĘè£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)

```
TypeError: Expected an int
>>>
```

ä¡IJäÿžè¿ŞåĚëïijŇæŔŘè£řåŹÍçŽĎæŕŘäÿĂäÿłæŰźæşŢäijŽæŐěåŘŮäÿĂäÿłæŞ■ä¡IJåőđä¿ŃãĂĆ äÿžäžĘåőđçŐřèŕůæśĆæŞ■ä¡IJïijŇäijŽçŻÿåžŤçŽĎæŞ■ä¡IJåőđä¿ŃåžŢåśĆçŽĎå■ŮåĚÿ(__dict__åśđæĂğ)ãĂĆ æŔŘè£řåŹÍçŽĎ `self.name` åśđæĂğå■ŸåĆÍäžĘåIJÍåőđä¿Ňå■ŮåĚÿäÿ■èćńåőđéŹĚä¡£çŤÍåĹřçŽĎkeyãĂĆ

èőÍèőž

æŔŘè£řåŹÍåŔŕåőđçŐřåď'ğéČÍåĹĘPythonçśžçL'źæĂğäÿ■çŽĎåžŢåśĆé■ŤæşŢïijŇ åŇĚæŃň `@classmethod` ãĂĄ`@staticmethod` ãĂĄ`@property` ïijŇçŤŽèĞşæŸŕ `__slots__` çL'źæĂğãĂĆ

éĂŽè£ĞåőŽäźL'äÿĂäÿłæŔŘè£řåŹÍïijŇä¡ãåŔŕäžěåIJÍåžŢåśĆæ■ŢèŐůæäÿå£ČçŽĎåőđä¿ŃæŞ■ä¡IJ(get, set, delete)ïijŇåźűäÿŤåŔŕåőŇåĚÍèĞłåőŽäźL'åőČäžňçŽĎèąŇäÿžãĂĆ è£ŹæŸŕäÿĂäÿłåijžåď'ğçŽĎåůěåĚůïijŇæIJL'äžĘåőČä¡ãåŔŕäžěåőđçŐřå¿Ĺåď'ŽéńŸçžğåŁ§èČ¡ïijŇåźűäÿŤåőČä

æŔŘè£řåŹÍçŽĎäÿĂäÿłæŕŤè¿ČåŻřæČŠçŽĎåIJřæŰźæŸŕåőČåŔłèČ¡åIJÍçśžçžğåĹńèćńåőŽäźL'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__()` æŰźæşŢåőđçŐřètůæİěæŕŤçIJŃäÿŁåŐżèęĄåď'■æİĆå¿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'■æİĆçŽĎåŐ§åŻää¡Šçżşäž Őåőđä¿ŃåŔŸéĞŔåŠŇçśžåŔŸéĞŔçŽĎ åęĆæđIJäÿĂäÿłæŔŘè£řåŹÍèćńå¡ŞåĄŽäÿĂäÿłçśžåŔŸéĞŔæİèè£éŮőïijŇéĆčäźĹ `instance` åŔĆæŢřèćńèő¿ç¡őæĹŘ `None` ãĂĆ è£Źçğ■æČĚåĘţäÿŃïijŇæăĞåĞĘåĄŽæşŢåřśæŸŕçőĂå■ŢçŽĎè£ŤåŻđè£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>
>>>
```

æŔŘè£řåŹĺéĂŽåÿÿæŸŕéĆčäžŻä¡£çŤĺåĹřèčĚéěřåŹĺæĹŰåĚČçśżçŽĎåď ğåđŃæąĘæđűäÿ■çŽĎäÿĂäÿłçżĎ
äÿ¿äÿłä¿Ńå■ŘïijŇäÿŃéĺćæŸŕäÿĂäžŻæŻť éńŸçžğçŽĎå§žäžŐæŘŔè£řåŹĺçŽĎäžčçăĄïijŇåźűæűĽ åŔŁåĹřäÿĂ

```
# 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
```

æIJĂåŘŐèęĄæŇĞåĞžçŽĎäÿĂçĆźæŸŕïijŇåęĆæđIJä¡ăåŔłæŸŕæČşçőĂå■ŢçŽĎèĞłåőŽäźĽæ§Řäÿłçśżç
è£Źçğ■æČĚåĘţäÿŃä¡£çŤĺ8.6åřŔèŁĆäžŃçż■çŽĎpropertyæŁĂæIJŕäijŽæŻť åŁăåőźæŸŞãĂĆ
å¡ŞçÍŃåžŔäÿ■æIJL å¿Ĺåď ŽéĞ■åď ■äžčçăĄçŽĎæŮűåĂŹæŘŔè£řåŹĺåřśå¿ĹæIJL çŤĺäžĘ
(æŕŤåęĆä¡ăæČşåIJĺä¡ăäžčçăĄçŽĎå¿Ĺåď ŽåIJřæŰźä¡£çŤĺæŘŔè£řåŹĺæŘŔä¿ŻçŽĎåŁ§èČ¡æĹŰèĂĚårĘåőČä¡Ł

10.10 8.10 ä¡£çŤĺåžűè£§èőąçőŮåśđæĂğ

éŮőécŸ

ä¡ăæČşåřĘäÿĂäÿłåŔłèŕžåśđæĂğåőŽäźĽæĹŘäÿĂäÿłpropertyïijŇåźűäÿŤåŔłåIJĺèő£éŮőçŽĎæŮűåĂŹæĽ
ä¡ĘæŸŕäÿĂæŮęèćńèő£éŮőåŘŐïijŇä¡ăåÿŇæIJŻçżŞæđIJåĂijèćńçijŞå■ŸèţůæÍëïijŇäÿ■çŤĺæŕŘæňąéČ¡åŐżèőą

èğčåĘşæŰźæąĹ

åőŽäźĽäÿĂäÿłåžűè£§åśđæĂğçŽĎäÿĂçğ■énŸæŢĹæŰźæşŢæŸŕéĂŽè£Ğä¡£çŤĺäÿĂäÿłæŔŔè£řåŹĺçśżïijŇ

```
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
```

äÿŃéİćåIJĺäÿĂäÿłäžd'äžŠçŐŕåćČäÿ■æijŤçd'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
>>>
```

äžŤçżĘèğĆåŕ§ä¡ääijŽåŔŚçŐřæűĹæĄŕ `Computing area` åŠŇ `Computing perimeter` äžĚäžĚåĞžçŐřäÿĂæňąãĂĆ

èőĺèőž

å¿Ĺåď˙ŽæŮűůåĂŹïijŇæđĎéĂääÿĂäÿłåžűè£§èőąçőŮåšđæĂğçŽĎäÿžèęĄçŽőçŽĎæŸŕäÿžäžĘæŔŘå■ĞæĂ
ä¿ŇåęĆïijŇä¡ääŔŕäžěéĄ£åĚ■èőąçőŮè£ŹäžŽåšđæĂğåĂijïijŇéŹď˙éİđä¡ăçIJ§çŽĎéIJĂèęĄåőČäžñãĂĆ
è£ŹéĞŇæijŤçď˙žçŽĎæŮźæąĹåřśæŸŕçŤĺæİěåőđçŐřè£ŹæăůçŽĎæŢĹæđIJçŽĎïijŇ
åŘłäÿ■è£ĞåőČæŸŕéĂŽè£ĞäžěéİđåÿÿéńŸæŢĹçŽĎæŮźåijŔä¡£çŤĺæŔŘè£řåŹĺçŽĎäÿĂäÿłçš¿åęŹçĽ˙źæĂğæİě

æ■čåęĆåIJĺåĚűäžŮåřŔèŁĆ(åęĆ8.9åřŔèŁĆ)æĽ˙Ăè őšçŽĎéĆčæăůïijŇå¡ŞäÿĂäÿłæŔŘè£řåŹĺèćńæŤ¿åĚěäÿ
æŕŘæňąèő£éŮőåšđæĂğæŮűůåőČçŽĎ `__get__()` ãĂĄ`__set__()` åŠŇ `__delete__()` æŰźæşŢåřśäijŽèćńèğęåŔŚãĂĆ äÿ■è£ĞïijŇåęĆæđIJäÿĂäÿłæŔŘè£řåŹĺäžĚäžĚåŘłåőŽäźĽäžĘäÿĂäÿł `__get__()` æŰźæşŢçŽĎèŕİïijŇåőČæŕŤéĂŽåÿÿçŽĎåĚůæIJĽæŻť˙åijśçŽĎçżŚåőŽãĂĆ çĽ˙źåĹńåIJŕïijŇåŘłæIJĽ˙å¡Şèćńèő£éŮőåšđæĂğäÿ■åIJĺåőđä¿ŇåžŢåśĆçŽĎå■ŮåĚÿäÿ■æŮű `__get__()` æŰźæşŢæĽ˙■äijŽèćńèğęåŔŚãĂĆ

`lazyproperty` çśżåĽľçŤĺè£ŹäÿĂçĆźïijŇä¡£çŤĺ `__get__()` æŰźæşŢåIJĺåőđä¿Ňäÿ■å■ŸåĆĺèőąçőŮåĞžæİěçŽĎåĂijïijŇ è£Źäÿłåőđä¿Ňä¡£çŤĺçŻÿåŘŇçŽĎåŘ■å■Ůä¡IJäÿžä
è£ŹæăůäÿĂæİěïijŇçżŞæđIJåĂijèćńå■ŸåĆĺåIJĺåőđä¿Ňå■ŮåĚÿäÿ■åźűäÿŤäžěåŘŐářśäÿ■éIJĂèęĄåĘ■åŐžèőąçő
ä¡ääŔŕäžěåřÌèŕŢæŻť˙æůśåĚěçŽĎä¿Ňå■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■æŰźæąĹæIJĽäÿĂäÿłåřŔçijžéŹůåřśæŸŕèőąçőŮåĞžçŽĎåĂijèćńåĹŻåžžåŘŐæŸŕåŔŕäžěèćńä£őæŤ

```
>>> c.area
Computing area
50.26548245743669
>>> c.area = 25
>>> c.area
25
>>>
```

åęĆædIJä¡ăæŃĚå£Čè£ŹäÿłéŭőécŸïijŇéĆčäźĹåŔŕäżěä¡£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
```

åęĆædIJä¡ăä¡£çŤĺè£Źäÿłçl'ĹæIJñïijŇåŕšäijŽåŔŠçŐřçŐřåIJĺä£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£ĚéążèćńåőŽåŘŠåĹřă
getter åĞ¡æŢřäÿŁåŐżãĂĆ è£Źäÿłèů§äźŇåL'■çőĂå■ŢçŽĎåIJĺåőđä¿Ńå■ŮåĚÿäÿ■æ§ěæL'¿åĂijçŽĎæŰźæąĺ
åęĆædIJæČşèŐůåŔŰæŻt'åd'ŽåĚşäžŐpropertyåŠŇåŔŕçőąçŘĘåśđæĂğçŽĎä£ąæĄŕïijŇåŔŕäżěåŔĆèĂČ8.6åřŔ

10.11 8.11 çőĂåŇŰæŢřæ■őçżŞæđĎçŽĎåĹiåğŃåŇŰ

éŮőécŸ

ä¡ăåĘźäžĘå¿Ĺåd'ŽäžĚäżĚçŤĺä¡IJæŢřæ■őçżŞæđĎçŽĎçśżïijŇäÿ■æČşåĘŹåd'łåd'ŽçČęäžžçŽĎ
__init__() åĞ¡æŢř

èğčåĘşæŰźæąĹ

åŔŕäžěåIJĺäÿĂäÿłå§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£Źäÿłå§žçśż:

```
# 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¿ŇïijŽ

```
>>> 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■ŮåŔĆæŢřïijŇåŔŕäžěårĘåĚşéŤőå■ŮåŔĆæŢřèő¿ç¡őäÿžåőđä¿ŃåšđæÀ

```
class Structure2:
    _fields = []
```

(continues on next 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)

```
            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¡ăéIJĂèęĄä¡£çŤÍåď ğéĞŔå¿ĹåřŔçŽĎæŢřæ■őçżŞæđĎçśżçŽĎæŮűåĂŹïijŇ çŻÿæŕŤæĽŃåůěäýĂäýłäýłåőŽäźĽ `__init__()` æŰźæşŢèĂŇåůšïijŇä¡£çŤÍè£ŹçğæŰźåijŔåŔŕäžěåď ğåď ğ

åIJÍäýŁéÍćçŽĎåőđçŐřäý■æĹŚäžňä¡£çŤÍäžĘ `setattr()` åĞ¡æŢřçśżèő¿ç¡őåśđæĂğåĂijïijŇ ä¡ăåŔŕèČ¡äý■æČşçŤÍè£ŹçğæŰźåijŔïijŇèĂŇæŸŕæČşçŻťæŐëæŻťæŰřåőć

```
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£ŹäźŞåŔŕäžěæ■čåýÿåůëä¡IJïijŇä¡ĘæŸŕå¡ŞåőŽäźĽå■ŘçśżçŽĎæŮűåĂŹéŮőécŸåřśæİěäžĘãĂĆ å¡ŞäýĂäýłå■ŘçśżåőŽäźĽäžĘ `__slots__` æĹŰèĂĚéĂŽè£Ğproperty(æĹŰæŔŔè£řåŹĺ)æİěåŇĚècĚæ§Řäýłå éĆčäźĹçŻťæŐëèő£éŮőåőđä¿Ńå■ŮåĚÿåřśäý■èţůä¡IJçŤÍäžĘãĂĆæĹŚäžňäýŁéÍćä¡£çŤÍ `setattr()` äijŽæŸ¿å¿ŮæŻťéĂŽçŤÍäžŽïijŇåŽääýžåőČäź§éĂĆçŤÍäžŐå■ŘçśżæČĚåĘţãĂĆ

è£Źçğ■æŰźæşŢåŤŕäýĂäý■åě¡çŽĎåIJŕæŰźåřśæŸŕåŕźæ§ŘäžZIDEèĂŇèĺĂïijŇåIJĺæŸ¿çď žåýőåŁľåĞ¡æŢ

```
>>> help(Stock)
Help on class Stock in module __main__:
class Stock(Structure)
...
| Methods inherited from Structure:
|
| __init__(self, *args, **kwargs)
|
...
>>>
```

åŘŕäžěåŘĆèĂČ9.16åřŔèŁĆæİěåijžåĹűåIJĺ `__init__()` æŰźæşŢäÿ■æŇĞåőŽåŘĆæŢřçŽĎçśżåđŃç■¿åŘ■ãĂĆ

10.12 8.12 åőŽäźĽæŐěåŘčæĹŰèĂĚæŁ¡èśąå§žçśż

éŮőécŸ

ä¡ăæČşåőŽäźĽäÿĂäÿłæŐěåŘčæĹŰæŁ¡èśąçśżïijŇåźűäÿŤéĂŽè£ĞæĽğèąŇçśżåđŃæčĂæ§ěæİěçąőä£İå■Ř

èğčåĘşæŰźæąĹ

ä¡£çŤĺ `abc` æĺąåİŮåŘŕäžěå¿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¡èśąçśżçŽĎäÿĂäÿłçL'źçĆźæŸŕåőČäÿ■èČ¡çŻť'æŐěèćńåőđä¿ŃåŇŰïijŇæŕŤåęĆä¡ăæČşåČŔäÿŃéİćè£Ź

```
a = IStream() # TypeError: Can't instantiate abstract class
              # IStream with abstract methods read, write
```

æŁ¡èśąçśżçŽĎçŻőçŽĎåřśæŸŕèől'åĹńçŽĎçśżçżğæL'£åőČåźűåőđçŐřçL'źåőŽçŽĎæŁ¡èśąæŰźæşŢïijŽ

```
class SocketStream(IStream):
    def read(self, maxbytes=-1):
        pass

    def write(self, data):
        pass
```

æŁ¡èśąå§žçśżçŽĎäÿĂäÿłäÿżèęĄçŤĺéĂŤæŸŕåIJĺäžčçăĄäÿ■æčĂæ§ěæ§ŘäžŽçśżæŸŕåŘęäÿžçL'źåőŽçśżåđ

```
def serialize(obj, stream):
    if not isinstance(stream, IStream):
        raise TypeError('Expected an IStream')
    pass
```

éŹđ'äžEçżğæL'£è£Źçğ■æŰźåijŔåđ'ŰïijŇè£ŸåŔŕäžěéĂŽè£ĞæşĺåĘŇæŰźåijŔæİěèől'æ§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¡æşÍèğčéİŹæĂĄæŰźæşŢãĂĄçżæŰźæşŢåŠŇ properties ãĂĆ ä¡ääŔłéIJĂä£İèŕĄè£Źäÿłæşĺèğčçť ğéİääIJĺåĞ¡æŢřåŐŽäźĽ'åĽ■å■şåŔŕïijŽ

```
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

æăĞåĞĘäžŞäÿ■æIJL'å¿Ĺåď'ŽçŤĺåĹřæŁ¡èśąå§žçśżçŽĎåIJřæŰźãĂĆcollections æĺąåİŮåőŽäźĽ'äžĘå¿Ĺåď'Žèů§åőźåŹĺåŠŇè£■äžčåŹĺ(åžŔåĹŮãĂĄæŸäářĎãĂĄéŻĘåŘĹç■L')æIJL'åĚşçŽĎæŁ numbers äžŞåőŽäźĽ'äžĘèů§æŢřå■Ůåŕźèśą(æŢ'æŢřãĂĄæţőçĆźæŢřãĂĄæIJL'çŘĘæŢřç■L')æIJL'åĚşçŽĎå§ äžŞåőŽäźĽ'äžĘå¿Ĺåď'Žèů§I/OæŞ■ä¡IJçŻÿåĚşçŽĎå§žçśżãĂĆ

ä¡ääŔŕäžěä¡£çŤĺécĎåőŽäźĽ'çŽĎæŁ¡èśąçśżæİěæĽ'ğèąŇæŻť'éĂŽçŤĺçŽĎçśżåđŃæčĂæ§ěïijŇä¿ŃåęĆïijŽ

```
import collections

# Check if x is a sequence
if isinstance(x, collections.Sequence):
...

# Check if x is iterable
```

(continues on next 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):
```

åř¡çőąABCsåŔŕäžěèőľ'æĹŚäžňå¿ĹæŰźä¿£çŽĎåĄŽçśžåđŃæčĂæ§ěïijŇä¡ĘæŸŕæĹŚäžňåIJĺäžčçăĄäý■æĹ
åŻăäýžPythonçŽĎæIJňèť'ĺæŸŕäýĂéŮĺåŁĺæĂĄçijŰçĺŃèŕ■èĺĂïijŇåĚűçŽőçŽĎåřśæŸŕçżŹä¡ăæŽť åď'ŽçĄţæť'żą
åijžåĹűçśžåđŃæčĂæ§ěæĹŰèőľä¡ääžčçăĄåŔŸå¿ŮæŽť'åď'■æİĆïijŇè£ŹæăůåĄŽæŮăåijĆäžŐèĹ■æIJňæśĆæIJ

10.13 8.13 åőđçŐřæŢřæ■őæĺąåđŃçŽĎçśžåđŃçžęæİ§

éŮőécŸ

ä¡ăæČşåőŽäźL'æ§ŘäžŽåIJĺåśđæĂğèţŃåĂijäýŁéİćæIJL'éŹŘåĹűçŽĎæŢřæ■őçżŞæđĎãĂĆ

èğčåĘşæŰźæąĹ

åIJĺè£ŹäýłéŮőécŸäý■ïijŇä¡äéIJĂèęĄåIJĺåŕźæ§ŘäžŽåőđä¿ŃåśđæĂğèţŃåĂijæŮűè£ŽèąŇæčĂæ§ěãĂĆ
æĽĂäžěä¡äèęĄèĞłåőŽäźL'åśđæĂğèţŃåĂijåĞ¡æŢřïijŇè£Źçğ■æČĚåĘţäýŃæIJĂåě¡ä¡£çŤĺæŔŔè£řåŹĺãĂĆ

äýŃéİćçŽĎäžčçăĄä¡£çŤĺæŔŔè£řåŹĺåőđçŐřäžĘäýĂäýłçşżçż§çśžåđŃåŠŇèţŃåĂijéłŇèŕĄæąĘæđűïijŽ

```
# 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)

```
# 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£ŹäžŻçśżåřśæŸŕäą¡äèęĄåĹŻåżžçŽĎæŢřæ■őæĺąåđŃæĹŰçśżåđŃçşżçż§çŽĎå§žçąĂæđĎåżžæĺąåİŮãĂĆäÿŃéİćåřśæŸŕæĹŚäžňåőđéŹĚåőŽäźĹ’çŽĎåŘĎçğ■äÿ■åŘŃçŽĎæŢřæ■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
```

çĎűåŘŐä¡£çŤĺè£ŹäžŻèĞłåőŽäźĹ’æŢřæ■őçśżåđŃïijŇæĹŚäžňåőŽäźĹ’äÿĂäÿłçśżïijŽ

```
class Stock:
    # Specify constraints
    name = SizedString('name', size=8)
    shares = UnsignedInteger('shares')
    price = UnsignedFloat('price')
```

(continues on next page)

```
    def __init__(self, name, shares, price):
        self.name = name
        self.shares = shares
        self.price = price
```

çĎűåŘŐæţŃèŕŢè£Źäÿłçśżç ŽĎåśđæĂğèţŃåĂijçžęæİ§ïijŇåŔŕåŘŞçŐřåŕźæ§ŘäžŽåśđæĂğçŽĎèţŃåĂijè£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'äÿĂäžŽæŁĂæIJŕåŔŕäžěçőĂåŇŰäÿŁéİćçŽĎäžčçăĄïijŇåĚűäÿ■äÿĂçğ■æŸŕä¡£çŤĺçśżèčĚéěřåŹĺ

```
# 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)

```
# 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
```

åŔęåď Ű äÿ Ă ç ğ ■ æ Ű ź å ij Ŕ æ Ÿ ŕ ä ¡ £ ç Ť Í å Ě Č ç ś ż ï ij Ž

```
# 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
```

èőÍèőž

æIJňèŁĆä¡£çŤÍäžĘå¿Ĺåď Žéń Ÿ çžğæŁĂæIJŕïijŇåŇĚæŃňæŔŔè£řåŹÍãĂĄæůůåĚěçśżãĂĄsuper() çŽĎä¡£çŤÍãĂĄçśżèčĚéěřåŹÍåŠŇåĚČçśżãĂĆ äÿ■åŔŕèČ¡åIJÍè£ŹéĞŇäÿĂäÿĂèŕęçżĘåśŢåijÄæİěèőšïijŇä¡ĘæŸä¡ĘæŸŕïijŇæĹŠåIJÍè£ŹéĞŇè£ŸæŸŕèęĄæŔŘäÿĂäÿŃåĞăäÿłéIJĂèęĄæşÍæĎŔçŽĎçĆźãĂĆ

éęŰåĚĹïijŇåIJÍ Descriptor å§žçśżäÿ■ä¡ăäijŽçIJŃåĹřæIJL'äÿł __set__() æŰźæşŢïijŇå■ť'æšąæIJL'çŻÿåžŤçŽĎ __get__() æŰźæşŢãĂĆ åęĆæđIJäÿĂäÿłæŔŔè£řäżĚäżĚæŸŕäżŐåžŢåśĆåőđä¿Ńå■ŮåĚÿäÿ■èŐůåŘŰæ§Řäÿłåśđæ Ă ğå Ă ijçŽĎèŕÍïijŇéĆ __get__() æŰźæşŢãĂĆ

æL'ĂæIJL'æŔŔè£řåŹÍçśżéČ¡æŸŕå§žäžŐæůůåĚěçśżæİèåőđçŐřçŽĎãĂĆæŕŤåęĆ Unsigned åŠŇ MaxSized èęĄèů§åĚűäżŰçżğæL'£èĞł Typed çśżæůůåĚěãĂĆ è£ŹéĞŇåĹl'çŤÍåď Žçżğæ L'£æİèåőđçŐřçŻÿåžŤçŽĎåŁ§èČ¡ãĂĆ

æůůåĚěçśżçŽĎäÿĂäÿłæŕŤè¿ČéŽ¿çŘĘèğčçŽĎåIJřæŰźæŸŕïijŇèřČçŤÍ

super() åĞ¡æŢřæŮűïijŇä¡ăåźűäÿ■ç§ěéĄŞçľűçń§èęĄèřČçŤĺåŞłäÿłåĚůä¡Şçśžã ĂĆ
ä¡ăéIJĂèęĄèů§åĚűäżŰçśžçżŞåŘĹåŘŐæĽ■èČ¡æ■čçąőçŽĎä¡£çŤĺïijŇäź§åřśæŸŕå£Ěéążå ŘĹä¡IJæĽ■èČ¡äžğçŤ

ä¡£çŤĺçśžèčĚéěřåŹĺåŠŇåĚČçśžéĂŽåÿÿåŔŕäžěçőĂåŇŰäżčçăĄãĂĆäÿŁéİćäÿďäÿłä¿Ńå■Řäÿ■ä¡ääijŽåŔŚ

```
# Normal
class Point:
    x = Integer('x')
    y = Integer('y')

# Metaclass
class Point(metaclass=checkedmeta):
    x = Integer()
    y = Integer()
```

æĽĂæIJL'æŰźæşŢäÿ■ïijŇçśžèčĚéěřåŹĺæŰźæąĹåžŤèřěæŸŕæIJĂçĄţæť'żåŠŇæIJĂéńŸæŸŐçŽĎãĂĆ éęŰåĚĹïijŇåőČåźűäÿ■ä¿İèţŰäżżä¡ŢåĚűäżŰæŰřçŽĎæŁĂæIJŕïijŇæŕŤåęĆåĚČçśžãĂĆåĚűæňąïijŇèčĚéěřåŹĺå

æIJĂåŘŐïijŇèčĚéěřåŹĺè£ŸèČ¡ä¡IJäÿžæůůåĚěçśžçŽĎæŻ£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)

```
    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
```

èő¿ç¡őäÿĂäÿłçőĂå■ŢçŽĎçśżåđŃåśđæĂğçŽĎåĂijïijŇèčĚéěřåŹĺæŰźåijŔèęĄæŕŤäźŃåL’■çŽĎæůůåĚěçśżçŽĎ
çŐřåIJĺä¡ăåžŤèŕěåžĘåźÿèĞłåůśèŕżåőŇäžĘæIJňèŁĆåĚĺéČĺåĘĚåőźäžĘåŘğïij§^_^

10.14 8.14 åőđçŐřèĞłåőŽäźL’åőźåŹĺ

éŮőécŸ

ä¡ăæČşåőđçŐřäÿĂäÿłèĞłåőŽäźL’çŽĎçśżæİěæĺąæŃ§åĘĚç¡őçŽĎåőźåŹĺçśżåŁ§èČ¡ïijŇærŤåęĆåĹŮèąĺåŠ

èğčåĘşæŰźæąĹ

collections åőŽäźL’äžĘå¿Ĺåd’ŽæŁ¡èśąå§žçśżïijŇå¡Şä¡ăæČşèĞłåőŽäźL’åőźåŹĺçśżçŽĎæŮűåĂŹåőČ
ærŤåęĆä¡ăæČşèől’ä¡ăçŽĎçśżæŤŕæŇĄè£■äžčïijŇéĆčåřśèől’ä¡ăçŽĎçśżçžğæL’£
collections.Iterable å■şåŔřïijŽ

```
import collections
class A(collections.Iterable):
    pass
```

äÿ■è£Ğä¡ăéIJĂèęĄåőđçŐř collections.Iterable
æL’ĂæIJL’çŽĎæŁ¡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¡ăåŘłèęĄåőđçŐř __iter__() æŰźæşŢåřśäÿ■äijŽæŁěéŤŹäžĘ(åŔĆèĂČ4.2åŠŇ4.7åřŔèŁĆ)ãĂĆ

ä¡ăåŔŕäžěåĚĹèŕŢçİĂåŐžåőđä¿ŃåŇŰäÿĂäÿłåŕźèśąïijŇåIJĺéŤŹèŕŕæŔŘçď’žäÿ■åŔŕäžěæL’¿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__
>>>
```

äÿŃéIćæŸŕäÿĂäÿłçőĂå■ŢçŽĎçď’žä¿ŃïijŇçżğæL’£èĞłäÿŁéIćSequenceæŁ¡èśąçśżïijŇåźűäÿŤåőđçŐřåĚČ

```
class SortedItems(collections.Sequence):
    def __init__(self, initial=None):
        self._items = sorted(initial) if initial is not None else []
```

(continues on next 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 æăůïijŇæŤŕæŇĄæĽ ĂæIJĽåÿÿç

è£ŹéĞŇéÍćä¡£çŤĺåĹřäžĘ `bisect` æÍąåİŮïijŇåőČæŸŕäÿĂäÿłåIJÍæŐŠåžŔåĹŮèąÍäÿ■æŔŠåĚěåĚČçť ăçŽ

èőÍèőž

ä¡£çŤÍ `collections` äÿ■çŽĎæŁ¡èśąå§žçśżåŔŕäžěçąőä£Íä¡äèĞłåőŽäźĽçŽĎåőźåŹĺåőđçŐřäžĘæĽĂæI
ä¡ăçŽĎèĞłåőŽäźĽåőźåŹĺäijŽæżąèűşåď ğéČĺåĹĘçśżåđŇæčĂæ§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¡èśąçśżäijŽäÿžäÿĂäžŽåÿÿèğĄåőźåŹĺæŞ■ä¡IJæŔŘä¿ŻéżŸèőď çŽĎåőđçČ
è£ŹæăůäÿĂæÍěä¡ăåŔłéIJĂèęĄåőđçŐřéĆčäžŽä¡ăæIJĂæĎ§åĚť èűčçŽĎæŰźæşŢå■şåŔŕãĂĆåĄĞèő¿ä¡ăçŽĎçśż
`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)

```
    # 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¡ăäijŽåŔŚçŐřåőČæŤŕæŇĄåĞäăźŐæĽ'ĂæIJL'çŽĎæäýå£Čå
äýŃéİćæŸŕä¡£çŤĺæijŤçđ'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
>>>
```

æIJňåřŔèŁĆåŔłæŸŕåŕźPythonæŁ¡èśąçśżåŁ§èČ¡çŽĎæŁŻçăŰåijŢçŐĽ'ãĂĆnumbers æĺąåİŮæŔŘä¿ŻäžĘäÿĂäÿłçśżäijijçŽĎèů§æŢt'æŢřçśżåđŃçŻÿåĚşçŽĎæŁ¡èśąçśżåđŃéŻĘåŘĹãĂĆ åŔŕäžěåŔĆèĂČ8.12åřŔèŁĆæİěæđĎéĂäæŻt'åd'ŽèĞłåőŽäźĽ'æŁ¡èśąå§žçśżãĂĆ

10.15 8.15 åśđæĂğçŽĎäżčçŘĘèő£éŮő

éŮőécŸ

ä¡ăæČşåřĘæ§Řäÿłåőđä¿ŃçŽĎåśđæĂğèő£éŮőäżčçŘĘåĹřåĘĚéČĺåŔęäÿĂäÿłåőđä¿Ńäÿ■åŐżïijŇçŻőçŽĎå

èğčåĘşæŰźæąĹ

çőĂå■Ţæİěèŕť'ïijŇäżčçŘĘæŸŕäÿĂçğ■çijŰçĺŃæĺąåijŔïijŇåőČåřĘæ§Řäÿłæş■ä¡IJè¡ňçğżçżŹåŔęåd'ŰäÿĂ æIJĂçőĂå■ŢçŽĎå¡ćåijŔåŔŕèČ¡æŸŕåČŔäÿŃéİćè£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
```

åęĆæđIJäżĚäžĚåřśäÿd'äÿłæŰźæşŢéIJĂèęĄäżčçŘĘïijŇéĆčäźĹåČŔè£Źæăůåę̆Źåřśèűşåd'§äžĘãĂĆä¡ĘæŸ éĆčäźĹä¡£çŤĺ `__getattr__()` æŰźæşŢæĹŰèőÿæĹŰæŻt'åě¡äžŻïijž

```
class B2:
    """ä¡£çŤĺ__getattr__çŽĎäżčçŘĘïijŇäżčçŘĘæŰźæşŢærŤè¿Čåd'ŽæŮűåĂŹ"""

    def __init__(self):
```

(continues on next 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¡£çŤĺæijŢçd'ži

```
b = B()
b.bar() # Calls B.bar() (exists on B)
b.spam(42) # Calls B.__getattr__('spam') and delegates to A.spam
```

åŔęåd'ŰäÿĂäÿłäżčçŘĘä¿Ńå■ŘæŸŕåőđçŐřäżčçŘĘæĺąåijŔïijŇä¿ŃåęĆïijŽ

```
# 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¡ăåŔłéIJĂèęĄçŤĺåőČæİěåŇĚèčĔäÿŃåĔűäżŰçśżå■şåŔŕïijŽ

```
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£ĞèĞłåőŽäźL’åśđæĂğèő£éŮőæŰźæşŢïijŇä¡ääŔŕäżěçŤĺäÿ■åŘŇæŰźåijŔèĞłåőŽäźL’äżčçŘĘçśżèąŇ

èőÍèőž

äżčçŘĘçśżæIJL’æŮűåĂŹåŔŕäżěä¡IJäÿžçżğæL’£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¡£çŤĺäżčçŘĘçŽĎèŕİïijŇåřśæŸŕäÿŃéİćè£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)

```
    def bar(self):
        print('B.bar')
    def __getattr__(self, name):
        return getattr(self._a, name)
```

å¡Şåőđç Őřäžčç ŘĘæĺąåijŔæŮűïijŇè£ŸæIJL'äžŽçżĘèŁĆéIJĂèęĄæşĺæĎŔãĂĆ éęŰåĚĹïijŇ__getattr__() åőđéŹĚæŸŕäÿĂäÿłåŘŐåď'ĞæŰźæşŢïijŇåŔłæIJL'åIJĺåśđæĂğäÿ■å■ŸåIJĺæŮú åŽäæ■ď'ïijŇåęĆæđIJäžčçŘĘçśżåőđä¿ŃæIJňèžńæIJL'è£ŹäÿłåśđæĂğçŽĎèŕĪïijŇéĆčäźĹäÿ■äijŽèğęåŔŚè£Źäÿłą åŔęåď'ŰïijŇ__setattr__() åŠŇ __delattr__() éIJĂèęĄéćİåď'ŰçŽĎé■ŤæşŢæİěåŇžåĹĘäžčçŘĘåőđ _obj çŽĎåśđæĂğãĂĆ äÿĂäÿłéĂŽåÿÿçŽĎçžęåőŽæŸŕåŔłäžčçŘĘéĆčäžŽäÿ■äžěäÿŃåĹŠçž£ _ åijĂåď't'çŽĎåśđæĂğ(äžčçŘĘçśżåŔłæŽt'éIJšèćńäžčçŘĘçśżçŽĎåĚňåĚšåśđæĂğ)ãĂĆ

è£ŸæIJL'äÿĂçĆźéIJĂèęĄæşĺæĎŔçŽĎæŸŕïijŇ__getattr__() åŕźäžŐåď'ğéČĺåĹĘäžěåŔŇäÿŃåĹŠçž£(__)åijĂåğŃåŠŇçżŞår¿çŽĎåśđæĂğåźűäÿ■éĂĆçŤĺãĂĆ æŕŤåęĆïijŇèĂČèŹŚåęĆäÿŃçŽĎçśžïijŽ

```
class ListLike:
    """__getattr__
↪åŕźäžŐåŔŇäÿŃåĹŠçž£åijĂåğŃåŠŇçżŞår¿çŽĎæŰźæşŢæŸŕäÿ■èČ¡çŤĺçŽĎïijŇéIJĂèęĄäÿĂäÿłäÿł
↪"""

    def __init__(self):
        self._items = []

    def __getattr__(self, name):
        return getattr(self._items, name)
```

åęĆæđIJæŸŕåĹŽåžžäÿĂäÿłListLikeåŕźèśąïijŇäijŽåŔŚçŐřåőČæŤŕæŇĄæŹőéĂŽçŽĎåĹŮèąĺæŰźæşŢïijŇ ä¡ĘæŸŕå■t'äÿ■æŤŕæŇĄlen()ãĂĄåĚČçt'äæ§ěæL'¿ç■L'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
>>>
```

äÿžäžĘèől'åőČæŤŕæŇĄè£ŹäžŽæŰźæşŢïijŇä¡åå£ĚéąžæL'ŃåŁĺçŽĎåőđçŐřè£ŹäžŽæŰźæşŢäžčçŘĘïijŽ

```
class ListLike:
    """__getattr__
↪åŕźäžŐåŔŇäÿŃåĹŠçž£åijĂåğŃåŠŇçżŞår¿çŽĎæŰźæşŢæŸŕäÿ■èČ¡çŤĺçŽĎïijŇéIJĂèęĄäÿĂäÿłäÿł
↪"""
```

(continues on next 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]
```

11.8åřŔèŁĆè£ŸæIJL'äÿĂäÿłåIJÍè£IJçÍŃæŰźæşŢèřČçŤÍçŐřåćČäÿ■ä¡£çŤÍäżčçŘĘçŽĎä¿Ńå■ŘãĂĆ

10.16 8.16 åIJÍçśżäÿ■åőŽäźĽåď̃ŽäÿłæđĎéĂäåŹÍ

éŮőécŸ

ä¡ăæČşåőđçŐřäÿĂäÿłçśżïijŇéŹď̃äžĘä¡£çŤÍ `__init__()` æŰźæşŢåď̃ŰïijŇè£ŸæIJL'åĚűäżŰæŰźåijŔåŔřäżěåĹİåğŃåŇŰåőČãĂĆ

èğčåĘşæŰźæąĹ

äÿžäžĘåőđçŐřåď̃Žäÿłæđ̃ĎéĂäåŹÍïijŇä¡ăéIJĂèęĄä¡£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)

```
        t = time.localtime()
        return cls(t.tm_year, t.tm_mon, t.tm_mday)
```

çŻťæŐěèřČçŤĺçśżæŰźæşŢå■şåŔŕïijŇäýŃéİćæŸŕä¡£çŤĺçd'žä¿ŃïijŽ

```
a = Date(2012, 12, 21) # Primary
b = Date.today() # Alternate
```

èőÍèőž

çśżæŰźæşŢçŽĎäýĂäýłäýżèęĄçŤĺéĂŤåřśæŸŕåőŽäźL'åd'ŽäýłæđĎéĂäåŹĺãĂĆåőČæŐěåŔŮäýĂäýł `class` ä¡IJäýžçňňäýĂäýłåŔĆæŢř(cls)ãĂĆ ä¡ăåžŤèřěæşĺæĎŔåĹřäžĘè£ŹäýłçśżèćńçŤĺæİěåĹŻåżžåźűè£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 åĹŻåżžäy■èřČçŤĺnitæŰźæşŢçŽĎåőđä¿Ń

éŮőécŸ

ä¡ăæČşåĹŻåżžäýĂäýłåőđä¿ŃïijŇä¡ĘæŸŕåýŇæIJŻçżŢè£ĞæL'ğèąŇ `__init__()` æŰźæşŢãĂĆ

èğčåĘşæŰźæąĹ

åŔŕäżěéĂŽè£Ğ `__new__()` æŰźæşŢåĹŻåżžäýĂäýłæIJłåĹİåğŃåŇŰçŽĎåőđä¿ŃãĂĆä¿ŃåęĆèĂČèŹŚåę

```
class Date:
    def __init__(self, year, month, day):
        self.year = year
        self.month = month
        self.day = day
```

äýŃéİćæijŤçd'žåęĆä¡Ţäy■èřČçŤĺ `__init__()` æŰźæşŢæİěåĹŻåżžè£ŹäýłDateåőđä¿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)

```
AttributeError: 'Date' object has no attribute 'year'
>>>
```

çżŞædIJåŔŕäžěçIJŃåĹřïijŇè£ŹäÿłDateåőđä¿ŃçŽĎåśđæĂğyearè£Ÿäÿ■å■ŸåIJĺïijŇæĽĂäżěä¡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¡ŞæĹŚäžňåIJĺåŔ■åžŔåĹŮåŕźèśąæĹŰèĂĚåőđçŐřæ§ŘäÿłçśżæŰźæşŢædĎéĂăåĞ¡æŢřæŮűéIJĂèęĄçżŢè
`__init__()` æŰźæşŢæİěåĹŻåżžåŕźèśąãĂĆ 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¡ăåŔ■åžŔåĹŮåŇŰJSONæŢřæ■őæŮűäžğçŤ§äÿĂäÿłåęĆäÿŃçŽĎå■ŮåĚÿåŕźèśąïijŽ

```
data = { 'year': 2012, 'month': 8, 'day': 29 }
```

åęĆædIJä¡ăæČşåřĘåőČè¡ñæ■ćæĹŘäÿĂäÿłDateçśżådŇåőđä¿ŃïijŇåŔŕäžěä¡£çŤĺäÿŁéİćçŽĎæŁĂæIJŕãĂĆ

å¡Şä¡ăéĂŽè£Ğè£Źçğ■éİđåÿÿèğĎæŰźåijŔæİěåĹŻåżžåőđä¿ŃçŽĎæŮűåĂŹïijŇæIJĂåě¡äÿ■èęĄçŻť'æŐěåĆ
åŘęåĹŹçŽĎèŕİïijŇåęĆædIJè£Źäÿłçśżä¡£çŤĺäžĘ `__slots__` ãĂĄproperties ãĂĄdescriptors æĹŰåĚűäžŰéńŸçžğæŁĂæIJŕçŽĎæŮűåĂŹäżčçăĄåřśäijŽåď śæŢĹãĂĆ
èĂŇè£ŹæŮűåĂŹä¡£çŤĺ `setattr()` æŰźæşŢäijŽèőľä¡ăçŽĎäżčçăĄåŔŸå¿ŮæŻť'åŁăéĂŽçŤĺãĂĆ

10.18 8.18 åĹľçŤÍMixinsæĽľåśŢçśżåŁ§èČ¡

éŮőécŸ

ä¡ăæIJL'å¿Ĺåd'ŽæIJL'çŤÍçŽĎæŰźæşŢïijŇæČşä¡£çŤÍåőČäžňæİěæL'l'åśŢåĚűäżŰçśżçŽĎåŁ§èČ¡ãĂCä¡Ęa
åŻăæ■d'ä¡ăäÿ■èČ¡çőĂå■ŢçŽĎåřĘè£ŹäžŻæŰźæşŢæŤ¿åĚëäÿĂäÿłå§žçśżïijŇçĎűåŘŐèćńåĚűäżŰçśżçżğæL'£ã

èğčåĘşæŰźæąĹ

éĂŽåÿÿå¡Şä¡ăæČşèĞłåőŽäźL'çśżçŽĎæŮűåĂŹäijŽçćřäÿŁè£ŹäžŻéŮőécŸãĂCåŘřèČ¡æŸŕæ§Řäÿłåž§æŔ
ä¡ăåŘřäžěåĹľçŤÍåőČäžňæİěædĎéĂää¡äèĞłåůşçŽĎçśżãĂĆ

åĄĞèő¿ä¡ăæČşæL'l'åśŢæŸăårĎårźèśąïijŇçżŹåőČäžňæůżåŁăæŮěå£ŮãĂĄåŤřäÿĂæĂğèő¿ç¡őãĂĄçśżådł

```
class LoggedMappingMixin:
    """
    Add logging to get/set/delete operations for debugging.
    """
    __slots__ = ()  #␣
↪ăůůåĚěçśżéČ¡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)

```
    '''
    __slots__ = ()

    def __setitem__(self, key, value):
        if not isinstance(key, str):
            raise TypeError('keys must be strings')
        return super().__setitem__(key, value)
```

è£ŹäžŻçśżå■ŢçŃñä¡£çŤÍèţůæİěæšąæIJL’äżżä¡ŢæĎŔäźL’ïijŇäžŇåőđäÿŁåęĆæđIJä¡ăåŐżåőđä¿ŃåŇŰäżżäåőČäžňæŸŕçŤÍæİěéĂŽè£ĞåđŽçżğæL’£æİěåŠŇåĚűäžŰæŸăářĎářźèśąæůůåĚëä¡£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¿Ńå■Řäÿ■ïijŇåŔŕäżěçIJŃåĹřæůůåĚěçśżèů§åĚűäžŰåůšå■ŸåIJÍçŽĎçśż(æŕŤåęĆdictãĂĄdefaultd
çżŞåŘĹåŘŐåŕśèČ¡åŔŚæŇěæ■čåÿÿåŁ§æŢĹäžĘãĂĆ

èőÍèőž

æůůåĚěçśżåIJÍæăĞåĞĘåžŞäÿ■å¿Ĺåď’ŽåIJřæŰźéČ¡åĞžçŐřè£ĞïijŇéĂŽåÿÿéČ¡æŸŕçŤÍæİěåČŔäÿŁéİćéĆč
åőČäžňäźŞæŸŕåď’Žçżğæĺ’£çŽĎäÿĂäÿłäÿżèęĄçŤÍéĂŤãĂĆæŕŤåęĆïijŇå¡Şä¡äçijŰåĘŹç¦ŚçžIJäżčçăĄæŮűåĂŹ
ä¡ăäijŽçżŔåÿÿä¡£çŤÍ socketserver æĺąåİŮäÿ■çŽĎ ThreadingMixIn
æİěçżŹåĚűäžŰç¡ŚçżIJçŻÿåĚşçśżåćđåŁăåď’Žçž£çÍŇæŤŕæŇĄãĂĆ
ä¿ŃåęĆïijŇäÿŇéİćæŸŕäÿĂäÿłåď’Žçž£çÍŇçŽĎXML-RPCæIJ■åŁąïijŽ

```
from xmlrpc.server import SimpleXMLRPCServer
from socketserver import ThreadingMixIn
class ThreadedXMLRPCServer(ThreadingMixIn, SimpleXMLRPCServer):
    pass
```

åŘŇæŮűåIJÍäÿĂäžŻåď’ğåđŃåžŞåŠŇæąĘæđűäÿ■äź§äijŽåŔŚçŐřæůůåĚěçśżçŽĎä¡£çŤÍïijŇçŤÍéĂŤåŘŇæ

åŕźäžŐæůůåĚěçśżïijŇæIJL’åĞăçĆźéIJĂèęĄèőřä¡ŔãĂĆéęŰåĚĹæŸŕïijŇæůůåĚěçśżäÿ■èČ¡çŻt’æŐèèćńåő

åËűæňąïijŇæůůåËěçśżæšąæIJL'èĞłåůśçŽĎçŁűæĂĄä£ąæĄřïijŇäź§åřśæŸŕèŕť'åőČäżňåźűæšąæIJL'åőŽäźL'
__init__() æŰźæşŢïijŇåźűäÿŤæšąæIJL'åőđä¿ŃåšđæĂğãĂĆ
è£Źäź§æŸŕäÿžäżĂäźĹæĹŚäżňåIJĺäÿŁéİćæŸŐçąőåőŽäźL'äžĘ __slots__ = () ãĂĆ

è£ŸæIJL'äÿĂçğ■åőđçŐřæůůåËěçśżçŽĎæŰźåijŔåřśæŸŕä¡£çŤĺçśżèčËéěřåŹĺïijŇåęĆäÿŃæL'Ăçd'žïijŽ

```
def LoggedMapping(cls):
    """çňňäžŇçğ■æŰźåijŔïijžä¡£çŤĺçśżèčËéěřåŹĺ"""
    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■åE■éIJĂèęĄä¡£çŤĺåď'ŽçżğæL'£äžEąĂĆ
åŔĆèĂČ8.13åřŔèŁĆæ§ěçIJŃæŻť'åď'ŽæůůåËěçśżåŠŇçśżèčËéěřåŹĺçŽĎä¿Ńå■ŘãĂĆ

10.19 8.19 åőđçŐřçŁűæĂĄåŕźèśąæĹŰèĂËçŁűæĂĄælJž

éŮőécŸ

ä¡ăæČşåőđçŐřäÿĂäÿłçŁűæĂĄæIJžæĹŰèĂËæŸŕåIJĺäÿ■åŘŇçŁűæĂĄäÿŃæL'ğèąŇæŞ■ä¡IJçŽĎåŕźèśąïijl

èğčåĘşæŰźæąĹ

åIJĺå¿Ĺåď'ŽçĺŃåžŔäÿ■ïijŇæIJL'äžŻåŕźèśąäijŽæăźæ■őçŁűæĂĄçŽĎäÿ■åŘŇæİěæL'ğèąŇäÿ■åŘŇçŽĎæŞ

```
class Connection:
    """æŹőéĂŽæŰźæąĹïijŇåě¡åd’ŽäÿłåĹd’æŰ■èŕ■åŔěïijŇæŢĹçŐĞä¡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¿Ĺåd’ŽçijžçĆźïijŇéęŰåĚĹæŸŕäžčçăĄåd’łåd’■æİĆäžĘïijŇåě¡åd’ŽçŽĎæİąäżűåĹd’æŰ■åŻääÿžäÿĂäžŻåÿÿèğĄçŽĎæŞ■ä¡IJærŤåęĆread()ãĂĄwrite()æŕŔæňąæL’ğèąŇåL’■éČ¡éIJĂèęĄæL’ğèąŇæčĂæş

äÿĂäÿłæŻt’åě¡çŽĎåŁđæşŢæŸŕäÿžæŕŔäÿłçŁűæĂĄåőŽäźL’äÿĂäÿłåŕźèśąïijŽ

```
class Connection1:
    """æŰřæŰźæąĹâĂŤâĂŤåŕźæŕŔäÿłçŁűæĂĄåőŽäźL’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)

```
    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)

```
        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

åęĆædIJäžčçăĄäÿ■åĞžçŐřåď'łåď'ŽçŽĎæİąäżűåĹď'æŰ■èŕ■åŔěçŽĎèŕİïijŇäžčçăĄåřšäijŽåŔŸå¿ŮéŽ¿äžě
è£ŹéĞŇçŽĎèğčåĘşæŰźæąĹæŸřåřĘæŕŔäÿłçŁűæĂĄæŁ¡åŔŰåĞžæİěåőŽäźĽ'æĹŘäÿĂäÿłçśžãĂĆ

è£ŹéĞŇçIJŃäÿŁåŐżæIJL'çĆźåěĞæĂłïijŇæŕŔäÿłçŁűæĂĄåŕźèśąéČ¡åŔłæIJL'éİŹæĂĄæŰźæşŢïijŇåźűæš
åőđéŹĚäÿŁïijŇæL'ĂæIJL'çŁűæĂĄä£ąæĄŕéČ¡åŔłå■ŸåĆĺåIJĺ `Connection`
åőđä¿Ńäÿ■ãĂĆ åIJĺå§žçśżäÿ■åőŽäźĽ'çŽĎ `NotImplementedError`
æŸŕäÿžäžĘçąőä£Ĺå■ŘçśżåőđçŐřäžĘçŻÿåžŢçŽĎæŰźæşŢãĂĆ è£ŹéĞŇä¡äæĹŰèőÿè£ŸæČşä¡£çŤĺ8.12åřŔèŁĆ

èő¿èőąæĺąåijŔäÿ■æIJL'äÿĂçğ■æĺąåijŔåŔńçŁűæĂĄæĺąåijŔïijŇè£ŹäÿĂåřŔèŁĆçőŮæŸŕäÿĂäÿłåĹİæ■ěåŀ

10.20 8.20 éĂŽè£Ğå■ŮçňęäÿšèřČçŤĺářźèśąæŰźæşŢ

éŮőécŸ

ä¡ăæIJL'äÿĂäÿłå■Ůçňęäÿšå¡ćåijŔçŽĎæŰźæşŢåŘ■çğřïijŇæČşéĂŽè£ĞåőČèřČçŤĺæ§Řäÿłářźèśąç ŽĎářźå

èğčåĘşæŰźæąĹ

æIJĂçőĂå■ŢçŽĎæČĚåĘţïijŇåŔŕäżěä¡£çŤĺ `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 Űäÿ Ăçğ■æŰźæşŢæŸŕä¡£çŤĺ `operator.methodcaller()` ïijŇä¿ŃåęĆïijŽ

```
import operator
operator.methodcaller('distance', 0, 0)(p)
```

å¡Şä¡ăéIJĂèęĄéĂŽè£ĞçŻÿåŘŇçŽĎåŔĆæŢřåď ŽæňąèřČçŤĺæ§Řäÿłæ ŰźæşŢæŮűïijŇä¡£çŤĺ `operator.methodcaller` åřśå¿ĹæŰźä¿£äžĘãĂĆ æŕŤåęĆä¡ăéIJĂèęĄæŐŠåžŔäÿĂçşżåĹŮçŽĎçĆźïijŇåř

```
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§ěæL'¿åšđæĂğïijŇçåŻăæ■d'ïijŇäÿžäžĘèřČçŤÍæ§ŘäÿłæŰźæşŢïijŇä¡ăåŘŕäżěéęŰåĚĹéĂŽè£Ğ `getattr()` æÍěæ§ěæL'¿åĹřè£ŹäÿłåšđæĂğïijŇçĎűåŘŐåĘ■åŐżäžěåĞ¡æŢřæŰźåijŔèřČçŤĺåőČå■şåŘŕãĂĆ

`operator.methodcaller()` åĹŻåžžäÿĂäÿłåŘŕèřČçŤĺåŕźèśąïijŇåźűåŘŇæŮűæŔŘä¿ŻæL'ĂæIJL'å£çĎűåŘŐèřČçŤÍçŽĎæŮűåĂŹåŘłéIJĂèęĄåřĘåőđä¿ŃåŕźèśąäijăéĂŠçžŹåőČå■şåŘŕïijŇæŕŤåęĆïijŽ

```
>>> p = Point(3, 4)
>>> d = operator.methodcaller('distance', 0, 0)
>>> d(p)
5.0
>>>
```

éĂŽè£ĞæŰźæşŢåŘ■çğřå■ŮçňęäÿšæÍèèřČçŤÍæŰźæşŢéĂŽåÿÿåĞžçŐřåIJÍéIJĂèęĄæÍąæŃ§ `case` èŕ■åŘěæĹŰåőđçŐřèő£éŮőèĂĚæÍąåijŔçŽĎæŮűåĂŹãĂĆ åŘĆèĂČäÿŃäÿĂåřŘèŁĆèŐůåŘŰæŻť'åđ'ŽéńŸçžğä¿Ńå■ŘãĂĆ

10.21 8.21 åőđçŐřèő£éŮőèĂĚæÍąåijŔ

éŮőécŸ

ä¡ăèęĄåđ'ĎçŘĘçŤšåđ'ğéĞŘäÿ■åŘŃçśžåđŃçŽĎåŕźèśąçżĎæĹŘçŽĎåđ'■æİĆæŢřæ■őçżŞæđĎïijŇæŕŘäÿĂæŕŤåęĆïijŇéĄ■åŐĘäÿĂäÿłæăŠå¡ćçżŞæđĎïijŇçĎűåŘŐæăźæ■őæŕŘäÿłèŁĆçĆźçŽĎçŻÿåžŤçŁűæĂĄæL'ğèąŃ

èğčåĘşæŰźæąĹ

è£ŹéĞŇéĄĞåĹřçŽĎéŮőécŸåIJÍçijŰçÍŃécĘå§§äÿ■æŸŕå¿ĹæŹőéĄ■çŽĎïijŇæIJL'æŮűåĂŹäijŽæđĎåžžäåĄĞèő¿ä¡ăèęĄåĘŹäÿĂäÿłèąÍçđ'žæŢřå■ęèąÍè¿¿åijŔçŽĎçÍŃåžŘïijŇéĆčäźĹä¡ăåŘŕèČ¡éIJĂèęĄåőŽäźL'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)

```
    pass

class Mul(BinaryOperator):
    pass

class Div(BinaryOperator):
    pass

class Negate(UnaryOperator):
    pass

class Number(Node):
    def __init__(self, value):
        self.value = value
```

çĎűåŘŐåĹľçŤĺè£ŹäžŽçśżæđĎđåžžåţŇåěŮæŢřæ■őçżŞæđĎïijŇåęĆäÿŃæĽ˙Ăçđ˙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£ŹæăůåĄŽçŽĎéŮőécŸæŸŕåŕźäžŐæŕŔäÿłèąĺè¿¿åijŔïijŇæŕŔæňąéČ¡èęĄéĞ■æŰřåőŽäźĽäÿĂéĄ■ïijŇæl
è£ŹéĞŇæĹŚäžňä¡£çŤĺèő£éŮőèĂĚæĺąåijŔåŔŕäžěè¿¿åĹře£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£ŹäÿłçśżïijŇåŔŕäžěåőŽäźĽäÿĂäÿłçśżçżğæĽ£åőČåźűäÿŤåőđçŐřåŘĎçğ■
`visit_Name()` æŰźæşŢïijŇåĚűäÿ■NameæŸŕnodeçśżåđŃãĂĆ
ä¿ŇåęĆïijŇåęĆæđIJä¡ăæČşæśĆèąĺè¿¿åijŔçŽĎåĂijïijŇåŔŕäžěè£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)

```
    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¡£çŤÍçd'žä¿ŃïijŽ

```
>>> e = Evaluator()
>>> e.visit(t4)
0.6
>>>
```

ä¡IJäýžäýĂäýłäý■åŘŇçŽĎä¿Ńå■ŘïijŇäýŃéİćåőŽäźĽ'äýĂäýłçśżåIJÍäýĂäýłæăĹäýŁéİćářĘäýĂäýłèąÍè¿¿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)

```
        self.instructions.append((instruction,))

    def visit_Negate(self, node):
        self.unaryop(node, 'NEG')
```

ä¡£çŤĺçd'žä¿ŃïijŽ

```
>>> s = StackCode()
>>> s.generate_code(t4)
[('PUSH', 1), ('PUSH', 2), ('PUSH', 3), ('PUSH', 4), ('SUB',),
('MUL',), ('PUSH', 5), ('DIV',), ('ADD',)]
>>>
```

èőÍèőž

åĹŽåijĂåğŃçŽĎæŮűåĂŹä¡ăåŔŕèČ¡äijŽåĘŹåd'ğéĞŔçŽĎif/elseèŕ■åŔěæİěåőđçŐřïijŇ è£ŹéĞŇèő£éŮőèĂĚæĺąåijŔçŽĎåě¡åd'ĎåřśæŸŕéĂŽè£Ğ `getattr()` æİèèŐůåŔŰçŻÿåžŤçŽĎæŰźæşŢïijŇåźűåĹl'çŤĺéĂŠå¡ŠæİěéĄ■åŐĘæL'ĂæIJL'çŽĎèŁĆçĆźïijŽ

```
def binop(self, node, instruction):
    self.visit(node.left)
    self.visit(node.right)
    self.instructions.append((instruction,))
```

è£ŸæIJL'äÿĂçĆźéIJĂèęĄæŇĞåĞžçŽĎæŸřïijŇè£Źçğ■æŁĂæIJŕäź§æŸŕåőđçŐřåĚűäżŰèŕ■èĺĂäÿ■switch æŕŤåęĆïijŇåęĆæđIJä¡ăæ■čåIJĺåĘŹäÿĂäÿłHTTPæąĘæđűïijŇä¡ăåŔŕèČ¡äijŽåĘŹè£ŹæăůäÿĂäÿłèŕůæśĆåĹĘåŔ

```
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£éŮőèĂĚæĺąåijŔäÿĂäÿłçijžçĆźåřśæŸŕåőČäÿěéĞ■ä¿İèţŰéĂŠå¡ŠïijŇåęĆæđIJæŢřæ■őçżŞæđĎåţŇåěŮ æIJL'æŮűåĂŹäijŽèűĚè£ĞPythonçŽĎéĂŠå¡ŠæůśåžęéŹŘåĹű(åŔĆèĂČ `sys.getrecursionlimit()`)ãĂĆ

åŔŕäžěåŔĆçĚğ8.22åřŔèŁĆïijŇåĹl'çŤĺçŤ§æĹŘåŹĺæĹŰè£■äžčåŹĺæİěåőđçŐřéİđéĂŠå¡ŠéĄ■åŐĘçőŮæşŢ

åIJĺèů§èğčæđŘåŠŇçijŰèŕŚçŻÿåĚşçŽĎçijŰçĺŃäÿ■ä¡£çŤĺèő£éŮőèĂĚæĺąåijŔæŸŕéİđåÿÿåÿÿèğĄçŽĎãĂĆ PythonæIJňèžńçŽĎ `ast` æĺąåİŮåĂijå¿ŮåĚşæşĺäÿŃïijŇåŔŕäžěåŐžçIJŃçIJŃæžŘçăĄãĂĆ 9.24åřŔèŁĆæijŤçd'žäžĘäÿĂäÿłåĹl'çŤĺ `ast` æĺąåİŮæİěåd'ĎçŘĘPythonæžŘäžčçăĄçŽĎä¿Ńå■ŘãĂĆ

10.22 8.22 äÿ■çŤĺéĂŠå¡Šåőđçŏřèő£éŮőèĂĚæĺąåijŔ

éŮőécŸ

ä¡ää¡£çŤĺèő£éŮőèĂĚæĺąåijŔéĄ■åŐĘäÿĂäÿłå¿Ĺæůśç ŽĎåţŇåěŮæăŠå¡ćæŢřæ■őçżŞæđĎïijŇåźűäÿŤåŻäää¡ăæČşæűĹéŹd'éĂŠå¡Šïij Ňåźűå ŘŇæŮűä£ÍæŇĄèő£éŮőèĂĚçijŰçÍŃæĺąåijŔãĂĆ

èğčåĘşæŰźæąĹ

éĂŽè£Ğåůğåę ŹçŽĎä¡£çŤĺçŤ§æĹŘåŹĺåŔŕäžěåIJĺæăŠéĄ■åŐĘæĹŰæŘIJçt'ćçőŮæşŢäÿ■æűĹéŹd'éĂŠå¡Šåij Í8.21åřŔèŁĆäÿ■ïijŇæĹŚäžňçžŹåĞžäžĘäÿĂäÿłèő£éŮőèĂĚçśżãĂĆäÿŃéİćæĹŚäžňåĽ'çŤĺäÿĂäÿłæăĹåŠŇçŤ§æĹŘåŹĺéĞ■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¡£çŤĺè£ŹäÿłçśżïijŇäź§èČ¡è¿¿åĹřçŻÿåŘŇçŽĎæŢĹæđIJãĂĆäžŇåőđäÿŁä¡ăåőŇåĚĺåŔŕäžěåřŁ

èĂČèŹŚåęĆäÿŃäżčçăĄïijŇéĄ■åŐĘäÿĂäÿłèąĺè¿¿åijŔçŽĎæăŚïijŽ

```
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)

```
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
```

åęĆæđIJåţŇåěŮåśĆæňąåd'łæůśéĆčäźĹäýŁè£řçŽĎEvaluatorářšäijŽåd'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£őæŤźäýŃäýŁéİćçŽĎEvaluatorïijŽ

```
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£ŘèąŇïijŇářšäý■äijŽæŁěéŤŹäžĘïijŽ

```
>>> a = Number(0)
>>> for n in range(1,100000):
...     a = Add(a, Number(n))
...
>>> e = Evaluator()
>>> e.visit(a)
4999950000
>>>
```

åęĆæđĲä¡ăè£ŸæČşæůżåŁăåĚűäžŰèĞłåőŽäźĽ'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
>>>
```

èőĺèőž

è£ŹäÿĂåřŘèŁĆæĹŚäżňæijŤçď'žäžĘçŤ§æĹŘåŹĺåŠŇå■ŔçĺŃåĲĺçĺŃåžŔæŐğåĹűæţĄæŰźéİćçŽĎåijžåď'g
éĄ£åĚ■éĂŠå¡ŠçŽĎäÿĂäÿłéĂŽåÿÿæŰźæşŢæŸŕä¡£çŤĺäÿĂäÿłæăĹæĹŰéŸ§åĹŮçŽĎæŢřæ■őçżŞæđĎãĂĆ
ä¿ŃåęĆïijŇæůśåžęäijŸåĚĹçŽĎéĄ■åŐĘçőŮæşŢïijŃçňňäÿĂæňąçćřåĹřäÿĂäÿłèŁĆçĆźæŮűåřĘåĚűåŐŃåĚěæ
æŰźæşŢçŽĎæäÿå£ČæĂİèůŕåřśæŸŕè£ŹæăůãĂĆ

åŘęåď'ŰäÿĂäÿłéĲĂèęĄçŘĘèğčçŽĎåřśæŸŕçŤ§æĹŘåŹĺäÿ■yieldèŕ■åŘěãĂĆå¡ŞçćřåĹřyieldèŕ■åŘěæŮűïij
äÿŁéİćçŽĎä¿Ńå■Ŕä¡£çŤĺè£ŹäÿłæŁĂæĲŕæİěäżčæŻ£äžĘéĂŠå¡ŠãĂĆä¿ŃåęĆïijŇäźŃåĽ■æĹŚäżňæŸŕè£Źæăů

```
value = self.visit(node.left)
```

çŐřåĲĺæ■ćæĹŘyieldèŕ■åŘěïijŽ

```
value = yield node.left
```

åőČäijŽåřĘ node.left è£ŤåŻđçżŹ visit() æŰźæşŢïijŇçĎűåŘŐ visit()

æŰźæşŢèřČçŤĺéĆčäÿłèŁĆçĆźçŻÿåžŤçŽĎ `visit_Name()` æŰźæşŢãĂĆ yield-
æŽĆæŮűåřEçĺŃåžŔæŐğåĹűåŹĺèőľ åĞžçżŹèřČçŤĺèĂĚïijŇå¡ŞæĽğèąŇåőŇåŘŐïijŇçżŞæđIJäijŽèţŃåĂijçżŹv

çIJŃåőŇè£ŹäÿĂåřŔèŁĆïijŇä¡ääź§èőÿæČşåŐżåŕżæĽ¿åĚűåőČæšąæIJL'yieldèŕ■åŘěçŽĎæŰźæąĹãĂĆä¡E
ä¿ŃåęĆïijŇäÿžäžĘæűĹéŹď'éĂŠå¡ŠïijŇä¡ää£Ěéążèę Ąçżt'æŁď'äÿĂäÿłæăĹçżŞæđĎïijŇåęĆæđIJäÿ■ä¡£çŤĺçŤ§a
åőđéŹĚäÿŁïijŇä¡£çŤĺyieldèŕ■åŘěåŘŕäžěèőľ'ä¡ääĘŹåĞžéİđåÿÿæijĆäžőçŽĎäžčçăĄïijŇåőČæűĹéŹď'äžĘéĂŠå¡

10.23 8.23 å¿łçŐŕåijŢçŤĺæŢřæ■őçżŞæđĎçŽĎåĘĚå■Ÿçőąçŕ Ę

éŮőécŸ

ä¡ăçŽĎçĺŃåžŔåĹŻåżžäžĘå¿Ĺåď'Žå¿łçŐŕåijŢçŤĺæŢřæ■őçżŞæđĎ(æŕŤåęĆæăŚãĂĄåŻ¿ãĂĄèğĆåŕ§èĂĚæ

èğčåĘşæŰźæąĹ

äÿĂäÿłçőĂå■ŢçŽĎå¿łçŐŕåijŢçŤĺæŢřæ■őçżŞæđĎä¿Ńå■ŔåŕśæŸŕäÿĂäÿłæăŚå¡ćçżŞæđĎïijŇåŔŇäžšèŁĆ
è£Źçğ■æČĚåĘţäÿŃïijŇåŘŕäżěèĂČèŹŚä¡£çŤĺ `weakref` åžŞäÿ■çŽĎåijśåijŢçŤĺãĂĆä¿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■æŸŕæČşæŰźåijŔåĚĄèőÿparentéİŹéżŸçżĹæ■ćãĂĆä¿ŃåęĆïijŽ

```
>>> root = Node('parent')
>>> c1 = Node('child')
>>> root.add_child(c1)
>>> print(c1.parent)
```

(continues on next page)

```
Node('parent')
>>> del root
>>> print(c1.parent)
None
>>>
```

èőÍèőž

å¿łçŐřåijŢçŤÍçŽĎæŢřæ■őçżŞæđĎåIJÍPythonäÿ■æŸŕäÿĂäÿłå¿ĹæčŸæĽŃçŽĎéŮőéćŸïijŇåŻääÿžæ■čåÿä¿ŇåęĆèĂČèŹŚåęĆäÿŇäžčçăĄïijŽ

```
# 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
```

äÿŇéÍćæĹŚäžňä¡£çŤÍè£ŹäÿłäžčçăĄæÍěåĄŽäÿĂäžŽåđČåIJ¿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)
>>>
```

åŘŕäžěçIJŃåĹřïijŇæIJĂåŘŐäÿĂäÿłçŽĎåĹăéŹđ'æŮűæĽŞå■řèŕ■åŘěæšąæIJL'åĞžçŐřãĂĆåŐ§åŻäæŸŕPyå¡ŞäÿĂäÿłåŕźèšąçŽĎåijŢçŤÍæŢřåŔŸæĹŘ0çŽĎæŮűåĂŹæĽ■äijŽçńŇå■şåĹăéŹđ'æŐĽ'ãĂĆèĂŇåŕźäžŐå¿łçŐřåŻäæ■đ'ïijŇåIJÍäÿŁéÍćä¿Ňå■ŘäÿæIJĂåŘŐéČÍåĹĘïijŇçĹűèŁĆçĆźåŠŇå■l'å■ŘèŁĆçĆźäžŠçŻÿæŇěæIJL'åŕźa

PythonæIJL'åŘęåđ'ŰçŽĎåđČåIJ¿åŻđæŤűåŹÍæÍěäÿŞéŮÍéŠĹåŕźå¿łçŐřåijŢçŤÍçŽĎïijŇä¡ĘæŸŕä¡ăæřÿè£IJåŘęåđ'Űä¡ăè£ŸåŘŕäžěæL'ŇåŁÍçŽĎèğęåŔŠåőČïijŇä¡ĘæŸŕäžčçăĄçIJŇäÿŁåŐżå¿ĹæŇńïijŽ

```
>>> import gc
>>> gc.collect() # Force collection
Data.__del__
Data.__del__
>>>
```

åęĆæđIJå¿łçŐŕåijŢçŤĺçŽĎåŕźèśąèĞłåůśè£ŸåőŽäźL'äžĘèĞłåůśçŽĎ __del__() æŰźæşŢïijŇéĆčäźĹäijŽèől'æČĚåĘţåŔŸå¿ŮæŻt'çş§çşŢãĂĆ åĄĞèő¿ä¡ăåČŔäÿŃéİćè£ŹæăůçżŹNodeåőŽäźL'èĞłåůśçŽĎ __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¡ăèŕŢçİĂåŐżè£ŘèąŇåőČäijŽåŔŚçŐřïijŇData.__del__ æűĹæĄŕæřÿè£IJäÿ■äijŽåĞžçŐřäžE,çŤŽèĞşåIJĺä¡ăåijžåĹűåĘĚå■ŸåŻđæŤűæŮűïijŽ

```
>>> a = Node()
>>> a.add_child(Node())
>>> del a # No message (not collected)
>>> import gc
>>> gc.collect() # No message (not collected)
>>>
```

åijśåijŢçŤĺæűĹéŹď'äžĘåijŢçŤĺå¿łçŐŕçŽĎè£ŹäÿłéŮőécŸïijŇæIJňèt'ĺæİèèőšïijŇåijśåijŢçŤĺåřśæŸŕäÿĂäÿ ä¡ăåŔŕäžěéĂŽè£Ğ weakref æİěåĹŽåžžåijśåijŢçŤĺãĆä¿ŃåęĆïijŽ

```
>>> import weakref
>>> a = Node()
>>> a_ref = weakref.ref(a)
>>> a_ref
<weakref at 0x100581f70; to 'Node' at 0x1005c5410>
>>>
```

äÿžäžĘèő£éŮőåijśåijŢçŤĺæL'ĂåijŢçŤĺçŽĎåŕźèśąïijŇä¡ăåŔŕäžěåČŔåĞ¡æŢřäÿĂæăůåŐżèřČçŤĺåőČå■şåŔ

çŤśäžŐåŐ§åğŃåŕźèśąçŽĎåijŢçŤĺèőąæŢřæšąæIJL’åćđåŁäïijŇéĆčäźĹåřśåŔŕäžěåŐżåĹäéŹď’åőČäžĘãĂĆä¿Ńåę

```
>>> print(a_ref())
<__main__.Node object at 0x1005c5410>
>>> del a
Data.__del__
>>> print(a_ref())
None
>>>
```

éĂŽè£Ğè£ŹéĞŇæijŤçď’žçŽĎåijśåijŢçŤĺæŁĂæIJŕïijŇä¡ăäijŽåŔŚçŐřäÿ■åĘ■æIJL’å¿łçŐŕåijŢçŤĺéŮőécŸ
ä¡ăè£ŸèČ¡åŔĆèĂČ8.25åřŔèŁĆåĚşäžŐåijśåijŢçŤĺçŽĎåŔęåď’ŰäÿĂäÿłä¿Ńå■ŘãĂĆ

10.24 8.24 èőľçśżæŤŕæŇĄæŕŤè¿ČæŞ■ä¡IJ

éŮőécŸ

ä¡ăæČşèőľæ§ŘäÿłçśżçŽĎåőđä¿ŃæŤŕæŇĄææĞåĞĘçŽĎæŕŤè¿Čè£ŘçőŮ(æŕŤåęĆ>=,!=,<=,<ç■L’)ïijŇä¡Ę

èğčåĘşæŰźæąĹ

PythonçśżåŕźæŕŔäÿłæŕŤè¿ČæŞ■ä¡IJéČ¡éIJĂèęĄåőđçŐřäÿĂäÿłçL’źæőŁæŰźæşŢæİěæŤŕæŇĄãĂĆ
ä¿ŃåęĆäÿžäžĘæŤŕæŇĄ>=æŞ■ä¡IJçňęïijŇä¡ăéIJĂèęĄåőŽäźĹ’äÿĂäÿł `__ge__()`
æŰźæşŢãĂĆ år¡çőąåőŽäźĹ’äÿĂäÿłæŰźæşŢæšąäżĂäźĹéŮőécŸïijŇä¡ĘåęĆæđIJèęĄä¡ăåőđçŐřæĹ’ĂæIJL’åŔŕè

èčĚéěřåŹÍ `functools.total_ordering` åřśæŸŕçŤĺæİěçőĂåŇŰè£Źäÿłåď’ĎçŘĘçŽĎãĂĆ
ä¡£çŤĺåőĆæİěèčĚéěřäÿĂäÿłæİěïijŇä¡ăåŔłéIJĂåőŽäźĹ’äÿĂäÿł `__eq__()` æŰźæşŢïijŇ
åď’ŰåŁăåĚűäžŰæŰźæşŢ(__lt__, __le__, __gt__, or __ge__)äÿ■çŽĎäÿĂäÿłå■şåŔŕãĂĆ
çĎűåŘŐèčĚéěřåŹĺäijŽèĞłåŁĺäÿžä¡ăåąńåĚĚåĚűåőĆærŤè¿ČæŰźæşŢãĂĆ

ä¡IJäÿžä¿Ńå■ŘïijŇæĹŚäžňædĎåžžäÿĂäžŻæĹ£å■ŘïijŇçĎűåŘŐçżŹåőČäžňáćđåŁääÿĂäžŻæĹ£éŮt’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)

```
    @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£ŹéĞŇæĹŚäžňåŔłæŸŕçżŹHouseçśżåőŽäźL'äžĘäÿd'äÿłæŰźæşŢïijŽ__eq__() åŠŇ __lt__() ïijŇåőČåřśèČ¡æŤŕæŇĄæL'ĂæIJL'çŽĎæŕŤè¿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 åőČåřśæŸŕåőŽäźĽäžĘäÿĂäÿłäžŐæŕŔäÿłæŕŤè¿ČæŤŕæŇĄæŰźæşŢåĹřæĽĂæIJĽéIJĂèęĄåőŽäźĽçŽĎåĚűäżŰ æŕŤåęĆä¡ääőŽäźĽäžĘ __le__() æŰźæşŢïijŇéĆčäźĹåőČåřśèćńçŤĺæİěæđĎåżžæĽĂæIJĽåĚűäżŰçŽĎéIJĂè åőđéŹĚäÿŁåřśæŸŕåIJÍçśžéĞŇéİćåČŔäÿŃéİćè£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¡ĘæŸŕä¡£çŤÍ @total_ordering åŔŕäžěçőĂåŇŰäžčçăĄïijŇä¡ŢäźŘèĂŇäÿ■äÿžåŚćãĂĆ

10.25 8.25 åĹŻåžžçijŞå■Ÿåőđä¿Ń

éŮőécŸ

åIJÍåĹŻåžžäÿĂäÿłçśžçŽĎåŕźèśąæŮűïijŇåęĆæđIJäźŃåL’■ä¡£çŤĺåŘŇæăůåŔĆæŢřåĹŻåžžè£Ğè£Źäÿłåŕźèś ä¡äæČşè£ŤåŻđåőČçŽĎçijŞå■Ÿåij ŢçŤÍãĂĆ

èğčåĘşæŰźæąĹ

è£ŹçğéĂŽåÿÿæŸŕåŻääÿžä¡ääÿŇæIJŻçŻÿåŘŇåŔĆæŢřåĹŻåžžçŽĎåŕźèśąæŮűå■Ţä¿ŃçŽĎãĂĆ åIJÍå¿Ĺåď’ŽåžŞäÿ■éČ¡æIJĽåőđéŹĚçŽĎä¿Ńå■ŘïijŇæŕŤåęĆ logging æÍąåİŮïijŇä¡£çŤÍçŻÿåŘŇçŽĎåŘ■çğřåĹŻåžžçŽĎ logger åőđä¿Ńæřÿè£IJåŔłæIJĽäÿĂäÿłãĂĆä¿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¡£çŤÍäÿĂäÿłåŠŇçśžæIJñèžńåĹĘåijĂçŽĎåůěåŐĆåĞ

```
# 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¡ăäijŽåŔŞçŐřèů§äźŃåL’■éĆčäÿłæŮěå£ŮåŕźèśąçŽĎåĹŻåžžèąŇäÿžæŸŕ

```
>>> a = get_spam('foo')
>>> b = get_spam('bar')
>>> a is b
False
>>> c = get_spam('foo')
>>> a is c
True
>>>
```

èőÍèőž

çijŰåĘŹäÿĂäÿłåůěåŐĆåĞ¡æŢřæİěä£őæŤźæŹőéĂŽçŽĎåőđä¿ŃåĹŻåžžèąŇäÿžéĂŽåÿÿæŸŕäÿĂäÿłæŕŤè¿ä¡ĘæŸŕæĹŚäżňè£ŸèČ¡åŘęæL’¿åĹřæŻt’äijŸéŻĚçŽĎèğčåĘşæŰźæąĹåŚćïij§

ä¿ŃåęĆïijŇä¡ăåŔŕèČ¡äijŽèĂČèŹŚéĞ■æŰřåőŽäźL’çśżçŽĎ `__new__()` æŰźæşŢïijŇářšåČŔäÿŃéİćè£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)

```
        print('Initializing Spam')
        self.name = name
```

åĹİçIJŃèţůæİěåě¡åČŕåŕŕäžěè¿¿åĹřéćĎæIJ§æŢĹæđIJïijŇä¡ĘæŸŕéŮőéćŸæŸŕ
__init__() æŕŕæňąéČ¡äijŽèćńěřČçŤĺïijŇäý■çőąè£Źäýłåőđä¿ŃæŸŕåŘęèćńçijŞå■ŸäžĘãĂĆä¿ŃåęĆïijŽ

```
>>> s = Spam('Dave')
Initializing Spam
>>> t = Spam('Dave')
Initializing Spam
>>> s is t
True
>>>
```

è£ŹäýłæĺŰèőÿäý■æŸŕä¡ăæČşèęĄçŽĎæŢĹæđIJïijŇåŻăæ■d'è£Źçğ■æŰźæşŢåźűäý■åŕŕåŘŰãĂĆ

äýŁéİćæĹŚäžňä¡£çŤĺåĹřäžĘåijśåijŢçŤĺèőąæŢřïijŇåŕźäžŐåđČåIJ¿åŻđæŤűæİèèőšæŸŕå¿ĹæIJL'åÿőåŁľçŽ
å¡ŞæĹŚäžňä£ÍæŇĄåőđä¿ŃçijŞå■ŸæŮűïijŇä¡ăåŘŕèČ¡åŘłæČşåIJĺçĺŃåžŔäý■ä¡£çŤĺåĹřåőČäžňæŮűæL'■ä£Íå■
äýĂäýł WeakValueDictionary åőđä¿ŃåŘłäijŽä£Íå■ŸéĆčäžŻåIJĺåĚűåőČåIJřæŰźè£ŸåIJĺèćńä¡£çŤĺçŽĎå
åŘęåĹŹçŽĎèŕÏïijŇåŘłèęĄåőđä¿Ńäý■åĘ■èćńä¡£çŤĺäžĘïijŇåőČåřśäžŐå■ŮåĚÿäý■èćńçğżéŹd'äžĘãĂĆèğĆår§ă

```
>>> 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)
[]
>>>
```

åŕźäžŐåd'ğéČĺåĹęçĺŃåžŔèĂŇåůšïijŇè£ŹéĞŇäžčçăĄåůšçżŘåd'§çŤĺäžĘãĂĆäý■è£Ğè£ŸæŸŕæIJL'äÿĂäž

éęŰåĚĹæŸŕè£ŹéĞŇä¡£çŤĺåĹřäžĘäÿĂäýłåĚĺåśĂåŔŸéĞŔïijŇåźűäýŤåůěåŐĆåĞ¡æŢřèů§çśżæŤ¿åIJĺäÿĂ

```
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)

```
            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 æÿĚæŹřïijŇåźűäýŤäź§æŻť çĄţæť żïijŇæĹŚäžňåŔŕäžěåćđåŁăæŻť åď ŽçŽĎçij

è£ŸæIJL’äýĂçĆźåřśæŸŕïijŇæĹŚäžňæŽť éIJšäžĘçśžçŽĎåőđä¿ŃåŇŰçżŹçŤĺæĹůïijŇçŤĺæĹůå¿Ĺåőźæ ŸŞ

```
>>> a = Spam('foo')
>>> b = Spam('foo')
>>> a is b
False
>>>
```

æIJL’åĞăçğ■æŰźåijŔåŔŕäžěéŸšæ■ćçŤĺæĹůè£Źæăůå ĄŽïijŇçňňäýĂäýłæŸŕåřĘçśžçŽĎåŘ■å■Ůä£őæŤźäý çňňäžŇçğ■åřśæŸŕèőľè£ŹäýłçśžçŽĎ `__init__()` æŰźæşŢæŁŻåĞžäýĂäýłåijĆåÿÿïijŇèőľåőČäÿ■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()` æĺěåĹŻåžžåőđä¿ŃïijŇèĂŇäÿ■æŸŕçŻť æŐěěřČçŤĺ `Spam()` æđĎéĂäåĞ¡æŢŕïijŽ

```
# ----------------------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)

```
        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£Źæăůç ŽĎæŰźæąĹåřśåůšçżŔèűşåď§åě¡äžĘãĂĆ
çijŞå■ŸåŠŇåĚűäžŰæđĎéĂăæĺąåijŔè£ŸåŔřäžěä¡£çŤÍ9.13åřŔèŁĆäÿ■çŽĎåĚČçśżåőđçŐřçŽĎæŻťäijŸéŻĚäÿ

11 çňňäźİçńäïijŽåĚČçijŰçÍŃ

è¡ŕäžűåijĂåŔŚéćĘå§§äÿ■æIJĂçżŔåĚÿçŽĎåŔčåď'ť'çęĚåřśæŸŕâĂIJdonâĂŹt repeat yourselfâĂİãĂĆ äź§åřśæŸŕèŕť'ïijŇäžżä¡ŢæŮűåĂŹå¡Şä¡ăçŽĎçÍŃåžŔäÿ■å■ŸåIJĺéńŸåžęéĞ■åď'■(æĹŰèĂĚæŸŕéĂ
åIJÍPythonå¡Şäÿ■ïijŇéĂŽåÿÿéČ¡åŔŕäžěéĂŽè£ĞåĚČçijŰçÍŃæİěèğčåĘşè£ŹçśżéŮőécŸãĂĆ
çőĂèĂŇèÍĂäźŃïijŇåĚČçijŰçÍŃåřśæŸŕåĚşäžŐåĹŽåžżæŞ■ä¡IJæžŘäžčçăĄ(æŕŤåęĆä£őæŤźãĂĄçŢ§æĹŘæĹŰ
äÿżèęĄæŁĂæIJŕæŸŕä¡£çŤÍèčĚéěřåŹÍãĂĄçśżèčĚéěřåŹÍåŠŇåĚČçśżãĂĆäÿ■è£Ğè£ŸæIJL'äÿĂäžŽåĚűäžŰæŁĂ
åŇĚæŃňç■¿åŘ■åŕźèśąãĂĄä¡£çŤÍ`exec()` æL'ğèąŇäžčçăĄäżěåŔŁåŕźåĘĚéČÍåĞ¡æŢŕåŠŇçśżçŽĎåŔ■åřĎæŁ
æIJňçńăçŽĎäÿżèęĄçŽőçŽĎæŸŕåŘŚåď'ğåőűäżŃçż■è£ŹäžŽåĚČçijŰçÍŃæŁĂæIJŕïijŇåźűäÿŤçżŹåĞžåőđä¿Ńæ

Contents:

11.1 9.1 åIJÍåĞ¡æŢŕäÿŁæůżåŁäåŇĚèčĚåŹÍ

éŮőécŸ

ä¡ăæČşåIJÍåĞ¡æŢŕäÿŁæůżåŁääÿĂäÿłåŇĚèčĚåŹÍïijŇåćđåŁäécİåď'ŰçŽĎæŞ■ä¡IJåď'ĎçŘĘ(æŕŤåęĆæŮěå

èğčåĘşæŰźæąĹ

åęĆæđIJä¡ăæČşä¡£çŤÍécİåď'ŰçŽĎäžčçăĄåŇĚèčĚäÿĂäÿłåĞ¡æŢŕïijŇåŔŕäžěåőŽäźĽäÿĂäÿłèčĚéěřåŹÍåĞ

```
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
```

äÿŃéÍćæŸŕä¡£çŤÍèčĚéěřåŹÍçŽĎä¿Ńå■ŘïijŽ

```
>>> @timethis
... def countdown(n):
...     '''
...     Counts down
...     '''
...     while n > 0:
...         n -= 1
...
>>> countdown(100000)
countdown 0.008917808532714844
>>> countdown(10000000)
countdown 0.87188299392912
>>>
```

èőÍèőž

äÿĂäÿłèčĚéěřåŹÍåřśæŸŕäÿĂäÿłåĞ¡æŢřïijŇåőČæŐěåŔŮäÿĂäÿłåĞ¡æŢřä¡IJäÿžåŔĆæŢřåźűè£ŤåŻđäÿĂäÿ
å¡Şä¡ăåČŔäÿŃéÍćè£ŹæăůåĘŹïijŽ

```
@timethis
def countdown(n):
    pass
```

èů§åČŔäÿŃéÍćè£ŹæăůåĘŹåĚűåőđæŢĹæđIJæŸŕäÿĂæăůçŽĎïijŽ

```
def countdown(n):
    pass
countdown = timethis(countdown)
```

éąžä¿£èŕť'äÿĂäÿŃïijŇåĘĚç¡őçŽĎèčĚéěřåŹÍæŕŤåęĆ `@staticmethod`, `@classmethod`, `@property` åŐ§çŘĘäź§æŸŕäÿĂæăůçŽĎãĂĆ

ä¿ÑåęĆïijÑäÿÑéİćè£Źäÿď äÿłäžčçăĄçL’Ğæőţæ Ÿŕç■L’äžůçŽĎïijŽ

```
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¡æŢřäÿ■ïijÑ èčĚéěřåŹÍåĘĚéČĺåőŽäźL’äžĘäÿĂäÿłä¡£çŢÍ `*args` åŠŇ `**kwargs` æİěæŐěåŔŮäżżæĎŔåŔĆæŢřçŽĎåĞ¡æŢřãĂĆ åIJÍè£Źäÿłå Ğ¡æŢřéĞŇéİćèřČçŢĺäžĘąåŐ§ågŃåĞ¡æŢřåźűåřĘąĚűçżŞæđIJè£ŤåŻđïijÑäÿ■è£Ğä¡ăè£ŸåŔŕäžěæůżçĎűåŘŐè£Źäÿłæ ŰřçŽĎåĞ¡æŢřåŇĚèčĚåŹÍèćńä¡IJäÿžçżŞæđIJè£ŤåŻđæİěäžčæŻ£åŐ§ågŃåĞ¡æŢřãĂĆ

éIJĂèęĄåijžèřČçŽĎæŸŕècĚéěřåŹÍåźűäÿ■äijŽä£őæŤźåŐ§ågŃåĞ¡æŢřçŽĎåŔĆæŢřç■¿åŘ■äżěåŔŁè£ŤåŻ ä¡£çŢÍ `*args` åŠŇ `**kwargs` çŻőçŽĎåřśæŸŕçąőä£İäżżä¡ŢåŔĆæŢréČ¡èČ¡éĂĆçŢÍãĂĆ èĂŇè£ŤåŻđçżŞæđIJåĂijå§žæIJńéČ¡æŸŕèřČçŢÍåŐ§ågŃåĞ¡æŢř `func(*args, **kwargs)` çŽĎè£ŤåŻđçżŞæđIJïijÑåĚűäÿ■funcåřśæŸŕåŐ§ågŃåĞ¡æŢřãĂĆ

åĹŽåijĂågŃå■ęäźăèčĚéěřåŹÍçŽĎæŮűåĂŹïijÑäijŽä¡£çŢÍäÿĂäžŻçőĂå■ŢçŽĎä¿Ńå■Řæİèèŕť æŸŐïijÑæŕ äÿ■è£ĞåőđéŹĚåIJžæŹŕä¡£çŢÍæŮűïijÑè£ŸæŸŕæIJL’äÿĂäžŻçżĘèŁĆéŮőécŸèęĄæşÍæĎŔçŽĎãĂĆ æŕŤåęĆäÿŁéİćä¡£çŢÍ `@wraps(func)` æşÍèğčæŸŕå¿ĹéĞ■èęĄçŽĎïijÑ åőĆèČ¡ä£İçŢŹåŐ§ågŃåĞ¡æŢřçŽĎåĚČæŢřæ■ő(äÿÑäÿĂåřŔèŁĆäijŽèőšåĹř)ïijÑæŰřæL’ŃçżŘåÿÿäijŽå£¡çŢĕè æŐëäÿÑæİěçŽĎåĞăäÿłåřŔèŁĆæĹŚäžňäijŽæŻť åŁăæůśåĚęçŽĎèőšèğčèČĚéěřåŹÍåĞ¡æŢřçŽĎçżĘèŁĆéŮőécŸ

11.2 9.2 åĹŽåżžèčĚéěřåŹÍæŮűä£İçŢŹåĞ¡æŢřåĚČä£ąæĄŕ

éŮőécŸ

ä¡ăåĘŹäžĘäÿĂäÿłèčĚéěřåŹÍä¡IJçŤÍåIJÍæ§Řäÿłå Ğ¡æŢřäŸŁïijÑä¡ĘæŸŕè£Źäÿłå Ğ¡æŢřçŽĎéĞ■èęĄçŽĎåĚ

èğčåĘşæŰźæąĹ

äżżä¡ŢæŮűåĂŹä¡ăåőŽäźL’èčĚéěřåŹÍçŽĎæŮűåĂŹïijÑéČ¡åžŤèŕěä¡£çŤÍ `functools` åžŞäÿ■çŽĎ `@wraps` èčĚéěřåŹÍæİěæşÍèğčåžŢåśĆåŇĚèčĚåĞ¡æŢřãĂĆä¿ÑåęĆïijŽ

```
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)

```
        start = time.time()
        result = func(*args, **kwargs)
        end = time.time()
        print(func.__name__, end-start)
        return result
    return wrapper
```

äÿŃéİćæĹŚäżňä¡£çŤĺè£ŹäÿłèćńåŇĚèčĚåŘŐçŽĎåĞ¡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'>}
>>>
```

èőĺèőž

åIJĺçijŰåĘŹèčĚéěřåŹĺçŽĎæŮűåĂŹåd’■åĹűåĚČä£ąæĄŕæŸŕäÿĂäÿłéİđåÿÿéĞ■èęĄçŽĎéČĺåĹĘãĂĆåęĆæ
`@wraps` ïijŇ éĆčäźĹä¡ääijŽåŔŚçŐřèćńèčĚéěřåĞ¡æŢřäÿćåd’śäžĘæL’ĂæIJL’æIJL’çŤĺçŽĎä£ąæĄŕãĂĆærŤåęĆ
`@wraps` åŘŐçŽĎæŢĹæđIJæŸŕäÿŃéİćè£ŹæăůçŽĎïijŽ

```
>>> countdown.__name__
'wrapper'
>>> countdown.__doc__
>>> countdown.__annotations__
{}
>>>
```

`@wraps` æIJL’äÿĂäÿłéĞ■èęĄçL’źå¿ĄæŸŕåőČèČ¡èől’ä¡äéĂŽè£ĞåšđæĂğ
`__wrapped__` çŻt’æŐèèő£éŮőèćńåŇĚèčĚåĞ¡æŢřãĂĆä¿ŃåęĆ:

```
>>> countdown.__wrapped__(100000)
>>>
```

`__wrapped__` åšđæĂğè£ŸèČ¡èől’èćńèčĚéěřåĞ¡æŢřæ■čçąőæŽt’éIJšåžŢåśĆçŽĎåŔĆæŢřç■¿åŘ■ä£ąæA

```
>>> from inspect import signature
>>> print(signature(countdown))
(n:int)
>>>
```

äÿĂäÿłå¿ĹæŹőéĄ■çŽĎéŮőécŸæŸŕæĂŐæăůèőľèčĚéěřåŹĺåŐżçŻťæŐěåď■åĹűåŐ§ågŃåĞ¡æŢřçŽĎåŔĆåęĆæđIJæČşèĞłåůśæĽŃåŁĺåőđçŐřçŽĎèŕÍéIJĂèęĄåĄŽåď ğéĞŔçŽĎåůěä¡IJïijŇæIJĂåě¡åřśçőĂå■ŢçŽĎä¡£çŤ
`@wraps` èčĚéěřåŹĺãĂĆ éĂŽè£ĞåžŢåśĆçŽĎ `__wrapped__` åśđæĂğèő£éŮőåĹřåĞ¡æŢřç■¿åŘ■ä£ąæĄřãĂĆæŻť åď ŽåĚşäžŐç■¿åŘ■çŽĎåĘĚåőźåŔŕäżěåŔĆèĂČ9.16åřŔèŁ

11.3 9.3 èğčéŹď äÿĂäÿłèčĚéěřåŹĺ

éŮőécŸ

äÿĂäÿłèčĚéěřåŹĺåůšçżŔä¡IJçŤĺåIJĺäÿĂäÿłåĞ¡æŢřäŸŁïijŇä¡ăæČşæŠď éŤĂåőČïijŇçŻť æŐèèő£éŮőåŐ§åğ

èğčåĘşæŰźæąĹ

åĄĞèő¿èčĚéěřåŹĺæŸŕéĂŽè£Ğ `@wraps` (åŔĆèĂČ9.2åřŔèŁĆ)æİěåőđçŐřçŽĎïijŇéĆčäźĹä¡ääŔŕäżěéĂŽè
`__wrapped__` åśđæĂğæİèèő£éŮőåŐ§åğŃåĞ¡æŢřïijŽ

```
>>> @somedecorator
>>> def add(x, y):
...     return x + y
...
>>> orig_add = add.__wrapped__
>>> orig_add(3, 4)
7
>>>
```

èőĺèőž

çŻť æŐèèő£éŮőæIJłåŇĚèčĚçŽĎåŐ§ågŃåĞ¡æŢřåIJĺèřČèŕŢãĂĄåĘĚçIJĄåŠŇåĚűäżŮåĞ¡æŢřæŞ■ä¡IJæŮ
ä¡ĘæŸŕæĹŚäżňè£ŹéĞŇçŽĎæŰźæąĹäżĚäżĚéĂĆçŤĺäžŐåIJĺåŇĚèčĚåŹĺäÿ■æ■čçąőä¡£çŤĺäžĘ
`@wraps` æĹŰèĂĚçŻť æŐèèő¿ç¡őäžĘ `__wrapped__` åśđæĂğçŽĎæČĚåĘţãĂĆ

åęĆæđIJæIJL ådď Žäÿłå ŇĚèčĚåŹĺïijŇéĆčäźĹèő£éŮő `__wrapped__`
åśđæĂğçŽĎèąŇäÿžæŸŕäÿ■åŕŕéćĎç§ěçŽĎïijŇåžŤèŕěéĄ£åĚ■è£Źæăůå ĄžãĂĆ
åIJĺPython3.3äÿ■ïijŇåőČäijŽçŢèè£ĞæĽĂæIJL çŽĎåŇĚèčĚåśĆïijŇæŕŤåęĆïijŇåĄĞåęĆä¡ăæIJL åęĆäÿŃçŽĎ

```
from functools import wraps

def decorator1(func):
    @wraps(func)
    def wrapper(*args, **kwargs):
```

(continues on next 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
>>>
```

æIJĂåŘŐèęĄèŕťçŽDæŸŕïijŇåźűäÿ■æŸŕæĽĂæIJĽçŽDèčĚéěřåŹĺéČ¡ä¡£çŤĺäžĘ `@wraps` ïijŇåŽäæ■d'è£ŹéĞŇçŽDæŰźæąĹåźűäÿ■åĚĺéČĺéĂĆçŤĺãĂĆçĽźåĹńçŽDïijŇåĘĚç¡őçŽDèčĚéěřåŹĺ `@staticmethod` åŠŇ `@classmethod` åřśæšąæIJĽéĄţå¿łè£Źäÿłçžęåőž (åőČäżňæŁŁåŐ§ågŃåĞ¡æŢřå■ŸåĆĺåIJĺåśđæĂg `__func__` äÿ■)ãĂĆ

11.4 9.4 åőŽäźĽäÿĂäÿłåÿęåŔĆæŢřçŽDèčĚéěřåŹĺ

éŮőécŸ

ä¡ăæČşåőŽäźĽäÿĂäÿłåŔŕäžěæŐěåŔŮåŔĆæŢřçŽDèčĚéěřåŹĺ

èğčåĘşæŰźæąĹ

æĹŚäžňçŤĺäÿĂäÿłä¿Ńå■ŘèŕęçżĕéŸŘè£řäÿŃæŐěåŔŮåŔĆæŢřçŽĎåď ĎçŘĘè£ĞçÍŃãĂĆ åĄĞèő¿ä¡ăæČşåĘŹäÿĂäÿłèčĚéěřåŹÍïijŇçżŹåĞ¡æŢřæůžåŁăæŮěå£ŮåŁ§èČ¡ïijŇåŘŇæŮűåĚĄèőÿçŤĺæĹůæŇ äÿŃéİćæŸŕè£ŹäÿłèčĚéěřåŹÍçŽĎåőŽäźĽåŠŇä¡£çŤĺçď 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!')
```

åĹİçIJŃèţůæİěïijŇè£Źçğ■åőđçŐřçIJŃäÿŁåŐżå¿Ĺåď■æİĆïijŇä¡ĘæŸŕæăÿå£ČæĂİæČşå¿ĹçőĂå■ŢãĂĆ æIJĂåď ŰåśĆçŽĎåĞ¡æŢř `logged()` æŐěåŔŮåŔĆæŢřåźűåřĘåőČäžňä¡IJçŤĺåIJĺåĘĚéČĺçŽĎèčĚéěřåŹĺåĞ¡æ åĘĚåśĆçŽĎåĞ¡æŢř `decorate()` æŐěåŔŮäÿĂäÿłåĞ¡æŢřä¡IJäÿžåŔĆæŢřïijŇçĎűåŘŐåIJĺåĞ¡æŢřäŁéİćæŤ è£ŹéĞŇçŽĎåĚşéŤőçĆźæŸŕåŇĚèčĚåŹĺæŸŕåŔŕäžěä¡£çŤĺäijåéĂŠçžŹ `logged()` çŽĎåŔĆæŢřçŽĎãĂĆ

èőĺèőž

åőŽäźĽäÿĂäÿłæŐěåŔŮåŔĆæŢřçŽĎåŇĚèčĚåŹĺçIJŃäÿŁåŐżæŕŤè¿Čåď■æİĆäÿżèęĄæŸŕåŻääÿžåžŢåśĆ

```
@decorator(x, y, z)
def func(a, b):
```

(continues on next page)

```
    pass
```

èčĚéěřåŹĺåď'ĎçŘĘè£ĞçÍŃèů§äÿŃéİćçŽĎèřČçŤÍæŸŕç■L'æŢĹçŽĎ;

```
def func(a, b):
    pass
func = decorator(x, y, z)(func)
```

decorator(x, y, z) çŽĎè£ŤåŻđçżŞæđIJå£ĚéążæŸŕäÿĂäÿłåŔŕèřČçŤĺåŕźèśąïijŇåőČæŐěåŔůäÿĂ åŔŕäżěåŔĆèĂČ9.7åřŔèŁĆäÿ■åŔęåď'ŰäÿĂäÿłåŔŕæŐěåŔůåŔĆæŢřçŽĎåŇĚèčĚåŹĺä¿Ńå■ŘãĂĆ

11.5 9.5 åŔŕèĞłáőŽäźĽåśđæĂğçŽĎèčĚéěřåŹĺ

éŮőécŸ

ä¡ăæČşåĘŹäÿĂäÿłèčĚéěřåŹĺæİěåŇĚèčĚäÿĂäÿłåĞ¡æŢřïijŇåźűäÿŤåĚĄèőÿçŤĺæĹůæŔŘä¿ŻåŔĆæŢřåIJĺè

èğčåĘşæŰźæąĹ

åijŢåĚěäÿĂäÿłèő£éŮőåĞ¡æŢřïijŇä¡£çŤĺ nonlocal æİěä£őæŤźåĘĚéČĺåŔŸéĞŔãĂĆ çĎűåŘŐè£Źäÿłèő£éŮőåĞ¡æŢřèćńä¡IJäÿžäÿĂäÿłåśđæĂğètŇåĂijçżŹåŇĚèčĚåĞ¡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)

```
            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!')
```

äÿŃéİćœŸŕäžď'äžŠçŐŕåćČäÿŃçŽĎä¡£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` æÌěä£őæŤźåĞ¡æŢřåĘĚéČÍçŽĎåŔŸéĞŔãĂĆ

è£ŸæIJL'äÿĂäÿłäžd'äžžåŘČæČŁçŽĎåIJřæŰźæŸŕèő£éŮőåĞ¡æŢřäijŽåIJÍåd'ŽåśĆèčĚéěřåŹÍéŮt'äijäæŠ■ `@functools.wraps` æşÍèğč)ãĂĆ ä¿ŃåęĆïijŇåĄĞèő¿ä¡ăåijŢåĚěåŔęåd'ŰäÿĂäÿłèčĚéěřåŹÍïijŇæŕŤåęĆ9.2å `@timethis` ïijŇåČŔäÿŃéİćè£Źæăůïijž

```
@timethis
@logged(logging.DEBUG)
def countdown(n):
    while n > 0:
        n -= 1
```

ä¡ăäijŽåŔŠçŐřèő£éŮőåĞ¡æŢřä¿ÌæŮğæIJL'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¡ăè£ŸäijŽåŔŠçŐřå■şä¡£èčĚéěřåŹÍåČŔäÿŃéİćè£ŹæăůäžěçŻÿåŔ■çŽĎæŰźåŘŠæŐŠæŤ¿ïijŇæŢĹæđIJäź

```
@logged(logging.DEBUG)
@timethis
def countdown(n):
    while n > 0:
        n -= 1
```

è£ŸèČ¡éĂŽè£Ğä¡£çŤÍlambdaèąÍè¿¿åijŔäžčçăĄæÌěěőľèő£éŮőåĞ¡æŢřçŽĎè£ŤåŻđäÿ■åŘŇçŽĎèő¿åőŽå

```
@attach_wrapper(wrapper)
def get_level():
    return level

# Alternative
wrapper.get_level = lambda: level
```

äÿĂäÿłæŕŤè¿ČéŽ¿çŘĘèğčçŽĎåIJřæŰźåřśæŸŕåŕźäžŐèő£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)

```
# Attach adjustable attributes
wrapper.level = level
wrapper.logmsg = logmsg
wrapper.log = log
```

è£ŹäÿłæŰźæşŢäź§åŔŕèČ¡æ■čåÿÿåůěä¡IJïijŇä¡ĘåĽ■æŔŘæŸŕåőČå£Ěéążæ
åęĆæđIJåőČçŽĎäÿŁéİćè£ŸæIJĽåŔęåď ŰçŽĎèčĚéěřåŹÍ(æŕŤåęĆäÿŁéİćæŔŘåĽřçŽĎ
@timethis ä¿Ňå■Ř)ïijŇéĆčäźĹåőČäijŽéŽŘèŮŔåžŢåśĆåśđæĂğïijŇä¡£å¿Ůä£őæŤźåőČäžňæšąæIJĽäżżä¡Ţ
èĂŇéĂŽè£Ğä¡£çŤĺèő£éŮőåĞ¡æŢřåŕśèČ¡éĄ£åĚ■è£ŹæăůçŽĎåśĂéŹŘæĂğãĂĆ

æIJĂåŘŐæŔŘäÿĂçĆźïijŇè£ŹäÿĂåřŔèŁĆçŽĎæŰźæąĹäź§åŔŕäżěä¡IJäÿž9.9åřŔèŁĆäÿ■èčĚéěřåŹĺçśżçŽĺ

11.6 9.6 åÿęåŔŕéĂĽåŔĆæŢřçŽĎèčĚéěřåŹĺ

éŮőécŸ

ä¡ăæČşåĘŹäÿĂäÿłèčĚéěřåŹĺïijŇæŮćåŔŕäżěäÿ■äijăåŔĆæŢřçżŹåőČïijŇæŕŤåęĆ
@decorator ïijŇ äź§åŔŕäżěäijăéĂŠåŔŕéĂĽåŔĆæŢřçżŹåőČïijŇæŕŤåęĆ
@decorator(x,y,z) ãĂĆ

èğčåĘşæŰźæąĹ

äÿŃéİćæŸŕ9.5åřŔèŁĆäÿ■æŮěå£ŮèčĚéěřåŹĺçŽĎäÿĂäÿłä£őæŤźçĽĹæIJñïijŽ

```
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)

```
@logged
def add(x, y):
    return x + y

@logged(level=logging.CRITICAL, name='example')
def spam():
    print('Spam!')
```

åŔŕäżěçIJŃåĹřïijŇ`@logged`èčĚéěřåŹĺåŔŕäżěåŘŇæŮűäÿ■åÿęåŔĆæŢřæĹŰåÿęåŔĆæŢřãĂĆ

èőĺèőž

è£ŹéĞŇæŔŘåĹřçŽĎè£ŹäÿłéŮőéćŸåřśæŸŕéĂŽåÿÿæĽĂèŕť çŽĎçijŰçĺŃäÿĂèĞť æĂğéŮőéćŸãĂĆ
å¡ŞæĹŠäžňä¡£çŤĺèčĚéěřåŹĺçŽĎæŮűåĂŹïijŇåď ğéČĺåĹĘçĺŃåžŔåŠŸäźăæČŕäžĘèęĄäźĹäÿ■çżŹåőČäžňäijäéĂ
åĚűåőđäżŐæŁĂæIJŕäÿŁæİěèőšïijŇæĹŠäžňåŔŕäżěåőŽäźĽäÿĂäÿłæĽĂæIJĽåŔĆæŢřéČ¡æŸŕåŔŕéĂĽçŽĎèčĚ

```
@logged()
def add(x, y):
    return x+y
```

ä¡ĘæŸŕïijŇè£Źçğ■åĘŹæşŢåźűäÿ■çňęåŘĹæĹŠäžňçŽĎäźăæČŕïijŇæIJĽæŮűåĂŹçĺŃåžŔåŠŸå£ŸèőŕåŁää
è£ŹéĞŇæĹŠäžňåŘŠä¡ăåśŢçď žäžĘåęĆä¡ŢäžěäÿĂèĞť çŽĎçijŰçĺŃéčŐæăijæİěåŘŇæŮűæżąèűşæšąæIJĽæŃňå

äÿžäžĘçŘĘèğčäžčçăĄæŸŕåęĆä¡Ţåůěä¡IJçŽĎïijŇä¡áéIJĂèęĄéİđåÿÿçE§æĆĽèčĚéěřåŹĺæŸŕåęĆä¡Ţä¡IJçŤ
åŕźäžŐäÿĂäÿłåČŔäÿŃéİćè£ŹæăůçŽĎçőĂå■ŢèčĚéěřåŹĺïijŽ

```
# Example use
@logged
def add(x, y):
    return x + y
```

è£ŹäÿłèřČçŤĺåžŔåĹŮèů§äÿŃéİćç■L'äžůïijŽ

```
def add(x, y):
    return x + y

add = logged(add)
```

è£ŹæŮűåĂŹïijŇèćńèčĚéěřåĞ¡æŢřäijŽèćńå¡ŞåĄŽçňňäÿĂäÿłåŔĆæŢřçŻť æŐëäijäéĂŠçżŹ
`logged`èčĚéěřåŹĺãĂĆ åŻăæ■d'ïijŇ`logged()` äÿ■çŽĎçňňäÿĂäÿłåŔĆæŢřåŕśæŸŕèćńåŇĚèčĚåĞ¡æŢřæIJňè

èĂŇåŕźäžŐäÿĂäÿłäÿŃéİćè£ŹæăůæIJĽåŔĆæŢřçŽĎèčĚéěřåŹĺïijŽ

```
@logged(level=logging.CRITICAL, name='example')
def spam():
    print('Spam!')
```

èřČçŤĺåžŔåĹŮèů§äÿŃéİćç■L'äžůïijŽ

```
def spam():
    print('Spam!')
spam = logged(level=logging.CRITICAL, name='example')(spam)
```

åĹİåğŃèřČçŤĺ `logged()` åĞ¡æŢřæŮűïijŇèćńåŇĚèčĚåĞ¡æŢřåźűæšąæIJĽäijăéĂŠè£ŻæİěãĂĆ åŻăæ■ďåIJĺèčĚéěřåŹĺåĘĚïijŇåőČå£ĚéążæŸřåŘréĂĽçŽĎãĂĆè£ŹäÿłåŔ■è£ĞæİěäijŽè£ńä¡£åĚűäžŰåŔĆæŢřåźűäÿŤïijŇä¡Ęè£ŹäžŻåŔĆæŢřèćńäijăéĂŠè£ŻæİěåŘŐïijŇèčĚéěřåŹĺèęĄè£ŤåŻđäÿĂäÿłæŐěåŘŮäÿĂäÿłåĞ¡æŤäÿžäžĘè£ŹæăůåĄŽïijŇæĹŠäžňä¡£çŤĺäžĘäÿĂäÿłæŁĂåůğïijŇåřśæŸřåĽ'çŤĺ `functools.partial` ãĂĆ åőČäijŽè£ŤåŻđäÿĂäÿłæIJłåőŇåĚĺåĹİåğŃåŇŰçŽĎèĞłèžńïijŇéŹď'äžĘèćńåŇĚèčĚåĞ¡æŢřåď'åŘřäžěåŘĆèĂČ7.8åřŘèŁĆèŐůåŘŰæŻť'åď'Ž `partial()` æŰźæşŢçŽĎç§èèřĘãĂĆ

11.7 9.7 åĹľçŤĺèčĚéěřåŹĺåijžåĹűåĞ¡æŢřäŸŁçŽĎçśžåđŃæčĂæ§ě

éŮőécŸ

ä¡IJäÿžæ§Řçğ■çijŰçÍŃèğĎçžęïijŇä¡ăæČşåIJĺářźåĞ¡æŢřåŘĆæŢřè£ŻèąŇåijžåĹűçśžåđŃæčĂæ§ěãĂĆ

èğčåĘşæŰźæąĹ

åIJÍæijŤçď'žåőđéŹĚäżčçăĄåĽ■ïijŇåĚĹèřť'æŸŐæĹŚäžňçŽĎçŻőæăĞïijŽèČ¡ářźåĞ¡æŢřåŘĆæŢřçśžåđŃè

```
>>> @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'>
>>>
```

äÿŃéİćæŸřä¡£çŤĺèčĚéěřåŹĺæŁĂæIJřæİěåőđçŐř `@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)

```
        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
```

åŘŕäžěçIJŃåĞžè£ŹäýłèčĚéěřåŹĺéİđåýÿçĄţæť'žïijŇæŮćåŘŕäžěæŇĞåőŽæĽ'ĂæIJĽ'åŘĆæŢřçśžåđŃïijŇäź
åźűäýŤåŘŕäžěéĂŽè£Ğä¡■ç¡őæĹŰåĚşéŤőå■ŮæİěæŇĞåőŽåŘĆæŢřçśžåđŃãĂĆäýŃéİćæŸŕä¡£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¿ŃïijŇåijŢåĚěäžEå¿Ĺåď'ŽéĞ■èęĄçŽĎæęĆå£ţãĂĆ

éęŰåĚĹïijŇèčĚéěřåŹĺåŘłäijŽåIJĺåĞ¡æŢřåőŽäźĽ'æŮűèćńěřČçŤĺäÿĂæňąãĂĆ
æIJĽ'æŮűåĂŹä¡ăåŐżæŐĽ'èčĚéěřåŹĺçŽĎåŁ§èČ¡ïijŇéĆčäźĹä¡ăåŘłéIJĂèęĄçőĂå■ŢçŽĎè£ŤåŻđèćńèčĚéěřåĞ¡
äÿŃéİćçŽĎäžčçăĄäÿ■ïijŇåęĆæđIJåĚĺåśĂåŔŸéĞŔãĂĂ__debug__
èćńèő¿ç¡őæĹŘäžĘFalse(å¡Şä¡ăä¡£çŤĺ-OæĹŰ-OOåŘĆæŢřçŽĎäijŸåŇŰæĺąåijŔæĽ'ğèąŇçĺŃåžŔæŮű)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¡æŢřçŽĎåŔĆæŢřç■¿åŘ■è£ŻèąŇäžĘæčĂæ§ěïijŇæĹŚäžňä¡£çŤ
inspect.signature() åĞ¡æŢřãĂĆ çőĂå■ŢæİěëőšïijŇåőČè£ŘèąŇä¡ăæŔŘåŔŰäÿĂäÿłåŔŕěřČçŤĺåŕźèšąę

```
>>> 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'>
>>>
```

èčĚéěřåŹĺçŽĎåijĂåğŃéČĺåĹĘïijŇæĹŚäžňä¡£çŤĺäžĘ bind_partial()
æŰźæşŢæİěæL'ğèąŇäżŐæŇĞåőŽçśžåđŃåĹřåŘ■çğřçŽĎéČĺåĹĘçżŚåőŽãĂĆ
äÿŃéİćæŸŕä¿Ńå■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¡ăåŔŕäžěæşĺæĎŔåĹřçijžåď'śçŽĎåŔĆæŢřèćńå£¡çŢëäžĘ(æŕŤåęĆåźűæš
äÿ■è£ĞæIJĂéĞ■èęĄçŽĎæŸŕåĹŻåżžäžĘäÿĂäÿłæIJL'åžŔå■ŮåĚÿ bound_types.
arguments ãĂĆ è£ŹäÿłåŮåĚÿäijŽåřĘåŔĆæŢřåŘ■äžěåĞ¡æŢřç■¿åŘ■äÿ■çŽÿåŘŇéąžåžŔæŸăåřĎåĹřæŇĞå
åIJĺæĹŚäžňçŽĎèčĚéěřåŹĺä¿Ńå■Řäÿ■ïijŇè£ŹäÿłæŸăåřĎåŇĚåŘńäžĘæĹŚäžňèęĄåijžåĹűæŇĞåőŽçŽĎçśžåđŃ

åIJĺèčĚéěřåŹĺåĹŻåżžçŽĎåőđéŹĚåŇĚèčĚåĞ¡æŢřäÿ■ä¡£çŤĺåŕžĘ
sig.bind() æŰźæşŢãĂĆ bind() èů§ bind_partial()
çśžäijijïijŇä¡ĘæŸŕåőČäÿ■åĚĄèöÿå£¡çŢëäżžä¡ŢåŔĆæŢřãĂĆåŻăæ■d'æIJL'äžĘäÿŃéİćçŽĎçżŞæđIJïijŽ

```
>>> bound_values = sig.bind(1, 2, 3)
>>> bound_values.arguments
OrderedDict([('x', 1), ('y', 2), ('z', 3)])
```

(continues on next page)

```
>>>
```

ä¡£çŤÍè£ŹäÿłæŸăåřĎæĹŚäžňåŘŕäžěå¿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£ŸæIJL'çĆźåřŘçŚŢçŰţïijŇåőČåŕźäžŐæIJL'éżŸèőď'åĂijçŽĎåŘĆæŢřåźűäÿ■éĂĆæŕŤåęĆäÿŃéİćçŽĎäžčçăĄåŘŕäžěæ■čåÿÿåůěä¡IJïijŇår¡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]
>>>
```

æIJĂåŘŐäÿĂçĆźæŸŕåĚşäžŐéĂĆçŤÍèčĚéěřåŹĺåŘĆæŢřåŠŇåĞ¡æŢřæşÍèğčäźŃéŮť'çŽĎäžL'èőžãĂĆ
ä¿ŃåęĆïijŇäÿžäżĂäźĹäÿ■åČŔäÿŃéİćè£ŹæăůåĘŹäÿĂäÿłèčĚéěřåŹĺåÍèæ§ěæL'¿åĞ¡æŢřäÿ■çŽĎæşÍèğčåŚćïij§

```
@typeassert
def spam(x:int, y, z:int = 42):
    print(x,y,z)
```

äÿĂäÿłåŔŕèČ¡çŽĎåŐ§åŻăæŸŕåęĆæđIJä¡£çŤÍäžĘåĞ¡æŢřåŘĆæŢřæşÍèğčïijŃéĆčäźĹåřśèćńéŹŘåĹűäžĘãĄ
åęĆæđIJæşÍèğčèćńçŤÍæÍěåĄŽçśżåđŃæčĂæ§ěåřśäÿ■èČ¡åĄŽåĚűäžŰäžŃæČĚäžĘãĂĆèĂŃäÿŤ
`@typeassert` äÿ■èČ¡åĘ■çŤÍäžŐä¡£çŤÍæşÍèğčåĄŽåĚűäžŰäžŃæČĚçŽĎåĞ¡æŢřäžĘãĂĆ
èĂŃä¡£çŤÍäÿŁéİćçŽĎèčĚéěřåŹĺåŘĆæŢřçĄţæť'żæĂgåď'gåď'ŽäžĘïijŇäź§æŻť'åŁäéĂŽçŤÍãĂĆ

åŔŕäžěåIJÍPEP 362äžěåŘŁ `inspect` æÍąåİŮäÿ■æL'¿åĹřæŻť'åď'ŽåĚşäžŐåĞ¡æŢřåŘĆæŢřåŕźèśąçŽĎä£ą

11.8 9.8 åřĘèčĚéěřåŹĺåőŽäźĽäÿžçśżçŽĎäÿĂéČĺåĹĘ

éŮőécŸ

ä¡äæČşåIJĺçśżäÿ■åőŽäźĽ'èčĚéěřåŹĺïijŇåźűåřĘåĚűä¡IJçŤĺåIJĺåĚűäżŰåĞ¡æŢřæĹŰæŰźæşŢäÿŁãĂĆ

èğčåĘşæŰźæąĹ

åIJĺçśżéĞŇéİćåőŽäźĽ'èčĚéěřåŹĺå¿ĹçőĂå■ŢïijŇä¡ĘæŸřä¡äéęŰåĚĹèęĄçąőèőd'åőČçŽĎä¡£çŤĺæŰźåijŔãÁäÿŃéİćæĹŚäżňçŤĺä¿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
```

äÿŃéİćæŸřäÿĂä¡£çŤĺä¿Ńå■ŘïijŽ

```
# As an instance method
a = A()
@a.decorator1
def spam():
    pass
# As a class method
@A.decorator2
def grok():
    pass
```

äžŤçżĘèğĆåŕ§åŘŕäžěåŘŚçŐřäÿĂäÿłæŸřåőđä¿ŃěřČçŤĺïijŇäÿĂäÿłæŸŕçśżèřČçŤĺãĂĆ

èőÍèőž

åIJÍçśžäÿ■åőŽäźL'èčĚéěřåŹÍåĹİçIJŃäÿŁåŐżåě¡åČŔå¿ĹåěĞæĂłïijŇä¡ĘæŸŕåIJÍæăĞåĞĘåžŞäÿ■æIJL'å¿Ĺ
çL'źåĹńçŽĎïijŇ`@property` èčĚéěřåŹÍåőđéŹĚäÿŁæŸŕäÿĂäÿłçśžïijŇåőČéĞŃéİćåőŽäźL'äžĘäÿL'äÿłæŰźæşŢ
`getter(), setter(), deleter()`,æŕŔäÿĂäÿłæŰźæşŢéČ¡æŸŕäÿĂäÿłèčĚéěřåŹÍãĂĆä¿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§åŽăæŸŕåŘĎçğ■äÿ■åŘŇçŽĎèčĚéěřåŹÍæŰźæşŢäijŽåI
`property` åőđä¿ŃäÿŁæŞ■ä¡IJåőČçŽĎçŁűæĂĄãĂĆ åŻăæ■đ'ïijŇäżżä¡ŢæŮűåĂŹåŔłèęĄä¡ăçćřåĹřéIJĂèęĄå

åIJÍçśžäÿ■åőŽäźL'èčĚéěřåŹÍæIJL'äÿłéŽ¿çŘĘèğčçŽĎåIJřæŰźåřśæŸŕåŕźäžŐéćÍåđ'ŰåŔĆæŢř
`self` æĹŰ `cls` çŽĎæ■čçąőä¡£çŤÍãĂĆ åř¡çőąæIJĂåđ'ŰåśĆçŽĎèčĚéěřåŹÍåĞ¡æŢřæŕŤåęĆ
`decorator1()` æĹŰ `decorator2()` éIJĂèęĄæŔŘä¿ŻäÿĂäÿł `self`
æĹŰ `cls` åŔĆæŢřïijŇ ä¡ĘæŸŕåIJÍäÿđ'äÿłèčĚéěřåŹÍåĘĚéČÍèćńåĹŻåżžçŽĎ
`wrapper()` åĞ¡æŢřåźűäÿ■éIJĂèęĄåŇĚåŘńè£ŹäÿŁ `self` åŔĆæŢřãĂĆ
ä¡ăåŤŕäÿĂéIJĂèęĄè£ŹäÿłåŔĆæŢřæŸŕåIJÍä¡ăçąőåőđèęĄèő£éŮőåŇĚèčĚåŹÍäÿ■è£Źäÿłåőđä¿ŃçŽĎæ§ŘäžŽéČ

åŕźäžŐçśžéĞŇéİćåőŽäźL'çŽĎåŇĚèčĚåŹÍè£ŸæIJL'äÿĂçĆźæŕŤè¿ČéŽ¿çŘĘèğčïijŇåřśæŸŕåIJÍæűL'åŘŁåÍ
ä¿ŃåęĆïijŇåĄĞèő¿ä¡ăæČşèőľ'åIJÍAäÿ■åőŽäźL'çŽĎèčĚéěřåŹÍä¡IJçŤÍåIJÍå■ŘçśžBäÿ■ãĂĆä¡ăéIJĂèęĄåČŔäÿŇ

```
class B(A):
    @A.decorator2
    def bar(self):
        pass
```

äź§åřśæŸŕèŕť'ïijŇèčĚéěřåŹÍèęĄèćńåőŽäźL'æĹŘçśžæŰźæşŢåźűäÿŤä¡ăå£ĚéąžæŸ¿åijŘçŽĎä¡£çŤÍçĹűçśż
ä¡ăäÿ■èČ¡ä¡£çŤÍ `@B.decorator2` ïijŇåŽăäÿžåIJÍæŰźæşŢåőŽäźL'æŮűïijŇè£ŹäÿłçśžBè£ŸæšąæIJL'èćńåĹŻ

11.9 9.9 åřĘèčĚéěřåŹÍåőŽäźL'äÿžçśż

éŮőécŸ

ä¡ăæČşä¡£çŤÍäÿĂäÿłèčĚéěřåŹÍåŐżåŇĚèčĚåĞ¡æŢřïijŇä¡ĘæŸŕåÿŇæIJŻè£ŤåŽđäÿĂäÿłåŔŕèřČçŤÍçŽĎåő
ä¡ăéIJĂèęĄèőľ'ä¡ăçŽĎèčĚéěřåŹÍåŔŕäžěåŘŇæŮűåůěä¡IJåIJÍçśžåőŽäźL'çŽĎåĘĚéČÍåŠŇåđ'ŰéČÍãĂĆ

èğčåĘşæŰźæąĹ

äÿžäžĘåřĘèčĚéěřåŹĺåőŽäźL’æĹŘäÿĂäÿłåőđä¿ŃïijÑä¡ăéIJĂèęĄçąőä£ÌåőČåőđçŐřäžĘ
__call__() åŠŇ __get__() æŰźæşŢãĂĆ ä¿ŃåęĆïijÑäÿŃéÌćçŽĎäžčçăĄåőŽäźL’äžĘäÿĂäÿłçśżïijÑåőČå

```
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¡ăåŘŕäžěåřĘåőČå¡ŞåĄŽäÿĂäÿłæŹőéĂŽçŽĎèčĚéěřåŹĺæÌěä¡£çŤĺïijÑåIJĺçśżéĞŇéÌćæĹŰåď Ůéİćéč¡åŕŕ

```
@Profiled
def add(x, y):
    return x + y

class Spam:
    @Profiled
    def bar(self, x):
        print(self, x)
```

åIJĺäžď’äžŠçŐřáćČäÿ■çŽĎä¡£çŤĺçď’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

åřĘèčĚéěřåŹÍåőŽäźL’æĹŘçśżéĂŽåÿÿæŸŕå¿ĹçőĂå■ŢçŽĎãĂĆä¡ĘæŸŕè£ŹéĞŇè£ŸæŸŕæIJL’äÿĂäžŽçżĘ

éęŰåĚĹïijŇä¡£çŤÍ `functools.wraps()` åĞ¡æŢřçŽĎä¡IJçŤÍèů§äźŃåL’■è£ŸæŸŕäÿĂæăůïijŇåřĘèćńå

åĚűæňąïijŇéĂŽåÿÿå¿Ĺåőźæ ŸŞäijŽå£¡èğĘäÿŁéİćçŽĎ `__get__()` æŰźæşŢãĂĆåęĆæđIJä¡ăå£¡çŢěåőČïijŇä£ÌæŇĄåĚűäžŰäžčçăĄäÿ■åŔŸåĘ■æňąè£ŘèąŇïijŇ ä¡ăäijŽåŔŞçŐřå¡Şä¡ăåŐżèřČçŤÍèćńčĚéěřåőđä¿ŃæŰźæşŢæŮűåĞžçŐřå¿ĹåěĞæĂłçŽĎéŮőécŸãĂĆä¿ŃåęĆï

```
>>> s = Spam()
>>> s.bar(3)
Traceback (most recent call last):
...
TypeError: bar() missing 1 required positional argument: 'x'
```

åĞžéŤŹåŐ§åŽăæŸŕå¡ŞæŰźæşŢåĞ¡æŢřåIJÍäÿĂäÿłçśżäÿ■èćńæ§ěæL’¿æŮűïijŇåőČäžňçŽĎ `__get__()` æŰźæşŢä¿Íæ■őæŔŘè£řåŹÍå■ŘèőőèćńěřČçŤÍïijŇ åIJÍ8.9åřŘèŁĆåůšçżŘèőšè£řè£ĞæŔŘè£řåŹÍå■Řèőőäž ĘãĂĆåIJÍè£ŹéĞŇïijŇ`__get__()` çŽĎçŽőçŽĎæŸŕåĹŽåžžäÿĂäÿłçżŚåőŽæŰźæşŢårźèśą (æIJĂçżĹäijŽçżŹè£ŹäÿłæŰźæşŢäijåéĂŠselfåŔĆæŢř)ă

```
>>> s = Spam()
>>> def grok(self, x):
...     pass
...
>>> grok.__get__(s, Spam)
<bound method Spam.grok of <__main__.Spam object at 0x100671e90>>
>>>
```

`__get__()` æŰźæşŢæŸŕäÿžäžĘçąőä£ÍçżŚåőŽæŰźæşŢårźèśąèČ¡èćńæ■čçąőçŽĎåĹŽåžžãĂĆ `type.MethodType()` æĽŃåŁĺåĹŽåžžäÿĂäÿłçżŚåőŽæŰźæşŢæÍěä¡£çŤÍãĂĆåŔłæIJL’å¡Şåőđä¿Ńèćńä¡£çŤ åęĆæđIJè£ŹäÿłæŰźæşŢæŸŕåIJÍçśżäÿŁéİćæÍěèő£éŮőïijŇ éĆčäźĹ `__get__()` äÿ■çŽĎinstanceåŔĆæŢřäijŽèćńèő¿ç¡őæĹŘNoneåźűçŻť’æŐěè£ŤåŻđ `Profiled` åőđä¿ŃæIJňèžńãĂĆè£ŹæåůçŽĎèřÍæĹŚäžňářśåŔřäžěæŔŘåŔŰåőČçŽĎ `ncalls` åśđæĂğäžĘãĂĆ

åęĆæđIJä¡ăæČşéĄ£åĚ■äÿĂäžŻæůůäźśïijŇäź§åŔřäžěèĂČèŹŚåŔęåď’ŰäÿĂäÿłä¡£çŤÍéŮ■åŇĚåŠŇ `nonlocal` åŔŸéĞŘåőđçŐřçŽĎèčĚéěřåŹÍïijŇè£ŹäÿłåIJÍ9.5åřŘèŁĆæIJL’èőšåĹřãĂĆä¿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)

```
# Example
@profiled
def add(x, y):
    return x + y
```

è£ŹäÿłæŰźåijŔèů§äźŃåL’■çŽĎæŢĹæđIJåĞääźŐäÿĂæăůïijŇéŹď’äžĘåŕźäžŐ `ncalls` çŽĎèő£éŮőçŐřåIJÍæŸŕéĂŽè£ĞäÿĂäÿłèćńçżŚåőŽäÿžåšđæĂğçŽĎåĞ¡æŢřæÌěåőđçŐřïijŇä¿ŃåęĆïijŽ

```
>>> add(2, 3)
5
>>> add(4, 5)
9
>>> add.ncalls()
2
>>>
```

11.10 9.10 äÿžçśżåŠŇéİŹæĂĄæŰźæşŢæŔŘä¿ŻèčĕéěřåŹÍ

éŮőécŸ

ä¡ăæČşçżŹçśżæĹŰéİŹæĂĄæŰźæşŢæŔŘä¿ŻèčĕéěřåŹÍãĂĆ

èğčåĘşæŰźæąĹ

çżŹçśżæĹŰéİŹæĂĄæŰźæşŢæŔŘä¿ŻèčĕéěřåŹÍæŸŕå¿ÍçőĂå■ŢçŽĎïijŇäÿ■è£ĞèęAçąőä£ÌèčĕéěřåŹÍåIJ `@classmethod` æĹŰ `@staticmethod` äźŃåL’■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)

```
        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¡IJïijŇåŔłäÿ■è£Ğåćđå£ääžĘéćİåď ŰçŽĎèőąæŮ

```
>>> 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¿ŇåęĆïijŇåĄĞèő¿ä¡ääČŔäÿŇéİć

```
class Spam:
    @timethis
    @staticmethod
    def static_method(n):
        print(n)
        while n > 0:
            n -= 1
```

éĆčäźĹä¡ăèřČçŤÍè£ŹäÿłéİŹæĂĄæŰźæşŢæŮűůåřśäijŽæŁěéŤŹïijŽ

```
>>> Spam.static_method(1000000)
Traceback (most recent call last):
```

(continues on next 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 åőđéŹĚäýŁåźűäý■äijŽåĹŻåžžåŔŕçŻť'æŐěěřČçŤĺçŽĎåŕźèšąïijŇ
èĂŇæŸŕåĹŻåžžçĽźæőŁçŽĎæŔŔè£ŕåŹĺåŕźèšą(åŔĆèĂČ8.9åřŔèŁĆ)ãĂĆåŻäæ■d'å¡Şä¡äèŕŢçİĂåIJĺåĚűäžŰèč
çąőä£Ìè£Źçğ■èčĚéěřåŹĺåĞžçŐřåIJÍèčĚéěřåŹĺéŞ¿äý■çŽĎçňňäýĂäýłä¡■ç¡őåŔŕäžěä£őåd'■è£ŹäýłéŮőécŸãĂĆ

å¡ŞæĹŚäžňåIJĺæŁ¡èšąå§žçšżäý■åőŽäźL'çšżæŰźæşŢåŠŇéİŹæĂĄæŰźæşŢ(åŔĆèĂČ8.12åřŔèŁĆ)æŮűïijŇ
ä¿ŇåęĆïijŇåęĆæđIJä¡äæČşåőŽäźL'äýĂäýłæŁ¡èšąçšżæŰźæşŢïijŇåŔŕäžěä¡£çŤĺçżäijijäýŇéİćçŽĎäžčçăĄïijŽ

```
from abc import ABCMeta, abstractmethod
class A(metaclass=ABCMeta):
    @classmethod
    @abstractmethod
    def method(cls):
        pass
```

åIJÍè£Źæőţäžčçă Ąäý■ïijŇ@classmethod èů§ @abstractmethod
äýd'èĂĚçŽĎéąžäžŔæŸŕæIJL'èőšçl'űçŽĎïijŇåęĆæđIJä¡äèřČæ■ćåőČäžňçŽĎéąžäžŔåřśäijŽåĞžéŤŹãĂĆ

11.11 9.11 èčĚéěřåŹĺäýžèćńåŇĚèčĚåĞ¡æŢřåćđåŁäåŔĆæŢř

éŮőécŸ

ä¡äæČşåIJÍèčĚéěřåŹĺäý■çżŹèćńåŇĚèčĚåĞ¡æŢřåćđåŁäéćÍåd'ŰçŽĎåŔĆæŢřïijŇä¡ĘæŸŕäý■èČ¡å¡śåŞ■è£Ź

èğčåĘşæŰźæąĹ

åŔŕäžěä¡£çŤĺåĚşéŤőå■ŮåŔĆæŢřæÍěçżŹèćńåŇĚèčĚåĞ¡æŢřåćđåŁäéćÍåd'ŰåŔĆæŢřãĂĆèĂČèŹŚäýŇéÍć

```
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■đ'ïijŇæIJL'æŮűåĂŹåőČåŔřäžěéĄ£åĚ■äÿĂäžŽéĞ■åđ'■äžčçăĄãĂĆä¿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¡ăåŔřäžěářĘåĚűéĞ■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)

```
@optional_debug
def b(x, y, z):
    pass

@optional_debug
def c(x, y):
    pass
```

è£Źçğ■åőđçŐřæŰźæąĹäźŃæĽ Ăäżěèą Ňå¿ŮéĂŽïijŇåIJĺäžŐåijžåĹűåĚşéŤőå■ŮåŔĆæŢřå¿Ĺåőźæ ŸŞèćńa
`*args` åŠŇ `**kwargs` åŔĆæŢřçŽĎåĞ¡æŢřäÿ■ãĂĆ éĂŽè£Ğä¡£çŤĺåijžåĹűåĚşéŤőå■ŮåŔĆæŢřïijŇåőČèćńä
åźűäÿŤæŐěäÿŃæİěäżĚäżĚä¡£çŤĺåĽľä¡ŹçŽĎä¡■ç¡őåŠŇåĚşéŤőå■ŮåŔĆæŢřåŐżèřČçŤĺè£Źäÿłå Ğ¡æŢřæŮűïij
äźş åřśæŸŕèŕťïijŇåőČåźűäÿ■äijŽèćńçžşåĚěåĹř `**kwargs` äÿ■åŐżãĂĆ

è£ŸæIJL'äÿĂäÿłéŽ¿çĆźåřśæŸŕåęĆä¡ŢåŐżåď ĎçŘĘèćńæůżåŁăçŽĎåŔĆæŢřäÿŐèćńåŇĚèčĚåĞ¡æŢřåŔĆ
ä¿ŃåęĆïijŇåęĆæđIJèčĚéěřåŹÍ `@optional_debug` ä¡IJçŤĺåIJĺäÿĂäÿłåůšçżŔæŃěæIJL'äÿĂäÿł
`debug` åŔĆæŢřçŽĎåĞ¡æŢřäŸŁæŮűäijŽæIJL'éŮőécŸãĂĆ è£ŹéĞŇæĹŚäżňáćđåŁääžĘäÿĂæ■ěåŘ■å■ŮæčĂæ

äÿŁéİćçŽĎæŰźæąĹè£ŸåŔŕäżěæŻť åőŇç¿ŐäÿĂçĆźïijŇåŻääÿžçš¿æŸŐçŽĎçÍŃåžŔåŚŸåžŤèŕěåŔŚçŐřäž

```
>>> @optional_debug
... def add(x,y):
...     return x+y
...
>>> import inspect
>>> print(inspect.signature(add))
(x, y)
>>>
```

éĂŽè£ĞåęĆäÿŃçŽĎä£őæŤźïijŇåŔŕäżěèğčåĘşè£ŹäŷłéŮőécŸïijŽ

```
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¡

éŮőécŸ

ä¡ăæČşéĂŽè£ĞåŘ■çIJĄæÍŰèĂĚéĞ■åĘŹçśżåőŽäźL'çŽĎæ§ŘéČĺåĹĘæİěä£őæŤźåőČçŽĎèąŇäÿžïijŇä¡E

èğčåĘşæŰźæąĹ

è£Źçğ■æČĚåĘţåŘŕèČ¡æŸŕçśżèčĚéěřåŹÍæIJĂåě¡çŽĎä¡£çŤÍåIJžæŹŕäžĘãĂĆä¿ŃåęĆïijŇäÿŃéİćæŸŕäÿĂä __getattribute__ çŽĎçśżèčĚéěřåŹÍïijŇ åŘŕäžěæĽŞå■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
```

äÿŃéİćæŸŕä¡£çŤÍæŢĹæđIJïijŽ

```
>>> a = A(42)
>>> a.x
getting: x
42
>>> a.spam()
getting: spam
>>>
```

èőÍèőž

çśżèčĚéěřåŹÍéĂŽåÿÿåŘŕäżěä¡IJäÿžåĚűäżŰéńŸçžğæŁĂæIJŕæŕŢåęĆæůůåĚěæĹŰåĚČçśżçŽĎäÿĂçğ■éİđă
æŕŢåęĆïijŇäÿŁéİćçđ'žä¿Ńäÿ■çŽĎåŔęåđ'ŰäÿĂçğ■åőđçŐřä¡£çŤĺåĹřçżğæL'£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¡ăåřśå£Ěéążç§ěéĄŞæŰźæşŢèřČ
äžěåŔŁåĚűåőČ8.7åřŔèŁĆäżŃçż■çŽĎçżğæL'£ç§ěèřĘãĂĆ æ§Řçğ■çÍŃåžęäÿŁæİèèőšïijŇçśżèčĚéěřåŹÍæŰźæą
åŻääÿžäžŰåźűäÿ■ä¿İèţŰ `super()` åĞ¡æŢřãĂĆ

åęĆæđIJä¡ăçşżæČşåIJÍäÿĂäÿłçśżäÿŁéİćä¡£çŤĺåd'ŽäÿłçśżèčĚéěřåŹÍïijŇéĆčäźĹåřśéIJĂèęĄæşÍæĎŔäÿŃé
ä¿ŃåęĆïijŇäÿĂäÿłèčĚéěřåŹÍAäijŽåřĘåĚűèčĚéěřçŽĎæŰźæşŢåőŇæŢt'æŻ£æ■ćæĹŘåŔęäÿĂçğ■åőđçŐřïijŇ
èĂŇåŔęäÿĂäÿłèčĚéěřåŹÍBåŘłæŸŕçőĂå■ŢçŽĎåIJÍåĚűèčĚéěřçŽĎæŰźæşŢäÿ■æůżåŁăçĆźéĆİåd'ŰéĂżè¿ŚãĂ
éĆčäźĹè£ŹæŮűåĂŹèčĚéěřåŹÍAåřśéIJĂèęĄæŤ¿åIJÍèčĚéěřåŹÍBçŽĎåL'■éİćãĂĆ

ä¡ăè£ŸåŔŕäžěåŻđéą¿äÿĂäÿŃ8.13åřŔèŁĆåŔęåd'ŰäÿĂäÿłåĚşäžŐçśżèčĚéěřåŹÍçŽĎæIJL'çŤÍçŽĎä¿Ńå■Ř

11.13 9.13 ä¡£çŤĺåĚČçśżæŐğåĹűåőđä¿ŃçŽĎåĹŻåžž

éŮőécŸ

ä¡ăæČşéĂŽè£ĞæŤźåŔŸåőđä¿ŃåĹŻåžžæŰźåijŔæİèåőđçŐřå■Ţä¿ŃãĂĄçijŞå■ŸæĹŰåĚűäžŰçśżäijijçŽĎ

èğčåĘşæŰźæąĹ

PythonçÍŃåžŔåŚŸéČ¡ç§ěéĄŞïijŇåęĆæđIJä¡ăåőŽäźL'äžĘäÿĂäÿłçśżïijŇåřśèČ¡åČŔåĞ¡æŢřäÿĂæăůçŽĎè

```
class Spam:
    def __init__(self, name):
        self.name = name

a = Spam('Guido')
b = Spam('Diana')
```

åęĆædIJä¡ăæČşèĞłåőŽäźL'è£Źäÿłæ■ěéłd'ïijŇä¡ăåŔŕäžěåőŽäźL'äÿĂäÿłåĚČçśżåźűèĞłåůśåődçŐř __call__() æŰźæşŢãĂĆ

äÿžäžĘæijŤçd'žïijŇåĄĞèő¿ä¡ääÿ■æČşäżżä¡Ţäžžå ĹŻåżžè£Źäÿłçśżç ŽĎåődä¿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£ŹæăůçŽĎèŕÍïijŇçŤÍæĹůåŔłèČ¡èřČçŤÍè£Źäÿłçśżç ŽĎéİŹæĂĄæŰźæşŢïijŇèĂŇäÿ■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¡ăæČşåődçŐřå■Ţä¿ŃæÍąåijŔïijĹåŔłèČ¡åĹŻåżžåŤŕäÿĂåődä¿ŃçŽĎçśżïijL'ïijŇåődçĆ

```
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)

```
    def __init__(self):
        print('Creating Spam')
```

éĆčäźĹSpamçśżåřśåŘłèČ¡åĹŻåżžåŤŕäýĂçŽĎåőđä¿ŃäžĘïijŇæijŤçd’žåęĆäýŃïijŽ

```
>>> a = Spam()
Creating Spam
>>> b = Spam()
>>> a is b
True
>>> c = Spam()
>>> a is c
True
>>>
```

æIJĂåŘŐïijŇåĄĞèő¿ä¡ăæČşåĹŻåżž8.25åřŔèŁĆäý■éĆčæăůçŽĎçijŞå■Ÿåőđä¿ŃãĂĆäýŃéİćæĹŚäžňåŔŕä

```
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§æİěæţŃèŕŢäýĂäýŃïijŽ

```
>>> 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

åĹľçŤĺåĚČçśżåőđçŐřåď'Žçğ■åőđä¿ŃåĹŻåżžæĺąåijŔéĂŽåÿÿèęĄæŕŤäÿ■ä¡£çŤĺåĚČçśżçŽĎæŰźåijŔäijŸ

åĄĞèő¿ä¡ääÿ■ä¡£çŤĺåĚČçśžïijÑä¡ăåŔřèČ¡éIJĂèęĄåřEçśżéŽŘèŮŔåIJÍæ§ŘäžŽåůěåŐĆåĞ¡æŢřåŘŐéİćãÄ
æŕŤåęĆäÿžäžĘåőđçŐřäÿĂäÿłå■Ţä¿ŃïijÑä¡ää¡ăåŔřèČ¡äijŽåČŔäÿŃéİćè£ŹæăůåĘŹïijŽ

```
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
```

ař¡çőąä¡£çŤĺåĚČçśżåŔřèČ¡äijŽæűĽåŘŁåĹřæŕŤè¿ČéńŸçžğçĆźçŽĎæŁĂæIJŕïijÑä¡ĘæŸŕåőČçŽĎäžčçăĄ

æŻť'åď'ŽåĚşäžŐåĹŻåżžçijŞå■Ÿåőđä¿ÑãĂĄåijśåijŢçŤĺç■L'åĘĚåőźïijÑèŕůåŔĆèĂČ8.25åřŔèŁĆãĂĆ

11.14 9.14 æ■ŢèŐůçśżçŽĎåśđæĂğåőŽäźĽéąžåžŔ

éŮőécŸ

ä¡ăæČşèĞłåŁĺèőřå¡ŢäÿĂäÿłçśžäÿ■åśđæĂğåŠÑæŰźæşŢåőŽäźĽçŽĎéąžåžŔïijÑ
çĎűåŘŐåŔŕäżčåĹľçŤĺåőČæİěåĄŽå¿Ĺåď'ŽæŞ■ä¡IJïijĹæŕŤåęĆåžŔåĹŮåŇŰãĂĄæŸăåřĎåĹřæŢřæ■őåžŞç■L'ç

èğčåĘşæŰźæąĹ

åĹľçŤĺåĚČçśżåŔŕäżčå¿Ĺåőźæ ŸŞçŽĎæ■ŢèŐůçśżçŽĎåőŽäźĽ'ä£ąæĄřãĂĆäÿŃéİćæŸŕäÿĂäÿłä¿Ńå■ŘïijÑ

```
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)

```
        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£Źäÿ łåĚČçśżäÿ■ïijŇæL’ğèąŇçśżäÿżä¡ŞæŮűæŔŔè£řåŹÍçŽĎåőŽäźL’éąžåžŔäijŽèćńäÿĂäÿł
OrderedDict``æ■ŢèŐůåĹřïijŇ çŤŞæĹŘçŽĎæIJL’åžŔåŘ■çğřäżŐå■ŮåĚÿäÿ■æŔŘåŔŰåĞžæİě
``_order äÿ■ãĂĆè£ŹæăůçŽĎèŕÍçśżäÿ■çŽĎæŰźæşŢåŔŕäżěéĂŽè£Ğåď’Žçğ■æŰźåijŔæİěä¡£çŤĺåőČãĂĆ
ä¿ŇåęĆïijŇäÿŃéÍćæŸŕäÿĂäÿłçőĂå■ŢçŽĎçśżïijŇä¡£çŤÍè£ŹäÿłæŐŠåžŔå■ŮåĚÿæİěåőđçŐřårĘäÿĂäÿłçśżåőđă

```
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
```

æĹŚäżňåIJÍäžd’äžŠåijŔçŐŕåćČäÿ■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

æIJňèŁĆäÿĂäÿłåĚşéŤőçĆźåřśæŸŕŕOrderedMetaåĚČçśżäÿ■åőŽäźL’çŽĎ ‘‘ __prepare__()‘‘ æŰźæşŢãĂĆ è£ŹäÿłæŰźæşŢäijŽåIJĺåijĂåğŃåőŽäźL’çśżåŠŇåőČçŽĎçĹűçśżçŽĎæŮűåĂŹèćńæL’ğè æĹŚäžňè£ŹéĞŇéĂŽè£Ğè£ŤåŻđäžĘäÿĂäÿłOrderedDictèĂŇäÿ■æŸŕäÿĂäÿłæŹőéĂŽçŽĎå■ŮåĚÿïijŇåŔŕäžěå

åęĆæđIJä¡ăæČşæđĎéĂăèĞłåůśçŽĎçśżå■ŮåĚÿåŕźèśąïijŇåŔŕäžěå¿Ĺåőźæ ŸŞçŽĎæL’l’åśŢè£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)
```

äÿŃéÍćæĹŚäžňæţŃèŕŢéĞ■åď■çŽĎåőŽäźL’äijŽåĞžçŐřäžĂäźĹæČĚåĘţïijŽ

```
>>> class A(metaclass=OrderedMeta):
...  def spam(self):
...  pass
...  def spam(self):
...  pass
```

(continues on next 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■èęĄïijŇåřśæŸŕåIJĺ `__new__()` æŰźæşŢäÿ■åŕźäžŐåĚČçśżäÿ■èćńä£őæŤźå■ŮåĚÿçŽĎåď'ĎçŘĘãĂĆ åř¡çőąçśżä¡£çŤĺäžĘåŔęåď'ŰäÿĂäÿłå■ŮåĚÿæİěåőŽäźL'ïijŇåIJĺæđĎéĂăæIJĂçźĹçŽĎ `class` åŕźèśąçŽĎæŮűåĂŹïijŇ æĹŚäżňäž■çĎűéIJĂèęĄåřĘè£Źäÿłå■ŮåĚÿè¡ňæ■ćäÿžäÿĂäÿłæ■čçąőçŽĎ `dict` åőđä¿ŃãĂĆ éĂŽè£ĞèŕåŔě `d = dict(clsdict)` æİěåőŇæĹŘè£ŹäÿłæŢĹæđIJãĂĆ

åŕźäžŐå¿Ĺåď'ŽåžŤçŤĺçĺŃåžŔèĂŇåůšïijŇèČ¡åď'§æ■ŢèŐůçśżåőŽäźL'çŽĎéąžåžŔæŸŕäÿĂäÿłçIJŃäijijäÿ■ä¿ŃåęĆïijŇåIJĺåŕźèśąåĚşçşżæŸăåřĎäÿ■ïijŇæĹŚäżňéĂŽåÿÿäijŽçIJŃåĹřäÿŃéİćè£Źçğ■æŰźåijŔåőŽäźL'çŽĎç

```
class Stock(Model):
    name = String()
    shares = Integer()
    price = Float()
```

åIJĺæąĘæđűåžŢåśĆïijŇæĹŚäżňå£Ěéążæ■ŢèŐůåőŽäźL'çŽĎéąžåžŔæİěåřĘåŕźèśąæŸăåřĎåĹřåĚČçżĎæĹŰ `as_csv()` çŽĎåŁ§èČ¡ïijL'ãĂĆ è£ŹèŁĆæijŤçď'žçŽĎæŁĂæIJŕéİđåÿÿçőĂå■ŢïijŇåźűäÿŤéĂŽåÿÿäijŽæŕŤåĚű

11.15 9.15 åőŽäźL'æIJL'åŔŕéĂL'åŔĆæŢřçŽĎåĚČçśż

éŮőécŸ

ä¡ăæČşåőŽäźL'äÿĂäÿłåĚČçśżïijŇåĚĄèőÿçśżåőŽäźL'æŮűæŔŘä¿ŻåŔŕéĂL'åŔĆæŢřïijŇè£ŹæăůåŔŕäžěæĆ

èğčåĘşæŰźæąĹ

åIJĺåőŽäźL'çśżçŽĎæŮűåĂŹïijŇPythonåĚĄèőÿæĹŚäżňä¡£çŤĺ "metaclass" åĚşéŤőå■ŮåŔĆæŢřæİěæŇĞåőŽçL'źåőŽçŽĎåĚČçśżãĂĆ ä¿ŃåęĆä¡£çŤĺæŁ¡èśąå§žçśżïijŽ

```
from abc import ABCMeta, abstractmethod
class IStream(metaclass=ABCMeta):
    @abstractmethod
    def read(self, maxsize=None):
        pass

    @abstractmethod
```

(continues on next page)

```
    def write(self, data):
        pass
```

çĎűèĂŇïijŇåIJÍèĞłåőŽäźL'åĚČçśżäÿ■æĹŚäžňè£ŸåŔŕäžěæŔŘä¿ŻåĚűäżŰçŽĎåĚşéŤőå■ŮåŔĆæŢřïijŇå

```
class Spam(metaclass=MyMeta, debug=True, synchronize=True):
    pass
```

äÿžäžĘä¡£åĚČçśżæŤŕæŇĄè£ŹäžŻåĚşéŤőå■ŮåŔĆæŢřïijŇä¡ăå£Ěéążçąőä£ÍåIJÍ
__prepare__() , __new__() åŠŇ __init__() æŰźæşŢäÿ■
éČ¡ä¡£çŤĺåijžåĹűåĚşéŤőå■ŮåŔĆæŢřãĂĆářšåČŔäÿŃéÍćè£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

çżŹäÿĂäÿłåĚČçśżæůżåŁăåŔŕéĂL'åĚşéŤőå■ŮåŔĆæŢřéIJÄèęĄä¡ăåőŇåĚÍåijĐæĞĆçśżåĹŻåżžçŽĎæL'Ăa
åŽääÿžè£ŹäžŻåŔĆæŢřäijŽèćńäijåéĂŠçżŹæŕŔäÿĂäÿłçŻÿåĚşçŽĎæŰźæşŢãĂĆ
__prepare__() æŰźæşŢåIJÍæL'ĂæIJL'çśżåőŽäźL'åijĂåğŇæL'ğèąŇåL'■éęŮåĚÍèćńèřČçŤÍïijŇçŤÍæÍěåĹŻ
éĂŽåÿÿæÍěëőšïijŇè£ŹäÿłæŰźæşŢåŔłæŸŕçőÄå■ŢçŽĎè£ŤåŽđäÿĂäÿłå■ŮåĚÿæĹŰåĚűäżŰæŸäářĎářźèśąãĂĆ
__new__() æŰźæşŢèćńçŤÍæÍěåőđä¿ŃåŇŰæIJĂçżĹçŽĎçśżåŕźèśąãĂĆåőČåIJÍçśżçŽĎäÿżä¡Şèćńæ L'ğèąŇåő
__init__() æŰźæşŢæIJĂåŘŐèćńèřČçŤÍïijŇçŤÍæÍěæL'ğèąŇåĚűäżŰçŽĎäÿĂäžŽåĹÍåğŇåŇŰåůëä¡IJãĂĆ

å¡ŞæĹŚäžňædĎéĂäåĚČçśżçŽĎæŮŭåĂŹïijŇéĂŽåÿÿåŔłéIJÄèęĄåőŽäźL'äÿĂäÿł
__new__() æĹŰ __init__() æŰźæşŢïijŇä¡Ęäÿ■æŸŕäÿď'äÿłéČ¡åőŽäźL'ãĂĆ
ä¡ĘæŸŕïijŇåęĆædIJéIJÄèęĄæŐěåŔŮåĚűäżŰçŽĎåĚşéŤőå■ŮåŔĆæŢřçŽĎèŕÍïijŇè£Źäÿď'äÿłæŰźæşŢåřśèęĄå
éżŸèőď'çŽĎ __prepare__() æŰźæşŢæŐěåŔŮäżżæĎŔçŽĎåĚşéŤőå■ŮåŔĆæŢřïijŇä¡ĘæŸŕäijŽå£¡çŢěåő

æĽĂäżĕåŔŧæĲĽå¡Şè£Źäžŷé¢Ĭåď Ůç ŽĎåŔĆæŢřåŔŕèČ¡äijŽå¡śåŞ■åĹŕçśżåŠ¡åŘ■çĺ žéŮť çŽĎåĹŻåżžæŮűä¡ä
__prepare__() æŰźæşŢãĂĆ

éĂŽè£Ğä¡£çŤĺåijžåĹűåĚşéŤőå■ŮåŔĆæŢřïijŇåĲĺçśżçŽĎåĹŻåżžè£ĞçĺŃä■æĹŠä żňå£Ěéąžé ĂŽè£ĞåĚş

ä¡£çŤĺåĚşéŤőå■ŮåŔĆæŢřéĚ■ç¡őäÿĂäÿłåĚČçśżè£ŸåŔŕäżěèğĘä¡ĲåŕźçśżåŔŸéĞŔçŽĎäÿÁçğ■æŽ£äżčæ

```python
class Spam(metaclass=MyMeta):
    debug = True
    synchronize = True
    pass
```

åřĘè£ŹäžŻåśđæĂğåőŽäźĽäÿžåŔĆæŢřçŽĎåě¡åď ĎåĲĺäžŐåőČäžňäÿ■äijŽæśąæ§Şçśżç ŽĎåŘ■çğřçľ žéŮť
è£ŹäžŻåśđæĂğäžĚåŔłäżŐåśđäžŐçśżçŽĎåĹŻåżžéŸűæőţïijŇèĂŇäÿ■æŸŕçśżäÿ■çŽĎèŕ■åŔěæĽ ğèąŇéŸűæ
åŔęåď ŰïijŇåőČäžňåĲĺ __prepare__() æŰźæşŢäÿ■æŸŕåŔŕäżěèćńèő£éŮőçŽĎïijŇåŻăäÿžè£ŹäÿłæŰźæşŢ
ä¡ĘæŸŕçśżåŔŸéĞŔåŔłèČ¡åĲĺåĚČçśżçŽĎ __new__() åŠŇ __init__()
æŰźæşŢäÿ■åŔŕèğĄãĂĆ

11.16 9.16 *argsåŠŇ**kwargsçŽĎåijžåĹűåŔĆæŢřç■¿åŘ■

éŮőéćŸ

ä¡ăæĲĽäÿĂäÿłåĞ¡æŢřæĹŰæŰźæşŢïijŇåőČä¡£çŤĺ*argsåŠŇ**kwargsä¡ĲäÿžåŔĆæŢřïijŇè£Źæăůä¡£å¿Ů
ä¡ĘæĲĽæŮűåĂŹä¡ăæČşæčĂæ§ěäijăéĂŠè£ŻæİĕçŽĎåŔĆæŢřæŸŕäÿ■æŸŕæ§Řäÿłä¡ăæČşèęĄçŽĎçśżåđŃãĂĆ

èğčåĘşæŰźæąĹ

åŕźäżżä¡ŢæűĽåŔŁåĹřæŞ■ä¡ĲåĞ¡æŢřèřČçŤĺç■¿åŘ■çŽĎéŮőéćŸïijŇä¡ăéČ¡åžŤèŕěä¡£çŤĺ
inspect æĺąåİŮäÿ■çŽĎç■¿åŘ■çĽźæĂğãĂĆ æĹŠäżňæĲĂäÿžèęĄåĚşæşĺäÿď äÿłçśżïijŽSignature
åŠŇ Parameter ãĂĆäÿŃéİćæŸŕäÿĂäÿłåĹŻåżžåĞ¡æŢřåĽ■éİćçŽĎäžď äžŠä¿Ńå■ŘïijŽ

```python
>>> 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¡ăåřśåŔŕäżěä¡£çŤĺåőČçŽĎ bind()
æŰźæşŢå¿Ĺåőźæŷ§çŽĎåřĘåőČçżŠåőŽåĹř *args åŠŇ **kwargs äÿŁåŐżãĂĆ
äÿŃéİćæŸŕäÿĂäÿłçőĂå■ŢçŽĎæijŤçď žïijŽ

```python
>>> def func(*args, **kwargs):
...     bound_values = sig.bind(*args, **kwargs)
```

(continues on next 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'
>>>
```

åŘŕäžěçIJŃåĞžæÌěïijŇéĂŽè£ĞåřEç■¿åŘ■åŠÑäijăéĂŠçŽĎåŔĆæŢřçżŠåőŽèţůæÌěïijŇåŔŕäžěåijžåĹűåĞ¡ä

äýŃéÌćæŸŕäýĂäýłåijžåĹűåĞ¡æŢřç■¿åŘ■æŻťˇåĚůä¡ŞçŽĎä¿Ńå■ŘãĂĆåIJÍäžčçăĄäý■ïijŇæĹŚäžňåIJÍå§žç
`__init__()` æŰźæşŢïijŇ çĎűåŘŐæĹŚäžňåijžåĹűæL’ĂæIJL’çŽĎå■Řçśżå£ĚéążæŔŘä¿ŻäýĂäýłçL’źåőŽçŽĹ

```
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)

```
    __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')
```

äÿŃéÌćæŸŕä¡£çŤÍè£ŹäÿŁ `Stock` çśžçŽĎçď˙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'
>>>
```

èőÍèőž

åIJÍæĹŚäžňéIJÄèęĄæđĎåžžéĂŽçŤÍåĞ¡æŢřåžŞãĂĄçijŰåĘŹècĚéěřåŹÍæĹŰåőđçŐřäžčçŘĘçŽĎæŮűåĂŹ
`*args` åŠŇ `**kwargs` çŽĎä¡£çŤÍæŸŕå¿ĹæŹőéĄ■çŽĎãĂĆ
ä¡ĘæŸŕïijŇè£ŹæăůçŽĎåĞ¡æŢřæIJL’äÿĂäÿłçijžçĆźåřśæŸŕå¡Şä¡äæČşèęĄåőđçŐřèĞłåůśçŽĎåŘĆæŢřæčĂéłŇ
è£ŹæŮűåĂŹæĹŚäžňåŘŕäžěéĂŽè£ĞäÿĂäÿłç■¿åŘ■åŕźèśąæİěçőĂåŇŰåőČãĂĆ

åIJÍæIJĂåŘŐçŽĎäÿĂäÿłæŰźæąĹåőđä¿ŃäÿïijŇæĹŚäžňè£ŸåŘŕäžěéĂŽè£Ğä¡£çŤÍèĞłåőŽäźL’åĚČçśžæİ

```
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)

```
        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■çŽĎæŮűåĂŹïijŇářĘç■¿åŘ■å■ŸåĆĺåIJĺçL'źåőŽçŽĎåšđæĂğ `__signature__` äÿ■éĂŽåÿÿæŸŕå¿ĹæIJL'çŤĺçŽĎãĂĆ è£ŹæăůçŽĎèŕİïijŇåIJĺä¡£çŤĺ `inspect` æĺąåİŮæL'ğèąŇåĘĚçIJAçŽĎäžčçăĄářśèČ¡åŔŚçŐřç■¿åŘ■åźűářĘąőČä¡IJäÿžèřČçŤĺçžęåőŽãĂĆ

```
>>> import inspect
>>> print(inspect.signature(Stock))
(name, shares, price)
>>> print(inspect.signature(Point))
(x, y)
>>>
```

11.17 9.17 åIJĺçśżäÿŁåijžåĹűä¡£çŤĺçijŰçíŃèğĎçžę

éŮőécŸ

ä¡ăçŽĎçíŃåžŔåŇĚåŘńäÿĂäÿłå¿Ĺåď'ğçŽĎçśżçżğæL'£ä¡ŞçşżïijŇä¡ääÿŇæIJŻåijžåĹűæL'ğèąŇæ§ŘäžŻçij

èğčåĘşæŰźæąĹ

åęĆædIJä¡ăæČşçŻŚæŐğçśżçŽĎåőŽäźL'ïijŇéĂŽåÿÿåŔŕäžěéĂŽè£ĞåőŽäźL'äÿĂäÿłåĚČçśżãĂĆäÿĂäÿłå§ `type` åźűéĞ■åőŽäźL'åőČçŽĎ `__new__()` æŰźæşŢ æĹŰèĂĚæŸŕ `__init__()` æŰźæşŢãĂĆærŤåęĆïijŽ

```
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)
```

åŔęäÿĂçğ■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¡£çŤĺè£ŹäÿłåĚČçśżïijŇä¡äéĂŽåÿÿèęĄåřĘåőČæŤ¿åĹřåĹřäÿĂäÿłéąűçžğçĹűçśżåőŽäźĽäÿ■ïijŇçĹ

```
class Root(metaclass=MyMeta):
    pass

class A(Root):
    pass

class B(Root):
    pass
```

åĚČçśżçŽĎäÿĂäÿłåĚşéŤőçĽźçĆźæŸŕåőČåĚĄèőÿä¡ääIJĺåőŽäźĽçŽĎæŮűåĂŹæčĂæ§ěçśżçŽĎåĘĚåőźã
`__init__()` æŰźæşŢäÿ■ïijŇ ä¡ääŔŕäžěå¿Ĺè¡żæİ¿çŽĎæčĂæ§ěçśżå■ŮåĚÿãĂĄçĹűçśżç■Ľ'ç■Ľ'ãĂĆåźűäÿŤi
åŽăæ■d'ïijŇäÿĂäÿłæąĘæđűçŽĎæđĎåžžèĂĚåřśèČ¡åIJĺåd'ğåđŃçŽĎçžğæĽ'£ä¡Şçşžäÿ■éĂŽè£ĞçżŹäÿĂäÿłéąű

ä¡IJäÿžäÿĂäÿłåĚůä¡ŞçŽĎåžŤçŤĺä¿Ńå■ŘïijŇäÿŃéİćåőŽäźĽäžĘäÿĂäÿłåĚČçśżïijŇåőČäijŽæŃŠçżİäżżä¡Ţ

```
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'énŸçžğåŠŇåőđçŤĺçŽĎä¿Ńå■ŘïijŇäÿŃéİćæIJĽäÿĂäÿłåĚČçśżïijŇåőČçŤĺæİěæčĂæţŃéĞ■è¡¡

```
from inspect import signature
import logging

class MatchSignaturesMeta(type):

```

(continues on next 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
```

åęĆædIJä¡ăè£ŘèąŇè£Źæőţäżčçă Ąïij Ňářśäij Žå¿ŮåĹřäÿŃéİćè£ŹæăůçŽĎè¿ŞåĞžçżŞædIJïijŽ

```
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£ąæĄřářźäžŐæ■ŢèŐůäÿĂäžŻå¿őåęŹçŽĎçÍŃåžŔbugæŸřå¿ĹæIJL’çŤÍçŽĎãĂĆä¿ŃåęĆïij
éĆčäźĹå¡Şå■ŘçśżæŤźåŔŸåŔĆæŢřåŘ■å■ŮçŽĎæŮűåĂŹářśäijŽèřČçŤÍåĞžéŤŹãĂĆ

èőÍèőž

åIJÍåď ğåđŃéİćåŘŚåŕźèśąçŽĎçÍŃåžŔäÿ■ïijŇéĂŽåÿÿåřĘçśżçŽĎåőŽäźĽæŤ¿åIJÍåĚČçśżäÿ■æŐğåĹűæŸŕ
åĚČçśżåŔŕäžěçŻŚæŐğçśżçŽĎåőŽäźĽ'ïijŇè■ęåŚŁçijŰçÍŃäžžåŚŸæ§ŘäžŻæšąæIJĽæşÍæĎŔåĹřçŽĎåŔŕèČ¡åĞ

æIJĽäžžåŔŕèČ¡äijŽèŕť'ïijŇåČŔè£ŹæăůçŽĎéŤŹèŕŕåŔŕäžěéĂŽè£ĞçÍŃåžŔåĹĘæđŘåůěåĚůæĹŰIDEåŐżå
ä¡ĘæŸŕïijŇåęĆæđIJä¡ăåIJÍæđĎåżžäÿĂäÿłæąĘæđűæĹŰåĞ¡æŢřåžŞä¿ŻåĚűäžŰäžžä¡£çŤÍïijŇéČčäźĹä¡ăæšąåŁ
åŻăæ■d'ïijŇåŕźäžŐè£Źçğ■çśżåđŃçŽĎçÍŃåžŔïijŇåęĆæđIJåŔŕäżěåIJÍåĚČçśżäÿ■åĄŽæčĂæţŃæĹŰèőÿåŔŕäžěś

åIJÍåĚČçśżäÿ■éĂĽæŃl'éĞ■æŰřåőŽäźĽ `__new__()` æŰźæşŢè£ŸæŸŕ `__init__()` æŰźæşŢåŘŰåĘşäžŐä¡ăæČşæĂŐæăůä¡£çŤÍçżŞæđIJçśżãĂĆ `__new__()` æŰźæşŢåIJÍçśżåĹŻåżžäźŃåĽ■èćńèřČçŤÍïijŇéĂŽåÿÿçŤÍäžŐéĂŽè£Ğæ§Řçğ■æŰźåijŔïijĹærŤåęĆéĂŽè£ĞæŤ
èĂŇ `__init__()` æŰźæşŢæŸŕåIJÍçśżèćńåĹŻåżžäźŃåŘŐèćńèřČçŤÍïijŇå¡Şä¡ăéIJĂèęĄåőŇæŢť'æđĎåżžçśż
åIJÍæIJĂåŘŐäÿĂäÿłä¿Ńå■Řäÿ■ïijŇè£ŹæŸŕå£ĚèęĄçŽĎïijŇåŻăäÿžåőČä¡£çŤÍäžĘ `super()`
åĞ¡æŢřæİěæŘIJçť'ćäźŃåĽ■çŽĎåőŽäźĽãĂĆ åőČåŔłèČ¡åIJÍçśżçŽĎåőđä¿Ńèćńå ĹŻåżžäźŃåŘŐïijŇåźűäÿŤçŻÿ

æIJĂåŘŐäÿĂäÿłä¿Ńå■Řè£ŸæijŤçď'žäžĘPythonçŽĎåĞ¡æŢřç■¿åŘ■åŕźèśąçŽĎä¡£çŤÍãĂĆ
åőđéŹĚäÿŁïijŇåĚČçśżåřĘæŕŘäÿłåŔŕèřČçŤÍåőŽäźĽæŤ¿åIJÍäÿĂäÿłçśżäÿ■ïijŇæŘIJçť'ćåĽ■äÿĂäÿłåőŽäźĽ'ïijÍ
çĎűåŘŐéĂŽè£Ğä¡£çŤÍ `inspect.signature()` æİěçőĂå■ŢçŽĎærŤè¿ČåőČäžñçŽĎèŘČçŤÍç■¿åŘ■ãĂĆ

æIJĂåŘŐäÿĂçĆźïijŇäžčçăĄäÿ■æIJĽäÿĂèąŇä¡£çŤÍäžĘ `super(self, self)`
åźűäÿ■æŸræŐŠçĽĹéŤŹèŕŕãĂĆ å¡Şä¡£çŤÍåĚČçśżçŽĎæŮűåĂŹïijŇæĹŚäžňèęĄæŮűåĹżèőŕä¡ŔäÿĂçĆźåŕśæŸ
`self` åőđéŹĚäÿŁæŸŕäÿĂäÿłçśżåŕźèśąãĂĆ åŻăæ■d'ïijŇè£Źæİąèŕ■åŔěåĚűåőđåŕśæŸŕçŤÍæİěåŕżæĽ¿ä¡■äžŐçż
`self` çĹűçśżçŽĎåőŽäźĽãĂĆ

11.18 9.18 äžěçijŰçÍŃæŰźåijŔåőŽäźĽçśż

éŮőécŸ

ä¡ăåIJÍåĘŹäÿĂæőţäžčçăĄïijŇæIJĂçżĹéIJĂèęĄåĹŻåżžäÿĂäÿłæŰřçŽĎçśżåŕźèśąãĂĆä¡äèĂČèŹŚåřĘçśżç
åźűäÿŤä¡£çŤÍåĞ¡æŢřærŤåęĆ `exec()` æİěæĽ'ğèąŇåőČïijŇä¡ĘæŸŕä¡ăæČşåŕżæĽ¿äÿĂäÿłæŻť'åŁääijŸéŻĚçŻ

èğčåĘşæŰźæąĹ

ä¡ăåŔŕäžěä¡£çŤÍåĞ¡æŢř `types.new_class()` æİěåĹİåğŃåŇŰæŰřçŽĎçśżåŕźèśąãĂĆ
ä¡äéIJĂèęĄåĄçŽĎåŔłæŸræŔŘä¿Żçśżç ŽĎåŘ■å■ŮãĂĄçĹűçśżåĚČçżĎãĂĄåĚşéŤőå■ŮåŔĆæŢřïijŇäžěåŔŁ

```
# 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)

```
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■æŰźåijŘäijŽæđĎåžžäýĂäýłæŹőéĂŽçŽĎçśžåŕźèśąïijŇåźűäýŤæŇĽçĚğä¡ăçŽĎæIJ§æIJŻåůěä¡IJïj

```
>>> s = Stock('ACME', 50, 91.1)
>>> s
<stock.Stock object at 0x1006a9b10>
>>> s.cost()
4555.0
>>>
```

è£Źçğ■æŰźæşŢäý■ïijŇäýĂäýłæŕŤè¿ČéŽ¿çŘĘèğčçŽĎåIJřæŰźæŸŕåIJÍèřČçŤĺåőŇ `types.new_class()` åŕź `Stock.__module__` çŽĎèţŃåĂijãĂĆ æŕŘæňąå¡ŞäýĂäýłçśžèćńåőŽäźĽ'åŘŐïijŇåőČçŽĎ `__module__` åśđæĂğåŇĚåŘńåőŽäźĽ'åőČçŽĎæĺąåİŮåŘ■ãĂĆ è£Źäýłå Ř■å■ŮçŤĺäžŐçŤ§æĹŘ `__repr__()` æŰźæşŢçŽĎè¿ŞåĞžãĂĆåőČåŘŇæăůäź§èćńçŤĺäžŐå¿Ĺåd'ŽåžŞïijŇæŕŤåęĆ `pickle` ãĂĆ åŻăæ■d'ïijŇäýžäžĘèől'ä¡ăåĹŻåžžçŽĎçśžæŸŕâĂIJæ■čçąőâĂİçŽĎïijŇä¡ăéIJĂèęAçąőä£Ìè£Źäýł

åęĆædIJä¡ăæČşåĹŻåžžçŽĎçśžéIJĂèęĄäýĂäýłäý■åŘŇçŽĎåĚČçśžïijŇåŕŕäžěéĂŽè£Ğ `types.new_class()` çňňäýĽ'äýłåŔĆæŢřäijăéĂŠçžŹåőČãĂĆä¿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'>
>>>
```

çňňäýĽ'äýłåŔĆæŢřè£ŸåŕŕäžěåŇĚåŘńåĚűäžŰçŽĎåĚşéŤőå■ŮåŔĆæŢřãĂĆæŕŤåęĆïijŇäýĂäýłçśžçŽĎåč

```
class Spam(Base, debug=True, typecheck=False):
    pass
```

éĆčäźĹåŕŕäžěåŕĘåĚűç£żèŕŚæĹŘåęĆäýŃçŽĎ `new_class()` èřČçŤĺå¡ćåijŕïijŽ

```
Spam = types.new_class('Spam', (Base,),
                       {'debug': True, 'typecheck': False},
                       lambda ns: ns.update(cls_dict))
```

new_class() çňňåŻŻäÿłåŔĆæŢřæIJĂçěđçğŸïijŇåőČæŸŕäÿĂäÿłçŤĺæİěæŐěåŔŮçśżåŠ¡åŘ■çľžéŮťçŻ
éĂŽåÿÿè£ŹæŸŕäÿĂäÿłæŹőéĂŽçŽĎå■ŮåĚÿïijŇä¡ĘæŸŕåőČåőđéŹĚäÿŁæŸŕ
__prepare__() æŰźæşŢè£ŤåŽđçŽĎäżżæĎŔåŕźèśąïijŇè£ŹäÿłåIJĺ9.14åřŔèŁĆåůšçżŔäžŃçż■è£ĞäžĘãĂĆ
è£ŹäÿłåĞ¡æŢřéIJĂèęĄä¡£çŤĺäÿŁéİćæijŤçď'žçŽĎ update()
æŰźæşŢçżŹåŠ¡åŘ■çľžéŮťåćđåŁääĘĚåőźãĂĆ

èőĺèőž

å¿Ĺåď'ŽæŮűåĂŹåęĆæđIJèČ¡æđĎéĂäæŰřçŽĎçśżåŕźèśąæŸŕå¿ĹæIJL'çŤĺçŽĎãĂĆ
æIJL'äÿłå¿ĹçĘ§æĆL'çŽĎä¿Ńå■ŘæŸŕèřČçŤĺ collections.namedtuple()
åĞ¡æŢřïijŇä¿ŃåęĆïijŽ

```
>>> Stock = collections.namedtuple('Stock', ['name', 'shares',
↪'price'])
>>> Stock
<class '__main__.Stock'>
>>>
```

namedtuple() ä¡£çŤĺ exec() èĂŇäÿ■æŸŕäÿŁéİćäżŃçż■çŽĎæŁĂæIJŕãĂĆä¡ĘæŸŕïijŇäÿŃéİćéĂŽè
æĹŚäżňçŻť'æŐěåĹŽåżžäÿĂäÿłçśżïijŽ

```
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)

```
    cls.__module__ = sys._getframe(1).f_globals['__name__']
    return cls
```

è£ŹæőţäžčçăąçŽĎæIJĂåŘŐéČĺåĹĘä¡£çŤĺäžĘäÿĂäÿłæL'ĂèřŞçŽĎâĂİæąĘæđűé■ŤæşŢâĂİïijŇéĂŽè£Ğè
sys._getframe() æİěèŐůåŘŰèřČçŤĺèĂĚçŽĎæĺąåİŮåŘ■ãĂĆ
åŔęåd'ŰäÿĂäÿłæąĘæđűé■ŤæşŢä¿Ńå■ŘåIJĺ2.15åřŔèŁĆäÿ■æIJL'äžŃçż■è£ĞãĂĆ

äÿŃéİćçŽĎä¿Ńå■ŘæijŤçd'žäžĘåL'■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¡ăåŔŕèČ¡åČŔéĂŽè£ĞçŻt'æŐěåőđä¿ŃåŇŰäÿĂäÿłåĚČçśżæİěçŻt'æŐěåĹŻåžžäÿĂäÿłçśżïijŽ

```
Stock = type('Stock', (), cls_dict)
```

è£Źçğ■æŰźæşŢçŽĎéŮőécŸåIJĺäžŐåőČå£¡çŢěäžĘäÿĂäžŻåĚşéŤőæ■ěéłd'ïijŇæŕŤåęĆåŕźäžŐåĚČçśżäÿ■
__prepare__() æŰźæşŢçŽĎèŕČçŤĺãĂĆ éĂŽè£Ğä¡£çŤĺ types.new_class()
ïijŇä¡ăåŔŕäžěä£ÌèŕĄæL'ĂæIJL'çŽĎå£ĚèęĄåĹİåğŃåŇŰæ■ěéłd'éČ¡èČ¡å¿ŮåĹřæL'ğèąŇãĂĆ
æŕŤåęĆïijŇtypes.new_class() çňňåŻŻäÿłåŔĆæŢřçŽĎåŻđèŕČåĞ¡æŢřæŐěåŔŮ
__prepare__() æŰźæşŢè£ŤåŻđçŽĎæŸăåřĎåŕźèśąãĂĆ

åęĆæđIJä¡ăäžĚäżĚåŔłæŸŕæČşæL'ğèąŇåĞĘåd'Ğæ■ěéłd'ïijŇåŔŕäžěä¡£çŤĺ types.
prepare_class() ãĂĆä¿ŃåęĆïijŽ

```
import types
metaclass, kwargs, ns = types.prepare_class('Stock', (), {'metaclass
↪': type})
```

åőČäijŽæ§ěæL'¿åŘĹéĂĆçŽĎåĚČçśżåźűèřČçŤĺåőČçŽĎ __prepare__()
æŰźæşŢãĂĆ çĎűåŘŐè£ŹäÿłåĚČçśżä£Ìå■ŸåőČçŽĎåĚşéŤőå■ŮåŔĆæŢřïijŇåĞĘåd'ĞåŚ¡åŘ■çl'žéŮt'åŘŐèćńè

æŻt'åd'Žä£qæĄŕ, èŕůåŔĆèĂČ PEP 3115 , äžěåŘŁ Python documentation .

11.19 9.19 åIJÍåőŽäźĽçŽĎæŮűåĂŹåĹİåğŃåŇŰçśžçŽĎæĹŘåŚŸ

éŮőécŸ

ä¡ăæČşåIJÍçśżèćńåőŽäźĽçŽĎæŮűåĂŹåřśåĹİåğŃåŇŰäÿĂéČĺåĹĘçśžçŽĎæĹŘåŚŸïijŇèĂŇäÿ■æŸŕèęAç

èğčåĘşæŰźæąĹ

åIJÍçśżåőŽäźĽæŮűåřśæĽğèąŇåĹİåğŃåŇŰæĹŰèő¿ç¡őæŞ■ä¡IJæŸŕåĚČçśžçŽĎäÿĂäÿłåĚÿåđŃåžŤçŤÍåIJè£ŹæŮűåĂŹä¡ăåŔŕäżěæĽğèąŇäÿĂäžŽéćİåď˙ŰçŽĎæŞ■ä¡IJãĂĆ

äÿŃéİćæŸŕäÿĂäÿłä¿Ńå■ŘïijŇåĹľçŤÍè£ŹäÿłæĂİèůŕæİěåĹŻåžžçśżäijijäžŐ `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£ŹæőţäžčçăĄåŔŕäżěçŤÍæİěåőŽäźĽçőĂå■ŢçŽĎå§žäžŐåĚČçżĎçŽĎæŢřæ■őçżŞædĎïijŇąęĆäÿŃæĽĂç

```
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)

```
>>> s.shares = 23
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
AttributeError: can't set attribute
>>>
```

èőÍèőž

è£ŹäÿĂåřŔèŁĆäÿ■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■ěéłd'æŸŕäżĂäźĹæŮűåĂŹåŔŞ `StructTupleMeta` äÿ■çŽĎ `__init__()` æŰźæşŢåŔłåIJĺæŕŔäÿłçśżèćńåőŽäźĽ'æŮűèćńèřČçŤĺäÿĂæňąãĂ `cls` åŔĆæŢřåřśæŸŕéĆčäÿłèćńåőŽäźĽçŽĎçśżãĂĆåőđéŹĚäÿŁïijŇäÿŁè£řäżčçăĄä¡£çŤĺäžĘ `_fields` çśżåŔŸéĞŔæİěä£İå■ŸæŰřçŽĎèćńåőŽäźĽçŽĎçśżïijŇ çĎűåŘŐçżŹåőČåĘ■æůżåŁääÿĂçĆźæŰřçŽĎäÿIJèě£ãĂĆ

`StructTuple` çśżä¡IJäÿžäÿĂäÿłæŹőéĂŽçŽĎå§žçśżïijŇä¿ŻåĚűäżŰä¡£çŤĺèĂĚæİěçżğæĽ£ãĂĆ è£Źäÿłçśżäÿ■çŽĎ `__new__()` æŰźæşŢçŤĺæİěædĎéĂăæŰřçŽĎåőđä¿ŃãĂĆ è£ŹéĞŇä¡£çŤĺ `__new__()` åźűäÿ■æŸŕå¿Ĺåÿÿèğ„ïijŇäÿżèęĄæŸŕåŻääÿžæĹŚäžňèęĄä£őæŢźåĚČçżĎçŽĎèřČçŤĺç■¿åŘ■ïijŇ ä¡£å¿ŮæĹŚäżňåŔŕäżèåČŔæŹőéĂŽçŽĎåőđä¿ŃèřČçŤĺéĆčæůåĹŻåżžåőđä¿ŃãĂĆåřśåČŔäÿŃéÍćè£ŹæůïijŽ

```
s = Stock('ACME', 50, 91.1) # OK
s = Stock(('ACME', 50, 91.1)) # Error
```

èů§ `__init__()` äÿ■åŘŇçŽĎæŸŕïijŇ `__new__()` æŰźæşŢåIJĺåőđä¿ŃèćńåĹŻåżžäźŃåL'■èćńèğęåŔŚ çŤśäžŐåĚČçżĎæŸŕäÿ■åŔŕä£őæŢźçŽĎïijŇæL'Ääžěäÿ̆ĂæŮęåőČäżňèćńåĹŻåżžäžĘåřśäÿ■åŔŕèČ¡åŕźåőČåĄŽä `__init__()` äijŽåIJĺåőđä¿ŃåĹŻåżžçŽĎæIJĂåŘŐèćńèğęåŔŚïijŇ è£ŹæůçŽĎèŕİæĹŚäżňåřśåŔŕäžěåĄŽæĹŚäżňæČşåĄŽçŽĎäžĘãĂĆè£Źäź§æŸŕäÿžäżĂäźĹ `__new__()` æŰźæşŢåůšçżŔèćńåőŽäźĽ'äžĘãĂĆ

åř¡çőąæIJňèŁĆå¿Ĺç§■ïijŇè£ŸæŸŕéIJĂèęĄä¡ăèČ¡äžŤçżĘçăŤèŕžïijŇæůśåĚěæĂİèĂČPythonçśżæŸŕåęĆä è£ŸæIJL'åŕśæŸŕåĚČçśżåŠŇçśżçŽĎåŘĎäÿłäÿ■åŘŇçŽĎæŰźæşŢçl'űçń§åIJĺäżĂäźĹæŮűåĂŽèćńèřČçŤĺãĂĆ

PEP 422 æŔŘä¿ŻäžĘäÿĂäÿłèğčåĘşæIJňèŁĆéŮőécŸçŽĎåŔęåď'ŰäÿĂçğ■æŰźæşŢãĂĆ ä¡ĘæŸŕïijŇæĹłæ■ćåĽræĹŚåĘŹè£ŹæIJňäźęçŽĎæŮűåĂŹïijŇåőČè£ŸæšąèćńéĞĞçžşåŠŇæŐěåŔŮãĂĆ åř¡çőąåęĆæ■ď'ïijŇåęĆæđIJä¡ăä¡£çŤĺçŽĎæŸŕPython 3.3æĹŮæŻť'énŸçŽĎçL'ĹæIJňïijŇéĆčäźĹè£ŸæŸŕåĂijå

11.20 9.20 åĹľçŤĺåĞ¡æŢřæşÍèğčåőđçŐřæŰźæşŢéĞ■è¡¡

éŮőécŸ

ä¡ăåůšçżŔå■ęè£ĞæĂŐæăůä¡£çŤĺåĞ¡æŢřåŔĆæŢřæşÍèğčïijŇéĆčäźĹä¡ăåŔŕèČ¡äijŽæČşåĹľçŤĺåőČæÍěåőđ ä¡ĘæŸŕä¡ääÿ■çąőåőŽåžŤèŕěæĂŐæăůåŐżåőđçŐřïijĹæĹŮèĂĚåĹřåžŢèąŇå¿ŮéĂŽäÿ■ïijL'ãĂĆ

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
```

äÿŃéÌćæŸŕæĹŚäżñçňňäÿĂæ■ěçŽĎåřÌèŕŢïijŇä¡£çŤĺåĹřäžĘäÿĂäÿłåĚČçśżåŠŇæŔŔè£řåŹĺïijŽ

```
# 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)

```
            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)

```
    '''
    def __new__(cls, clsname, bases, clsdict):
        return type.__new__(cls, clsname, bases, dict(clsdict))

    @classmethod
    def __prepare__(cls, clsname, bases):
        return MultiDict()
```

äýžäžĘä¡£çŤĺè£Źäýłçśżïij Ňä¡ääŔŕäżěåČŔäýŃéĺćè£ŹæăůåĘŹïijŽ

```
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)
```

äýŃéĺćæŸŕäýĂäýłäžd'äžŠçd'žä¿ŃæĺěéłŇèŕĄåőČèČ¡æ■čçąőçŽĎåůěä¡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)

```
>>> # Get today's date
>>> e = Date()
>>> e.year
2012
>>> e.month
12
>>> e.day
3
>>>
```

èőĺèőž

åİęçŹ¡æİěèőšïijŇçŻÿåŕźäžŐéĂŽåÿÿçŽĎäžčçăĄèĂŇåůšæIJňèŁĆä¡£çŤĺåĹřäžĘå¿Ĺåď'ŽçŽĎé■ŤæşŢäżčçă
ä¡ĘæŸŕïijŇåőČå■t'èČ¡èől'æĹŚäžňæůšåĚěçŘĘèğčåĚČçśżåŠŇæŔŔè£řåŹĺçŽĎåžŢåśĆåůěä¡IJåŐ§çŘęïijŇ
åźűèČ¡åŁăæůšåŕźè£ŹäžŻæęĆå£ţçŽĎå■řèśąăĂĆåŻăæ■d'ïijŇářśçőŮä¡ăáźűäÿ■äijŽçńŇå■şåŐżäžŤçŤĺæIJňèŁĆ
åőČçŽĎäÿĂäžŻåžŢåśĆæĂİæČşå■t'äijŽå¡śåŞ■åĹřåĚűåőČæúĽ'åŔŁåĹřåĚČçśżãĂĄæŔŔè£řåŹĺåŠŇåĞ¡æŢřæş

æIJňèŁĆçŽĎåőđçŐřäÿ■çŽĎäÿżèęĄæĂİèůŕåĚűåőđæŸŕå¿ĹçőĂå■ŢçŽĎãĂĆMutipleMeta
åĚČçśżä¡£çŤĺåőČçŽĎ __prepare__() æŰźæşŢ æİěæŔŘä¿ŻäÿĂäÿłä¡IJäÿž MultiDict
åőđä¿ŃçŽĎèĞłåőŽäźĽå■ŮåĚÿãĂĆè£Źäÿłèů§æŹőéĂŽå■ŮåĚÿäÿ■äÿĂæăůçŽĎæŸŕïijŇ
MultiDict äijŽåIJĺåĚČçt'ăèćńèő¿ç¡őçŽĎæŮűåĂŹæčĂæ§ěæŸŕåŘęåůšçżŘå■ŸåIJĺïijŇåęĆæđIJå■ŸåIJĺçŽĎ
MultiMethod åőđä¿Ńäÿ■åŘĹåźűãĂĆ

MultiMethod åőđä¿ŃéĂŽè£ĞæđĎåžžäžŐçśżåđŃç■¿åŘ■åĹřåĞ¡æŢřçŽĎæŸäåřĎæİěæŤűéŻĘæŰźæşŢ
åIJĺè£ŹäÿłæđĎåžžè£ĞçĺŃäÿ■ïijŇåĞ¡æŢřæşĺèğčèćńçŤĺæİěæŤűéŻĘè£ŹäžŻç■¿åŘ■çĎűåŘŐæđĎåžžè£ŹäÿłæŸ
è£Źäÿłè£ĞçĺŃåIJĺ MultiMethod.register() æŰźæşŢäÿ■åőđçŐřãĂĆ
è£Źçğ■æŸäåřĎçŽĎäÿĂäÿłåĚşéŤőçL'źçĆźæŸŕåŕźäžŐåď'ŽäÿłæŰźæşŢïijŇæL'ĂæIJL'åŔĆæŢřçśżåđŃéČ¡å£Ěé

äÿžäžĘèől' MultiMethod åőđä¿ŃæĺąæŃ§äÿĂäÿłèřČçŤĺïijŇåőČçŽĎ
__call__() æŰźæşŢèćńåőđçŐřäžĘãĂĆ è£ŹäÿłæŰźæşŢäżŐæL'ĂæIJL'æŐŠéŹd' slef
çŽĎåŔĆæŢřäÿ■æđĎåžžäÿĂäÿłçśżåđŃåĚČçżĎïijŇåIJĺåĘĚéČĺmapäÿ■æ§ěæL'¿è£ŹäÿłæŰźæşŢïijŇ
çĎűåŘŐèřČçŤĺçŽÿåžŤçŽĎæŰźæşŢãĂĆäÿžäžĘèČ¡èől' MultiMethod
åőđä¿ŃåIJĺçśżåőŽäźĽæŮűæ■čçąőæŞ■ä¡IJïijŇ__get__() æŸŕå£Ěéążå¿ŮåőđçŐřçŽĎãĂĆ
åőČèćńçŤĺæİěæđĎåžžæ■čçąőçŽĎçżŠåőŽæŰźæşŢãĂĆæŕŤåęĆïijŽ

```
>>> 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¡£çŤĺåĘşéŤőå■

```
>>> 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¡ĘæŸŕåőČéIJĂèęĄäÿĂäÿłåőŇåĚĺäÿ■
éŮőécŸåIJĺäžŐåĚşéŤőå■ŮåŔĆæŢřçŽĎåĞžçŐřæŸŕæšąæIJL'éąžåžŔçŽĎãĂĆå¡ŞåőČèů§ä¡■ç¡őåŔĆæŢřæůůåĚ
éĆčä¡äçŽĎåŔĆæŢřářśäijŽåŔŸå¿ŮæŕŤè¿ČæůůäźśäžĘïijŇè£ŹæŰűåĂŹä¡ääÿ■å¿Ůäÿ■åIJĺ
`__call__()` æŰźæşŢäÿ■åĚĹåŐżåĄŽäÿłæŐŠåžŔãĂĆ

åŘŇæăůåŕźäžŐçżğæL'£äź§æŸŕæIJL'éŹŘåĹűçŽĎïijŇä¿ŃåęĆïijŇçśżäijijäÿŃéİćè£Źçğ■äžčçăĄåřśäÿ■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■Řçśżåőđä¿ŃïijĹæŕŤåęĆBçŽĎåőđä¿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)

```
TypeError: No matching method for types (<class '__main__.B'>,)
>>>
```

ä¡IJäýžä¡£çŤĺåĚČçśżåŠŇæşĺèğčçŽĎäýĂçğ■æŻ£äżčæŰźæąĹïijŇåŕŕäżěéĂŽè£ĞæŔŕè£řåŹĺæİěåőđçŐřç

```
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
```

äýžäžĘä¡£çŤĺæŔŕè£řåŹĺçL’ĹæIJňïijŇä¡ăéIJĂèęĄåČŔäýŃéİćè£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)

```
    def bar(self, s, n = 0):
        print('Bar 2:', s, n)
```

æŔŔè£řåŹÍæŰźæąĹåŘŇæăůäź§æIJL'åL'■éÍćæŔŘåĹřçŽĎéŹŘåĹűïijĹäý■æŤŕæŇĄåĚşéŤőå■ŮåŔĆæŢřåŠ

æL'ĂæIJL'äžŃçL'ľéČ¡æŸŕåźşç■L'çŽĎïijŇæIJL'åě¡æIJL'åİŔïijŇäź§èőýæIJĂåě¡çŽĎåŁđæşŢåřśæŸŕåIJÍæŹ
äý■è£ĞæIJL'äžŻçL'źæőŁæČĚåĘţäýŃè£ŸæŸŕæIJL'æĎŔäźL'çŽĎïijŇæŕŤåęĆå§žäžŐæĺąåijŔåŇźéĚ■çŽĎæŰźä
äý¿äýłä¿Ńå■ŘïijŇ8.21åřŔèŁĆäý■çŽĎè£éŮőèĂĚæĺąåijŔåŔŕäżëä£őæŤźäýžäýĂäýłä¡£çŤĺæŰźæşŢéĞ■è¡¡çŽ
ä¡ĘæŸŕïijŇéŹď'äžĘè£Źäýłäžëåď'ŰïijŇéĂŽåÿÿäý■åžŤèŕëä¡£çŤĺæŰźæşŢéĞ■è¡¡ïijĹåřśçőĂå■ŢçŽĎä¡£çŤĺäý■å

åIJÍPythonçď¿åŇžåŕźäžŐåőđçŐřæŰźæşŢéĞ■è¡¡çŽĎèőÍèőžåůšçżŔçŤśæİěåůšäźĚãĂĆ
åŕźäžŐåijŢåŘŚè£ŹäýłäžL'èőžçŽĎåŐ§åŻäïijŇåŔŕäżëåŔĆèĂČäýŃGuido van
RossumçŽĎè£ŹçŕĞå■Žåőćïij Ž Five-Minute Multimethods in Python

11.21 9.21 éĄ£åĚ■éĞ■åď'■çŽĎåśđæĂğæŰźæşŢ

éŮőécŸ

ä¡ăåIJÍçśżäý■éIJĂèęĄéĞ■åď'■çŽĎåőŽäźL'äýĂäžŻæL'ğèąŇçŻýåŘŇéĂżè¿ŚçŽĎåśđæĂğæŰźæşŢïijŇæŕŤ

èğčåĘşæŰźæąĹ

èĂČèŹŚäýŃäýĂäýłçőĂå■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)

```
            raise TypeError('age must be an int')
        self._age = value
```

åŔŕäžěçIJŃåĹřïijŇäýžäžĘåőđçŐřåśđæĂğåĂijçŽĎçśžåđŃæčĂæ§ěæĹŚäżňåĘŹäžĘå¿Ĺåď'ŽçŽĎéĞ■åď'■ä
åŔłèęĄä¡ääžěåŘŐçIJŃåĹřçśżäijijè£Źæăůç ŽĎäžčçăĄïijŇä¡ăéČ¡åžŤèŕěæČşåŁđæşŢåŐżçőĂåŇŰåőČãĂĆ
äýĂäýłåŔŕèąŇçŽĎæŰźæşŢæŸŕåĹŻåžžäýĂäýłåĞ¡æŢřçŤĺæİěåőŽäźĽåśđæĂğåźűè£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
```

èőÍèőž

æIJňèŁĆæĹŚäżňæijŤçď'žåĘĚéČĺåĞ¡æŢřæĹŰèĂĚéŮ■åŇĚçŽĎäýĂäýłéĞ■èęĄçL'źæĂğïijŇåőČäžňå¿Ĺå
`typed_property()` çIJŃäýŁåŐżæIJL'çĆźéŽ¿çŘĘèğčïijŇåĚűåőđåőČæL'ĂåĄŽçŽĎäžĚäżĚåŕśæŸŕäýžä¡ăç
åŽăæ■ď'ïijŇå¡ŞåIJÍäýĂäýłçśżäÿ■ä¡£çŤĺåőČçŽĎæŮűåĂŹïijŇæŢĹæđIJèů§åřĘåőČéĞŇéİćçŽĎäžčçăĄæŤ¿åĹř
åř¡çőąåśđæĂğçŽĎ `getter` åŠŇ `setter` æŰźæşŢèő£éŮőäžĘæIJňåIJřåŔŸéĞŔąęĆ `name` ,
`expected_type` äžěåŔŁ `storate_name` ïijŇè£ŹäýŁå¿Ĺæ■čåÿÿïijŇè£ŹäžŽåŔŸéĞŔçŽĎåĂijäijŽä£İå■Ÿ

æĹŚäżňè£ŸåŔŕäžěä¡£çŤĺ `functools.partial()` æİěçĺ■çĺ■æŤźåŔŸäýŃè£ŹäýŁä¿Ńå■ŘïijŇå¿ĹæIJI

```
from functools import partial

String = partial(typed_property, expected_type=str)
Integer = partial(typed_property, expected_type=int)

# Example:
```

(continues on next page)

```
class Person:
    name = String('name')
    age = Integer('age')

    def __init__(self, name, age):
        self.name = name
        self.age = age
```

åĚűåőđä¡ăåŔŕäžěåŔŚçŐřïijŇè£ŹéĞŇçŽĎäžčçăĄèů§8.13åřŔèŁĆäÿ■çŽĎçśžåđŃçşżçż§æŔŔè£řåŹĺäžčçă

11.22 9.22 åőŽäźĽäÿŁäÿŃæŰĞçőąçŘĘåŹĺçŽĎçőÅå■ŢæŰźæşŢ

éŮőécŸ

ä¡ăæČşèĞłåůśåŐżåőđçŐřäÿĂäÿłæŰřçŽĎäÿŁäÿŃæŰĞçőąçŘĘåŹĺïijŇäžěä¿£ä¡£çŤĺwithèí■åŔěãĂĆ

èğčåĘşæŰźæąĹ

åőđçŐřäÿĂäÿłæŰřçŽĎäÿŁäÿŃæŰĞçőąçŘĘåŹĺçŽĎæIJĂçőÅå■ŢçŽĎæŰźæşŢåřśæŸŕä¡£çŤĺ contexlib æĺąåİŮäÿ■çŽĎ @contextmanager èčĚéěřåŹĺãĂĆ äÿŃéİćæŸŕäÿĂäÿłåőđçŐřäžĘäžčçăĄåİŮèőąæŮűåŁ§èČ¡çŽĎäÿŁäÿŃæŰĞçőąçŘĘåŹĺä¿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
```

åIJĺåĞ¡æŢř timethis() äÿ■ïijŇyield äźŃåĽ■çŽĎäžčçăĄäijŽåIJĺäÿŁäÿŃæŰĞçőąçŘĘåŹĺäÿ■ä¡IJäÿ __enter__() æŰźæşŢæĽğèąŇïijŇ æĽĂæIJĽåIJĺ yield äźŃåŘŐçŽĎäžčçăĄäijŽä¡IJäÿž __exit__() æŰźæşŢæĽğèąŇãĂĆ åęĆæđIJåĞžçŐřäžĘåijĆåÿÿïijŇåijĆåÿÿäijŽåIJĺyield-èí■åŔěéĆčéĞŇæŁŻåĞžãĂĆ

äÿŃéİćæŸŕäÿĂäÿłæŻť'åŁäéńŸçžğäÿĂçĆźçŽĎäÿŁäÿŃæŰĞçőąçŘĘåŹĺïijŇåőđçŐřäžĘåĹŮèąĺåŕźèśąäÿŁç

```
@contextmanager
def list_transaction(orig_list):
    working = list(orig_list)
    yield working
    orig_list[:] = working
```

è£ŹæőţäžčçăÇžĎä¡IJçŤĺæŸŕäżżä¡ŢåŕźåLŮèąĺçŽĎä£őæŤźåŔłæIJL’å¡ŞæL’ĂæIJL’äžčçăĄè£ŘèąŇåőŇæ
äÿŃéÌćæĹŚäžňæÌěæijŤçd’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]
>>>
```

èőÍèőž

éĂŽåÿÿæČĚåĘţäÿŃïijŇåęĆæđIJèęĄåĘŹäÿĂäÿłäÿŁäÿŃæŰĞçőąçŘĘåŹĺïijŇä¡ăéIJĂèęĄåőŽäźL’äÿĂäÿłç
`__enter__()` åŠŇäÿĂäÿł `__exit__()` æŰźæşŢïijŇåęĆäÿŃæL’Ăçd’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))
```

åř¡çőąè£Źäÿłäź§äÿ■éŽ¿åĘŹïijŇä¡ĘæŸŕçŻÿæŕŤè¿ČåĘŹäÿĂäÿłçőĂå■ŢçŽĎä¡£çŤĺ
`@contextmanager` æşĺèğčçŽĎåĞ¡æŢřèĂŇèĺĂè£ŸæŸŕçĺ■æŸ¿äźŔåŠşãĂĆ

`@contextmanager` åžŤèŕěäžĚäżĚçŤĺæİěåĘŹèĞłåŇĚåŘńçŽĎäÿŁäÿŃæŰĞçőąçŘĘåĞ¡æŢřãĂĆ åęĆæđIJä¡ăæIJL'äÿĂäžŻåŕźèśą(æŕŤåęĆäÿĂäÿłæŰĞäżűãĂĄç¡ŚçżIJè£đæŐěæĹŰéŤĄ)ïijŇéIJĂèęĄæŤŕæŇĄ `with` èŕ■åŘěïijŇéĆčäźĹä¡ăåřśéIJĂèęĄå■ŢçŃňåőđçŐř `__enter__()` æŰźæşŢåŠŇ `__exit__()` æŰźæşŢãĂĆ

11.23 9.23 åIJĺåśĂéČĺåŔŸéĞŔå§§äÿ■æĽğèąŇäžčçăĄ

éŮőécŸ

ä¡ăæČşåIJĺä¡£çŤĺèŇČåŻť'åĘĚæĽğèąŇæ§Řäÿłäžčçă ĄçĽĞæőţïijŇåźűäÿŤåÿŇæIJŻåIJĺæĽğèąŇåŘŐæĽĂ

èğčåĘşæŰźæąĹ

äÿžäžĘçŘĘèğčè£ŹäÿłéŮőécŸïijŇåĚĹèŕŢèŕŢäÿĂäÿłçőĂå■ŢåIJžæŹŕãĂĆéęŰåĚĹïijŇåIJĺåĚĺåśĂåŚ¡åŘ■ç

```
>>> 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
>>>
```

åŔŕäžěçIJŃåĞžïijŇæIJĂåŘŐæŁŻåĞžäžĘäÿĂäÿłNameErroråijĆåÿÿïijŇåřśèů§åIJĺ `exec()` èŕ■åŘěäžŐæšąæĽğèąŇè£ĞäÿĂæăůãĂĆ èęĄæŸŕä¡ăæČşåIJĺåŘŐéĺćçŽĎèőąçŐůäÿ■ä¡£çŤĺåĹř `exec()` æĽğèąŇçżŞæđIJçŽĎèŕĺåřśäijŽæIJL'éŮőécŸäžĘãĂĆ

äÿžäžĘä£őæ■čè£ŹæăůçŽĎéŤŹèŕŕïijŇä¡ăéIJĂèęĄåIJĺèřČçŤĺ `exec()` äźŃåL'■ä¡£çŤĺ `locals()` åĞ¡æŢřæĺěå¿ŮåĹřäÿĂäÿłåśĂéČĺåŔŸéĞŔå■ŮåĚÿãĂĆ äźŃåŘŐä¡ăåřśèČ¡äžŐåśĂéČĺå■ŮåĚÿäÿ■èŐůåŔŰä£őæŤźè£ĞåŘŐçŽĎåŔŸéĞŔåĂijäžĘãĂĆä¿ŃåęĆïijŽ

```
>>> def test():
...     a = 13
...     loc = locals()
...     exec('b = a + 1')
...     b = loc['b']
```

(continues on next page)

```
...     print(b)
...
>>> test()
14
>>>
```

èőÍèőž

åőđéŹĚäÿŁåŕźäžŐ `exec()` çŽĎæ■čçąőä¡£çŤÍæŸŕæŕŤè¿ČéŽ¿çŽĎãĂĆåď´gåď´ŽæŢřæČĚåĘţäÿŃå¡Şä¡ăè
`exec()` çŽĎæŮűåĂŹïijŇ è£ŸæIJĽ´åŔęåď´ŰæŻť´åě¡çŽĎèğčåĘşæŰźæąĹïijĹæŕŤåęĆèčĚéěřåŹÍãĂĄéŮ■åŇĚã

çĎűèĂŇïijŇåęĆæđIJä¡ăäż■çĎűěęĄä¡£çŤÍ `exec()` ïijŇæIJňèŁĆåĹŮåĞžäžĘäÿĂäžŻåęĆä¡Ţæ■čçąőä¡£ç
éżŸèőď´æČĚåĘţäÿŃïijŇ`exec()` äijŽåIJÍèřČçŤÍèĂĚåśĂéČÍåŠŇåĚÍåśĂèŇČåŻť´åĘĚæĽ´ğèąŇäžčçăĄãĂĆçĎű
äijăéĂŠçżŹ `exec()` çŽĎåśĂéČÍèŇČåŻť´æŸŕæŃůèť´Ìåőđé ŹĚåśĂéČÍåŔŸéĞŔçżĎæĹŘçŽĎäÿĂäÿłå■ŮåĚÿãĂ
åŻăæ■d´ïijŇåęĆæđIJ `exec()` åęĆæđIJæĽ´ğèąŇäžĘä£őæŤźæŞ■ä¡IJïijŇè£Źçğ■ä£őæŤźåŘŐçŽĎçżŞæđIJåŕźåć
äÿŃéÌćæŸŕåŔęåď´ŰäÿĂäÿłæijŤçď´žåőČçŽĎä¿Ńå■ŘïijŽ

```
>>> def test1():
...     x = 0
...     exec('x += 1')
...     print(x)
...
>>> test1()
0
>>>
```

äÿŁéÌćäżčçăĄéĞŇïijŇå¡Şä¡ăèřČçŤÍ `locals()` èŐůåŔŰåśĂéČÍåŔŸéĞŔæŮűïijŇä¡ăèŐůå¿ŮçŽĎæŸŕäij
`exec()` çŽĎåśĂéČÍåŔŸéĞŔçŽĎäÿĂäÿłæŃůèť´ÌãĂĆ éĂŽè£ĞåIJÍäżčçăĄæĽ´ğèąŇåŘŐåőąæ§ěè£ŹäÿłåŮ■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
>>>
```

äžŤçżĘèğĆåŕ§æIJĂåŘŐäÿĂæ■ěçŽĎè¿ŞåĞžïijŇéŹď´éÌđä¡ăåřĘ `loc`
äÿ■èćńä£őæŤźåŘŐçŽĎåĂijæĽ´ŃåŁÍèţŃåĂijçżŹxïijŇåŘęåĹŹxåŔŸéĞŔåĂijæŸŕäÿ■äijŽåŔŸçŽĎãĂĆ

åIJÍä¡£çŤÍ `locals()` çŽĎæŮűåĂŹïijŇä¡ăéIJĂèęĄæşÍæĎŔæŞ■ä¡IJéąžåžŘãĂĆærŔæňąåőČèćńèřČçŤÍçż
`locals()` äijŽèŐůåŔŰåśĂéČÍåŔŸéĞŔåĂijäÿ■çŽĎåĂijåźűèęĘçŻŰå■ŮåĚÿäÿ■çŽÿåžŤçŽĎåŔŸéĞŔãĂĆ
èŕůæşÍæĎŔèğĆåŕ§äÿŃäÿŃéÌćè£ŹäÿłèŕŢéłŃçŽĎè¿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}
>>>
```

æşÍæĎŔæIJÄåŘŐäÿĂæňąèřČçŤÍ `locals()` çŽĎæŮűåĂŹxçŽĎåĂijæŸŕåęĆä¡Ţèćńèę€çŻŰæŐL'çŽĎã

ä¡IJäÿž `locals()` çŽĎäÿĂäÿłæŻ£äžčæŰźæąĹïijŇä¡ăåŔŕäžëä¡£çŤĺä¡äèĞłåůšçŽĎå■ŮåĚÿïijŇåźűårĘåő `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¡£çŤÍ `exec()` çŽĎæIJĂä¡şåőđèůţãĂĆ ä¡ăåŔłéIJÄèęĄä£ÌèŕĄåĚĺåśĂåŠŇåśĂéČĺå■ŮåĚÿåIJĺåŘŐéÍćäžčçăĄèő£éŮőæŮűåůšçżŔèćńåĹÌåğŃåŇŰãĂĆ

è£ŸæIJL'äÿĂçĆźïijŇåIJĺä¡£çŤÍ `exec()` äźŃåL'■ïijŇä¡ăåŔŕèČ¡éIJÄèęĄéŮőäÿŃèĞłåůśæŸŕåŘęæIJL'åĚú åď ğåď ŽæŢřæČĚåĘţäÿŃå¡Şä¡ăèęĄèĂČèŹŚä¡£çŤÍ `exec()` çŽĎæŮűåĂŹïijŇ è£ŸæIJL'åŔęåď ŰæŻť'åě¡çŽĎèğčåĘşæŰźæąĹïijŇærŤåęĆèčĚéěřåŹĺãĂĄéÙ■åŇĚãĂĄåĚČçśżïijŇæĹŰåĚűäżÚ

11.24 9.24 èğčæđŘäÿŐåĹĘæđŘPythonæžŘçăĄ

éŮőécŸ

ä¡ăæČşåĘŹèğčæđŘåźűåĹĘæđŘPythonæžŘäžčçăĄçŽĎçÍŃåžŔãĂĆ

èğčåĘşæŰźæąĹ

åď ğéČĺåĹĘçÍŃåžŔåŚŸç§ěéĄŞPythonèČ¡åď §èőąçőÙæĹŰæL'ğèąŇå■Ůçňęäÿšå¡ćåijŔçŽĎæžŘäžčçăĄãĂ

```
>>> 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
>>>
```

åř¡çőąåęĆæ■ď˙ïijŇast æĺąåİŮèČ¡èćńçŤĺæİěåřĘPythonæžŘçăAçijŰèřŚæĹŘäÿĂäÿłåŔřèćńåĹĘæđŘçŽĎa

```
>>> 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¡äèĞłåůśæŻt˙åd˙ŽçŽĎå■ęäźăïijŇåőČæŸŕçŤśäÿĂçşžåĹŮASTèŁĆçĆźżĎ
åĹĘæđŘè£ŹäžŻèŁĆçĆźæIJĂçőĂå■ŢçŽĎæŰźæşŢåřśæŸŕåőŽäźL˙äÿĂäÿłèő£éŮőèĂĚçśżïijŇåőđçŐřå¿Łåd˙Ž
visit_NodeName() æŰźæşŢïijŇ NodeName() åŇźéĚ■éĆčäžŻä¡äæĎ§åĚt˙èűčçŽĎèŁĆçĆźãĂĆäÿŃéIćæ

```
import ast

class CodeAnalyzer(ast.NodeVisitor):
    def __init__(self):
        self.loaded = set()
        self.stored = set()
        self.deleted = set()
```

(continues on next 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)
```

åęĆædIJä¡ăè£Řèą Ňè£ŹäýłçÍŃåžŔïijŇä¡ăäijŽå¿ŮåĹřäýŃéÍćè£ŹæăůçŽĎè¿ŞåĞžïijŽ

```
Loaded: {'i', 'range', 'print'}
Stored: {'i'}
Deleted: {'i'}
```

æIJĂåŘŐïijŇASTåŔřäżěéĂŽè£Ğ `compile()` åĞ¡æŢřæÍěçijŰèŕŚåźűæL'ğèąŇãĂĆä¿ŃåęĆïijŽ

```
>>> exec(compile(top,'<stdin>', 'exec'))
0
1
2
3
4
5
6
7
8
9
>>>
```

èőÍèőž

å¡Şä¡ăèČ¡åď§åĹĘæđŘæžŘäžčçăĄåźűäžŐäÿ■èŐůåŔŰä£ąæĄŕçŽĎæŮűůåĂŹïijÑä¡ăářśèČ¡åĘŹå¿Ĺåď Žäžč
ä¿ÑåęĆïijÑçŻÿæŕŤçŻšçŻőçŽĎäijăéĂŠäÿĂäžŻäžčçăĄçĽĞæőţåĹřçśžäijij
exec() åĞ¡æŢřäÿ■ïijÑä¡ăåŔŕäžěåĚĹářĘåőČè¡ňæ■ćæĹŘäÿĂäÿłASTïijÑ
çĎűůåŘŐèğĆår§åőČçŽĎçżĘèŁĆçIJÑåőČåĹřåžŢæŸŕæĂŐæăůåĄŽçŽĎãĂĆ
ä¡ăè£ŸåŔŕäžěåĘŹäÿĂäžŻåůěåĚůæİěæ§ěçIJÑæ§ŘäÿłæĺąåİŮçŽĎåĚĺéČĺæžŘçăĄïijÑåźűäÿŤåIJĺæ■ď å§žçąĂäÿ

éIJĂèęĄæşĺæĎŔçŽĎæŸŕïijÑåęĆædIJä¡ăç§ěéĄŞèĞłåůśåIJĺåźšåŢěïijÑä¡ăè£ŸèČ¡åď§éĞ■åĘŹASTæİěèą
äÿÑéİćæŸŕäÿĂäÿłèčĚéěřåŹĺä¿Ńå■ŘïijÑåŔŕäžěéĂŽè£ĞéĞ■æŰřèğčæđŘåĞ¡æŢřä¡ŞæžŘçăĄãĂĄ
éĞ■åĘŹASTåźűéĞ■æŰřåĹŽåžžåĞ¡æŢřäžčçăĄåŕźèśąæİěåřĘåĚĺåśĂèő£éŮőåŔŸéĞŔéŹ■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)

```
        # 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¡ăåŔŕäžěåČŔäÿŃéİćè£ŹæăůåĘŹïijŽ

```
INCR = 1
@lower_names('INCR')
def countdown(n):
    while n > 0:
        n -= INCR
```

èčĚéěřåŹÍäijŽářĘ `countdown()` åĞ¡æŢřéĞ■åĘŹäÿžçśżäijijäÿŃéİćè£ŹæăůåŕŘïijŽ

```
def countdown(n):
    __globals = globals()
    INCR = __globals['INCR']
    while n > 0:
        n -= INCR
```

åIJÍæĂğèČ¡æţŃèŕŢäÿ■ïijŇåőČäijŽèőľåĞ¡æŢřè£ŘèąŇå£ń20%

çŐřåIJÍïijŇä¡ăæŸŕäÿ■æŸŕæČşäÿžä¡ăæĽĂæIJL'çŽĎåĞ¡æŢřéČ¡åŁääÿŁè£ŹäÿłèčĚéěřåŹÍåŚćïij§æĹŰèőÿä
ä¡ĘæŸŕïijŇè£Źå■t'æŸŕåŕźäžŐäÿĂäžŻéńŸçžğæŁĂæIJŕæŕŤåęĆASTæŞ■ä¡IJãĂĄæžŘçăĄæŞ■ä¡IJç■Ľç■Ľç■ŽĎ

æIJňèŁĆåŔŮåŔęåď'ŰäÿĂäÿłåIJÍ `ActiveState` äÿ■åď'ĎçŘĘPythonå■ŮèŁĆçăĄçŽĎçńäèŁĆçŽĎåŘŕçc
ä¡£çŤÍASTæŸŕäÿĂäÿłæŻť'åŁäéńŸçžğçĆźçŽĎæŁĂæIJŕïijŇåźűäÿŤäź§æŻť'çőĂå■ŢäžŻãĂĆåŔĆèĂČäÿŃéİćäÿ

11.25 9.25 æŃĘèğčPythonå■ŮèŁĆçăĄ

éŮőécŸ

ä¡ăæČşéĂŽè£ĞářĘä¡ăçŽĎäżčçăĄåŔ■çijŰèŕŚæĹŘä¡ŐçžğçŽĎå■ŮèŁĆçăĄæİěæ§ěçIJŃåőČåžŢåśĆçŽĎå

èğčåĘşæŰźæąĹ

`dis` æÍąåİŮåŔŕäżěèćńçŤÍæİěè¿ŞåĞžäżżä¡ŢPythonåĞ¡æŢřçŽĎåŔ■çijŰèŕŚçżŞædIJãĂĆä¿ŃåęĆïijŽ

```
>>> def countdown(n):
...     while n > 0:
...         print('T-minus', n)
...         n -= 1
...     print('Blastoff!')
...
>>> import dis
>>> dis.dis(countdown)
...
>>>
```

èőÍèőž

å¡Şä¡äæČşèęAç§ěéĄŞä¡ăçŽĎçÍŃåžŘåžŢåśĆçŽĎè£ŘèąŇæIJžåĹűçŽĎæŮűåĂŹïijŇdis æÍąåİŮæŸřå¿ĹæIJL'çŤÍçŽĎãĂĆæŕŤåęĆåęĆæđIJä¡äæČşèŕŢçİĂçŘĘèğčæĂğèČ¡çL'źå¿ĄãĂĆ èćń dis() åĞ¡æŢřèğčæđŘçŽĎåŐ§åğŃå■ŮèŁĆçăĄåęĆäÿŃæL'Ăçđ'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¡äæČşèĞłåůśèğčéĞŁè£Źæőţäžčçă ĄïijŇä¡äéIJĂèęĄä¡£çŤÍäÿĂäžŻåIJÍ opcode æÍąåİŮäÿ■åőŽäźL'çŽĎåÿÿéĞŘãĂĆä¿ŃåęĆïijŽ

```
>>> c = countdown.__code__.co_code
>>> import opcode
>>> opcode.opname[c[0]]
>>> opcode.opname[c[0]]
'SETUP_LOOP'
>>> opcode.opname[c[3]]
'LOAD_FAST'
>>>
```

åěĞæĂłçŽĎæŸřïijŇåIJÍ dis æÍąåİŮäÿ■åźűæšąæIJL'åĞ¡æŢřèől'ä¡ääžěçijŰçÍŃæŰźåijŔå¿Ĺåőźæ ŸŞçŽĎäÿ■è£ĞïijŇäÿ ŃéİćçŽĎçŢ§æĹŘåŹÍåĞ¡æŢřåŘŕäžěåřĘåŐ§åğŃå■ŮèŁĆçăĄåžŘåĹŮè¡ñæ■ćæĹŘ opcodes åŠŇåŔĆæŢřãĂĆ

```
import opcode

def generate_opcodes(codebytes):
    extended_arg = 0
    i = 0
    n = len(codebytes)
    while i < n:
```

(continues on next 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¡£çŤĺæŰźæşŢåęĆäÿŃïijŽ

```
>>> for op, oparg in generate_opcodes(countdown.__code__.co_code):
...     print(op, opcode.opname[op], oparg)
```

è£Źçğ■æŰźåijŔå¿ĹåřŚæIJL'äžžç§ěéĄŞïijÑä¡ăåŔŕäżěåĹľçŤĺåőČæŻ£æ■ćäżżä¡Ţä¡äæČşèęĄæŻ£æ■ćçŽĎäÿŃéİćæĹŚäżñçŤĺäÿĂäÿłçđ'žä¿ŃæİěæijŤçđ'žæŢť'äÿłè£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¡ăåŔŕäžěåČŔè£ŹæăůèĂ■åđ'ğæŃŻèől'èğčéĞŁåŹĺåěŤæžČãĂĆä¡ĘæŸŕïijŇåŕźäžŐçijŰåĘŹæŻt'énŸçžğäi
äžŰäžñåŔŕèČ¡çIJ§çŽĎéIJĂèęĄéĞ■åĘŹå■ÛèŁĆçăĄãĂĆæIJñèŁĆæIJĂåŘŐçŽĎéČĺåĹĘæijŤçđ'žäžĘè£ŹäÿłæŸ
this code on ActiveState

12 çňňå■ĄçńăïijŽæĺąåİŮäÿŐåŇĚ

æĺąåİŮäÿŐåŇĚæŸŕäżżä¡Ţåď ğåđŃçĺŃåžŔçŽĎæäÿå£ČïijŇåřśè£đPythonåőĽèčĚçĺŃåžŔæIJňèžńäź§æŸŕ

Contents:

12.1 10.1 æđĎåżžäÿĂäÿłæĺąåİŮçŽĎåśĆçžğåŇĚ

éŮőéćŸ

ä¡ăæČşåřĘä¡ăçŽĎäżčçăĄçżĎçżĞæĹŘçŤśå¿Ĺåď ŽåĹĘåśĆæĺąåİŮæđĎæĹŘçŽĎåŇĚãĂĆ

èğčåĘşæŰźæąĹ

åřĄèčĚæĹŘåŇĚæŸŕå¿ĹçőĂå■ŢçŽĎãĂĆåIJĺæŰĞäżűçşżçż§äÿŁçżĎçżĞä¡ăçŽĎäżčçăĄïijŇåźűçąőä£Ƀæŕ
ä¿ŃåęĆïijŽ

```
graphics/
    __init__.py
    primitive/
        __init__.py
        line.py
        fill.py
        text.py
    formats/
        __init__.py
        png.py
        jpg.py
```

äÿĂæŮęä¡ăåĄŽåĹřäžĘè£Źäÿ ĂçĆźïijŇä¡ăåžŤèŕéèČ¡åď §æĽ ğèąŇåŘĎçğ■importèŕ■åŔĕïijŇåęĆäÿŃïijŽ

```
import graphics.primitive.line
from graphics.primitive import line
import graphics.formats.jpg as jpg
```

èőĺèőž

åőŽäźĽæĺąåİŮçŽĎåśĆæňąçżŞæđĎåřśåČŔåIJĺæŰĞäżűçşżçż§äÿŁåżžçńŃçŻőå¡ŢçżŞæđĎäÿĂæăůåőźæŸ
æŰĞäżű__init__.pyçŽĎçżŐçŽĎæŸŕèęĄåŇĚåŘńäÿ■åŘŇè£ŘèąŇçžğåĹńçŽĎåŇĚçŽĎåŘŕéĂĽçŽĎåĹİäğŃåŇ
äÿ¿äÿłä¿Ńå■ŘïijŇåęĆæđIJä¡ăæĽ ğèąŇäžĘèŕ■åŘĕimport graphicsïijŇ æŰĞäżűgraphics/__init__.pyåřĘèćńåřijåĚě,åżžçńŃgraphicsåŚ¡åŘ■çľ žéŮť çŽĎåĘĚåőźãĂĆåČŘimport graphics.format.jpgè£ŹæăůåřijåħëïijŇæŰĞäżűgraphics/__init__.pyåŠŇæŰĞäżűgraphics/formats/__init__.py

çżİåď ğéČĺåĹĘæŮűåĂŹèőľ__init__.pyçľ žçİĂåřśåě¡ãĂĆä¡ĘæŸŕæIJĽäžŻæČĚåĘţäÿŃåŘńèČ¡åŇĚåŘńäž
äÿ¿äÿłä¿Ńå■ŘïijŇ__init__.pyèČ¡åď §çŤĺæİěèĞłåŁĺåŁăè¡¡å■ŘæĺąåİŮ:

```
# graphics/formats/__init__.py
from . import jpg
from . import png
```

åČŔè£Źæăůäÿ Ăäÿłæ ŰĞäžű,çŤĺæĹůåŔŕäžëäžĚäžĚéĂŽè£Ğimport grahpics.formatsæİěäžčæŻ£import graphics.formats.jpgäžěåŔŁimport graphics.formats.pngãĂĆ

__init__.pyçŽĎåĚűäžŰåÿÿçŤĺçŤĺæşŢåŇĚæŃňåřĘåď Žäÿłæ ŰĞäžűåŘĹåźűåĹřäÿ Ăäÿłéäžè¿ŚåŠ¡åŘ■çl'žé

æŢŔéŤŘçŽĎçĺŃåžŔåŚŸäijŽåŘŚçŐřïijŇå■şä¡£æšąæIJL'__init__.pyæŰĞäžűå■ŸåIJĺïijŇpythonäž■çĎűä

12.2 10.2 æŐğåĹűæĺąåİŮèćńåĚĺéČĺåŕijåĚěçŽĎåĘĚåőź

éŮőécŸ

å¡Şä¡£çŤĺâĂŹfrom module import *âĂŹ èŕ■åŔěæŮűïijŇåÿŇæIJŻåŕźäžŐæĺąåİŮæĹŰåŇĚåŕijåĞžçŽĎçňę

èğčåĘşæŰźæąĹ

åIJĺä¡äçŽĎæĺąåİŮäÿ■åőŽäźL'äÿĂäÿłåŔŸéĞŔ __all__ æİěæŸŐçąőåIJřåĹŮåĞžéIJĂèęĄåŕijåĞžçŽĎåĘĚå

äÿ¿äÿłä¿Ńå■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¡ĘæŸŕåIJĺåőŽäźL'äžĘåď ğéĞŔåŔŸéĞŔåŘ■çŽĎæĺąåİŮäÿ■écŚçźĄä¡£çŤĺãĂĆ åęĆæđIJä¡ääÿ■åĄŽäžżä¡ŢäžŃ, è£ŹæăůçŽĎåŕijåĚěåřĘäijŽåŕijåĚěæL'ĂæIJL'äÿ■äžěäÿŃåĹŠçž£åijĂåď t'çŽĎã åŘęäÿĂæŰźéIć,åęĆæđIJåőŽäźL'äžĘ __all__ , éĆčäźĹåŔłæIJL'èćńåĹŮäÿ¿åĞžçŽĎäÿIJèě£äijŽèćńåŕijåĞžãĂĆ

åęĆæđIJä¡ăåřĘ __all__ åőŽäźL'æĹŘäÿĂäÿłçl'žåĹŮèąĺ, æšąæIJL'äÿIJèě£åřĘèćńåŕijåĚěãĂĆ åęĆæđIJ __all__ åŇĚåŘńæIJłåőŽäźL'çŽĎåŘ■å■Ů, åIJĺåŕijåĚěæŮűåijŢètůAttributeErrorãĂĆ

12.3 10.3 ä¡£çŤĺçŻÿåŕźèůŕå¿ĎåŘ■åŕijåĚěåŇĚäÿ■å■ŘæĺąåİŮ

éŮőécŸ

åřĘäžčçăĄçżĎçżĞæĹŘåŇĚ,æČşçŤĺimportèŕ■åŔěäžŐåŔęäÿĂäÿłåŇĚåŘ■æšąæIJL'çąňçijŰçăĄè£ĞçŽĎå

èğčåĘşæŰźæąĹ

ä¡£çŤĺåŇĚçŽĎçŻÿåŕźåŕijåĚěïijŇä¡£äÿĂäÿłæĺąåİŮåŕijåĚěåŘŇäÿĂäÿłåŇĚçŽĎåŔęäÿĂäÿłæĺąåİŮ
äÿ¿äÿłä¿Ńå■ŘïijŇåĄĞèő¿åIJĺä¡ăçŽĎæŰĞäžűçşżçż§äÿŁæIJL'mypackageåŇĚïijŇçżĎçżĞåęĆäÿŇïijŽ

```
mypackage/
    __init__.py
    A/
        __init__.py
        spam.py
        grok.py
    B/
        __init__.py
        bar.py
```

åęĆæđIJæĺąåİŮmypackage.A.spamèęĄåŕijåĚěåŘŇçŻőå¡ŢäÿŇçŽĎæĺąåİŮgrokïijŇåőČåžŤèŕěåŇĚæŃňçŽ

```
# mypackage/A/spam.py
from . import grok
```

åęĆæđIJæĺąåİŮmypackage.A.spamèęĄåŕijåĚëäÿ■åŘŇçŻőå¡ŢäÿŇçŽĎæĺąåİŮB.barïijŇåőČåžŤèŕěä¡£çŤĺ

```
# mypackage/A/spam.py
from ..B import bar
```

äÿd'äÿłimportèŕ■åŔééČ¡æšąåŇĚåŘńéąűåśĆåŇĚåŘ■ïijŇèĂŇæŸŕä¡£çŤĺäžĘspam.pyçŽĎçŻÿåŕźèůŕå¿ĎãĂ

èőÍèőž

åIJĺåŇĚåĘĚïijŇæŮćåŔŕäžěä¡£çŤĺçŻÿåŕźèůŕå¿Ďäź§åŔŕäžěä¡£çŤĺçżÌåŕźèůŕå¿ĎæÌěåŕijåĚěãĂĆ
äÿ¿äÿłä¿Ńå■ŘïijŽ

```
# mypackage/A/spam.py
from mypackage.A import grok # OK
from . import grok # OK
import grok # Error (not found)
```

åČŔmypackage.Aè£Źæăůä¡£çŤĺçżÌåŕźèůŕå¿ĎåŘ■çŽĎäÿ■åĹl'äźŃåd'ĎæŸŕè£ŹåřĘéąűåśĆåŇĚåŘ■çąňçijŮ
äÿ¿äÿłä¿Ńå■ŘïijŇåęĆæđIJä¡ăæŤźåŔŸäžĘåŇĚåŘ■ïijŇä¡ăåřśå£ĚéążæčĂæ§ěæL'ĂæIJL'æŰĞäžűæÌěä£őæ■čæ
åŘŇæăůïijŇçąňçijŰçăĄçŽĎåŘ■çğřäijŽä¡£çğżåĹÍäžčçăĄåŔŸå¿ŮåŻŕéŽ¿ãĂĆäÿ¿äÿłä¿Ńå■ŘïijŇäź§èőÿæIJL'ä
åęĆæđIJä¡£çŤĺçŻÿåŕźåŕijåĚěïijŇéĆčäÿĂåĹĞéČ¡okïijŇçĎűèĂŇä¡£çŤĺçżÌåŕźèůŕå¿ĎåŘ■å¿ĹåŔŕèČ¡äijŽåĞžéŮ

importèŕ■åŔěçŽĎ . åŠŇ . . çIJŃèţůæİěå¿ĹæżŚçĺ¡, ä¡ĘåőČæŇĞåőŽçŻőå¡ŢåŘ■.äÿžå¡ŞåĽ■çŻőå¡ŢïijŇ..BäÿžçŻőå¡Ţ../BãĂĆè£Źçğ■èŕ■æşŢåŔłéĂĆçŤĺäžŐimport äÿ¿äÿłä¿Ńå■ŘïijŽ

```
from . import grok # OK
import .grok # ERROR
```

åř¡çőąä¡£çŤĺçŽÿåŕźåŕijåĚěçIJŃèţůæİěåČŔæŸŕæţŔèğĹæŰĞäžűçşżçż§ïijŇä¡ĘæŸŕäÿ■èČ¡åĹřåőŽäźĽåŇĚ

æIJĂåŘŐïijŇçŽÿåŕźåŕijåĚěåŔłéĂĆçŤĺäžŐåIJĺåŘĹéĂĆçŽĎåŇĚäÿ■çŽĎæĺąåİŮãĂĆård'åĚűæŸŕåIJĺéąűås ä¿ŃåęĆïijŽ

```
% python3 mypackage/A/spam.py # Relative imports fail
```

åŔęäÿĂæŰźéİćïijŇåęĆædIJä¡ăä¡£çŤĺPythonçŽĎ-méĂĽéąźæİěæĽğèąŇåĚĹåĽ■çŽĎèĎŽæIJñïijŇçŽÿåŕ ä¿ŃåęĆïijŽ

```
% python3 -m mypackage.A.spam # Relative imports work
```

æŻť'åd'ŽçŽĎåŇĚçŽĎçŽÿåŕźåŕijåĚěçŽĎèČŇæŹŕç§ěèŕE,èŕůçIJŃ PEP 328 .

12.4 10.4 åřĘæĺąåİŮåĹĘåĽšæĹŘåd'Žäÿłæ ŰĞäžű

éŮőécŸ

ä¡ăæČşåřĘäÿĂäÿłæĺąåİŮåĹĘåĽšæĹŘåd'Žäÿłæ ŰĞäžűãĂĆä¡ĘæŸŕä¡ăäÿ■æČşåřĘåĹĘçężçŽĎæŰĞäžűçż

èğčåĘşæŰźæąĹ

çĺŃåžŔæĺąåİŮåŔŕäżěéĂŽè£ĞåŔŸæĹŘåŇĚæİěåĹĘåĽšæĹŘåd'ŽäÿłçŃñçńŃçŽĎæŰĞäžűãĂĆèĂČèŹŚäÿ

```
# mymodule.py
class A:
    def spam(self):
        print('A.spam')

class B(A):
    def bar(self):
        print('B.bar')
```

åĄĞèő¿ä¡ăæČşmymodule.pyåĹĘäÿžäÿďäÿłæŰĞäžűïijŇæŕŔäÿłåőŽäźĽçŽĎäÿĂäÿłçśžãĂĆèęĄåĄŽåĹřè è£Źè£ŹäÿłçŻőå¡ŢäÿŃïijŇåĹŻåžžäżěäÿŃæŰĞäžűïijŽ

```
mymodule/
    __init__.py
    a.py
    b.py
```

åIJĺa.pyæŰĞäžűäÿ■æŘŠåĚěäżěäÿŃäžčçăĄïijŽ

```
# a.py
class A:
    def spam(self):
        print('A.spam')
```

åIJÍb.pyæŰĞäžűäÿ■æŔŠåĚěäžěäÿŃäžčçăĄïijŽ

```
# b.py
from .a import A
class B(A):
    def bar(self):
        print('B.bar')
```

æIJĂåŘŐïijŇåIJÍ __init__.py äÿ■ïijŇářĘ2äÿłæŰĞäžűçšŸåŘĹåIJÍäÿĂèţůïijŽ

```
# __init__.py
from .a import A
from .b import B
```

åęĆæđIJæŇL'çĚğè£ŹäžŻæ■ěéłd'ïijŇæL'Ăäžğç Ť§çŽĎåŇĚMyModuleářĘä¡IJäÿžäÿĂäÿłå■ŢäÿĂçŽĎéÄż

```
>>> import mymodule
>>> a = mymodule.A()
>>> a.spam()
A.spam
>>> b = mymodule.B()
>>> b.bar()
B.bar
>>>
```

èőÍèőž

åIJÍè£Źäÿłçńäè ŁĆäÿ■çŽĎäÿżèęĄéŮőécŸæŸŕäÿĂäÿłèő¿èőąéŮőécŸïijŇäÿ■çőąä¡ăæŸŕåŘęåÿŇæIJŻçŤÍæ

```
from mymodule.a import A
from mymodule.b import B
...
```

è£ŹæăůèČ¡åůëä¡IJïijŇä¡Ęè£ŹèőľçŤÍæĹůæL'£åŔŮæŻť'åd'ŽçŽĎèť§æŃĚïijŇçŤÍæĹůèęĄç§ěéĄŞäÿ■åŘŇ

```
from mymodule import A, B
```

åŕźåŘŐèĂĚèĂŇèÍĂïijŇèőľ mymoduleæĹŘäÿžäÿĂäÿłåd'ğçŽĎæžŘæŰĞäžűæŸŕæIJĂåÿÿèğĄçŽĎãĂĆä¡Ì è£ŹæăůåĄŽçŽĎåĚşéŤőæŸŕåĹŻåžžäÿĂäÿłåŇĚçŽőå¡ŢïijŇä¡£çŤÍ __init__.py æŰĞäžűæİěårĘæŕŔéČÍåĹĘçšŸåŘĹåIJÍäÿĂèţůãĂĆ

å¡ŞäÿĂäÿłæÍąåİŮèćńåĹĘąL'šïijŇä¡äéIJĂèęĄçL'źåĹńæşÍæĎŔäžd'åŔL'åijŢçŤÍçŽĎæŰĞäžűůŘ■ãĂĆäÿ¿äÿ from .a import A æİěèŐůåŔŰãĂĆ

æŢt'äÿłçńäèŁĆéČ¡ä¡£çŤÍåŇĚçŽĎçŻÿářźářijåĚěæİěéĄ£åĚ■årĘéąűåśĆæÍąåİŮåŘ■çąňçijŰçăĄåĹřæžŘäžč

ä¡IJäÿžè£ŹäÿĂçńäèŁĆçŽĎåžűäijÿïijŇåřĘäżŃçż■åžűè£§åŕijåĚěãĂĆåęĆåŻ¿æL'Ăçd'žïijŇ__init__.pyæŰ
èęĄåĄŽåĹřè£ŹäÿĂçĆźïijŇ__init__.pyæIJL'çżĘå¿őçŽĎåŔŸåŇŰïijŽ

```
# __init__.py
def A():
    from .a import A
    return A()

def B():
    from .b import B
    return B()
```

åIJÍè£ŹäÿłçL'ĹæIJñäÿ■ïijŇçśżAåŠŇçśżBècńæŻ£æ■ćäÿžåIJÍçňňäÿĂæňąèő£éŮőæŮűåŁäè¡¡æL'ĂéIJĂçŽĎ
ä¿ŃåęĆïijŽ

```
>>> import mymodule
>>> a = mymodule.A()
>>> a.spam()
A.spam
>>>
```

åžűè£§åŁäè¡¡çŽĎäÿżèęĄçijžçĆźæŸŕçżğæL'£åŠŇçśżåđŃæčĂæ§ěåŔŕèČ¡äijŽäÿ■æŰ■ãĂĆä¡ăåŔŕèČ¡äijŽ

```
if isinstance(x, mymodule.A): # Error
...

if isinstance(x, mymodule.a.A): # Ok
...
```

åžűè£§åŁäè¡¡çŽĎçIJ§åőđä¿Ńå■Ř, èğĄæăĞåĞĘåžŞ multiprocessing/__init__.py
çŽĎæžŘçăĄ.

12.5 10.5 åĹľçŤĺåŚ¡åŘ■çľžéŮťåŕijåĚěçŻőå¡ŢåĹĘæŢčçŽĎäżčçăĄ

éŮőécŸ

ä¡ăåŔŕèČ¡æIJL'åd'ğéĞŔçŽĎäżčçăĄïijŇçŤšäÿ■åŘŇçŽĎäžžæİěåĹĘæŢčåIJřçżt'æŁd'ãĂĆæŕŔäÿłéČĺåĹĘè

èğčåĘşæŰźæąĹ

äžŐæIJñèt'ĺäÿŁèőšïijŇä¡äèęĄåőŽäźL'äÿĂäÿłéąűçžğPythonåŇĚïijŇä¡IJäÿžäÿĂäÿłåd'ğéŻĘåŘĹåĹĘåijĂç

åIJÍçż§äÿĂäÿ■åŘŇçŽĎçŻőå¡ŢéĞŇçż§äÿĂçŻÿåŘŇçŽĎåŚ¡åŘ■çľžéŮťïijŇä¡ĘæŸŕèęĄåĹăåŐżçŤĺæİěåřŀ

```
foo-package/
    spam/
        blah.py
```

(continues on next page)

(çż■äÿŁéąţ)

```
bar-package/
    spam/
        grok.py
```

åIJÍè£Ź2äÿłçŽőå¡ŢéĞŃïijŃéČ¡æIJĽçİĂåĔśåŘŃçŽĎåŚ¡åŘ■çľžéŮť spamãĂĆåIJÍäżżä¡ŢäÿĂäÿłçŽőå¡ŢéČ

èőľ æĹŚäżňçIJŃçIJŃïijŇåęĆæđIJåřĘfoo-packageåŠŇbar-packageéČ¡åŁăåĹřpythonæĺąåİÜèůŕå¿Ďåźűåř İèŕŢåŕijåĚěäijŽåŔŚçŤ§äżĂäźĹ

```
>>> import sys
>>> sys.path.extend(['foo-package', 'bar-package'])
>>> import spam.blah
>>> import spam.grok
>>>
```

äÿď äÿłäÿ■åŘŇçŽĎåŇĔçŽőå¡ŢèćńåŘĹåźűåĹřäÿĂèţůïijŇä¡ääŔŕäżěåŕijåĚěspam.blahåŠŇspam.grokïijŇå

èőĺèőž

åIJÍè£ŹéĞŇåůĕä¡IJçŽĎæIJžåĹűèćńçğřäÿžâĂIJåŇĚåŚ¡åŘ■çľžéŮťâĂİçŽĎäÿĂäÿłçĽźå¿ĄãĂĆäżÕæIJñè

åŇĔåŚ¡åŘ■çľžéŮťçŽĎåĚşéŤőæŸŕçąőä£İéąűçžğçŽőå¡ŢäÿæšąæIJĽ__init__.pyæŰĞäżűæİěä¡IJäÿžåĚśåäÿ¿äÿłä¿Ńå■ŘïijŽ

```
>>> import spam
>>> spam.__path__
_NamespacePath(['foo-package/spam', 'bar-package/spam'])
>>>
```

åIJÍåőŽä¡■åŇĔçŽĎå■ŘçżĎäżűæŮűïijŇçŽőå¡Ţ__path__åřĘèćńçŤĺåĹř(ä¿Ńåę Ć, å¡Şåŕijå Ěěspam.grokæĹŰèĂĚspam.blahçŽĎæŮűåĂŹ).

åŇĔåŚ¡åŘ■çľžéŮťçŽĎäÿĂäÿłéĞ■èęAçL'źçĆźæŸŕäżżä¡ŢäžžéČ¡åŔŕäżěçŤĺèĞłåůśçŽĎäżčçăĄæİěæL'ľå

```
my-package/
    spam/
        custom.py
```

åęĆæđIJä¡ääřĘä¡ăçŽĎäżčçăĄçŻőå¡ŢåŠŇåĚűäżŰåŇĔäÿĂèţůæůżåŁăåĹřsys.pathïijŇè£ŹåřĘæŮăçijİåIJřå

```
>>> import spam.custom
>>> import spam.grok
>>> import spam.blah
>>>
```

äÿĂäÿłåŇĔæŸŕåŘęèćńä¡IJäÿžäÿĂäÿłåŇĔåŚ¡åŘ■çľžéŮťçŽĎäÿżèęAæŰźæşŢæŸŕæčĂæ§ěåĚű__file__å

```
>>> spam.__file__
Traceback (most recent call last):
```

(continues on next 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£ąæĄŕåŔŕäžěæ§ěçIJŃ PEP 420.

12.6 10.6 éĞ■æŰřåŁäè¡¡æĺąåİŮ

éŮőécŸ

ä¡ăæČşéĞ■æŰřåŁäè¡¡åůšçżŔåŁäè¡¡çŽĎæĺąåİŮïijŇåŻääýžä¡ăářźåĚűæžŘçăĄè£ŻèąŇäžĘä£őæŤźãĂĆ

èğčåĘşæŰźæąĹ

ä¡£çŤĺimp.reload()æİěéĞ■æŰřåŁäè¡¡åĚĹåL'■åŁäè¡¡çŽĎæĺąåİŮãĂĆäÿ¿äÿłä¿Ńå■ŘïijŽ

```
>>> import spam
>>> import imp
>>> imp.reload(spam)
<module 'spam' from './spam.py'>
>>>
```

èőÍèőž

éĞ■æŰřåŁäè¡¡æĺąåİŮåIJĺåijĂåŔŚåŠŇèřČèŕŢè£ĞçÍŃäÿ■åÿÿåÿÿå¿ĹæIJL'çŤĺãĂĆä¡ĘåIJĺçŤ§äžğçŐŕåćČä

reload()æŞęéŹd'äžĘæĺąåİŮåžŢåśĆå■ŮåĚÿçŽĎåĘĚåőźïijŇåźűéĂŽè£ĞéĞ■æŰřæL'ğèąŇæĺąåİŮçŽĎæžŘ

åř¡çőąåęĆæ■d'ïijŇreload()æšąæIJL'æŻť'æŰřåČŔâĂİfrom module import nameâĂİè£ŹæăůäĄ£çŤĺimportèŕ■åŘěåŕijåĚěçŽĎåőŽäźL'ãĂĆäÿ¿äÿłä¿Ńå■ŘïijŽ

```
# spam.py
def bar():
    print('bar')

def grok():
    print('grok')
```

çŐřåIJĺåŘŕåŁĺäžd'äžŠåijŔäijŽèŕİïijŽ

```
>>> import spam
>>> from spam import grok
>>> spam.bar()
```

(continues on next page)

```
bar
>>> grok()
grok
>>>
```

äÿ■éĂĂåĞžPythonä£őæŤźspam.pyçŽĎæžŘçăĄïijŇářEgrok()åĞ¡æŢřæŤźæĹŘè£Źæăůïij Ž

```
def grok():
    print('New grok')
```

çŐřåIJĺåŻđåĹřäžd'äžŠåijŔäijŽèŕİïijŇéĞ■æŰřåŁäè¡¡æĺąåİŮïijŇářİèŕŢäÿŃè£Źäÿłåőđél Ñïij Ž

```
>>> 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
>>>
```

åIJÍè£Źäÿłä¿Ńå■Řäÿ■ïijŇä¡ăçIJŃåĹřæIJL'2äÿłçL'ĹæIJńçŽĎgrok()åĞ¡æŢřèćńåŁäè¡¡ãĂĆéĂŽåÿÿæİèèŕt'ï

åŻăæ■d'ïijŇåIJíçŤ§äžğçŐŕåćČäÿ■åŔŕèČ¡éIJĂèęĄéĄ£åĚ■éĞ■æŰřåŁäè¡¡æĺąåİŮãĂĆåIJĺäžd'äžŠçŐŕåćČ

12.7 10.7 è£ŘèąŇçŻőå¡ŢæĹŰåŐŃçijľæŰĞäžű

éŮőécŸ

æĆĺæIJL'äÿĂäÿłåůšæĹŘéŢ£äÿžåŇĚåŘńåd'Žäÿłæ ŰĞäžűçŽĎåžŤçŤĺïijŇåőČåůšè£IJäÿ■åĘ■æŸŕäÿĂäÿłçő

èğčåĘşæŰźæąĹ

åęĆædIJä¡ăçŽĎåžŤçŤĺçĺŃåžŔåůšçżŔæIJL'åd'Žäÿłæ ŰĞäžűïijŇä¡ääŔŕäžěæŁŁä¡ăçŽĎåžŤçŤĺçĺŃåžŔæŤ¿
äÿ¿äÿłä¿Ńå■ŘïijŇä¡ääŔŕäžěåČŔè£Źæăůå ĹŻåžžçŻőå¡Ţïij Ž

```
myapplication/
    spam.py
    bar.py
    grok.py
    __main__.py
```

åęĆædIJ__main__.pyå■ŸåIJĺïijŇä¡ääŔŕäžěçőĂå■ŢåIJřåIJĺéąűçžğçŻőå¡Ţè£ŘèąŇPythonèğčéĞŁåŹĺïijŽ

```
bash % python3 myapplication
```

èğčéĞŁåŹÍåřĘæL'ğèąŇ__main__.pyæŰĞäžűä¡IJäÿžäÿżçÍŃåžŔãĂĆ

åęĆæđIJä¡ăåřĘä¡ăçŽĎäžčçăĄæL'ŞåŇĚæĹŘzipæŰĞäžűïijŇè£Źçğ■æŁĂæIJŕåŘŇæăůäź§éĂĆçŤĺïijŇäÿ¿ă

```
bash % ls
spam.py bar.py grok.py __main__.py
bash % zip -r myapp.zip *.py
bash % python3 myapp.zip
... output from __main__.py ...
```

èőÍèőž

åĹŻåžžäÿĂäÿłçŻőå¡ŢæĹŰzipæŰĞäžűåźűæůżåŁă__main__.pyæŰĞäžűæİěårĘäÿĂäÿłæŽť åď ğçŽĎPytho

çŤśäžŐçŻőå¡ŢåŠŇzipæŰĞäžűäÿŐæ■čåÿÿæŰĞäžűæIJL'äÿĂçĆźäÿ■åŘŇïijŇä¡ăåŔŕèČ¡è£ŸéIJĂèęĄáćđå

```
#!/usr/bin/env python3 /usr/local/bin/myapp.zip
```

12.8 10.8 èŕżåŔŰä¡■äžŐåŇĚäÿ■çŽĎæŢřæ■őæŰĞäžű

éŮőécŸ

ä¡ăçŽĎåŇĚäÿ■åŇĚåŘńäžčçăĄéIJĂèęĄåŐżèŕżåŔŰçŽĎæŢřæ■őæŰĞäžűãĂĆä¡ăéIJĂèęĄåř¡åŔŕèČ¡åIJřç

èğčåĘşæŰźæąĹ

åĄĞèő¿ä¡ăçŽĎåŇĚäÿ■çŽĎæŰĞäžűçżĎçżĞæĹŘåęĆäÿŃïijŽ

```
mypackage/
    __init__.py
    somedata.dat
    spam.py
```

çŐřåIJĺåĄĞèő¿spam.pyæŰĞäžűéIJĂèęĄèŕżåŔŰsomedata.datæŰĞäžűäÿ■çŽĎåĘĚåőźãĂĆä¡ăåŔŕäžěçŤĺ

```
# spam.py
import pkgutil
data = pkgutil.get_data(__package__, 'somedata.dat')
```

çŤśæ■d'äžğçŤ§çŽĎåŔŸéĞŔæŸŕåŇĚåŘńèřěæŰĞäžűçŽĎåŐ§ågŃåĘĚåőźçŽĎå■ŮèŁĆå■Ůçňęäÿšã

èőÍèőž

èęĄèŕżåŔŰæŢřæ■őæŰĞäżűïijŇä¡ääŔŕèČ¡äijŽåĂ¿åŘŚäžŐçijŰåĘŹä¡£çŤĺåĘĚç¡őçŽĎI/OåŁ§èČ¡çŽĎäžčçăĄïijŇåęĆopen()ãĂĆä¡ĘæŸŕè£Źçğ■æŰźæşŢäź§æIJL'äÿĂäžŻéŮőécŸãĂĆ

éęŰåĚĹïijŇäÿĂäÿłåŇĚåŕźèğčéĞŁåŹĺçŽĎå¡ŞåL'■åůëä¡IJçŻőå¡ŢåĞääźŐæšąæIJL'æŐğåĹűæİČãĂĆåŻäæ

çňňäžŇïijŇåŇĚéĂŽåÿÿåőL'èčĚä¡IJäÿž.zipæĹŰ.eggæŰĞäżűïijŇè£ŹäžŻæŰĞäżűåźűäÿ■åČŔåIJĺæŰĞäżű

pkgutil.get_data()åĞ¡æŢřæŸŕäÿĂäÿłèŕżåŔŰæŢřæ■őæŰĞäżűçŽĎéńŸçžğåůěåĚůïijŇäÿ■çŤĺçőąåŇĚæŸŕå

get_data()çŽĎçňňäÿĂäÿłåŔĆæŢřæŸŕåŇĚåŘńåŇĚåŘ■çŽĎå■Ůçňęäÿšã ĂĆä¡ääŔŕäžěçŻt'æŐěä¡£çŤĺåŇĚ

12.9 10.9 åřĘæŰĞäżűåď źåŁääĚěåĹřsys.path

éŮőécŸ

ä¡ăæŮăæşŢåŕijåĚěä¡ăçŽĎPythonäžčçăĄåŻääÿžåőČæL'ĂåIJĺçŽĎçŻőå¡Ţäÿ■åIJĺsys.pathéĞŇãĂĆä¡ăæČş

èğčåĘşæŰźæąĹ

æIJL'äÿd'çğ■åÿÿçŤĺçŽĎæŰźåijŔåřĘæŰřçŻőå¡ŢæůżåŁääĹřsys.pathãĂĆçňňäÿĂçğ■ïijŇä¡ääŔŕäžěä¡£çŤĺ

```
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', ...]
>>>
```

åIJĺèĞłåőŽäźL'åžŤçŤĺçĺŃåžŔäÿ■ïijŇè£ŹæăůçŽĎçŐŕåćČåŔŸéĞŔåŔŕåIJĺçĺŃåžŔåŘŕåŁĺæŮűèő¿ç¡őæĹŰ

çňňäžŇçğ■æŰźæşŢæŸŕåĹŻåžžäÿĂäÿł.pthæŰĞäżűïijŇåřĘçŻőå¡ŢåĹŮäÿ¿åĞžæİěïijŇåČŔè£ŹæăůïijŽ

```
# myapplication.pth
/some/dir
/other/dir
```

è£Źäÿł.pthæŰĞäżűéIJĂèęĄæŤ¿åIJĺæ§ŘäÿłPythonçŽĎsite-packagesçŽőå¡ŢïijŇéĂŽåÿÿä¡■äžŐ/usr/local/lib/python3.3/site-packages æĹŰèĂĚ ~/.local/lib/python3.3/sitepackagesãĂĆå¡ŞèğčéĞŁåŹĺåŘŕåŁĺæŮűïijŇ.pthæŰĞäżűéĞŇåĹŮäÿ¿åĞžæİěçŽĎå■ŸåIJ

èőÍèőž

æŕŤètůèť źåŁŽåIJřæL'¿æŰĞäżűïijŇä¡ääŔŕèČ¡äijŽåĂ¿åŘŚäžŐåĘŹäÿĂäÿłäžčçăĄæL'ŃåŁĺèřČèŁĆsys.pat

```
import sys
sys.path.insert(0, '/some/dir')
sys.path.insert(0, '/other/dir')
```

èŹ¡çĎűè£ŹèČ¡âĂĲåůěä¡ĲâĂİïijÑä¡ĘæŸŕåĲĺåőđèůţäÿ■æđĄäÿžèĎĘåijśïijÑåžŤåř¡éĞŔéĄ£åĚ■ä¡£çŤĺâĂ

```
import sys
from os.path import abspath, join, dirname
sys.path.insert(0, join(abspath(dirname(__file__)), 'src'))
```

è£ŹåřĘsrcçŻőå¡ŢæůżåŁăåĹřpathéĞŇïijÑåŠŇæĽ§èąŇæŔŠåĚěæ■ěéłď çŽĎäżčçăĄåĲĺåŘŇäÿĂäÿłçŻőå¡Ţ

site-packagesçŻőå¡ŢæŸŕçňňäÿĽæŰźåŇĚåŠŇæĺąåİŮåőĽèčĚçŽĎçŻőå¡ŢãĂĆåęĆæđĲä¡ăæĽŃåŁĺåőĽèč packagesçŻőå¡ŢãĂĆèŹ¡çĎűçŤĺäžŐéĚ■ç¡őpathçŽĎ.pthæŰĞäżűå£ĚéążæŤ¿ç¡őåĲĺsite-packageséĞŇïijÑä¡ĘåőČéĚ■ç¡őçŽĎèůŕå¿ĎåŔŕäžěæŸŕçşżç§äÿŁäżżä¡Ţä¡ăåÿÑæĲŻçŽĎçŻőå¡ŢãĂĆåŽăæ■d'

12.10 10.10 éĂŽè£Ğå■Ůçňęäÿšå Ř■åřijåĚěæĺąåİŮ

éŮőéćŸ

ä¡ăæČşåřijåĚěäÿĂäÿłæĺąåİŮïijÑä¡ĘæŸŕæĺąåİŮçŽĎåŘ■å■ŮåĲĺå■Ůçňęäÿšé ĞŇãĂĆä¡ăæČşåŕźå■Ůçňęäÿ

èğčåĘşæŰźæąĹ

ä¡£çŤĺimportlib.import_module()åĞ¡æŢřæĽŃåŁĺåřijåĚěåŘ■å■Ůäÿžå■Ůçňęäÿšçż ŹåĞžçŽĎäÿĂäÿłæ

```
>>> 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■ŢåĲřæĽ§èąŇåŠŇimportçŻÿåŘŇçŽĎæ■ěéłď ïijÑä¡ĘæŸŕè£ŤåŽđçŢ§æĹŘç

åęĆæđĲä¡ăæ■čåĲĺä¡£çŤĺçŽĎåŇĚïijŇimport_module()äź§åŔŕçŤĺäžŐçŻÿåŕźåřijåĚěãĂĆä¡ĘæŸŕïijÑä¡ăé

```
import importlib
# Same as 'from . import b'
b = importlib.import_module('.b', __package__)
```

èőĺèőž

ä¡£çŤĺimport_module()æĽŃåŁĺåřijåĚěæĺąåİŮçŽĎéŮőéćŸéĂŽåÿÿåĞžçŐřåĲĺäžěæ§Ŕçğ■æŰźåijŔçijŰå

åIJÍæÛğçŽĎäžčçăĄïijŇæIJL'æŮűä¡ăäijŽçIJŃåĹřçŤÍäžŐářijåĚěçŽĎåĘĚåžžåĞ¡æŢř__import__()ãĂĆår¡ç
éĂŽåÿÿæŻť'åőźæŸŞä¡£çŤÍãĂĆ

èĞłåőŽäźL'årijåĚèè£ĞçÍŃçŽĎéńŸçžğåőđä¿ŃèğĄ10.11årŕÈłĆ

12.11 10.11 éĂŽè£ĞéŠľå■Řè£lJçÍŃåŁäè¡¡æÍąåİů

éŮőécŸ

ä¡ăæČşèĞłåőŽäźL'PythonçŽĎimportèŕ■åŘěïijŇä¡£å¿ŮåőČèČ¡äžŐè£IJçÍŃæIJžåŹÍäÿŁéİćéĂŔæŸŐçŽĎå

èğčåĘşæŰźæąĹ

éęŰåĚĹèęĄæŔŘåĞžæİěçŽĎæŸŕåőL'åĚĺéŮőécŸãĂĆæIJňèŁĆèőÍèőžçŽĎæĂİæČşåęĆæđIJæšąæIJL'äÿĂ
äź§åřśæŸŕèŕť'ïijŇæĹŚäžňçŽĎäÿżèęĄçŽőçŽĎæŸŕæůšåĚěåĹĘæđŘPythonçŽĎimportèŕ■åŔěæIJžåĹűãĂĆ
åęĆæđIJä¡ăçŘĘèğčäžĘæIJňèŁĆåĘĚéČĺåŐ§çŘĘïijŇä¡ăåřśèČ¡åd'§äÿžåĚűäžŰäžžä¡ŢçŻőçŽĎèĂŇèĞłåőŽäźL'i
æIJL'äžĘè£ŹäžŻïijŇèől'æĹŚäžňçżğçż■åŘŚåL'■èţřãĂĆ

æIJňèŁĆæăÿå£ČæŸŕèő¿èőąåŕijåĚěèŕ■åŘěçŽĎæL'l'åśŢåŁ§èČ¡ãĂĆæIJL'å¿Ĺåd'Žçğ■æŰźæşŢåŘŕäžěåĄŽ
äÿ■è£ĞäÿžäžĘæijŤçd'žçŽĎæŰźä¿£ïijŇæĹŚäžňåijĂåğŃåĚĹæđĎéĂăäÿŃéİćè£ŹäÿłPythonäžčçăĄçżŞæđĎïijŽ

```
testcode/
    spam.py
    fib.py
    grok/
        __init__.py
        blah.py
```

è£ŹäžŻæŰĞäžűçŽĎåĘĚåőźåźűäÿ■éĞ■èęĄïijŇäÿ■è£ĞæĹŚäžňåIJÍæŕŘäÿłæŰĞäžűäÿ■æŤ¿åĚěäžĘåřŚéĞ
è£Źæăůä¡ăåŘŕäžěæţŃèŕŢåőČäžňåźűæ§ěçIJŃå¡ŞåőČäžňècńåŕijåĚěæŮűçŽĎè¿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)

```
# grok/blah.py
print("I'm grok.blah")
```

è£ŹéĞŇçŽĎçŻőçŽĎæŸŕåĚĄèőÿè£ŹäžŽæŰĞäżűä¡IJäÿžæÍąåİŮèćńè£IJçÍŃèő£éŮőãĂĆ
äź§èőÿæIJĂçőĂå■ŢçŽĎæŰźåijŔářśæŸŕářĘåőČäžňåŘŠåÿČåĹřäÿĂäÿłwebæIJ■åŁąåŹÍäÿŁéİćãĂĆåIJÍtestcode

```
bash % cd testcode
bash % python3 -m http.server 15000
Serving HTTP on 0.0.0.0 port 15000 ...
```

æIJ■åŁąåŹÍè£ŘèąŇèţůæİěåŘŐåĘ■åŘŕåŁÍäÿĂäÿłå■ŢçŃňçŽĎPythonèğčéĞŁåŹÍãĂĆ
çąőä£İä¡ăåŔŕäžěä¡£çŤÍ `urllib` èő£éŮőåĹře£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)
>>>
```

äžŐè£ŹäÿłæIJ■åŁąåŹÍåŁăè¡¡æžŘäžčçăĄæŸŕæŐěäÿŃæİěæIJňèŁĆçŽĎå§žçąĂãĂĆ
äÿžäžĘæŻ£äžčæL'ŃåŁÍçŽĎéĂŽè£Ğ `urlopen()` æİěæŤűéŻĘæžŘæŰĞäżűïijŇ
æĹŠäžňéĂŽè£ĞèĞłåőŽäźL'importèŕ■åŘěæİěåIJÍåŘŐåŘŕèĞłåŁÍåÿőæĹŠäžňåĄŽåĹřãĂĆ

åŁăè¡¡è£IJçÍŃæÍąåİŮçŽĎçňňäÿĂçğ■æŰźæşŢæŸŕåĹŻåžžäÿĂäÿłæŸ¿åijŔçŽĎåŁăè¡¡åĞ¡æŢřæİěåőŇæĹŘ

```
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¡¡æžŘäžčçăĄïijŇåźűä¡£çŤÍ `compile()`
åřĘåĚűçijŰèŕŠåĹřäÿĂäÿłäžčçăĄåŕźèśąäÿ■ïijŇ çĎűåŘŐåIJÍäÿĂäÿłæŰřåĹŻåžžçŽĎæÍąåİŮåŕźèśąçŽĎå■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¡äæĽĂèğĄïijŇåŕźäžŐçőĂå■ŢçŽĎæĺąåİŮè£Źäÿłæ Ÿŕèą Ňå¿ŮéĂŽçŽĎãĂĆ
äÿ■è£ĞåőČåźűæšąæIJĽåţŇåĚěåĹřéĂŽåÿÿçŽĎimportèŕ■åŔěäÿ■ïijŇåęĆæđIJèęĄæŤŕæŇĄæŻť'éńŸçžğçŽĎçż

äÿĂäÿłæŻť'éĚůçŽĎåĄŽæşŢæŸŕåĹŽåżžäÿĂäÿłèĞłåőŽäźĽ'åŕijåĚěåŹĺãĂĆçňňäÿĂçğ■æŰźæşŢæŸŕåĹŽåż

```
# 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)

```
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)

```
            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)

```
            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)
```

äýŃéÍćæŸŕäýĂäýłäžď'äžŠäijŽèŕÍïijŇæijŤçď'žäžĘåęĆä¡Ţä¡£çŤĺåL'■éÍćçŽĎäżčçăĄïijŽ

```
>>> # 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)

```
I'm spam
>>> import grok.blah
I'm grok.__init__
I'm grok.blah
>>> grok.blah.__file__
'http://localhost:15000/grok/blah.py'
>>>
```

è£ŹäÿłçL'źæőŁçŽĎæŰźæąŁäijŽåőL'èčĔäÿĂäÿłçL'źåĹńçŽĎæ§ěæL'¿åŹÍ UrlMetaFinder åőđä¿ŃïijŇ ä¡IJäÿž sys.meta_path äÿ■æIJĂåŘŐçŽĎåőđä¡ŞãĂĆ å¡ŞæĺąåİŮèćńåŕijåĚěæŮűïijŇäijŽä¿İæ■ő sys.meta_path äÿ■çŽĎæ§ěæL'¿åŹĺåőŽä¡■æĺąåİŮãĂĆ åIJÍè£Źäÿłä¿Ńå■ŘäÿïijŇUrlMetaFinder åőđä¿ŃæŸŕæIJĂåŘŐäÿĂäÿłæ§ěæL'¿åŹĺæŰźæąĹïijŇ å¡ŞæĺąåİŮåIJĺäżżä¡ŢäÿĂäÿłæŹőéĂŽåIJřæŰźéČ¡æL'¿äÿ

ä¡IJäÿžåÿÿèğĄçŽĎåőđçŐřæŰźæąĹïijŇUrlMetaFinder çśżåŇĚèčĚåIJĺäÿĂäÿłçŤĺæĹůæŇĞåőŽçŽĎURLäÿŁãĂĆ åIJĺåĘĚéČĺïijŇæ§ěæL'¿åŹĺéĂŽè£ĞæŁŞåŔŰæŇĞåčåŕijåĚěçŽĎæŮűåĂŹïijŇæĺąåİŮåŘ■äijŽèů§åůšæIJL'çŽĎéŞ¿æŐěä¡IJåŕźærŤãĂĆåęĆædIJæL'¿åĹřäžĘäÿĂäÿłåÑäÿĂäÿłå■ŢçŃňçŽĎ UrlModuleLoader çśżèćńçŤĺæİěäżŐè£IJçÍŃæIJžåŹĺäÿŁåŁäè¡¡æžŘäżčçăĄåźűåĹŻåżżè£ŹéĞŇçijŞå■ŸéŞ¿æŐěçŽĎäÿĂäÿłåŐ§åŽăæŸŕéĄ£åĚ■äÿ■å£ĚèęĄçŽĎHTTPèŕůæśĆéĞ■åd'■åŕijåĚěãĂĆ

èĞłåőŽäźL'åŕijåĚěçŽĎçňňäžŇçğ■æŰźæşŢæŸŕçijŰåĘŹäÿĂäÿłéŠl'å■ŘçŻt'æŐěåţŇåĚěåĹř sys.path åŘŸéĞŘäÿ■åŐżïijŇ èŕĘåĹńæ§ŘäžŻçŻőå¡ŢåŚ¡åŘ■æĺąåijŔãĂĆ åIJÍ urlimport.py äÿ■æůżåŁäåęĆäÿŇçŽĎçśżåŠŇæŤŕæŇĄåĞ¡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)

```
            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¡£çŤĺè£Źäÿłèůŕå¿Ďæ§ěæL’¿åŹĺïijŇä¡ăåŔłéIJĂèęĄåIJĺ `sys.path` äÿ■åŁăåĚěURLéŞ¿æŐěãĂĆä¿ŃåęĆïijŽ

```
>>> # 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'
>>>
```

åËşéŤőçĆźářśæŸŕ `handle_url()` åĞ¡æŢřïijŇåőČèćńæůżåŁääĹřäžĘ `sys.path_hooks` åŔŸéĞŔäÿ■ãĂĆ å¡Ş `sys.path` çŽĎåőđä¡Şèćńåď ĎçŘĘæŮűïijŇäijŽèřČçŢĺ `sys.path_hooks` äÿ■çŽĎåĞ¡æŢřãĂĆ åęĆæđIJäżża¡ŢäÿĂäÿłåĞ¡æŢřè£ŤåŻđäžĘäÿĂäÿłæ§ěæL'¿åŹĺářźèśą `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)

```
        return fib(n-1) + fib(n-2)
>>>
```

èőÍèőž

åIJÍèŕęçżĘèőÍèőžäźŃåL’■ïijŇæIJL’çĆźèęĄåijżèřČçŽĎæŸŕïijŇPythonçŽĎæÍąåİŮãĂĄåŇĚåŠŇåŕijåĚěæIJ
å■şä¡£çżŔéłŇäÿřåŕŇçŽĎPythonçÍŃåžŔåŚŸäź§å¿ÍåřŚèČ¡çš¿éĂŽáőČäžñãĂĆ
æĹŚåIJÍè£ŹéĞŇæŐÍè■ŘäÿĂäžŽåĂijçŽĎåŐżèŕżçŽĎæŰĞæąčåŠŇäźęçś■ïijŇåŇĚæŃň im-
portlib module åŠŇ PEP 302. æŰĞæąčåĘĚåőźåIJÍè£ŹéĞŇäÿ■äijŽèćńéĞ■åd’■æŔŘåĹřïijŇäÿ■è£ĞæĹŚåIJÍè£Ź

éęŰåĚĹïijŇåęĆædIJä¡ăæČşåĹŽåžžäÿĂäÿłæŰřçŽĎæÍąåİŮåŕźèśąïijŇä¡£çŤÍ imp.
new_module() åĞ¡æŢřïijŽ

```
>>> import imp
>>> m = imp.new_module('spam')
>>> m
<module 'spam'>
>>> m.__name__
'spam'
>>>
```

æÍąåİŮåŕźèśąéĂŽåÿÿæIJL’äÿĂäžŽæIJ§æIJŻåśđæĂğïijŇåŇĚæŃň __file__
ïijÍè£ŘèąŇæÍąåİŮåŁäè¡¡èŕ■åŘęçŽĎæŰĞäžűåŘ■ïijL’ åŠŇ __package__ (åŇĚåŘ■)ãĂĆ

åĚűæňąïijŇæÍąåİŮäijŽèćńèğčéĞŁåŹÍçijŞå■ŸèţůæİěãĂĆæÍąåİŮçijŞå■ŸåŔŕäžěåIJÍå■ŮåĚÿ
sys.modules äÿ■èćńæL’¿åĹřãĂĆ åŽääÿžæIJL’äžĘè£ŹäÿłçijŞå■ŸæIJžåĹűïijŇéĂŽåÿÿåŔŕäžěårĘçijŞå■ŸåŠ

```
>>> import sys
>>> import imp
>>> m = sys.modules.setdefault('spam', imp.new_module('spam'))
>>> m
<module 'spam'>
>>>
```

åęĆædIJçżŹåőŽæÍąåİŮåůšçżŔå■ŸåIJÍéĆčäźĹářšäijŽçŽt’æŐěèŐůå¿ŮåůšçżŕèćńåĹŽåžžè£ĞçŽĎæÍąåİŮï

```
>>> 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
>>>
```

çŤšäžŐåĹŽåžžæÍąåİŮå¿ĹçőĂå■ŢïijŇå¿Ĺåőźæ ŸŞçijŰåĘŹçőĂå■ŢåĞ¡æŢřæŕŤåęĆçńňäÿĂéČĺåĹĘçŽĎ
load_module() åĞ¡æŢřãĂĆ è£ŹäÿłæŰźæąĹçŽĎäÿĂäÿłçijžçĆźæŸŕå¿ĹéŽ¿åd’ĎçŘĘåd’■æİĆæČĚåĘţæŕŤ

äÿžäžĘåď ĎçŘĘäÿĂäÿłåŇĚïijŇä¡ãèęĄéĞ■æŰřåőđçŐřæŹőéĂŽimportèí■åŔěçŽĎåžŢåśĆéĂżè¿Śïij Ĺæŕ Ťåę Ća
æĽğèąŇéĆčäžŻæŰĞäžűïijŇèő¿ç¡őèůřå¿Ďç■Ľ'ïijĽ'ãĂĆè£Źäÿłåď'■æİĆæĂğåřśæŸŕäÿžäżĂäźĹæIJĂåě¡çŻť'æŐ

æĽ'ľ'åśŢimportèí■åŔěå¿ĹçőĂå■ŢïijŇä¡ĘæŸŕäijŽæIJL'å¿Ĺåď'Žçğżå Ł Íæ Ş■ä¡IJãĂĆ
æIJĂéńŸåśĆäÿŁïijŇåŕijåĚěæŞ■ä¡IJèćńäÿĂäÿłä¡■äžŐsys.meta_pathåĹŮèąĺäÿ■çŽĎâĂIJåĚČèůřå¿ĎâĂİæ§ěæ
åęĆæđIJä¡ãè¿ŞåĞžåőČçŽĎåĂijïijŇäijŽçIJŃåŁřäÿŃéİćè£Źæăůïijž

```
>>> from pprint import pprint
>>> pprint(sys.meta_path)
[<class '_frozen_importlib.BuiltinImporter'>,
<class '_frozen_importlib.FrozenImporter'>,
<class '_frozen_importlib.PathFinder'>]
>>>
```

å¡ŞæĽğèąŇäÿĂäÿłèí■åŔěæŕŤåęĆ `import fib` æŮűïijŇèğčéĞŁåŹĺäijŽéĄ■åŐĘsys.mata_pathäÿ■çŽĎ
èřČçŤĺåőČäžňçŽĎ `find_module()` æŰźæşŢåőŽä¡■æ■čçąőçŽĎæĺąåİŮåŁãè¡¡åŹĺãĂĆ
åŔŕäžěéĂŽè£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
>>>
```

æşĺæĎŔçIJŃ `find_module()` æŰźæşŢæŸŕæĂŐæăůåIJĺæŕŔäÿĂäÿłåŕijåĚěåřśèćńèğęåŔŠçŽĎãĂĆ
è£ŹäÿłæŰźæşŢäÿ■çŽĎpathåŔĆæŢřçŽĎä¡IJçŤĺæŸŕåď'ĎçŘĘåŇĚãĂĆ
åď'ŽäÿłåŇĚèćńåŕijåĚěïijŇåřśæŸŕäÿĂäÿłåŔŕåIJĺåŇĚçŽĎ `__path__`
åśđæĂğäÿ■æĽ'¿åĹřçŽĎèůřå¿ĎåĹŮèąĺãĂĆ èęĄæĽ'¿åĹřåŇĚçŽĎå■ŘçżĎäžűåřśèęĄæčĂæ§ěè£ŹäžŻèůřå¿ĎãĂ
æŕŤåęĆæşĺæĎŔåŕźäžŐ `xml.etree` åŠŇ `xml.etree.ElementTree`
çŽĎèůřå¿ĎéĚ■ç¡őïijž

```
>>> 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)

```
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
>>>
```

åIJĺ `sys.meta_path` äÿŁæ§ěæL'¿åŹĺçŽĎä¡■ç¡ő å¿ĹéĞ■èęĄïijŇářĘåőČäżŐéŸ§åď'ţğżåĹřéŸ§år¿ïijŇ

```
>>> del sys.meta_path[0]
>>> sys.meta_path.append(Finder())
>>> import urllib.request
>>> import datetime
```

çŐřåIJĺä¡ăçIJŃäÿ■åĹřäżżä¡Ţè¿ŞåĞžäžĘïijŇåŻääÿžářijåĚěèćńsys.meta_pathäÿ■çŽĎåĚűäżŮåőđä¡Şåď'Đç
è£ŹæŮűåĂŹïijŇä¡ääŔłæIJL'åIJĺářijåĚěäÿ■å■ŸåIJĺæĺąåİŮçŽĎæŮűåĂŹæL'■èČ¡çIJŃåĹřåőČèćńèğęåŔŚïijŽ

```
>>> 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£ĞäÿĂääÿłæ■ŢèŐůæIJłç§ěæĺąåİŮçŽĎæ§ěæL'¿åŹĺïijŇè£ŹäÿłæŸŕ
`UrlMetaFinder` çśżçŽĎåĚşéŤõãĂĆ äÿĂääÿł `UrlMetaFinder` åőđä¿ŃèćńæůżåŁääĹř
`sys.meta_path` çŽĎæIJńář¿ïijŇä¡IJäÿżæIJĂåŘŐäÿĂääÿłæ§ěæL'¿åŹĺæŰźæąĹãĂĆ
åęĆæđIJèćńèřůæśĆçŽĎæĺąåİŮåŘ■äÿ■èČ¡åőŽä¡■ïijŇářšäijŽèćńè£Źäÿłæ§ěæL'¿åŹĺåď'ĐçŘĘæŐL'ãĂĆ
åď'ĐçŘĘåŇĚçŽĎæŮűåĂŹéIJĂèęĄæşĺæĎŔïijŇåIJĺpathåŔĆæŢřäÿ■æŇĞåőŽçŽĎåĂijéIJĂèęĄèćńæčĂæ§ěïijŃ
åęĆæđIJäÿ■æŸŕïijŇèřěå■ŘæĺąåİŮå£Ěéąžå¡ŠåšđäžŐåĚűäżŰæ§ěæL'¿åŹĺåźűèćńå£¡çŢěæŐL'ãĂĆ

åŕźäžŐåŇĚçŽĎåĚűäżŰåď'ĐçŘĘåŔŕåIJĺ `UrlPackageLoader`
çśżäÿ■èćńæL'¿åĹřãĂĆ è£Źäÿłçśżäÿ■äijŽåŕijåĚěåŇĚåŘ■ïijŇèĂŇæŸŕåŐžåŁäè¡¡åŕźåžŢçŽĎ
`__init__.py` æŰĞäźűãĂĆ åőČäź§äijŽèő¿ç¡ő æĺąåİŮçŽĎ `__path__`
åšđæĂğïijŇè£ŹäÿĂæ■ěå¿ĹéĞ■èęĄïijŃ åŻääÿžåIJĺåŁäè¡¡åŇĚçŽĎå■ŘæĺąåİŮæŮűè£ŹäÿłåĂijäijŽèćńäijăçżŹå
`find_module()` èřČçŢĺãĂĆ å§žäžŐèůŕå¿ĐçŽĎåŕijåĚěéŠl'å■ŘæŸŕè£ŹäžŽæĂİæČşçŽĎäÿĂääÿłæL'l'åśŢïijŃ
æĹŚäżňéČ¡ç§ěéĄŞïijŇ`sys.path` æŸŕäÿĂääÿłPythonæ§ěæL'¿æĺąåİŮçŽĎçŻő å¡ŢåĹŮèąĺïijŇä¿ŃåęĆïijŽ

```
>>> from pprint import pprint
>>> import sys
```

(continues on next 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■çŽĎæŕŔäÿĂäÿłåőđä¡ŞéČ¡äijŽèćńéćÍåd'ŰçŽĎçżŚåőŽåĹřäÿĂäÿłæ§ěæL'¿åŹĺářźèśąäÿ
ä¡ăåŔŕäžěéĂŽè£Ğæ§ěçIJŃ `sys.path_importer_cache`
åŐżçIJŃäÿŃè£ŹäžŻæ§ěæL'¿åŹĺïijŽ

```
>>> 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` äijŽæŻt'åd'ğçĆźïijŃ
åŻääÿžåőČäijŽäÿžæL'ĂæIJL'èćńåŁäè¡¡äžčçăĄçŽĎçŻőå¡Ţèőřå¡ŢåőČäžńçŽĎæ§ěæL'¿åŹĺãĂĆ
è£ŹåŇĚæŃňåŇĚçŽĎå■ŘçŻőå¡ŢïijŃè£ŹäžŻéĂŽåÿÿåIJÍ `sys.path`
äÿ■æŸŕäÿ■å■ŸåIJĺçŽĎãĂĆ

èęĄæL'ğèąŇ `import fib` ïijŃäijŽéąžåžŔæčĂæ§ě `sys.path` äÿ■çŽĎçŻőå¡ŢãĂĆ
åŕźäžŐæŕŔäÿłçŻőå¡ŢïijŃåŘ■çğřâĂIJfibâĂİäijŽèćńäijăçżŹçŻÿåžŤçŽĎ `sys.path_importer_cache` äÿ■çŽĎæ§ěæL'¿åŹĺãĂĆ è£ŹäÿłåŔŕäžěèőľ'ä¡ăåĹŻåžžèĞłåůśçŽĎæ§ěæL'¿åŹĺáźűå

```
>>> 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)

```
>>> 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¡ääŔŕäżěäýžåŘ■å■ŮâĂIJdebugâĂİåĹŻåžžäýĂäýłæŰřçŽĎçijŞå■Ÿåőđä¡Şåźűå řĘåőČèő¿
`sys.path` äýŁçŽĎçňňäýĂäýłãĂĆ åIJÍæĽ ĂæIJĽæŐěäýŇæİěçŽĎåŕijåĚěäý■ïijŇä¡ääijŽçIJŇåĹřä¡ăçŽĎæ§ěa
äý■è£ĞïijŇçŤśäžŐåőČè£ŤåŻđ (None, [])ïijŇéĆčäźĹåď ĎçŘĘè£ŽçÍŇäijŽçżğçż■åď ĎçŘĘäýŇäýĂäýłåőđä¡ŞãA

`sys.path_importer_cache` çŽĎä¡£çŤĺèćńäýĂäýłå■ŸåĆĺåIJÍ `sys.path_hooks`
äý■çŽĎåĞ¡æŢřåĹŮèąÍæŐğåĹűãĂĆ èŕŢèŕŢäýŇéİćçŽĎä¿Ńå■ŘïijŇåőČäijŽæÿĚéŹď çijŞå■Ÿåźűçżź
`sys.path_hooks` æůżåŁääýĂäýłæŰřçŽĎèůŕå¿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¡ăæĽĂèğĄïijŇcheck_path() åĞ¡æŢřèćńæŕŔäýł `sys.path`
äý■çŽĎåőđä¡ŞěřČçŤĺãĂĆ äý■éą¿ïijŇçŤśäžŐæŁŻåĞžäžĘ `ImportError` åijĆåÿÿïijŇ
åŢěéČ¡äý■äijŽåŔŚçŤ§äžĘïijĹäzĚäżĚåřĘæčĂæ§ěè¡ńçğżåĹřsys.path_hooksçŽĎäýŇäýĂäýłåĞ¡æŢřïijĽãĂĆ

ç§ěéĄŞäžĘæĂŐæăůsys.pathæŸŕæĂŐæăůèćńåď ĎçŘĘçŽĎïijŇä¡ääřśèČ¡æđĎåžžäýĂäýłèĞłåőŽäźĽèůŕå¿

```
>>> def check_url(path):
...     if path.startswith('http://'):
...         return Finder()
...     else:
...         raise ImportError()
```

(continues on next 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£ŹåřśæŸŕæIJňèŁĆæIJĂåŘŐéČĺåĹĘçŽĎåĚşéŤőçĆźãĂĆäžŃåőđäÿŁïijŇäÿĂäÿłçŤĺæİěåIJĺsys.pathäÿ■æ
å¡ŞåőČäžňèćńçćřåĹřçŽĎæŮűåĂŹïijŇäÿĂäÿłæŰřçŽĎ `UrlPathFinder` åőđä¿ŃèćńåĹŻåžžåźűèćńæŤ¿åĚě `sys.path_importer_cache`. äźŃåŘŐïijŇæĽĂæIJĽéIJĂèęĄæčĂæ§ě `sys.path` çŽĎåŕijåĚěèŕ■åŔěéČ¡äijŽä¡£çŤĺä¡ăçŽĎèĞłåőŽäźĽæ§ěa

å§žäžŐèůŕå¿ĎåŕijåĚěçŽĎåŇĚåď ĎçŘEçĺ■å¿őæIJĽçĆźåď■æİĆïijŇåźűäÿŤèů§ `find_loader()` æŰźæşŢè£ŤåŻđåĂijæIJĽåĚşãĂĆ åŕźäžŐçőĂå■ŢæĺąåİŮïijŇ`find_loader()` è£ŤåŻđäÿĂäÿłåĚČçżĎ(loader, None)ïijŇ åĚűäÿ■çŽĎload-eræŸŕäÿĂäÿłçŤĺäžŐåŕijåĚěæĺąåİŮçŽĎåŁăè¡¡åŹĺåőđä¿ŃãĂĆ

åŕźäžŐäÿĂäÿłæŹőéĂŽçŽĎåŇĚïijŇ`find_loader()` è£ŤåŻđäÿĂäÿłåĚČçżĎ(loader, path)ïijŇ åĚűäÿ■çŽĎloaderæŸŕäÿĂäÿłçŤĺäžŐåŕijåĚěåŇĚïijĹåźűæĽğèąŇ__init__.pyïijĽçŽĎåŁăè¡¡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 ĎçŘEäÿĂäÿłåŚ¡åŘ■çľžéŮť åŇĚäÿ■æIJĽäÿĂäÿłåŘĹæşŢçŽĎåŇĚçŻőå¡ŢåŘ■ïijŇä¡ĘæŸŕäÿ■å■ŸåIJĺ__init__.pyæŰ è£ŹæăůçŽĎèŕİïijŇ`find_loader()` å£Ěéążè£ŤåŻđäÿĂäÿłåĚČçżĎ(None, path)ïijŇ pathæŸŕäÿĂäÿłçŻőå¡ŢåĹŮèąĺïijŇçŤśåőČæİěæđĎåžžåŇĚçŽĎåőŽäźĽæIJĽ__init__.pyæŰĞäžűçŽĎ__path__ åŕźäžŐè£Źçğ■æČĚåĘţïijŇåŕijåĚěæIJžåĹűäijŽçżğçż■åĽ■èąŇåŐżæčĂæ§ěsys.pathäÿ■çŽĎçŻőå¡ŢãĂĆ åęĆæđIJæĽ¿åĹřäžĘåŚ¡åŘ■çľžéŮť åŇĚïijŇæĽĂæIJĽçŽĎçżŞæđIJèůŕå¿ĎèćńåŀăåĹřäÿĂètůæİěæđĎåžžæIJĂ åĚşäžŐåŚ¡åŘ■çľžéŮť åŇĚçŽĎæŻť åď Žä£qæĄŕèŕůåŔĆèĂČ10.5åřŔèŁĆãĂĆ

æĽĂæIJĽçŽĎåŇĚéČ¡åŇĚåŘńäžĘäÿĂäÿłåĘĚéČĺèůŕå¿Ďèő¿ç¡őïijŇåŔŕäžěåIJĺ__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£ŤåŻđåĂijæŐğåĹűçŽĎãĂĆ äÿ■è£ĞïijŇ__path__æŐěäÿŃæİěäź§èćńsys.path_hooksäÿ■çŽĎåĞ¡æŢ

åŽăæ■ď'ïijŇä¡ĘåŃĚçŽĎå■ŘçżĎäžűèćńåŁăè¡¡åŔŐïijŇä¡■äžŐ __path__ äÿ■çŽĎåőđä¡ŞäijŽèćń `handle_url()` åĞ¡æŢřæčĂæ§ěãĂĆ è£ŹäijŽåŕijèĞť æŰřçŽĎ `UrlPathFinder` åőđä¿ŃèćńåŻåžžåźűäÿèćńåŁăåĚěåĹř `sys.path_importer_cache` äÿ■ãĂĆ

è£ŸæIJĽäÿłéŽ¿çĆźåřśæŸŕ `handle_url()` åĞ¡æŢřäžěåŘŁåőČèů§åĘĚéČĺä¡£çŤĺçŽĎ `_get_links()` åĞ¡æŢřäźŃéŮť çŽĎäžď'äžŠãĂĆ åęĆæđIJä¡ăçŽĎæ§ěæĽ¿åŹĺåőđçŐřéIJĂèęĄä¡£çŤĺåĹřåĚű æIJĽåŔřèČ¡è£ŹäžŻæĺąåİŮäijŽåIJĺæ§ěæĽ¿åŹĺæŞ■ä¡IJæIJ§éŮť è£ŻèąŇæŻť åď'ŽçŽĎåŕijåĚěãĂĆ åőČåŔřäżěåŕijèĞť `handle_url()` åŠŃåĚűäżŰæ§ěæĽ¿åŹĺéČĺåĹĘéŹůåĚěäÿĂçğ■éĂŠå¡Šå¿łçŐŕçŁűæĂĄã äÿžäžĘèğčéĞŁè£Źçğ■åŔřèČ¡æĂğïijŇåőđçŐřäÿ■æIJĽäÿĂäÿłèćńåĹŻåżžçŽĎæ§ěæĽ¿åŹĺçijŞå■ŸïijĹæŕŔäÿĂ åőČåŔřäżěéĄ£åĚ■åĹŻåżžéĞ■åď'■æ§ěæĽ¿åŹĺçŽĎéŰőéćŸãĂĆ åŘęåď'ŰïijŇäÿŃéİćçŽĎäżčçăĄçĽĞæőţåŔřäżěçąőä£Ďæ§ěæĽ¿åŹĺäÿ■äijŽåIJĺåĹİåğŃåŇŰéŞ¿æŐěéŽĘåŘĹçŽ

```
# Check link cache
if self._links is None:
    self._links = [] # See discussion
    self._links = _get_links(self._baseurl)
```

æIJĂåŔŐïijŇæ§ěæĽ¿åŹĺçŽĎ `invalidate_caches()` æŰźæşŢæŸřäÿĂäÿłåůěåĚůæŰźæşŢïijŇçŤĺæİěæÿĚçŘĘåĘĚéČĺçijŞå■ŸãĂĆ è£ŹäÿłæŰźæşŢåĘ■çŤĺæĹůèřČçŤĺ `importlib.invalidate_caches()` çŽĎæŮűåĂŹèćńèğęåŔŠãĂĆ åęĆæđIJä¡ăæČşèőľURLåŕijåĚěèĂĚéĞ■æŰřèŕżåŔŰéŞ¿æŐěåĹŮèąĺçŽĎèŕİåŔŕä

åŕźæŕŤäÿŃäÿď'çğ■æŰźæąĹïijĹä£őæŤźsys.meta_pathæĹŰä¡£çŤĺäÿĂäÿłèůŕå¿ĎéŠľå■ŘïijĽãĂĆ ä¡£çŤĺsys.meta_pathçŽĎåŕijåĚěèĂĚåŔřäżěæŇĽçĚğèĞłåůśçŽĎéIJĂèęĄèĞłçŤśåď'ĎçŘĘæĺąåİŮãĂĆ ä¿ŃåęĉïijŇåőČäżňåŔřäżěäżŐæŢř■æőżåŞäÿ■åŕijåĚěæĹŰäžěäÿ■åŘŇäžŐäÿĂèĹňæĺąåİŮ/åŇĚåď'ĎçŘĘæŰźå è£Źçğ■èĞłçŤśåŘŇæăůæĎŘåŠşçİĂåŕijåĚěèĂĚéIJĂèęĄèĞłåůśè£ŻèąŇåĘĚéČĺçŽĎäÿĂäžŻçőąçŘĘãĂĆ åŘęåď'ŰïijŇå§žäžŐèůŕå¿ĎçŽĎéŠľå■ŘåŔłæŸŕéĂĆçŤĺäžŐåŕźsys.pathçŽĎåď'ĎçŘĘãĂĆ éĂŽè£Ğè£Źçğ■æĽľåśŢåŁăè¡¡çŽĎæĺąåİŮèů§æŹőéĂŽæŰźåijŔåŁăè¡¡çŽĎçĽźæĂğæŸŕäÿĂæăůçŽĎãĂĆ

åęĆæđIJåĹřçŐřåIJĺäÿžæ■ćä¡äè£ŸæŸŕäÿ■æŸŕå¿ĹæŸŐçŹ¡ïijŇéĆčäźĹåŔřäżěéĂŽè£Ğåćđåłăäÿ■äžŻæŮ

```
>>> 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)

```
I'm fib
>>>
```

æIJĂåŘŐïijŇåžžèőőä¡ăèŁśçĆźæŮűéŮt'çIJŃçIJŃ PEP 302 äžěåŘŁimportlibçŽĎæŰĞæąčãĂĆ

12.12 10.12 åŕijåĚěæĺąåİŮçŽĎåŘŇæŮűä£őæŤźæĺąåİŮ

éŮőécŸ

ä¡ăæČşçżŹæ§Řäýłåůšå■ŸåIJĺæĺąåİŮäý■çŽĎåĞ¡æŢřæůżåŁăèčĚéěřåŹĺãĂĆ äý■è£ĞïijŇåL'■æŔŘæŸŕè£ŹäýłæĺąåİŮåůšçżŔèćńåŕijåĚěåźűäýŤèćńä¡£çŤĺè£ĞãĂĆ

èğčåĘşæŰźæąĹ

è£ŹéĞŇéŮőécŸçŽĎæIJňèť'ĺåřśæŸŕä¡ăæČşåIJĺæĺąåİŮèćńåŁăè¡¡æŮűæL'ğèąŇæ§ŘäýłåŁĺä¡IJãĂĆ åŔŕèČ¡æŸŕä¡ăæČşåIJĺäýĂäýłæĺąåİŮèćńåŁăè¡¡æŮűèğęåŔŚæ§ŘäýłåŻđèřČåĞ¡æŢřæİěéĂŽç§ëä¡ããĂĆ

è£ŹäýłéŮőécŸåŔŕäžěä¡£çŤĺ10.11åřŔèŁĆäý■åŘŇæăůçŽĎåŕijåĚěéŠl'å■ŘæIJžåĹűæİěåőđçŐřãĂĆäýŃéIć

```
# 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)

```
        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¡ăářśåŔŕäžěä¡£çŤÍ `when_imported()` èčĚéěřåŹÍäžĘïijŇä¿ŃåęĆïijŽ

```
>>> 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'åőđéŹĚçŽĎä¿Ńå■ŘïijŇä¡ääŔŕèČ¡æČşåIJÍåůšå■ŸåIJÍçŽĎåőŽäźĽäÿŁéİćæůżåŁăèčĚé

```
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¿ÌètŰäžŐ10.11åřŔèŁĆäÿ■èőšè£řè£ĞçŽĎåŕijåĚéšĽå■ŘïijŇåźűçÍ■ä¡IJä£őæŤźãĂĆ

`@when_imported` èčĚéěřåŹÍçŽĎä¡IJçŤÍæŸŕæşÍåĘŇåIJÍåŕijåĚěæŮűèćńæ£Ăæť'żçŽĎåď'ĎçŘĘåŹÍåĞ¡a

èŕěèčĚéěřåŹÍæčĂæ§ěsys.modulesæİěæ§ěçIJŃæĺąåİŮæŸŕåŘęçIJ§çŽĎåůšçżŔèćńåŁăè¡¡äžĘãĂĆ
åęĆæđIJæŸŕçŽĎèŕÍïijŇèŕěåď˙ĎçŘĘåŹÍèćńçńŇå■şèřČçŤĺãĂĆäÿ■çĎűïijŇåď˙ĎçŘĘåŹÍèćńæůżåŁăåĹř
_post_import_hooks å■ŮåĚÿäÿ■çŽĎäÿĂäÿłåĹŮèąĺäÿ■åŐżãĂĆ
_post_import_hooks çŽĎä¡IJçŤĺářśæŸŕæŤűéŻĘæĽ˙ĂæIJĽçŽĎäÿžæŕŔäÿłæĺąåİŮæşĺåĘŇçŽĎåď˙ĎçŘĘ
äÿĂäÿłæĺąåİŮåŔŕäžěæşĺåĘŇåď˙Žäÿłåď˙ĎçŘĘåŹĺãĂĆ

èęĄèőľ˙æĺąåİŮåŕijåĚěåŘŐèğęåŔŚæůżåŁăçŽĎåŁĺä¡IJïijŇPostImportFinder
çśżèćńèő¿ç¡őäÿžsys.meta_pathçňňäÿĂäÿłåĚČçť˙ăãĂĆ åőČäijŽæ■ŢèŐůæĽ˙ĂæIJĽ˙æĺąåİŮåŕijåĚěæŞ■ä¡IJãĂĆ

æIJňèŁĆäÿ■çŽĎ PostImportFinder çŽĎä¡IJçŤĺåźűäÿ■æŸŕåŁăè¡¡æĺąåİŮïijŇèĂŇæŸŕèĞłåÿęåŕijåĚ
åőđéŹĚçŽĎåŕijåĚěèćńåğŤæť˙¿çżŹä¡■äžŐsys.meta_pathäÿ■çŽĎåĚűäżŰæ§ěæĽ˙¿åŹĺãĂĆ
PostImportLoader çśżäÿ■çŽĎ imp.import_module()
åĞ¡æŢřèćńéĂŠå¡ŠçŽĎèřČçŤĺãĂĆ äÿžäžĘéĄ£åĚ■éŹůåĚěæŮăçž£å¿łçŐŕïijŇPostImportFinder
ä£ÍæŇĄäžĘäÿĂäÿłæĽ˙ĂæIJĽèćńåŁăè¡¡è£ĞçŽĎæĺąåİŮéŻĘåŘĹãĂĆ
åęĆæđIJäÿĂäÿłæĺąåİŮåŘ■å■ŸåIJĺářśäijŽçŻť˙æŐěèćńå£¡çŢěæŐĽ˙ãĂĆ

å¡ŞäÿĂäÿłæĺąåİŮèćń imp.import_module() åŁăè¡¡åŐïijŇ
æĽ˙ĂæIJĽ˙åIJĺ_post_import_hooksèćńæşĺåĘŇçŽĎåď˙ĎçŘĘåŹÍèćńěŘČçŤĺïijŇä¡£çŤĺæŰřåŁăè¡¡æĺąåİŮä¡IJäÿž

æIJĽäÿĂçĆźéIJĂèęĄæşĺæĎŔçŽĎæŸŕæIJňæIJžäÿ■éĂĆçŤĺäžŐéĆčäžŻéĂŽè£Ğ imp.
reload() èćńæŸ¿åijŔåŁăè¡¡çŽĎæĺąåİŮãĂĆ äźßářśæŸŕèŕť˙ïijŇåęĆæđIJä¡ååŁăè¡¡äÿĂäÿłäźŃåĽ˙■åůšèćńåŁă
åŘęåď˙ŰïijŇèęĄæŸŕä¡ääžŐsys.modulesäÿ■åĹăéŹď˙æĺąåİŮçĎűåŘŐåĘ■éĞ■æŰřåŕijåĚěïijŇåď˙ĎçŘĘåŹĺåŔĹä

æŻť˙åď˙ŽåĚşäžŐåŕijåĚěåŘŐéŠĺ˙å■Ŕä£ąæĄŕèřůåŔĆèĂČ PEP 369.

12.13 10.13 åőĽèčĚçğĄæIJĽçŽĎåŇĚ

éŮőéćŸ

ä¡ăæČşèęĄåőĽèčĚäÿĂäÿłçňňäÿĽ˙æŰźåŇĚïijŇä¡ĘæŸŕæšąæIJĽ˙æİČéŹŘåřĘåőČåőĽèčĚåĹřçşżçż§Python
æĹŰèĂĚïijŇä¡ăåŔŕèČ¡æČşèęĄåőĽèčĚäÿĂäÿłä¿ŻèĞłåůśä¡£çŤĺçŽĎåŇĚïijŇèĂŇäÿ■æŸŕçşżçż§äÿŁéİćæĽ˙Ăa

èğčåĘşæŰźæąĹ

PythonæIJĽäÿĂäÿłçŤĺæĹůåőĽèčĚçŽőå¡ŢïijŇéĂŽåÿÿçśżäijijâĂİ~/.local/lib/python3.3/site-
packagesâĂİãĂĆ èęĄåijžåĹűåIJĺè£ŹäÿłçŽőå¡Ţäÿ■åőĽèčĚåŇĚïijŇåŕŕä¡£çŤĺåőĽèčĚéĂĽ˙éąźâĂIJ–userâĂİãĂ

```
python3 setup.py install --user
```

æĹŰèĂĚ

```
pip install --user packagename
```

åIJĺsys.pathäÿ■çŤĺæĹůçŽĎâĂIJsite-packagesâĂİçŽőå¡Ţä¡■äžŐçşżçż§çŽĎâĂIJsite-
packagesâĂİçŽőå¡ŢäźŃåĽ˙■ãĂĆ åŻäæ■ď˙ïijŇä¡ăåőĽèčĚåIJĺéĞŇéİćçŽĎåŇĚářśæŕŤçşżçż§åůšåőĽèčĚçŽĎåŇ
ïijĹář¡çőąåźűäÿ■æĂżæŸŕè£ŹæăůïijŇèęĄåŔŰåĘşäžŐçňňäÿĽ˙æŰźåŇĚçőąçŘĘåŹĺïijŇæŕŤåęĆdistributeæĹŰpi

èőÍèőž

éĂŽåÿÿåŇĚäijŽèćńåőL'èčĚåĹřçşžçż§çŽĎsite-packagesçŽőå¡Ţäÿ■åŐżïijŇèůŕå¿Ďçśżäijijâ˘IJ/usr/local/l
packagesâĂİãĂĆ äÿ■è£ĞïijŇè£ŹæăůåĄŽéIJĂèęĄæIJL'çőąçŘĘåŚŸæİČéŹŘåźűäÿŤä¡£çŤĺsudoåŚ¡äžd'ãĂĆ
åřśçőŮä¡ăæIJL'è£ŹæăůçŽĎæİČéŹŘåŐżæL'ğèąŇåŚ¡äžd'ïijŇä¡£çŤĺsudoåŐżåőL'èčĚäÿĂäÿłæŰřçŽĎïijŇåŕřèČ

åőL'èčĚåŇĚåĹřçŤĺæĹůçŽőå¡Ţäÿ■éĂŽåÿÿæŸŕäÿĂäÿłæIJL'æŢĹçŽĎæŰźæąĹïijŇåőČåĚĄèőÿä¡ääĹŽåžžä

åŘęåd'ŰïijŇä¡ăè£ŸåŕŕäžěåĹŽåžžäÿĂäÿłèŹŽæŃ§çŐŕåćČïijŇè£ŹäÿłæĹŚäžňåIJĺäÿŃäÿĂèŁĆäijŽèőšåĹřă

12.14 10.14 åĹŽåžžæŰřçŽĎPythonçŐŕåćČ

éŮőécŸ

ä¡ăæČşåĹŽåžžäÿĂäÿłæŰřçŽĎPythonçŐŕåćČïijŇçŤĺæİěåőL'èčĚæĺąåİŮåŠŇåŇĚãĂĆ
äÿ■è£ĞïijŇä¡ääÿ■æČşåőL'èčĚäÿĂäÿłæŰřçŽĎPythonåĚŃéŽĘïijŇäź§äÿ■æČşåŕźçşżçż§PythonçŐŕåćČäžğçŤ§

èğčåĘşæŰźæąĹ

ä¡ääŔŕäžěä¡£çŤĺ `pyvenv` åŚ¡äžd'åĹŽåžžäÿĂäÿłæŰřçŽĎâĂIJèŹŽæŃ§âĂİçŐŕåćČãĂĆ
è£ŹäÿłåŚ¡äžd'èćńåőL'èčĚåIJĺPythonèğčéĞŁåŹĺåŘŇäÿĂçŽőå¡ŢïijŇæĹŰWindowsäÿŁéİćçŽĎScriptsçŽőå¡Ţäÿ

```
bash % pyvenv Spam
bash %
```

äijăçżŹ `pyvenv` åŚ¡äžd'çŽĎåŘ■å■ŮæŸŕåřĘèęĄèćńåĹŽåžžçŽĎçŽőå¡ŢåŘ■ãĂĆå¡Şèćńå
ĹŽåžžåŘŐïijŇS

```
bash % cd Spam
bash % ls
bin include lib pyvenv.cfg
bash %
```

åIJĺbinçŽőå¡Ţäÿ■ïijŇä¡ääijŽæL'¿åĹřäÿĂäÿłåŔŕäžěä¡£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)

```
'/Users/beazley/Spam/lib/python3.3/site-packages']
>>>
```

è£ŹäÿłèğčéĞŁåŹÍçŽĎçL'źçĆźåřśæŸŕäżŰçŽĎsite-packagesçŻőå¡Ţèćńèő¿ç¡őäÿžæŰřåĹŻåżžçŽĎçŐŕåćČ
åęĆæđIJä¡ăèęĄåőL'èčĚçňňäÿL'æŰźåŇĚïijŇåőČäžňäijŽèćńåőL'èčĚåIJÍéĆčéĞŇïijŇèĂŇäÿ■æŸŕéĂŽåÿÿçşżçż
packagesçŻőå¡ŢãĂĆ

èőÍèőž

åĹŻåżžèŹŽæŃ§çŐŕåćČéĂŽåÿÿæŸŕäÿžäžĘåőL'èčĚåŠŇçőąçŘĘçňňäÿL'æŰźåŇĚãĂĆ
æ■čåęĆä¡ăåIJÍä¿Ńå■Řäÿ■çIJŃåĹřçŽĎéĆčæăůïijŇsys.path
åŘŸéĞŔåŇĚåŘńæÌèèĞłäžŐçşżçż§PythonçŽĎçŻőå¡ŢïijŇ èĂŇ site-
packagesçŻőå¡ŢåůšçżŕèćńéĞ■åőŽä¡■åĹřäÿĂäÿłæŰřçŽĎçŻőå¡ŢãĂĆ

æIJL'äžĘäÿĂäÿłæŰřçŽĎèŹŽæŃ§çŐŕåćČïijŇäÿŃäÿĂæ■ěårśæŸŕåőL'èčĚäÿĂäÿłåŇĚçőąçŘĘåŹÍïijŇæŕŤå
ä¡ĘåőL'èčĚè£ŹæăůçŽĎåůěåĚůåŠŇåŇĚçŽĎæŮűåĂŹïijŇä¡ăéIJĂèęĄçąőä£Ìä¡ää¡£çŤÍçŽĎæŸŕèŹŽæŃ§çŐŕåćČ
åőČäijŽårĘåŇĚåőL'èčĚåĹřæŰřåĹŻåżžçŽĎsite-packagesçŻőå¡Ţäÿ■åŐžãĂĆ

år¡çőąäÿĂäÿłèŹŽæŃ§çŐŕåćČçIJŃäÿŁåŐżæŸŕPythonåőL'èčĚçŽĎäÿĂäÿłåd'■åĹűïijŇ
äÿ■è£ĞåőČåőđéŹĚäÿŁåŘłåŇĚåŘńäžĘåřŚéĞŘåĞääÿłæŰĞäžűåŠŇäÿĂäžŽçňęåŘůéŞ¿æŐěãĂĆ
æL'ĂæIJL'æăĞåĞĘåžŞåĞ¡æŰĞäżűåŠŇåŘŕæL'ğèąŇèğčéĞŁåŹÍéČ¡æÌěèĞłåŐ§æÌěçŽĎPythonåőL'èčĚãĂĆ
åŽăæ■d'ïijŇåĹŻåżžè£ŹæăůçŽĎçŐŕåćČæŸŕå¿ĹåőźæŸ§çŽĎïijŇåźűäÿŤåĞääźŐäÿ■äijŽæűĹèĂŮæIJžåŹÍèţĎæ

éżŸèőd'æČĚåĘţäÿŃïijŇèŹŽæŃ§çŐŕåćČæŸŕçl'žçŽĎïijŇäÿ■åŇĚåŘńäżžä¡Ţéćİåd'ŰçŽĎçňňäÿL'æŰźåžŞ
åŘŕäžěä¡£çŤÍâĂIJ–system-site-packagesâĂÌéĂL'éąźæÌěåĹŻåżžèŹŽæŃ§çŐŕåćČïijŇä¿ŃåęĆïijŽ

```
bash % pyvenv --system-site-packages Spam
bash %
```

èů§åd'ŽåĚşäžŐ `pyvenv` åŠŇèŹŽæŃ§çŐŕåćČçŽĎä£ąæĄŕåŘŕäžěåŘĆèĂČ PEP 405.

12.15 10.15 åĹĘåŘŚåŇĚ

éŮőécŸ

ä¡ăåůšçżŘçijŰåĘŹäžĘäÿĂäÿłæIJL'çŤÍçŽĎåžŞïijŇæČşårĘåőČåĹĘäžńçżŹåĚűäżŰäžžãĂĆ

èğčåĘşæŰźæąĹ

åęĆæđIJä¡ăæČşåĹĘåŘŚä¡ăçŽĎäžčçăĄïijŇçňňäÿĂäżűäžŃårśæŸŕçżŹåőČäÿĂäÿłåŤŕäÿĂçŽĎåŘ■å■ŮïijŇ
ä¿ŃåęĆïijŇäÿĂäÿłåĚÿådŃçŽĎåĞ¡æŢŕåžŞåŇĚäijŽçśżäijijäÿŃéÌćè£Źæăůïijž

```
projectname/
    README.txt
    Doc/
        documentation.txt
```

(continues on next page)

```
    projectname/
        __init__.py
        foo.py
        bar.py
        utils/
            __init__.py
            spam.py
            grok.py
    examples/
        helloworld.py
        ...
```

èęĄèőľä¡ăçŽĎåŇĚåŘŕäžěåŘŚåÿČåĞžåŐżïijŇéęŰåĚĹä¡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■ěïijŇářśæŸŕåĹŻåžžäÿĂäÿł `MANIFEST.in` æŰĞäžűïijŇåĹŮåĞžæĽĂæIJĽåIJĺä¡ăçŽĎåŇĚäÿ

```
# MANIFEST.in
include *.txt
recursive-include examples *
recursive-include Doc *
```

çąőä£İ `setup.py` åŠŇ `MANIFEST.in` æŰĞäžűæŤ¿åIJĺä¡ăçŽĎåŇĚçŽĎæIJĂéąűçžğçŻőå¡Ţäÿ■ãĂĆäÿĂæŮęä¡ăåůšçżŘåĄŽäžĘè£ŹäžŻïijŇä¡ăářśåŘŕäžěåČŔäÿŃéİćè£ŹæăůæĽğèąŇåŚ¡äžďæİěåĹŻåžžäÿĂäÿłæžŘ

```
% bash python3 setup.py sdist
```

åőČäijŽåĹŻåžžäÿĂäÿłæŰĞäžűæŕŤåęĆâĂİprojectname-1.0.zipâĂİ æĹŰ âĂIJprojectname-1.0.tar.gzâĂİ, åĚůä¡Şä¿ÌètŰäžŐä¡ăçŽĎçşżçż§åźşåŘřãĂĆåęĆæđIJäÿĂåĹĞæ■čåÿÿïijŇè£ŹäÿłæŰĞäžűåřśåŘŕäžěåŘŚéĂĄçżŹåĹńäžža¡£çŤĺæĹŰèĂĚäÿŁäijăèĞş Python Package Index.

èőÍèőž

åŕźäžŐçžŕPythonäżčçăĄïijŇçijŰåĘŹäÿĂäÿłæŹőéĂŽçŽĎ `setup.py` æŰĞäžűéĂŽåÿÿå¿ĹçőĂå■ŢãĂĆ äÿĂäÿłåŘŕèČ¡çŽĎéŮőéćŸæŸŕä¡ăå£ĚéąžæĽŃåŁĺåĹŮåĞžæĽĂæIJĽædĐæäÿĂäÿłåÿÿèğĄéŤŹèŕŕåřśæŸŕäžĚäžĚåŘłåĹŮåĞžäÿĂäÿłåŇĚçŽĎæIJĂéąűçžğçŻőå¡ŢïijŇå£Ÿèőŕäž£åŇĚåŘńåŇ

è£Źäź§æŸŕäÿžäżĂäźĹåIJĺ `setup.py` äÿ■åŕźäžŐåŇĚçŽĎèŕť'æŸŐåŇĚåŘńäžĘåĹŮèąĺ `packages=['projectname', 'projectname.utils']`

åď'ğéČĺĺĹĘPythonçĺŃåžŔåŚŸéČ¡ç§ěéĄŞïijŇæIJL'å¿Ĺåď'ŽçňňäÿL'æŰźåŇĚçőąçŘĘåŹĺä¿ŻéĂL'æŃl'ïijŇ
æIJL'äžŻæŸŕäÿžäżĘæŻ£äżčæăĞåĞĘåžŞäÿ■çŽĎdistutilsãĂĆæşĺæĎŔåęĆæđIJä¡ää¿ÌèţŰè£ŹäžŻåŇĚïijŇ
çŤĺæĹůåŔŕèČ¡äÿ■èČ¡åőL'èčĚä¡ăçŽĎè¡ŕäżűïijŇéŹď'éİđäžŰäżňåůšçżŔäžŇåĚĹåőL'èčĚè£ĞæL'ĂéIJĂèęĄçŽĎå
æ■čåŻääęĆæ■ď'ïijŇä¡ăæŻť'åžŤèŕěæŮűåĹżèőŕä¡ŔèűŁçőĂå■Ţèűŀåě¡çŽĎéĄŞçŘĘãĂĆ
æIJĂåě¡èől'ä¡ăçŽĎäžčçăĄä¡£çŤĺæăĞåĞĘçŽĎPython 3åőL'èčĚãĂĆ
åęĆæđIJåĚűäżŮåŇĚäź§éIJĂèęĄçŽĎèŕİïijŇåŘŕäžěéĂŽè£ĞäÿĂäÿłåŘŕéĂL'éąźæİěæŤŕæŇĄãĂĆ

åŕźäžŐæűL'åŘŁåĹřCæL'l'åśŢçŽĎäžčçăĄæL'ŞåŇĚäÿŐåĹĘåŘŠåŕśæŻť'åď'■æİĆçĆźäžĘãĂĆ
çňň15çńääŕźåĚşäžŐCæL'l'åśŢçŽĎè£ŹæŰźéİćç§ěèŕĘæIJL'äÿĂäžŻèŕęçżĘèőšèğčïijŇçL'źåĹńæŸŕåIJĺ15.2åřŔè

13 çňňå■ĄäÿĂçńăïijŽç¡ŚçżIJäÿŐWebçijŰçÍŃ

æIJňçńăæŸŕåĚşäžŐåIJĺç¡ŚçżIJåžŢçŤĺåŠŇåĹĘåÿČåijŔåžŢçŤĺäÿ■ä¡£çŤĺçŽĎåŘĎçğ■äÿżécŸãĂĆäÿżécŸã

Contents:

13.1 11.1 ä¡IJäÿžåőćæĹůçńŕäÿŐHTTPælJ■åŁąäžď'äžŠ

éŮőécŸ

ä¡ăéIJĂèęĄéĂŽè£ĞHTTPå■ŔèőőäžěåőćæĹůçńŕçŽĎæŰźåijŔèő£éŮőåď'Žçğ■æIJ■åŁąãĂĆä¿ŃåęĆïijŇäÿ

èğčåĘşæŰźæąĹ

åŕźäžŐçőĂå■ŢçŽĎäžŃæČĚæİěèŕť'ïijŇéĂŽåÿÿä¡£çŤĺ `urllib.request` æĺąåİŮåřšåď'§äžĘãĂĆä¿ŃåęĆïijŇåŘŠéĂĄäÿĂäÿłçőĂå■ŢçŽĎHTTP GETèŕůæśĆåĹřè£IJçĺŃçŽĎæIJ■åŁąäÿŁïijŇåŘŕäžěè£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)

```
u = request.urlopen(url+'?' + querystring)
resp = u.read()
```

åęĆædIJä¡ăéIJĂèęĄä¡£çŤĺPOSTæŰźæşŢåIJĺèřůæśĆäÿżä¡Şäÿ■åŘŠéĂĄæ§ěèřćåŘĆæŢřïijŇåŘřäżěářĘåŘ
urlopen() åĞ¡æŢřïijŇářšåČŔè£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()
```

åęĆædIJä¡ăéIJĂèęĄåIJĺåŘŠåĞžçŽĎèřůæśĆäÿ■æŘŘä¿ŻäÿĂäžŻèĞłåőŽäźĽçŽĎHTTPåd'ť'ïijŇä¿ŃåęĆä£
user-agent å■Ůæőţ,åŘřäżěåĹŻåžžäÿĂäÿłåŇĚåŘńå■Ůæőţå˘ijçŽĎå■ŮåĚÿïijŇåźűåĹŻåžžäÿĂäÿłRequestå
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()
```

åęĆædIJéIJĂèęĄäžd'äžŠçŽĎæIJ■åŁąæŕŤäÿŁéİćçŽĎä¿Ńå■ŘéČ¡èęĄåd'■æİĆïijŇäź§èőÿåžŤèŕěåŐżçIJŃç
requests åžŞïijĹhttps://pypi.python.org/pypi/requestsïijL'ãĂĆä¿ŃåęĆïijŇäÿŃéİćè£Źäÿłçdžä¿ŃéĞĞçŤĺreques

```
import requests

```

(continues on next 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¿ŮäÿĂæŔŘçŽĎçĽźæĂğářśæŸŕåőČèČ¡äžěåď’Žçğ■æŰźåijŔäžŐèŕůæ
`resp.text` åÿęçżŹæĹŚäžňçŽĎæŸŕäžěUnicodeèğčçăĄçŽĎåŞ■åžŤæŰĞæIJňãĂĆä¡ĘæŸŕïijŇåęĆæđIJåŐżèě
`resp.content` ïijŇářśäijŽå¿ŮåĹřåŐ§ågŃçŽĎäžŇè£ŻåĹűæŢřæ■őãĂĆåŔęäÿĂæŰźéİćïijŇåęĆæđIJèő£éŮć
`resp.json` ïijŇéĆčäźĹářśäijŽå¿ŮåĹřJSONæäijåijŔçŽĎåŞ■åžŤåĘĚåőźãĂĆ

äÿŃéİćè£Źäÿłçđ’žä¿ŃåĹl’çŤÍ `requests` åžŞåŔŚèţůäÿĂäÿłHEAD-èŕůæśĆïijŇåźűäžŐåŞ■åžŤäÿ■æŔŘåŔŰåĞžäÿĂäžŻHTTPåđ’t’æŢřæ■őçŽĎå■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']
```

äÿŃéİćæŸŕäÿĂäÿłåĹl’çŤÍrequestséĂŽè£Ğå§žæIJňèőd’èŕĄçŹżå¡ŢPypiçŽĎä¿Ńå■ŘïijŽ

```
import requests

resp = requests.get('http://pypi.python.org/pypi?:action=login',
                    auth=('user','password'))
```

äÿŃéİćæŸŕäÿĂäÿłåĹl’çŤÍrequestsåřĘHTTP cookiesäžŐäÿĂäÿłèŕůæśĆäijåéĂŠåĹřåŔęäÿĂäÿłçŽĎä¿Ńå■

```
import requests

# First request
```

(continues on next 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■ŘæŸŕçŤĺrequestsäÿŁäijăåĘĚåőźïijŽ

```
import requests
url = 'http://httpbin.org/post'
files = { 'file': ('data.csv', open('data.csv', 'rb')) }

r = requests.post(url, files=files)
```

èőĺèőž

åŕźäžŐçIJ§çŽĎå¿ĹçőĂå■ŢHTTPåőćæĹůçńŕäžčçăĄïijŇçŤĺåĘĚç¡őçŽĎ urllib æĺąåİŮéĂŽåÿÿåřśèűşåď§äžĘãĂĆä¡ĘæŸŕïijŇåęĆæđIJä¡ăèęĄåĄŽçŽĎäÿ■äžĚäžĚåŘłæŸŕçőĂå■ŢçŽĎGETæĹ requests åď ğæŸ¿èžńæĽŃçŽĎæŮűåĂŹäžĘãĂĆ

ä¿ŃåęĆïijŇåęĆæđIJä¡ăåĘşåőŽåİŽæŇĄä¡£çŤĺæăĞåĞĘçŽĎçÍŃåžŔåžŞèĂŇäÿ■èĂČèŹŚåČŔ requests è£ŹæăůçŽĎçňñäÿĽæŰźåžŞïijŇéĆčäźĹäź§èőÿářśäÿ■å¿Ůäÿ■ä¡£çŤĺåžŢåśĆçŽĎ http.client æĺąåİŮæİěåőđçŐřèĞłåůśçŽĎäžčçăĄãĂĆærŤæŰźèŕť ïijŇäÿŃéİćçŽĎäžčçăĄåśŢçď žäžĘåęĆä

```
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£Ěéąžçij ŰåĘŹæűĽ'åŔŁäžčçŘĘãĂĄèőď'èŕĄãĂĄcookiesäžěåŔŁåĚűäžŰäÿĂäžŽ urllib åřśæŸ¿å¿ŮçĽźåĹńåĹńæĽ'■åŠŇåŢřåŮęãĂĆærŤæŰźèŕť ïijŇäÿŃéİćè£Źäÿłçđ'žä¿Ńåőđç ŐřåIJÍPython

```
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)

```
resp = u.read()

# From here. You can access more pages using opener
...
```

åİęçŹ¡èŕť'ïijŇæL'ĂæIJL'çŽĎè£ŹäžŻæŞ■ä¡IJåIJÍ requests åžŞäÿ■éČ¡åŔŸå¿ŮçőĂå■ŢçŽĎåď'ŽãĂĆ

åIJÍåijĂåŔŚè£ĞçÍŇäÿ■æţŇèŕŢHTTPåőćæĹůçńŕäžčçăĄåÿÿåÿÿæŸŕå¿Ĺäžď'äžžæšőäÿğçŽĎïijŇåŻääÿžæĽ //httpbin.orgïijL'ãĂĆè£ŹäÿłçńŹçĆźäijŽæŐěæŤűåŔŚåĞžçŽĎèřůæśĆïijŇçĎűåŘŐäžěJSONçŽĎå¡ćåijŔářĘçŻÿ

```
>>> 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¡ŚçńŹäÿŁåĄŽåőđéłŇåÿÿåÿÿæŸŕåŔŕåŔŰçŽĎåŁđæşŢãĂĆårď'åĚűæŸŕå¡ŞæĹŚäžňéÍcåŕź3æňąçŹ

år¡çőąæIJňèŁĆæšąæIJL'æűL'åŔŁïijŇ request åžŞè£Ÿårźèőÿåď'ŽéńŸçžğçŽĎHTTPåőćæĹůçńŕå■Ŕèőő requests æĺąåİŮçŽĎæŰĞæąčïijĹhttp://docs.python-requests.org)èt'ÍéĞŔå¿ĹéńŸïijĹåİęçŹ¡èŕť'ærŤåIJÍè£Źç§■ç§■çŽĎäÿĂèŁĆçŽĎçŕĞáźĚäÿ■æL'ĂæŔŘä¿ŻçŽĎäžżä¡Ţä£ą

13.2 11.2 åĹŻåžžTCPælJ■åŁąåŹÍ

éŮőécŸ

ä¡ăæČşåőđçŐřäÿĂäÿłæIJ■åŁąåŹÍïijŇéĂŽè£ĞTCPå■ŔèőőåŠŇåőćæĹůçńŕéĂŽä£ąãĂĆ

èğčåĘşæŰźæąĹ

åĹŻåžžäÿĂäÿłTCPæIJ■åŁąåŹÍçŽĎäÿĂäÿłçőĂå■ŢæŰźæşŢæŸŕä¡£çŤÍ socketserver åžŞãĂĆä¿ŃåęĆïijŇäÿŇéÍćæŸŕäÿĂäÿłçőĂå■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)

```
        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■ïijÑä¡ăåőŽäźL'äžĘäÿĂäÿłçL'źæőŁçŽĎåď'ĎçŘĘçśżïijÑåőđçŐřäžĘäÿĂäÿł handle() æŰźæşŢïijÑçŤÍæİěäÿžåőćæĹůçńŕè£đæŐěæIJ■åŁąãĂĆ request åśđæĂğæŸŕåőćæĹůçńŕsocketïijÑclient_address æIJL'åőćæĹůçńŕåIJřåİĂãĂĆ äÿžäžĘæţŃèŕŢè£ŹäÿłæIJ■åŁąåŹĺïijÑè£ŘèąÑåőČåźűæL'Şåij ĂåŔęåď'ŰäÿĂäÿłPythonè£ŽçÍÑè£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'ŽæŮűåĂŹïijÑåŔŕäžěå¿Ĺåőźæ ŸŞçŽĎåőŽäźL'äÿĂäÿłäÿ■åŘŇçŽĎåď'ĎçŘĘåŹĺãĂĆäÿŃéĺćæŸŕäÿĂ StreamRequestHandler å§žçśżåřĘäÿĂäÿłçśżæŰĞäžűæŐěåŘčæŤ¿ç¡őåIJĺåžŢåśĆsocketäÿŁçŽĎä¿Ńå■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()
```

èőĺèőž

socketserver åŔŕäžěèőľ'æĹŚäžňå¿Ĺåőźæ ŸŞçŽĎåĹŻåžžçőĂå■ŢçŽĎTCPæIJ■åŁąåŹĺãĂĆ ä¡ĘæŸŕïijÑä¡ăéIJĂèęĄæşĺæĎŘçŽĎæŸŕïijÑéżŸèőď'æČĚåĘţäÿŃè£ŹçğğæIJ■åŁąåŹĺæŸŕå■Ţçž£çĺÑçŽĎïijÑäÿ åęĆæđIJä¡ăæČşåď'ĎçŘĘåď'ŽäÿłåőćæĹůçńŕïijÑåŔŕäžěåĹİåğŃåŇŰäÿĂäÿł ForkingTCPServer æĹŰèĂĚæŸŕ ThreadingTCPServer åŕźèśąãĂĆä¿ŃåęĆïijŽ

```
from socketserver import ThreadingTCPServer


if __name__ == '__main__':
    serv = ThreadingTCPServer(('', 20000), EchoHandler)
    serv.serve_forever()
```

ä¡£çŤÍforkæĹŰçž£çÍŃæIJ■åŁąåŹÍæIJL'äÿłæ¡IJåIJÍéŮőécŸåřśæŸŕåőČäžňäijŽäÿžæŕŔäÿłåőćæĹůçńŕè£đæçŤśäžŐåőćæĹůçńŕè£đæŐěæŢřæŸŕæšąæIJL'éŹŘåĹűçŽĎïijŇåŻäæ■ď'äÿĂäÿłæĄűæĎŘçŽĎéżŠåőćåŘŕäžěåŘŇ

åęĆæđIJä¡ăæŃĚå£Čè£ŹäÿłéůőécŸïijŇä¡ăåŘŕäžěåĹŻåžžäÿĂäÿłécĎåĚĹåĹĘéĚ■åď'gåřŘçŽĎåůěä¡IJçž£çä¡ăåĚĹåĹŻåžžäÿĂäÿłæŹőéĂŽçŽĎéIđçž£çÍŃæIJ■åŁąåŹĺïijŇçĎűåŘŐåIJÍäÿĂäÿłçž£çÍŃæśääÿ■ä¡£çŤÍ `serve_forever()` æŰźæşŢæİěåŘŕåŁĺåőČäžňãĂĆ

```
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()
```

äÿĂèĹňæIěèőšïijŇäÿĂäÿł `TCPServer` åIJÍåőđä¿ŃåŇŰçŽĎæŮűåĂŹäijŽçżŠåőŽåźűæ£Äæť'żçŻÿåžŤçŽ `socket` ãĂĆ äÿ■è£ĞïijŇæIJL'æŮűåĂŹä¡ăæČşéĂŽè£Ğèő¿ç¡őæ§ŘäžŻéĂL'éąźåŐżèřČæŢť'åžŢäÿŃçŽĎ *socket*‘ ïijŇåŘŕäžěèő¿ç¡őåŔĆæŢř `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¡őéąźïijŇåőČåĚĄèõÿæIJ■åŁąåŹĺéĞ■æçŤśäžŐèęĄèćńçżŘåÿÿä¡£çŤĺåĹřïijŇåőČèćńæŤ¿ç¡őåĹřçśżåŘŸéĞŔäÿ■ïijŇåŘŕäžěçŻť'æŐěåIJÍ `TCPServer` äÿŁéİćèő¿ç¡õãĂĆ åIJÍåőđä¿ŃåŇŰæIJ■åŁąåŹĺçŽĎæŮűåĂŹåŐżèő¿ç¡őåőČçŽĎåĂijïijŇåęĆäÿŃ

```
if __name__ == '__main__':
    TCPServer.allow_reuse_address = True
    serv = TCPServer(('', 20000), EchoHandler)
    serv.serve_forever()
```

åIJÍäÿŁéİćçď'žä¿Ńäÿ■ïijŇæĹŚäžňæijŤçď'žäžĘäÿď'çğ■äÿ■åŘŇçŽĎåď'ĎçŘĘåŹĺå§žçśżïijĹ `BaseRequestHandler` åŠŇ `StreamRequestHandler` ïijL'ãĂĆ `StreamRequestHandler` æŻť'åŁăçĄţæť'żçĆźïijŇèČ¡éĂŽè£Ğèő¿ç¡őåĚűäžŰçŽĎçśżåŘŸéĞŘæİěæŤŕæŇ

```
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£ŸéIJĂèęĄæşÍæĎŔçŽĎæŸŕåůÍåď ğéČÍåĹĘPythonçŽĎéńŸåśĆç¡ŚçżIJæĺąåİŮïijĹærŤåęĆI
RPCç■L’ïijL’éČ¡æŸŕåżžçńŇåIJÍ socketserver åŁ§èČ¡äźŇäÿŁãĂĆ
äź§åřśæŸŕèŕť’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 åĹŻåžžUDPæIJ■åŁąåŹÍ

éŮőécŸ

ä¡ăæČşåőđçŐřäÿĂäÿłå§žäžŐUDPå■ŔèőőçŽĎæIJ■åŁąåŹÍæİěäÿŐåőćæĹůçńŕéĂŽä£ąãĂĆ

èğčåĘşæŰźæąĹ

èů§TCPäÿĂæăůïijŇUDPæIJ■åŁąåŹÍäź§åŔŕäžěéĂŽè£Ğä¡£çŤÍ `socketserver` åžŞå¿ĹåőźæŸŞçŽĎèćńåĹŻåžžãĂĆ ä¿ŃåęĆïijŇäÿŃéİćæŸŕäÿĂäÿłçőĂå■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■äÿĂæăůïijÑä¡ăåĚĹåőŽäźĽäÿĂäÿłåőđçŐř `handle()` çĽźæőŁæŰźæşŢçŽĎçśžïijÑäÿžåőćæĹůçńŕè£đæŐěæIJ■åŁąãĂĆ è£ŹäÿłçśžçŽĎ `request` åśđæĂğæŸŕäÿĂäÿłåŇĚåŘńäžĘæŢřæ■őæŁěåŠŇåžŢåśĆsock-etåŕźèśąçŽĎåĚČçżĎãĂĆ`client_address` åŇĚåŘńäžĘåőćæĹůçńŕåIJŕåİĂãĂĆ

æĹŚäžňæİěæţŃèŕŢäÿŃè£ŹäÿłæIJ■åŁąåŹÍïijŇéęŰåĚĹè£ŘèąŇåőČïijŇçĎűåŘŐæĽŞåijĂåŔęåd'ŰäÿĂäÿłI

```
>>> 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¿¿çŽĎæŢřæ■őæŁě(æűĹæĄŕ)åŠŇåőćæĹůçńŕåIJŕåİĂãĂ
åőČèęĄçżŹåőćæĹůçńŕåŻđåŘŚäÿĂäÿłæŢřæ■őæŁěãĂĆåŕźäžŐæŢřæ■őæŁěçŽĎäijäéĂĄïijŇ
ä¡ăåžŤèŕěä¡£çŤÍsocketçŽĎ `sendto()` åŠŇ `recvfrom()` æŰźæşŢãĂĆ
åř¡çőąäijăçż§çŽĎ `send()` åŠŇ `recv()` äź§åŔŕäžěè¿¿åĹřåŘŇæăůçŽĎæŢĹædIJïijŇ
ä¡ĘæŸŕåĽ■éİćçŽĎäÿd'äÿłæŰźæşŢåŕźäžŐUDPè£đæŐěèĂŇèÍĂæŻť'æŹőéĄ■ãĂĆ

çŤśäžŐæšąæIJL'åžŢåśĆçŽĎè£đæŐěïijŇUPDæIJ■åŁąåŹĺçŻÿåŕźäžŐTCPæIJ■åŁąåŹĺæÌěèőšåőđçŐřètůæÌ
äÿ■è£ĞïijŇUDPåđ'l'çŤ§æŸŕäÿ■åŔŕéÌăçŽĎïijĹåŽääÿžéĂŽä£ąæšąæIJL'åžžçńŇè£đæŐěïijŇæűĹæĄŕåŔŕèČ¡äÿ
åŻăæ■đ'éIJĂèęĄçŤśä¡äèĞłåůśæÌěåĘşåőŽèřěæĂŐæăůåđ'ĎçŘĘäÿćåđ'śæűĹæĄŕçŽĎæČĚåĘţãĂĆè£Źäÿłåůšçżl
äÿ■è£ĞéĂŽåÿÿæÌěèŕť'ïijŇåęĆæđIJåŔŕéÌăæĂğåŕźäžŐä¡ăçÍŃåžŔå¿ĹéĞ■èęĄïijŇä¡äéIJĂèęĄåĂ§åŁl'äžŐåžŔåÍ
UDPéĂŽåÿÿèćńçŤĺåIJĺéĆčäžŽåŕźäžŐåŔŕéÌăäijäè¿ŞèęĄæśĆäÿ■æŸŕå¿ĹéńŸçŽĎåIJžåŘĹãĂĆä¿ŃåęĆïijŇåIJĺå
æŮăéIJĂè£ŤåŻđæĄćåđ'■äÿćåđ'śçŽĎæŢřæ■őåŇĚïijĹçÍŃåžŔåŔłéIJĂçőĂå■ŢçŽĎå£¡çŢěåőČåźűçżğçż■åŘŚåL

UDPServer çśżæŸŕå■Ţçž£çÍŃçŽĎïijŇäź§åřśæŸŕèŕť'äÿĂæňąåŔłèČ¡äÿžäÿĂäÿłåőćæĹůçńŕè£đæŐěæIJ■
åőđéŹĚä¡£çŤĺäÿ■ïijŇè£ŹäÿłæŮăèőžæŸŕåŕźäžŐUDPè£ŸæŸŕTCPéČ¡äÿ■æŸŕäżĂäźĹåđ'ğéŮőéćŸãĂĆ
åęĆæđIJä¡ăæČşèęĄåźűåŔŚæŞ■ä¡IJïijŇåŔŕäžěåőđä¿ŃåŇŰäÿĂäÿł ForkingUDPServer
æĹŰ ThreadingUDPServer åŕźèśąïijŽ

```
from socketserver import ThreadingUDPServer

   if __name__ == '__main__':
    serv = ThreadingUDPServer(('',20000), TimeHandler)
    serv.serve_forever()
```

çŻť'æŐěä¡£çŤĺ socket æÌěåőđçŐřäÿĂäÿłUDPæIJ■åŁąåŹĺäź§äÿ■éŽ¿ïijŇäÿŃéÍćæŸŕäÿĂäÿłä¿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§æĹŘåŕźåžŢçŽĎIPåIJřåÌĂéŻĘ

éŮőécŸ

ä¡ăæIJL'äÿĂäÿłCIDRç¡ŚçżIJåIJřåÌĂærŤåęĆâĂIJ123.45.67.89/27âĂÌïijŇä¡ăæČşåřĘåĚűè¡ňæ■ćæĹŘåőČa
ïijĹærŤåęĆïijŇâĂIJ123.45.67.64âĂÌ, âĂIJ123.45.67.65âĂÌ, âĂę, âĂIJ123.45.67.95âĂÌ)ïijL'

èğčåĘşæŰźæąĹ

åŔŕäžěä¡£çŤĺ ipaddress æÍąåÌŮå¿ĹåőźæŸŞçŽĎåőđçŐřè£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 ŰïijŇä¡ăè£ŸåŔŕäžěæL'ğèąŇç¡ŚçżIJæĹŘåŚŸæčĂæ§ëäźŃçśżçŽĎæŞ■ä¡IJïijŽ

```
>>> a = ipaddress.ip_address('123.45.67.69')
```

(continues on next page)

```
>>> a in net
True
>>> b = ipaddress.ip_address('123.45.67.123')
>>> b in net
False
>>>
```

äÿĂäÿłIPåIJřåİĂåŠŇç¡ŚçżIJåIJřåİĂèČ¡éĂŽè£ĞäÿĂäÿłIPæŐěåŔčæİěæŇĞåőŽïijŇä¿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 æÍąåİŮæIJL'å¿Ĺåď'ŽçśżåŔŕäżěèąÍçď'žIPåIJřåİĂãĂĄç¡ŚçżIJåŠŇæŐěåŔčãĂĆ
å¡Şä¡ăéIJĂèęĄæŞ■ä¡IJç¡ŚçżIJåIJřåİĂïijĹæŕŤåęĆèğčæđŘãĂĄæL'Şå■řãĂĄéłŇèŕĄç■L'ïijL'çŽĎæŮűåĂŹäijŽå

èęĄæşÍæĎŔçŽĎæŸŕïijŇipaddress æÍąåİŮèů§åĚűäžŰäÿĂäžŽåŠŇç¡ŚçżIJçŻÿåĚşçŽĎæÍąåİŮæŕŤåęĆ
socket åžŞäžď'éŻĘå¿ĹåřŚãĂĆ æL'ĂäžěïijŇä¡ääÿ■èČ¡ä¡£çŤÍ IPv4Address
çŽĎåőđä¿ŃæİěäžčæŻ£äÿĂäÿłåIJřåİĂå■ŮçňęäÿšïijŇä¡ăéęŰåĚĹå¿ŮæŸ¿åijŔçŽĎä¡£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'ŽçŻÿåĚşåĘĚåőźïijŇèŕůåŔĆèĂČ An Introduction to the ipaddress Module

13.5 11.5 åĹŻåżžäÿĂäÿłçőĂå■ŢçŽĎRESTæŐěåŔč

éŮőécŸ

ä¡ăæČşä¡£çŤÍäÿĂäÿłçőĂå■ŢçŽĎRESTæŐěåŔčéĂŽè£Ğç¡ŚçżIJè£IJçÍŃæŐğåĹűæĹŰèő£éŮőä¡ăçŽĎåžŤ

èğčåĘşæŰźæąĹ

æđĎåżžäÿĂäÿłRESTéčŐæăijçŽĎæŐěåŔčæIJĂçőĂå■ŢçŽĎæŰźæşŢæŸŕåĹŻåżžäÿĂäÿłå§žäžŐWSGIæă 3333ïijL'çŽĎå¿ĹářŔçŽĎåžŞïijŇäÿŃéİćæŸŕäÿĂäÿłä¿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¡ääŔłéIJĂèęĄçijŰåĘŹäÿ■åŘŇçŽĎåď˙ĎçŘĘåŹĺïijŇářśåČŔäÿŃéİćè£

```
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)

```
    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()
```

èęĄæţŃèŕŢäýŃè£ŹäýłæIJ■åŁąåŹĺïijŇä¡ăåŔŕäžěä¡£çŤĺäýĂäýłæţŔèğĹåŹĺæĹŰ `urllib` åŠŇåőČäžďäžŠãĂĆä¿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)

(çż■äÿŁéąţ)

```
<?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

åIJÍçijŰåĘŹRESTæŐěåŔčæŮűïijŇéĂŽåÿÿéČ¡æŸŕæIJ■åŁąäžŐæŹőéĂŽçŽĎHTTPèŕůæśĆãĂĆä¡ĘæŸŕè
è£ŹäžŻæŢřæ■őäžěåŘĎçğ■æäĞåĞĘæäijåijŔçijŰçăĄïijŇæŕŤåęĆXMLãĂĄJSONæĹŰCSVãĂĆ
åř¡çőąçÍŇåžŔçIJŇäÿŁåŐżå¿ĹçőĂå■ŢïijŇä¡ĘæŸŕäžěè£Źçğ■æŰźåijŔæŔŘä¿ŻçŽĎAPIåŕźäžŐå¿Ĺåď'ŽåžŤçŤÍç

ä¿ŇåęĆïijŇéŢ£æIJ§è£ŘèąŇçŽĎçÍŇåžŔåŔŕèČ¡äijŽä¡£çŤÍäÿĂäÿłREST
APIæİěåőđçŐřçŻŚæŐğæĹŰèŕŁæŰ■ãĂĆ åď'ğæŢřæ■őåžŤçŤÍçÍŇåžŔåŔŕäžěä¡£çŤÍRESTæİěæđĎåžžäÿĂäÿłæ
RESTè£ŸèČ¡çŤÍæİěæŐğåĹűçąňäžűěő¿åď'ĞæŕŤåęĆæIJžåŹÍäžžãĂĄäijäæĎ§åŹÍãĂĄåůěåŐĆæĹŰçĄŕæşąãĂĆ
æŻť éĞ■èęĄçŽĎæŸŕïijŇREST APIåůšçżŔèćńåď'ğéĞŔåőćæĹůçńŕçijŰçÍŇçŐŕåćČæL'ĂæŤŕæŇĄïijŇæŕŤåęĆ-
Javascript, Android, iOSç■L'ãĂĆ åŻäæ■d'ïijŇåĹl'çŤÍè£Źçğ■æŐěåŔčåŔŕäžěèől'ä¡ăåijĂåŔŠåĞžæŻť åŁăåď'■æ

äÿžäžĘåőđçŐřäÿĂäÿłçőĂå■ŢçŽĎRESTæŐěåŔčïijŇä¡ăåŔłéIJĂèől'ä¡ăçŽĎçÍŇåžŔäžčçăĄæżąęűşPython
WSGIèćńæăĞåĞĘåžŞæŤŕæŇĄïijŇåŘŇæŰűäź§èćńçżİåď'ğéČĺåĹĘçňňäÿL'æŰźwebæąĘæđűæŤŕæŇĄãĂĆ
åŻäæ■d'ïijŇåęĆæđIJä¡ăçŽĎäžčçăĄéĄţå¿łè£ŹäÿłæăĞåĞĘïijŇåIJĺåŘŐéIćçŽĎä¡£çŤÍè£ĞçÍŇäÿ■åřśäijŽæŻť åŁ

åIJÍWSGIäÿ■ïijŇä¡ăåŔŕäžěåČŔäÿŇéIćè£ŹæăůçžęåőŽçŽĎæŰźåijŔäžěäÿĂäÿłåŔŕèŘçŤÍåŕźèśąå¡ćåijŔæI

```
import cgi

def wsgi_app(environ, start_response):
    pass
```

environ åśđæĂğæŸŕäÿĂäÿłå■ŮåĚÿïijŇåŇĚåŘńäžĘäżŐwebæIJ■åŁąåŹĺåęĆA-
pache[åŘĆèĂČInternet RFC 3875]æŔŘä¿ŻçŽĎCGIæŐěåŘčäÿ■èŐůåŘŰçŽĎåĂijãĂĆ
èęĄåřĘè£ŹäžŻäÿ■åŘŇçŽĎåĂijæŔŘåŘŰåĞžæİěïijŇä¡ăåŔŕäžěåČŔè£ŹäźĹè£ŹæăůåĘŹïijŽ

```
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'žäžĘäÿĂäžŻåÿÿèğĄçŽĎåĂijãĂĆenviron['REQUEST_METHOD']
äžčèąÍèŕůæśĆçśžđŇåęĆGETãĂĄPOSTãĂĄHEADç■L'ãĂĆ environ['PATH_INFO']
èąÍçď'žèćńèŕůæśĆèţĎæžŘçŽĎèůŕå¿ĎãĂĆ èřČçŤÍ cgi.FieldStorage()
åŔŕäžěäžŐèŕůæśĆäÿ■æŔŘåŘŰæ§ěèŕćåŔĆæŢřåźűářĘåőČäžňæŤ¿åĚěäÿĂäÿłçśżå■ŮåĚÿåŕźèśąäÿ■äžěä¿£åŘŐ

start_response åŔĆæŢřæŸŕäÿĂäÿłäÿžäžĘåĹİåğŇåŇŰäÿĂäÿłèřůæśĆåŕźèśąèĂŇå£Ěéążèćńèř
çňňäÿĂäÿłåŔĆæŢřæŸŕè£ŤåŻđçŽĎHTTPçŁűæĂĄåĂijïijŇçňňäžŇäÿłåŔĆæŢřæŸŕäÿĂäÿł(åŘ■,åĂij)åĚĆçżĎ

```
def wsgi_app(environ, start_response):
    pass
    start_response('200 OK', [('Content-type', 'text/plain')])
```

äÿžäžĘè£ŤåŻđæŢřæ■őïijŇäÿĂäÿłWSGIçÍŃåžŔå£Ěéążè£ŤåŻđäÿĂäÿłå■ŮèŁĆå■ŮçňęäÿšäžŕåĹŮãĂĆåŕ

```
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£ŸåŔŕäžěä¡£çŤÍ yield ïijŽ

```
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£ŹéĞŇèęĄåijžèřĆçŽĎäÿĂçĆźæŸŕæIJÄåŘŐè£ŤåŻđçŽĎå£ĚéążæŸŕå■ŮèŁĆå■Ůçňęäÿšã
å¡ŞçĎűïijŇåźűæšąæIJL'èęĄæśĆä¡ăè£ŤåŻđçŽĎäÿĂåőŽæŸŕæŰĞæIJñïijŇä¡ăåŔŕäžěå¿Ĺè¡żæİ¿çŽĎçijŰåĘŹäÿ

åř¡çőąWSGIçÍŃåžŔéĂŽåÿÿèćńåőŽäźL'æĹŘäÿĂäÿłåĞ¡æŢřïijŇäÿ■è£Ğä¡ääź§åŔŕäžěä¡£çŤÍçśžåőđä¿ŃæĺÍ
__call__() æŰźæşŢãĂĆä¿ŃåęĆïijŽ

```
class WSGIApplication:
    def __init__(self):
        ...
    def __call__(self, environ, start_response)
       ...
```

æĹŚäžňåůšçżŔåIJÍäÿŁéİćä¡£çŤÍè£Źçğ■æŁĂæIJŕåĹŻåžž PathDispatcher çśžãĂĆ
è£ŹäÿłåĹĘåŘŚåŹÍäžĚäžĚåŘłæŸŕçőąçŘĘäÿĂäÿłå■ŮåĚÿïijŇårĘ(æŰźæşŢ,èůŕå¿Ď)åŕźæŸăårĎåĹråď'ĎçŘĘåŹÍ
å¡ŞäÿĂäÿłèŕůæśĆåĹřæİěæŮűïijŇåőČçŽĎæŰźæşŢåŠŇèůŕå¿Ďèćńæ ŔŘåŔŰåĞžæİěïijŇçĎűåŘŐèćńåĹĘåŘŚåĹ
åŘęåď'ŰïijŇäžžä¡Ţæ§èèŕćåŔŸéĞŔäijŽèćńèğčædŘåŘŐæŤ¿åĹřäÿĂäÿłå■ŮåĚÿäÿ■ïijŇäžě
environ['params'] å¡ćåijŔå■ŸåĆÍãĂĆ åŘŐéİcè£Źäÿłæ■ěéłď'åď'łåÿÿèğĄïijŇæL'Ääžěåžžèőőä¡ääIJÍåĹĘ
ä¡£çŤÍåĹĘåŘŚåŹÍçŽĎæŮűåĂŹïijŇä¡ääŔłéIJÄçőÄå■ŢçŽĎåĹŻåžžäÿĂäÿłåődä¿ŃïijŇçĎűåŘŐéĂŽè£ĞåőČæş
çijŰåĘŹè£ŹäžŽåĞ¡æŢřåžŤèŕěèűĚçžğçőÅå■ŢäžĘïijŇåŔłèęĄä¡äéĄţå¿ł
start_response() åĞ¡æŢřçŽĎçijŰåĘŹèğĎåĹŹïijŇåźűäÿŤæIJÄåŘŐè£ŤåŻđå■ŮèŁĆå■Ůçňęäÿšå■şåŔŕã

å¡ŞçijŰåĘŹè£Źçğ■åĞ¡æŢřçŽĎæŮűåĂŹè£ŸéIJÄæşÍæĎŔçŽĎäÿĂçĆźåŕśæŸŕåŕźäžŐå■ŮçňęäÿšæÍąæİ£çŽ
æšąäžžæĎ£æĎŔåĘŹéĆčçğ■åĹřåď'ĎæůûåĂŔĹçİÄ print() åĞ¡æŢř ãĂĄXM-
LåŠŇåď'ğéĞŔæäijåijŔåŇŰæŞ■ä¡IJçŽĎäžčçăĄãĂĆ æĹŚäžňäÿŁéİcä¡£çŤÍäžĘäÿL'åijŢåŔůåŇĚåŔńçŽĎéĆĎåĚ
è£Źçğ■æŰźåijŔçŽĎåŔŕäžěèőľæĹŚäžňå¿Ĺåőźæ ŸŞçŽĎåIJÍäžěåŘŐä£őæŤźè¿ŞåĞžæäijåijŔ(åŘłéIJÄèęĄä£őæ

æIJÄåŘŐïijŇä¡£çŤÍWSGIè£ŸæIJL'äÿĂäÿłå¿ĹéĞ■èęĄçŽĎéČĺåĹĘåŕśæŸŕæšąæIJL'äžĂäźĹåIJřæŰźæŸŕé

åŻääÿžæäĞåĞĘåŕźäžŐæIJ■åŁąåŹĺåŠŇæąĘæđűæŸŕäÿ■çńŇçŽĎïijŇä¡ăåŔŕäžěåřĘä¡ăçŽĎçĺŃåžŔæŤ¿åĚěäżäæĹŚäžňä¡£çŤĺäÿŃéİćçŽĎäžčçăĄæţŃèŕŢæţŃèŕŢæIJňèŁĆäžčçăĄïijŽ

```
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()
```

äÿŁéİćäžčçăĄåĹŻåżžäžĘäÿĂäÿłçőÅå■ŢçŽĎæIJ■åŁąåŹĺïijŇçĎűåŘŐä¡ăåřšåŔŕäžěæİěæţŃèŕŢäÿŃä¡ăçŽĎ
æIJĂåŘŐïijŇå¡Şä¡ăåĞĘåd'Ğè£ŻäÿĂæ■ěæL'l'åśŢä¡ăçŽĎçĺŃåžŔçŽĎæŮűåĂŹïijŇä¡ăåŔŕäžěä£őæŤźè£Źäÿłäżè

WSGIæIJňèžńæŸŕäÿĂäÿłå¿ĹåřŔçŽĎæăĞåĞĘãĂĆåŻäæ■d'åőČåźűæšąæIJL'æŔŘä¿ŻäÿĂäžŻénŸçžğçŽĎ
è£ŹäžŻä¡äèĞłåůšåőđçŐřętůæİěäź§äÿ■éŽ¿ãĂĆäÿ■è£ĞåęĆæđIJä¡äæČşèęĄæŻt'åd'ŽçŽĎæŤŕæŇĄïijŇåŔŕäžěè
WebOb æĹŰèĂĚ Paste

13.6 11.6 éĂŽè£ĞXML-RPCåőđçŐřçőÅå■ŢçŽĎè£IJçĺŃèřČçŤĺ

éŮőécŸ

ä¡äæČşæL'¿åĹřäÿĂäÿłçőÅå■ŢçŽĎæŰźåijŔåŐżæL'ğèąŇè£ŘèąŇåIJÍè£IJçĺŃæIJžåŹĺäÿŁéİćçŽĎPythonçĺ

èğčåĘşæŰźæąĹ

åőđçŐřäÿĂäÿłè£IJçĺŃæŰźæşŢèřČçŤĺçŽĎæIJĂçőÅå■ŢæŰźåijŔæŸŕä¡£çŤĺXML-
RPCãĂĆäÿŃéİćæĹŚäžňæijŤçd'žäÿĂäÿŃäÿĂäÿłåőđçŐřäžĘéŤő-
åĂijå■ŸåĆĺåŁ§èČ¡çŽĎçőÅå■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)

```
        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()
```

äÿŃéÍćæĹŚäżňäżŐäÿĂäÿłåőćæĹůçńŕæIJžåŹĺäÿŁéÍćæÍěèő£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
>>>
```

èőÍèőž

XML-RPC åŘŕäžěèőľ æĹŚäżňå¿Ĺåőźæ ŸŞçŽĎæđĎéĂääÿĂäÿłçőĂå■ŢçŽĎè£IJçÍŃèřČçŤÍæIJ■åŁąãĂĆä¡éĂŽè£ĞåőČçŽĎæŰźæşŢ `register_function()` æÍěæşĺåĘŇåĞ¡æŢřïijŇçĎűåŘŐä¡£çŤÍæŰźæşŢ `serve_forever()` åŘŕåŁĺåőČãĂĆ åIJÍäÿŁéÍćæĹŚäżňářĘè£ŹäžŻæ■ěéłď æŤ¿åIJÍäÿĂèţůåĘŹåĽřäÿĂäÿłçśż

```
from xmlrpc.server import SimpleXMLRPCServer
def add(x,y):
    return x+y

serv = SimpleXMLRPCServer(('', 15000))
```

(continues on next page)

```
serv.register_function(add)
serv.serve_forever()
```

XML-RPCæŽťéIJšåĞžæİěçŽĎåĞ¡æŢřåŔłèČ¡éĂĆçŢĺäžŐéČĺåĹĘæŢřæ■őçśżåđŃïijŇærŤåęĆå■Ůçňęäÿš
åŕźäžŐåĚűäżŰçśżåđŃåřšå¿ŮéIJĂèęĄåĄŽäžŻéćİåď˙ŰçŽĎåŁ§èŕ¿äžĘãĂĆ
ä¿ŃåęĆïijŇåęĆæđIJä¡ăæČşéĂŽè£Ğ XML-RPC äijăéĂŠäÿĂäÿłåŕźèšąåőđä¿ŃïijŇåőđéŹĚäÿŁåŔłæIJL’äżŰçŽĎ

```
>>> 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¡ăæČşèšąçŽĎäÿ■åď˙łäÿĂæăůïijŽ

```
>>> s.set('foo', b'Hello World')
>>> s.get('foo')
<xmlrpc.client.Binary object at 0x10131d410>

>>> _.data
b'Hello World'
>>>
```

äÿĂèĹňæİěèőšïijŇä¡ăäÿ■åžŤèŕěåřĘ XML-RPC æIJ■åŁąäžěåĚňåĚśAPIçŽĎæŰźåijŔæŽť˙éIJšåĞžæİěãĂĆ
åŕźäžŐè£ŹçğæČĚåĘţïijŇéĂŽåÿÿåĹĘåÿČåijŔåžŢçŤĺçÍŃåžŔäijŽæŸŕäÿĂäÿłæŻť˙åě¡çŽĎéĂĽ˙æŃľãĂĆ

XML-RPCçŽĎäÿĂäÿłçijžçĆźæŸŕåőČçŽĎæĂğèČ¡ãĂĆ`SimpleXMLRPCServer`
çŽĎåőđçŐřæŸŕå■Ţçž£çÍŃçŽĎïijŇ æĽ˙Ăäžěåőčäÿ■éĂĆåŘĹäžŐåď˙ğåđŃçÍŃåžŔïijŇår¡çőąæĹŚäžňåIJÍ11.2åř
åŔęåď˙ŰïijŇçŢśäžŐ XML-RPC åřĘæĽ˙ĂæIJL’æŢřæ■őéČ¡åžŔåĹŮåŇŰäÿžXMLæăijåijŔïijŇæĽ˙ĂäžěåőčäijŽ
ä¡ĘæŸŕåőČäź§æIJL’äijŸçĆźïijŇè£ŹçğæŰźåijŔçŽĎçijŰçăĄåŘŕäžěèćńçżÍåď˙ğéČĺåĹĘåĚűäżŰçijŰçÍŃèŕ■èÍĂ
éĂŽè£Ğä¡£çŢÍè£ŹçğæŰźåijŔïijŇåĚűäžŰèŕ■èÍĂçŽĎåőćæĹůçńŕçÍŃåžŔéČ¡èČ¡èő£éŮőä¡ăçŽĎæIJ■åŁąãĂĆ

èŹ¡çĎűXML-RPCæIJL’å¿Ĺåď˙ŽçijžçĆźïijŇä¡ĘæŸŕåęĆæđIJä¡ăéIJĂèęĄå£ńéĂ§æđĎåżžäÿĂäÿłçőĂå■Ţè£
æIJL’æŮűåĂŹïijŇçőĂå■ŢçŽĎæŰźæąĹåŕšåůšçżŔèűşåď˙§äžĘãĂĆ

13.7 11.7 åIJĺäÿ■åŘŇçŽĎPythonèğčéĞŁåŹĺäźŃéŮť˙äžď˙äžŠ

éŮőécŸ

ä¡ăåIJĺäÿ■åŘŇçŽĎæIJžåŹĺäÿŁéİcè£ŘèąŇçİĂåď˙Žäÿ łPythonèğčéĞŁåŹĺåőđä¿ŃïijŇåźűåÿŇæIJŻèČ¡åď˙§å

èğčåEşæŰźæąĹ

éĂŽè£Ğä¡£çŤÍ`multiprocessing.connection`æÍąåİŮåŔŕäžěå¿Ĺåőźæ ŸŞçŽĎåőđçŐřèğčéĞŁåŹÍ
äÿNéÍćæŸŕäÿĂäÿłçőĂå■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')
```

çĎŭåŘŐåőćæĹůçńŕè£đæŐěæIJ■åŁąåŹÍåźűåŔŠéĂĄæŭĹæĄŕçŽĎçőĂå■Ţçd'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■åŘŇçŽĎæŸŕïijŇæŕŔäÿłæŭĹæĄŕäijŽåőŇæŢt'ä£Íå■ŸïijĹæŕŔäÿĂäÿłéĂŽè£Ğsend()
åŘęåd'ŰïijŇæL'ĂæIJL'åŕźèśąäijŽéĂŽè£ĞpickleåžŘåĹŮåŇŰãĂĆåŻăæ■d'ïijŇäżżä¡ŢåĚijåőźpickleçŽĎåŕźèśą

èőÍèőž

çŻőåL'■æIJL'å¿Ĺåd'ŽçŤÍæÍěåőđçŐřåŘĎçğ■æŭĹæĄŕäijăè¿ŞçŽĎåŇĚåŠŇåĞ¡æŢřåžŞïijŇæŕŤåęĆZeroMQ
ä¡ăè£ŸæIJL'åŘęåd'ŰäÿĂçğ■éĂL'æŃl'åŕśæŸŕèĞłåůśåIJĺåžŢåśĆsocketå§žçąĂäźŃäÿŁæÍěåőđçŐřäÿĂäÿłæŭĹæ

ä¡ĘæŸŕä¡äæČşèęĄçőĂå■ŢäÿĂçĆźçŽĎæŰźæąĹïijŇéĆčäźĹè£ŹæŮűåĂŹ `multiprocessing.connection` åŕšæť¿äÿŁçŤĺåIJžäžĘãĂĆ äžĚäžĚä¡£çŤĺäÿĂäžŻçőĂå■ŢçŽĎèŕ■åŘěå■şåŔŕåőđçŐřåď Žäÿłèğčéĝåŕĺäžŕé Ůť çŽĎæűĹæĄŕéĂŽä£ąãĂĆ

åęĆæđIJä¡ăçŽĎèğčéĜŁåŹĺè£ŘèąŇåIJĺåŘŇäÿĂåŔřæIJžåŹĺäÿŁéÍćïijŇéĆčäźĹä¡ääŔŕäžĕä¡£çŤĺåŔęåď Űç
èęĄæČşä¡£çŤĺUNIXå§§åèŮæŐĕå■ŮæİĕåĹŻåžžäÿĂäÿłè£đæŐĕïijŇåŔłéIJĂçőĂå■ŢçŽĎåřĘåIJřåİĂæŤźåĘŹä

```
s = Listener('/tmp/myconn', authkey=b'peekaboo')
```

èęĄæČşä¡£çŤĺWindowsåŠ¡åŘ■çőąéĄŞæİĕåĹŻåžžè£đæŐĕïijŇåŔłéIJĂåČŔäÿŇéİćè£Źæăůä¡£çŤĺäÿĂäÿł

```
s = Listener(r'\\.\pipe\myconn', authkey=b'peekaboo')
```

äÿĂäÿłéĂŽçŤĺåĞĘåĹŹæŸŕïijŇä¡ääÿ■èęĄä¡£çŤĺ `multiprocessing` æİĕåőđçŐřäÿĂäÿłåŕźåď ŰçŽĎåĚňåĚśæIJ■åŁąãĂĆ `Client()` åŠŇ `Listener()` äÿ■çŽĎ `authkey` åŔĆæŢřçŤĺæİĕèőď èŕĄåŔŠèţůè£đæŐěçŽĎçżĹçńŕçŤĺæĹůãĂĆ
åęĆæđIJåŕĘéŠěäÿ■åŕźäijŽäžğçŤ§äÿĂäÿłåijĆåÿÿãĂĆæ■ď åď ŰïijŇèŕěæĺąåİÜæIJĂéĂĆåŔĹçŤĺæİĕåžžçńŃéŢ£
ä¿ŃåęĆïijŇäÿď äÿłèğčéĜŁåŹĺäźŃé Ůť åŔŕåŁĺåŔŐåŕśäijĂåğŃåžžçńŃè£đæŐĕåźűåIJĺåď ĎçŘĘæ§ŘäÿłéŮőéćŸ

åęĆæđIJä¡ăéIJĂèęĄåŕźåžŢåśĆè£đæŐĕåĄŽæŻť åď ŽçŽĎæŐğåĹűïijŇæŕŤåęĆéIJĂèęĄæŤŕæŇĄèűĚæŮű
ä¡æIJĂåĕ¡ä¡£çŤĺåŔęåď ŰçŽĎåžŞæĹŰèĂĚæŸŕåIJĺéńŸåśĆsocketäÿŁæİĕåőđçŐřè£ŹäžŻçĽ´źæĂğãĂĆ

13.8 11.8 åőđçŐřè£IJçĺŃæŰźæşŢèřČçŤĺ

éŮőéćŸ

ä¡ăæČşåIJĺäÿĂäÿłæűĹæĄŕäijăè¿ŞåśĆåęĆ `sockets` ãĂĄ`multiprocessing connections` æĹŰ `ZeroMQ` çŽĎå§žçąĂäźŃäÿŁåőđçŐřäÿĂäÿłçőĂå■ŢçŽĎè£IJçĺŃèřČçŤĺïijĹRPC

èğčåĘşæŰźæąĹ

åřĘåĞ¡æŢřèŕůæśĆãĂĄåŔĆæŢřåŠŇè£ŤåŻđåĂijä¡£çŤĺpickleçijŰçăĄåŘŐïijŇåIJĺäÿ■åŘŇçŽĎèğčéĜŁåŹ
äÿŃéİćæŸŕäÿĂäÿłçőĂå■ŢçŽĎPRCåď ĎçŘĘåŹĺïijŇåŔŕäžĕèćńæŢť 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)

```
                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¡£çŤĺè£Źäÿłåđ'ĎçŘĘåŹĺïijŇä¡ăéIJĂèęĄåřĘåőČåŁăåĚěåĹřäÿĂäÿłæűĹæĄŕæIJ■åŁąåŹĺäÿ■ãĂĆä¡ăæä¡ĘæŸŕä¡£çŤĺ multiprocessing åžŞæŸŕæIJĂçőĂå■ŢçŽĎãĂĆäÿŃéİćæŸŕäÿĂäÿłR-PCæIJ■åŁąåŹĺä¿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■åŁąåŹĺïijŇä¡ăéIJĂèęĄåĹŻåžžäÿĂäÿłåŕźåžŢçŽĎçŢĺæİ

```
import pickle

class RPCProxy:
    def __init__(self, connection):
        self._connection = connection
    def __getattr__(self, name):
```

(continues on next 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¡£çŤĺè£ŹäÿłäžčçŘĘçśżïijŇä¡ăéIJĂèęĄåřĘåĚűåŇĚèčĚåĹřäÿĂäÿłæIJ■åŁąåŹĺçŽĎè£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'
>>>
```

èęĄæşĺæĎŘçŽĎæŸŕå¿Ĺåď'ŽæűĹæĄŕåśĆïijĹæŕŤåęĆ `multiprocessing` ïijĽåůšçżŔä¡£çŤĺpickleåžŔåĹŮåŇŰäžĘæŢřæ■őãĂĆ åęĆæđIJæŸŕè£ŹæăůçŽĎèŕİïijŇåŕź `pickle.dumps()` åŠŇ `pickle.loads()` çŽĎèřĆçŤĺèęĄåŐżæŐĽ'ãĂĆ

èőĺèőž

`RPCHandler` åŠŇ `RPCProxy` çŽĎå§žæIJňæĂİèůŕæŸŕå¿ĹæŕŤè¿ČçőĂå■ŢçŽĎãĂĆ åęĆæđIJäÿĂäÿłåőćæĹůçńŕæČşèęĄèřČçŤĺäÿĂäÿłè£IJçĺŃåĞ¡æŢřïijŇæŕŤåęĆ `foo(1, 2, z=3)` ,äžčçŘĘçśżåĹŻåžžäÿĂäÿłåŇĚåŘńäžĘåĞ¡æŢřåŘ■åŠŇåŔĆæŢřçŽĎåĚČçżĎ `('foo', (1, 2), {'z': 3})` ãĂĆ è£ŹäÿłåĚČçżĎèćńpickleåžŔåĹŮåŇŰåŘŐéĂŽè£ĞçĄŚçżIJè£đæŐěåŔŚçŤ§åČ è£ŹäÿĂæ■ěåIJĺ `RPCProxy` çŽĎ `__getattr__()` æŰźæşŢè£ŤåŻđçŽĎ `do_rpc()` éŮ■åŇĚäÿ■åőŇæĹŘãĂĆ æIJ■åŁąåŹĺæŐěæŤűåŘŐéĂŽè£ĞpickleåŘ■åžŔåĹŮåŇŰæűĹæĄŕïijŇæ§ěæL'¿åĞ¡æ æL'ğèąŇçżŞædIJ(æĹŰåijĆåÿÿ)èćńpickleåžŔåĹŮåŇŰåŘŐè£ŤåŻđåŔŚéĂĄçżŹåőćæĹůçńŕãĂĆæĹŚäžňçŽĎåőŕ `multiprocessing` è£ŻèąŇéĂŽä£ąãĂĆ äÿ■è£ĞïijŇè£ŹçğæŰźåijŔåŘŕäžěéĂĆçŤĺäžŐåĚűäžŰäżżä¡Ţæű äžĚäžĚåŘłéIJĂèęĄåřĘè£đæŐěåŕźèśąæ■ćæĹŘåŘĹéĂĆçŽĎZeroMQçŽĎsocketåŕźèśąå■şåŘŕãĂĆ

çŤśäžŐåžŢåśĆéIJĂèęĄä¿ÌètŰpickleïijŇéĆčäźĹåőL'åĚĺéŮőécŸåŕśéIJĂèęĄèĂČèŹŚäžĘ ïijĹåŽääÿžäÿĂäÿłèĄłæŸŐçŽĎéżŚåőćåŘŕäžěåĹŻåžžçL'źåőŽçŽĎæűĹæĄŕïijŇèČ¡åď §èől'äżżæĎŘåĞ¡æŢřéĂŽ åŽăæ■ď'ä¡ăæřÿè£IJäÿ■èęĄåĚĄèőÿæÌěèĞłäÿ■ä£ąäžżæĹŰæIJłèőd'èŕĄçŽĎåőćæĹůçńŕçŽĎRPCãĂĆçL'źåĹńæ è£Źçğ■åŘłèČ¡åIJĺåĘĚéČĺèćńä¡£çŤĺïijŇä¡■äžŐéŸšçĄńáćŹåŘŐéİćåźűäÿŤäÿ■èęĄåŕźåď ŰæŽť éIJšãĂĆ

ä¡IJäÿžpickleçŽĎæŽ£äžčïijŇä¡ăäź§èőÿåŘŕäžěèĂČèŹŚä¡£çŤĺJSONãĂĄXMLæĹŰäÿĂäžŽåĚűäžŰçŽĎçi

ä¿ÑåęĆïijŇæIJňæIJžåőđä¿ÑåŔŕäžěå¿Ĺåőźæ˙ŸŞçŽĎæŤźåĘŹæĹŘJSONçijŰçăĄæŰźæąĹãĂĆè£˙ŸéIJĂèęĄåřĘ
`pickle.loads()` åŠŇ `pickle.dumps()` æŻ£æ■ćæĹŘ `json.loads()` åŠŇ `json.dumps()` å■şåŔŕïijŽ

```
# 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’■æİĆçŽĎéŮőéćŸæŸŕåęĆä¡ŢåŐżåď’ĎçŘĘåijĆåÿÿãĂĆèĞşåřŚïijŇå¡Şå
åŻăæ■d’ïijŇè£ŤåŻđçżŹåőćæĹůçńŕçŽĎåijĆåÿÿæL’Ăäžčèąĺ˛çŽĎåŘńäźL’åřśèęĄåě¡åě¡èő¿èőąäžĘãĂĆ
åęĆæđIJä¡ăä¡£çŤĺpickleïijŇåijĆåÿÿåŕźèśąåőđä¿ÑåIJĺåőćæĹůçńŕèČ¡è¢ńåŔ■äžŘåĹŮåŇŰåźűæŁŻåĞžãĂĆåęĆ
äÿ■è£ĞèĞşåřŚïijŇä¡ăäžŤèŕěåIJĺåŞ■äžŤäÿ■è£ŤåŻđåijĆåÿÿå■Ůçňęäÿšă ĂĆæĹŚäžňåIJĺJSONçŽĎä¿Ńå■Řäÿ■å

åŕźäžŐåĚűäżŰçŽĎRPCåőđçŐřä¿Ńå■ŘïijŇæĹŚæŐĺè■Řä¡ăçIJŃçIJŃåIJĺXML-RPCäÿ■ä¡£çŤĺçŽĎ `SimpleXMLRPCServer` åŠŇ `ServerProxy` çŽĎåőđçŐřïijŇäźş§åřśæŸŕ11.6åřŔèŁĆäÿ■çŽĎåĘĚåőźãĂĆ

13.9 11.9 çőÄå■ŢçŽĎåőćæĹůçńŕèőď'èŕĄ

éŮőécŸ

ä¡ăæČşåIJĺåĹĘåÿČåijŔçşżçż§äÿ■åőđçŐřäÿĂäÿłçőÄå■ŢçŽĎåőćæĹůçńŕè£đæŐěèőď'èŕĄåŁ§èČ¡ïijŇåŔĹä

èğčåĘşæŰźæąĹ

åŔŕäžěåĹl'çŤĺ hmac æĺąåİŮåőđçŐřäÿĂäÿłè£đæŐěæŔąæL'Ńïij Ňäž ŐèĂŇåőđçŐřäÿĂäÿłçőÄå■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£đæŐěåžžçńŇåŘŐïijŇæIJ■åŁąåŹĺçżŹåőćæĹůçńŕåŔŠéĂĄäÿĂäÿłéŽŔæIJžçŽĎ
os.urandom() è£ŤåŻđåĂijïijL'ãĂĆ åőćæĹůçńŕåŠŇæIJ■åŁąåŹĺåŘŇæŮűåĹl'çŤĺh-
macåŠŇäÿĂäÿłåŔłæIJL'åŔŇæŰźç§ěéĄŞçŽĎåŕĘéŠěæİèèőąçőŮåĞžäÿĂäÿłåŁăåŕĘåŞĹåÿŇåĂijãĂĆçĎűåŘŐå
æIJ■åŁąåŹĺéĂŽè£ĞæŕŤè¿Čè£ŹäÿłåĂijåŠŇèĞłåůśèőąçőŮçŽĎæŸŕåŘęäÿĂèĞt'æİěåĘşåőŽæŐěåŔŮæĹŰæŃŠ
hmac.compare_digest() åĞ¡æŢřãĂĆ ä¡£çŤĺè£ŹäÿłåĞ¡æŢřåŔŕäžěéĄ£åĚ■éĄ■åĹřæŮűéŮt'åĹĘæđŘæŤ
äÿžäžĘä¡£çŤĺè£ŹäžŽåĞ¡æŢřïijŇä¡ăéIJĂèęĄářĘåőČéŽĘæĹŘåĹřåůšæIJL'çŽĎç¡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)

```
        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¡£çŤĺåIJžæŹřæŸřåĘĚéČĺæűĹæĄřéĂŽä£ąçşżçż§åŠŇè£ŽçÍŇéŮť'éĂŽ
ä¿ŇåęĆïijŇåęĆæđIJä¡ăçijŰåĘŹçŽĎçşżçż§æűĽ'åŔŁåĹřäÿĂäÿłéŻĘç¿ď'äÿ■åď'Žäÿłåď'ĎçŘĘåŹĺäźŇéŮť'çŽĎéÀ
ä¡ääŔřäžěä¡£çŤĺæIJňèŁĆæŰźæąĹæÍěçąőä£ÍåŔłæIJL'èćńåĚĄèőÿçŽĎè£ŽçÍŇäźŇéŮť'æL'■èČ¡å¡ijæ■ď'éĂŽä£ą
äžŇåőđäÿŁïijŇå§žäžŐ hmac çŽĎèőď'èřĄèćń multiprocessing
æĺąåİŮä¡£çŤĺæÍěåőđçŐřå■Řè£ŽçÍŇçŻť'æŐěçŽĎéĂŽä£ąãĂĆ

è£ŸæIJL'äÿĂçĆźéIJĂèęĄåijžèřČçŽĎæŸŕè£đæŐěèőď'èřĄåŠŇåŁäărĘæŸřäÿď'çăĄäžŇãĂĆ
èőď'èřĄæĹŘåŁ§äźŇåŘŐçŽĎéĂŽä£ąæűĹæĄřæŸřäžěæŸŐæŰĞå¡ćåijŔåŘŠéĂĄçŽĎïijŇäżżä¡Ţäžžåŕłèę Ąæč

hmacèőď'èřĄçőŮæşŢå§žäžŐåŞĹåÿŇåĞ¡æŢřåęĆMD5åŠŇSHA-
1ïijŇåĚşäžŐè£ŹäÿłåIJÍIETF RFC 2104äÿ■æIJL'èŕęçżĘäżŇçż■ãĂĆ

13.10 11.10 åIJÍç¡ŚçżIJæIJ■åŁąäÿ■åŁååĚěSSL

éŮőécŸ

ä¡ăæČşåőđçŐřäÿĂäÿłå§žäžŐsocketsçŽĎç¡ŚçżIJæIJ■åŁąïijŇåőćæĹůçńřåŠŇæIJ■åŁąåŹĺéĂŽè£ĞSSLå■Ř

èğčåĘşæŰźæąĹ

ssl æĺąåİŮèČ¡äÿžåžŢåśĆsocketè£đæŐěæůżåŁăSSLçŽĎæŤŕæŇĄãĂĆ ssl.wrap_socket() åĞ¡æŢřæŐěåŘŮäÿĂäÿłåůšå■ŸåIJÍçŽĎsocketä¡IJäÿžåŘĆæŢřåźűä¡£çŤÍSSLåśĆæİěåŇĚèč ä¿ŇåęĆïijŇäÿŇéİćæŸŕäÿĂäÿłçőĂå■ŢçŽĎåžŤç■ŤæIJ■åŁąåŹĺïijŇèČ¡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))
```

äÿŃéÌćæĹŚäžňæijŤçď žäÿĂäÿłåőćæĹůçńřè£đæŐěæIJ■åŁąåŹĺçŽĎäžď äžŠä¿Ńå■ŘãĂĆåőćæĹůçńřäijŽèřü

```
>>> 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'ĎçŘĘåžŢåśĆsocketæŰźåijŔæIJL'äÿłéŮőécŸåřśæŸřåőČäÿ■èČ¡å¿Ĺåě¡çŽĎèů§æăĞåČ
ä¿ŃåęĆïijŇçżĺåď'ğéČĺåĹĘæIJ■åŁąåŹĺäžčçăĄïijĹHTTPãĂĄXML-
RPCç■L'ïijL'åőđéŹĚäÿŁæŸřå§žäžŐ socketserver åžŞçŽĎãĂĆ
åőćæĹůçńřäžčçăĄåIJĺäÿĂäÿłè¿ČéńŸåśĆäÿŁåőđçŐřãĂĆæĹŚäžňéIJĂèęĄåŔęåď'ŰäÿĂçğ■çĺ■å¿őäÿ■åŘŇçŽĎ

éęŰåĚĹïijŇårźäžŐæIJ■åŁąåŹĺèĂŇèĺĂïijŇåŔřäžěéĂŽè£ĞåČŔäÿŃéİćè£Źæăůä¡£çŤĺäÿĂäÿłmixinçśžæİě

```
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£ŹäÿłmixinçśžïijŇä¡ääŔřäžěářĘåőČèů§åĚűäžŰæIJ■åŁąåŹĺçśžæůůåŘĹãĂĆä¿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)

```
    kvserv.serve_forever()
```

ä¡£çŤÍè£ŹäÿłæIJ■åŁąåŹÍæŮűïijŇä¡ăåŔŕäžěä¡£çŤÍæŹőéĂŽçŽĎ `xmlrpc.client` æÍąåİŮæİěè£đæŐěåőČãĂĆ åŔłéIJĂèęĄåIJÍURLäÿ■æŇĞåőŽ `https:` å■şåŔŕïijŇä¿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
>>>
```

åŕźäžŐSSLåőćæĹůçńŕæİěèőšäÿĂäÿłæŕŤè¿Čåd'■æİĆçŽĎéŮőécŸæŸŕåęĆä¡Ţçąőèőd'æIJ■åŁąåŹÍèŕĄäźęäÿ■åźÿçŽĎæŸŕïijŇæŽĆæŮűè£ŸæšąæIJL'äÿĂäÿłæăĞåĞĘæŰźæşŢæİěèğčåĘşè£ŹäÿłéŮőécŸïijŇéIJĂèęĄèĞłäÿ■è£ĞïijŇäÿŃéİćçżŹåĞžäÿĂäÿłä¿Ńå■ŘïijŇçŤÍæİěåžžçńŃäÿĂäÿłåőL'åĚÍçŽĎXML-RPCè£đæŐěæİěçąőèőd'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)

```
s = ServerProxy('https://localhost:15000',
                transport=VerifyCertSafeTransport('server_cert.pem
↪'),
                allow_none=True)
```

æIJ■åŁąåŹĺåřĘèŕĄäźęåŔŠéĂĄçżŹåőćæĹůçńŕïijŇåőćæĹůçńŕæİěçąőèőď'åőČçŽĎåŘĹæşŢæĂğãĂĆè£ŹçğåęĆæđIJæIJ■åŁąåŹĺæČşèęĄçąőèőď'åőćæĹůçńŕïijŇåŔŕäžěåřĘæIJ■åŁąåŹĺåŘŕåŁĺäžčçăĄä£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()
```

äÿžäžĘèől'XML-RPCåőćæĹůçńŕåŔŠéĂĄèŕĄäźęïijŇä£őæŤź `ServerProxy` çŽĎåĹİåğŃåŇŰäžčçăĄåęĆäÿŃïijŽ

```
# 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 åŔŕèČ¡æIJĂåđ'ğçŽĎæŇŠæĹŸæŸŕåęĆä¡ŢäÿĂæ■ěæ■ěçŽĎèŐůåŔŰůåĹİåğŃéĚ■ç¡őkeyãĂĄèŕĄäźęåŠŇåĚűäžŰ

æĹŚèğčéĞŁäÿŃåĹřåžŢéIJĂèęĄåŢěïijŇæŕŔäÿĂäÿłSSLè£đæŐěçżĹçńŕäÿĂèĹňéČ¡äijŽæIJL'äÿĂäÿłçğĄé è£ŹäÿłèŕĄäźęåŇĚåŔńäžĘåĚňéŠěåźűåIJĺæŕŔäÿĂæňąè£đæŐěçŽĎæŮűåĂŹéČ¡äijŽåŘŚéĂĄçżŹårźæŰźãĂĆ åŕźäžŐåĚňåĚśæIJ■åŁąåŹĺïijŇåőČäžňçŽĎèŕĄäźęéĂŽåÿÿæŸŕèćńæİČåĺĄèŕĄäźęæIJžædĎæŕŤåęĆVerisignãĂA äÿžäžĘçąőèőď'æIJ■åŁąåŹĺç■¿åŘ■ïijŇåőćæĹůçńŕåŻđä£Ìå■ŸäÿĂäż¡åŇĚåŔńäžĘä£ąäżżæŐĹæİČæIJžædĎçŽĎä¿ŃåęĆïijŇwebæţŔèğĹåŹĺä£Ìå■ŸäžĘäÿżèęĄçŽĎèőď'èŕĄæIJžædĎçŽĎèŕĄäźęïijŇåźűä¡£çŤĺåőČæİěäÿżæŕŔäårźæIJňåřŔèŁĆçď'žä¿ŃèĂŇèĺĂïijŇåŔłæŸŕäÿžäžĘæţŃèŕŢïijŇæĹŠäżňåŔŕäžěåĹŻåżżèĞłç■¿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 %

åIJĺåĹŻåżžèŕĄäźęçŽĎæŮűåĂŹïijŇåŘĎäÿłåĂijçŽĎèő¿åőŽåŔŕäżěæŸŕäżżæĎŔçŽĎïijŇä¡ĘæŸŕâĂİComm
NameâĂIJçŽĎåĂijéĂŽåÿÿèęĄåŇĚåŘńæIJ■åŁąåŹĺçŽĎDNSäÿżæIJžåŘ■ãĂĆ
åęĆæđIJä¡ääŔłæŸŕåIJĺæIJňæIJžæţŃèŕŢïijŇéĆčäźĹåřśä¡£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■åŁąåŹÍçńŕäžčçăĄäÿ■ïijŇçğĄéŠěåŠŇèŕĄäźęæŰĞäżűäijŽèćńäijăçżŹSSLçŻÿåĚşçŽĎåŇĚèčĚåĞ¡
çğĄéŠěåžŤèŕěåIJÍä£Íå■ŸåIJÍæIJ■åŁąåŹÍäÿ■ïijŇåźűåŁääžěåőL’åĚÍä£ÍæŁd’ãĂĆ

åIJÍåőćæĹůçńŕäžčçăĄäÿ■ïijŇéIJĂèęĄä£Íå■ŸäÿĂäÿłåŘĹæşŢèŕĄäźęæŐĹæİČæŰĞäżűæİěçąőèőd’æIJ■åŁ
åęĆæđIJä¡ăæšąæIJL’è£Źäÿłæ̊ŰĞäżűïijŇä¡ăåŔŕäżěåIJÍåőćæĹůçńŕåd’■åĹűäÿĂäż¡æIJ■åŁąåŹÍçŽĎèŕĄäźęåźűä¡
è£đæŐěåžžçńŇåŘŐïijŇæIJ■åŁąåŹÍäijŽæŔŘä¿ŻåőČçŽĎèŕĄäźęïijŇçĎűåŘŐä¡ăåřśèČ¡ä¡£çŤÍåůšçżŔä£Íå■ŸçŽ

æIJ■åŁąåŹÍäź§èČ¡éĂL’æŃl’æŸŕåĘęęĄçąőèőd’åőćæĹůçńŕçŽĎèžńäż¡ãĂĆåęĆæđIJèęĄè£Źæăůå
æIJ■åŁąåŹÍäź§éIJĂèęĄä£Íå■ŸäÿĂäÿłèćńä£ąäżżèŕĄäźęæŐĹæİČæŰĞäżűæİěçąőèőd’åőćæĹůçńŕèŕĄäźęãĂĆ

åęĆæđIJä¡ăèęĄåIJÍçIJ§åőđçŐŕåćČäÿ■äÿžä¡ăçŽĎç¡ŚçżIJæIJ■åŁąåŁääÿŁSSLçŽĎæŤŕæŇĄïijŇè£ŹåřŔèŁ
ä¡ăè£ŸåžŤèŕěåŔĆèĂČåĚűäżŰçŽĎæŰĞæąčïijŇåĄŽåě¡èŁśèť źäÿ■åřŚæŮűéÚť’æİèæţŃèŕŢåőČæ■čåÿÿåůěä¡IJ
^_^

13.11 11.11 è£ŽçÍŃéŮť’äijăéĂŠSocketæŰĞäżűæŔŘè£řçňę

éŮőécŸ

ä¡ăæIJL’åd’ŽäÿłpythonèğčéĞŁåŹÍè£ŻçÍŃåIJÍåŘŇæŮűè£ŘèąŇïijŇä¡ăæČşåřĘæ§Řäÿłæ̊L’ŞåijĂçŽĎæŰĞä
ærŤåęĆïijŇåĄĞèő¿æIJL’äÿłæIJ■åŁąåŹÍè£ŻçÍŃçŻÿåžŤè£đæŐěèřůæśĆïijŇä¡ĘæŸŕåőđéŹĚçŽĎçŽÿåžŤéĂżè¿Ş

èğčåĘşæŰźæąĹ

äÿžäžĘåIJÍåd’Žäÿłè£ŻçÍŃäÿ■äijăéĂŠæŰĞäżűæŔŘè£řçňęïijŇä¡ăéęŰåĚĹéIJĂèęĄåřĘåőČäžňè£đæŐěåĹřă
èĂŇåIJÍwindowsäÿŁéÍćä¡ăéIJĂèęĄä¡£çŤĺåŞ¡åŘ■çőąéĄŞãĂĆäÿ■è£Ğä¡ăæŮäéIJĂçIJ§çŽĎéIJĂèęĄåŐżæŞ■ä¡
éĂŽåÿÿä¡£çŤÍ `multiprocessing` æĺąåİŮæİěåĹŻåżžè£ŹæăůçŽĎè£đæŐěäijŽæŻt’åőźæŸŞäÿĂäžŽãĂĆ

äÿĂæŮęäÿĂäÿłè£đæŐěèćńåĹŻåżžïijŇä¡ăåŔŕäżěä¡£çŤÍ `multiprocessing.reduction` äÿ■çŽĎ `send_handle()` åŠŇ `recv_handle()` åĞ¡æŢřåIJÍäÿ■åŘŇçŽĎåd’ĎçŘĘåŹÍçŻt’æŐěäijăéĂŠæŰĞäżűæŔŘè£řçňęãĂĆ
äÿŃéÍćçŽĎä¿Ńå■ŘæijŤçd’žäžĘæIJĂå§žæIJńçŽĎçŤÍæşŢïijŽ

```
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)

```
        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■ïijŇäÿd'äÿŁè£ŻçÍŃèćńåĹŻåżžåźűéĂŽè£ĞäÿĂäÿŁ `multiprocessing` çőąéĄŞè£đæŐěęţůæİěãĂĆ æIJ■åŁąåŹÍè£ŻçÍŃæL'ŞåijĂäÿĂäÿŁsocketåźűç■L'å¿Ěåőćæ åůěä¡IJè£ŻçÍŃäżĚäżĚä¡£çŤÍ `recv_handle()` åIJÍçőąéĄŞäÿŁéİćç■L'å¿ĚæŐěæŤűäÿĂäÿŁæŰĞäżűæŔŘè£řç å¡ŞæIJ■åŁąåŹÍæŐěæŤűåĹřäÿĂäÿŁè£đæŐëïijŇåőČářĘäžğçŤ§çŽĎsocketæŰĞäżűæŔŘè£řçňęéĂŽè£Ğ `send_handle()` äijåéĂŠçżŹåůěä¡IJè£ŻçÍŃãĂĆ åůěä¡IJè£ŻçÍŃæŐěæŤűåĹřsocketåŘŐåŘŠåőćæĹůçńŕåŻđå

åęĆæđIJä¡ăä¡£çŤÍTelnetæĹŰçśżäijijåůěåĚůè£đæŐěåĹřæIJ■åŁąåŹÍïijŇäÿŃéİćæŸŕäÿĂäÿŁæijŤçd'ża¿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■d'ä¿ŃæIJĂéĞ■èęĄçŽĎéČÍåĹĘæŸŕæIJ■åŁąåŹÍæŐěæŤűåĹŕçŽĎåőćæĹůçńŕsocketåőđéŹĚäÿŁèćńåŘęå æIJ■åŁąåŹÍäżĚäżĚåŘŁæŸŕåřĘåĚűè¡ňæL'Ńáźűå Ěşéů■æ■d'è£đæŐěïijŇçĎűåŘŐç■L'å¿ĚäÿŃäÿĂäÿŁè£đæŐěãĂ

èőÍèőž

åŕźäžŐåď ğéČÍåĹEçÍŃåžŔåŚŸæÌèèőšåIJÍäÿ■åŘŇè£ŻçÍŃäźŃéŮť äijăéĂŠæŰĞäżűæŔŔè£řçňęåě¡åČŔæšą ä¡ĘæŸŕïijŇæIJL'æŮűåĂŹåőČæŸŕædĐåžžäÿĂäÿłåŔŕæL'ľ åśŢçşżçż§çŽĎå¿ĹæIJL'çŤÍçŽĎåůěåĚůãĂĆä¿ŃåęĆ ä¡ååŔŕäžěæIJL'åď'ŽäÿłPythonèğčéĞŁåŹÍåőđä¿ŃïijŇåřĘæŰĞäżűæŔŔè£řçňęäijăéĂŠçżŹåĚűåőČèğčéĞŁåŹÍæ

send_handle() åŠŇ recv_handle() åĞ¡æŢřåŔłèČ¡åď§çŤÍäžŐ multiprocessing è£đæŐěãĂĆ ä¡£çŤÍåőČäžňæÌěäžčæŻ£çőąéĄŞçŽĎä¡£çŤÍïijĹåŔĆèĂČ11.7èŁĆïijL'ïijŇ ä¿ŃåęĆïijŇä¡ääŔŕäžěèőľæIJ■åŁąåŹÍåŠŇåůěä¡IJèĂĚåŘĎèĞłäžěå■ŢçŃňçŽĎçÍŃåžŔæÌěåŘŕåŁÍãĂĆäÿŃéÍćæŸ

```
# 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■åŁąåŹÍïijŇåŔłéIJĂèęĄæĽğèąŇ *python3 servermp.py /tmp/servconn 15000* ïijŇäÿŃéÍćæŸŕçŻÿåžŤçŽĎåůěä¡IJèĂĚäžčçăĄïijŽ

```
# workermp.py

from multiprocessing.connection import Client
from multiprocessing.reduction import recv_handle
import os
```

(continues on next 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£ŘèąŇåůěä¡IJèĂĚïijŇæĽ'ğèąŇæĽ'ğèąŇåŚ¡äžď *python3 workermp.py /tmp/servconn* . æŢĹædIJèů§ä¡£çŤĺPipe()ä¿Ńå■ŘæŸŕåőŇåĚĺäÿĂæăůçŽĎãĂĆ æŰĞäžűæŔŔè£řçňęçŽĎäijăéĂŠäijŽæűĽ'åŔŁåĹřUNIXå§§åěŮæŐěå■ŮçŽĎåĹŻåžžåŠŇåěŮæŐěå■ŮçŽĎ sendmsg() æŰźæşŢãĂĆ äÿ■è£Ğè£ŹçğæŁĂæIJŕåźűäÿ■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)

```
    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]))
```

äýŃéÌćæŸŕä¡£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)

```
        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¡ăæČşåIJĺä¡ăçŽĎçÍŃåžŔäÿ■äijäéĂŠæŰĞäżűæŔŔè£řçňęïijŇåżžèőőä¡ăåŔĆéŸĚåĚűäżŰäÿĂäžŽ
æŕŤåęĆ `Unix Network Programming by W. Richard Stevens (Prentice Hall, 1990)`。åIJÍWindowsäÿŁäijäéĂŠæŰĞäżűæŔŔè£řçňęèů§UnixæŸŕäÿ■äÿĂæăůçŽĎïijŇåżžèőőä¡ăçă
`multiprocessing.reduction` äÿ■çŽĎæžŘäżčçăĄçIJŃçIJŃåĚűåůěä¡IJåŐ§çŘĘãĂĆ

13.12 11.12 çŘĘèğčäžŃäżűéľśåŁÍçŽĎIO

éŮőécŸ

ä¡ăåžŤèŕěåůšçżŔåŘňè£Ğå§žäžŐäžŃäżűéľśåŁÍæĹŰåijĆæ■ěI/OçŽĎåŇĚïijŇä¡ĘæŸŕä¡ăè£Ÿäÿ■èČ¡åőŇåĚ
æĹŰèĂĚæŸŕåęĆæđIJä¡£çŤĺåőČçŽĎèŕÍäijŽåŕźä¡ăçŽĎçÍŃåžŔäžğçŤ§äżĂäźĹå¡śåŞ■ãĂĆ

èğčåĘşæŰźæąĹ

äžŃäżűéľśåŁÍI/OæIJňèťÍäÿŁæİěèőšåřśæŸŕåřĘå§žæIJňI/OæŞ■ä¡IJïijĹæŕŤåęĆèŕżåŠŇåĘŹïijL'è¡ňåŇŰäÿž
ä¿ŃåęĆïijŇå¡ŞæŢřæ■őåIJĺæ§ŘäÿłsocketäÿŁèćńæŐěåŘŮåŘŐïijŇåőČäijŽè¡ňæ■ćæĹŘäÿĂäÿł
`receive` äžŃäżűïijŇçĎűåŘŐèćńä¡ăåőŽäźĽçŽĎåŽđèřČæŰźæşŢæĹŰåĞ¡æŢřæİěåď ĎçŘĘãĂĆ
ä¡IJäÿžäÿĂäÿłåŔŕèČ¡çŽĎèţůåğŇçĆźïijŇäÿĂäÿłäžŃäżűéľśåŁÍçŽĎæąĘæđűåŔŕèČ¡äijŽäžěäÿĂäÿłåődçŐřäžĘä

```
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)

```
        '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£Źäÿłçśżç ŽĎåőđä¿Ńä¡IJäÿžæŔŠäžűèćńæŤ¿åĚèçśżäijijäÿŃéIćè£Źæăůç ŽĎäžŃäżűå¿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¡őèŕćæŰĞäżűæŔŔè£řçňę åIJĺèřČçŤĺ `select()` äźŃåĽ■ïijŇäžŃäżűå¿łçŐŕäijŽèŕćéŮőæĽĂæIJĽçŽĎåď ĎçŘĘåŹĺæİěåĘşåőŽåŞłäÿĂäÿ çĎűåŘŐåőČåřĘçżŞæđIJåĹŮèąĺæŔŘä¿ŻçżŹ `select()` ãĂĆçĎűåŘŐ `select()` è£ŤåŽđåĞĘåď ĞæŐěåŔŮæĹŰåŔŚéĂĄçŽĎåŕźèśąçżĎæĹŘçŽĎåĹŮèąĺãĂĆ çĎűåŘŐçŻÿåžŤçŽĎ `handle_receive()` æĹŰ `handle_send()` æŰźæşŢèćńèğęåŔŚãĂĆ

çijŰåĘŹåžŤçŤĺçĺŃåžŔçŽĎæŮűåĂŹïijŇ`EventHandler` çŽĎåőđä¿ŃäijŽèćńåĹŻåžžãĂĆä¿ŃåęĆïijŇäÿŃéIćæŸŕäÿď äÿłçőĂå■ŢçŽĎå§žäžŐUDPç¡ŚçżIJæIJ■åŁąçŽĎåđ

```
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)

```
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'ŰäÿĂäÿłPythonèğčéĞŁåŹĺè£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■åŁąåŹĺäijŽæŻt'åŁååď'■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)

(çż■äÿŁéąţ)

```
    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■åĹŮèąĺåćđåŁäåŠŇåĹăéŹď˙åőćæĹůçńŕçŽĎæŞ■ä¡IJã
åŕźæŕŔäÿĂäÿłè£đæŐëïijŇäÿĂäÿłæŰřçŽĎåď˙ĎçŘĘåŹÍèćńåĹŻåżžåźűåŁäåĹřåĹŮèąĺäÿ■ãĂĆå¡Şè£đæŐëèćńåĚ
åęĆæđIJä¡äè£ŘèąŇçÍŃåžŔåźűèŕŢçİĂçŤÍTelnetæĹŰçśżäijijåůěåĚůè£đæŐëïijŇåőČäijŽårĘä¡ääŔŠéĂĄçŽĎæű

èőÍèőž

åőđéŹĚäÿŁæL’ĂæIJL’çŽĎäžŃäžűéľ’śåŁÍæąĘæđűåŐ§çŘĘèů§äÿŁéÍćçŽĎä¿Ńå■ŘçŻÿåůőæŮăåĞããĂĆåőđ
ä¡ĘæŸŕåIJÍæIJĂæäÿå£ČçŽĎéČÍåŁĘïijŇéČ¡äijŽæIJL’äÿĂäÿłè¡őèŕćçŽĎå¿łçŐŕæİěæčĂæ§ěæť’żåŁÍsocketïijŇåź

äžŃäžűéľ’śåŁÍI/OçŽĎäÿĂäÿłåŔŕèČ¡åě¡åd’ĎæŸŕåőČèČ¡åd’ĎçŘĘéİđåÿÿåď’ğçŽĎåźűåŔŚè£đæŐěïijŇèĂŇ
äź§åřśæŸŕèŕť’ïijŇselect() èřČçŤÍïijĹæĹŰåĚűäżŰç■L’æŢÍçŽĎïijL’èČ¡çŻŚåŘňåď’ğéĞŔçŽĎsocketåźűåŞ■
åIJÍå¿łçŐŕäÿ■äÿĂæňąåď’ĎçŘĘäÿĂäÿłäžŃäžűïijŇåźűäÿ■éIJĂèęĄåĚűäżŰçŽĎåźűåŔŚæIJžåĹűãĂĆ

äžŃäžűéľ’śåŁÍI/OçŽĎçijžçĆźæŸŕæšąæIJL’çIJ§æ■čçŽĎåŘŇæ■ěæIJžåĹűãĂĆ
åęĆæđIJäżżä¡ŢäžŃäžűåď’ĎçŘĘåŹÍæŰźæşŢéŸżåąđæĹŰæL’ğèąŇäÿĂäÿłèĂŮæŮűèőąçőŮïijŇåőČäijŽéŸżåąđ
èřČçŤÍéĆčäžŽåźűäÿ■æŸŕäžŃäžűéľ’śåŁÍéčŐæäijçŽĎåžŞåĞ¡æŢřäź§äijŽæIJL’éŮőécŸïijŇåŘŇæăůèęĄæŸŕæ§

åŕźäžŐéŸżåąđæĹŰèĂŮæŮűèőąçőŮçŽĎéŮőécŸåŔŕäżěéĂŽè£ĞářĘäžŃäžűåŔŚéĂĄäÿłåĚűäżŰå■ŢçŃňç
äÿ■è£ĞïijŇåIJÍäžŃäžűå¿łçŐŕäÿ■åijŢåĚěåď’Žçž£çÍŃåŠŇåď’Žè£ŽçÍŃæŸŕæŕŤè¿ČæčŸæL’ŃçŽĎïijŇ
äÿŃéİćçŽĎä¿Ńå■ŘæijŤçď’žäžĘåęĆä¡Ţä¡£çŤÍ concurrent.futures
æÍąåİŮæİěåőđçŐŕïijŽ

```
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)

```
    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() æŰźæşŢèćńçŤĺæİěářĘåůëä¡IJæŔŘäžd'çżŹåŻđèřČåĞ¡æŢřæśäïijŇåd'ĎçŘĘåőŇ åőđéŹĚåůëä¡IJèćńæŔŘäžd'çżŹ ThreadPoolExecutor åőđä¿ŃãĂĆ äÿ■è£ĞäÿĂäÿłéŽ¿çĆźæŸŕå■ŔèřČèőąçőŮçżŞæđIJåŠŇäžŃäžűå¿łçŐřïijŇäÿžäžĘèğčåĘşåőČïijŇæĹŚäžňåĹŻåż å¡Şçž£çÍŃæśäåőŇæĹŘåůëä¡IJåŘŐïijŇåőČäijŽæL'ğèąŇçśžäÿ■çŽĎ _complete() æŰźæşŢãĂĆ è£ŹäÿłæŰźæşŢåĘ■æ§ŘäÿłsocketäÿŁåĘŹåĚěå■ŮèŁĆäźŃåL'■äijŽèőšæŇĆèţůçŽĎåŻđèřČåĞ¡æŢ fileno() æŰźæşŢè£ŤåŻđåŔęåđ'ŮçŽĎéĆčäÿłsocketãĂĆ åŻäæ■d'ïijŇè£Źäÿłå■ŮèŁĆèćńåĘŹåĚěæŮűïijŇå çĎűåŘŐ handle_receive() æŰźæşŢèćńæ£Äæť'żåźűäÿžæL'ĂæIJL'äźŃåL'■æŔŘäžd'çŽĎåůëä¡IJæL'ğèąŇ åİęçŹ¡èőšïijŇèŕť'äžĘè£ŹäźĹåd'Žè£đæĹŚèĞłåůśéČ¡æŹŢäžĘãĂĆ äÿŃéİćæŸŕäÿĂäÿłçőĂå■ŢçŽĎæIJ■åŁąåŹĺïijŇæijŤçd'žäžĘåęĆä¡Ţä¡£çŤĺçž£çÍŃæśäæİěåőđçŐřèĂŮæŮűçŽĎè

```
# 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■åŁąåŹĺïijŇçĎűåŘŐèřŢçİĂçŤĺåĚűåőČPythonçÍŃåžŔæİěæţŃèřŢåőČïijŽ

```
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¡ăåžŤèŕěèČ¡åIJĺäÿ■åŘŇçłŮåŔčäÿ■éĞ■åd'■çŽĎæĽğèąŇè£ŹäÿłçĺŃåžŔïijŇåźűäÿŤäÿ■äijŽå¡śåŞ■åĹřåĚ

åůšçżŔéŸĚèŕżåőŇäžĘè£ŹäÿĂåřŔèŁĆïijŇéĆčäźĹä¡ăåžŤèŕěä¡£çŤĺè£ŹéĞŇçŽĎäżčçăĄåŘŮïij§äź§èőÿäÿ■
äÿ■è£ĞïijŇåęĆæđIJä¡ăçŘĘèğčäžĘå§žæIJňåŐ§çŘĘïijŇä¡ăåřśèČ¡çŘĘèğčè£ŹäžŻæąĘæđűæĽĂä¡£çŤĺçŽĎæäÿ
ä¡IJäÿžåŕźåŻđèřČåĞ¡æŢřçijŰçĺŃçŽĎæŻ£äžčïijŇäžŃäżűél'śåŁĺçijŰçăĄæIJL'æŮűåĂŹäijŽä¡£çŤĺåĹŕå■ŔçĺŃï

13.13 11.13 åŔŚéĂĄäÿŐæŐěæŤűåď'ğåđŃæŢřçżĎ

éŮőécŸ

ä¡ăèęĄéĂŽè£Ğç¡ŚçżIJè£đæŐěåŔŚéĂĄåŠŇæŐěåŔŮè£đçż■æŢřæ■őçŽĎåď'ğåđŃæŢřçżĎïijŇåźűår¡éĞŔ

èğčåĘşæŰźæąĹ

äÿŃéİćçŽĎåĞ¡æŢřåĹl'çŤĺ `memoryviews` æİěåŔŚéĂĄåŠŇæŐěåŔŮåď'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:]
```

äÿžäžĘæţŃèŕŢçĺŃåžŔïijŇéęŰåĚĹåĹŻåžžäÿĂäÿłéĂŽè£Ğsocketè£đæŐěçŽĎæIJ■åŁąåŹĺåŠŇåőćæĹůçńŕç

```
>>> from socket import *
>>> s = socket(AF_INET, SOCK_STREAM)
>>> s.bind(('', 25000))
>>> s.listen(1)
>>> c,a = s.accept()
>>>
```

åIJĺåőćæĹůçńŕïijĹåŔęåď'ŰäÿĂäÿłèğčéĞŁåŹĺäÿ■ïijL'ïijŽ

```
>>> from socket import *
>>> c = socket(AF_INET, SOCK_STREAM)
>>> c.connect(('localhost', 25000))
>>>
```

æIJňèŁĆçŽĎçŻőæăĞæŸŕä¡ăèČ¡éĂŽè£Ğè£đæŐěäijăè¿ŞäÿĂäÿłèűĚåď ğæŢřçżĎãĂĆè£ŹçğæČĚåĘţçŽĎ
`array` æĺąåİŮæĹŰ `numpy` æĺąåİŮæİěåĹŻåžžæŢřçżĎïijŽ

```
# 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.])
>>>
```

èőÍèőž

åIJÍæŢřæ■őåŕĘéŻĘåđŃåĹĘåÿČåijŔèőąçőŮåŠŇåźşèąŇèőąçőŮçÍŃåžŔäÿ■ïijŇèĞłåůśåĘŹçÍŃåžŔæİěåőđç
äÿ■è£ĞïijŇèęĄæŸŕä¡ăçąőåőđæČşè£ŹæăůåĄŽïijŇä¡ăåŔŕèČ¡éIJĂèęĄåŕĘä¡ăçŽĎæŢřæ■ôè¡ňæ■ćæĹŘåŐ§åğŃ
ä¡ăåŔŕèČ¡è£ŸéIJĂèęĄåŕĘæŢřæ■ôåĹĞåL'šæĹŘåď'ŽäÿłåİŮïijŇåŽääÿžåď'ğéČĺåĹĘåŠŇç¡ŚçżIJçŻÿåĚşçŽĎåĞ¡

äÿĂçğ■æŰźæşŢæŸŕä¡£çŤĺæ§ŘçğæIJžåĹűåžŔåĹŮåŇŰæŢřæ■ôâĂŤâĂŤåŔŕèČ¡åŕĘåĚűè¡ňæ■ćæĹŘäÿĂ
äÿ■è£ĞïijŇè£ŹæăůæIJĂçżĹäijŽåĹŻåžžæŢřæ■őçŽĎäÿĂäÿłåď'■åĹűãĂĆ
åŕśçőŮä¡ăåŔłæŸŕéŽűçćŐçŽĎåĄŽè£ŹäžŻïijŇä¡ăçŽĎäžčçăĄæIJĂçżĹè£ŸæŸŕäijŽæIJL'åď'ğéĞŔçŽĎåŕŔåđŃå

æIJňèŁĆéĂŽè£Ğä¡£çŤĺåĘĚå■ŸèğĘåŻ¿åśŢçď'žäžĘäÿĂäžŻé■ŤæşŢæŞ■ä¡IJãĂĆ
æIJňèť'ĺäÿŁïijŇäÿĂäÿłåĘĚå■ŸèğĘåŻ¿åŕśæŸŕäÿĂäÿłåůšå■ŸåIJÍæŢřçżĎçŽĎèęĘçŽŰåśĆãĂĆäÿ■äżĚäžĚæŸŕé
åĘĚå■ŸèğĘåŻ¿è£ŸèČ¡äžëäÿ■åŘŇçŽĎæŰźåijŔè¡ňæ■ćæĹŘäÿ■åŘŇçśžåđŃæİěèąĺçŐřæŢřæ■őãĂĆ
è£ŹäÿłåŕśæŸŕäÿŇéİćè£Źäÿłèŕ■åŘëçŽĎçŽőçŽĎïijŽ

```
view = memoryview(arr).cast('B')
```

åőČæŐěåŘŮäÿĂäÿłæŢřçżĎ arråźűåŕĘåĚűè¡ňæ■ćäÿžäÿĂäÿłæŮäçňęåŘůå■ŮèŁĆçŽĎåĘĚå■ŸèğĘåŻ¿ãĂĆ
æŕŤåęĆ `socket.send()` æĹŰ `send.recv_into()` ãĂĆ
åIJĺåĘĚéČĺïijŇè£ŹäžŻæŰźæşŢèČ¡åď'§çŻť'æŐěæŞ■ä¡IJè£ŹäÿłåĘĚå■ŸåŇžå§§ãĂĆä¿ŃåęĆïijŇ`sock.send()` çŻť'æŐěäżŐåĘĚå■ŸäÿŸ■åŔŚçŤ§æŢřæ■ôèĂŇäÿ■éIJĂèęĄåď'■åĹűãĂĆ `send.recv_into()` ä¡£çŤĺè£ŹäÿłåĘĚå■ŸåŇžå§§ä¡IJäÿžæŐěåŘŮæŞ■ä¡IJçŽĎè¿ŞåĚěçijŞåĘšåŇžãĂĆ

åL'l'äÿŃçŽĎäÿĂäÿłéŽ¿çĆźåŕśæŸŕśocketåĞ¡æŢřåŔŕèČ¡åŘłæŞ■ä¡IJéČĺåĹĘæŢřæ■õãĂĆ
éĂŽåÿÿæİěèőšïijŇæĹŚäžňå¿Ůä¡£çŤĺå¿Ĺåď'Žäÿ■åŘŇçŽĎ `send()` åŠŇ `recv_into()`

æÌěäijåè¿ŞæŢť äÿłæŢřçżĎãĂĆ äÿ■çŤÍæŃĚå£ČïijŇæŕŔæňąæŞ■ä¡IJåŘŐïijŇèğĘåŻ¿äijŽéĂŽè£ĞåŔŠéĂĄæĹÚ
æŰřçŽĎèğĘåŻ¿åŘŇæăůäź§æŸŕåĘĚå■ŸèęĘçŻŰåśĆãĂĆåŻãæ■d'ïijŇè£ŸæŸŕæšąæIJL'äżżä¡ŢçŽĎåd'■åĹűæŞ

è£ŹéĞŇæIJL'äÿłéŮőécŸåřśæŸŕæŐěåŔŮèĂĚå£Ěéążäž ŇåĚĹç§ěéĄŞæIJL'åd'ŽåřŚæŢřæ■őèęĄècńåŔŠéĂ
äżěä¿£åőČèČ¡écĎåĹĘéĚ■äÿĂäÿłæŢřçżĎæĹŰèĂĚçąőä£ÌåőČèČ¡åřĘæŐěåŔŮçŽĎæŢřæ■őæŤ¿åĚěäÿĂäÿłåůš
åęĆæđIJæšąåŁđæşŢç§ěéĄŞçŽĎèŕÍïijŇåŔŠéĂĄèĂĚåřśå¿ŮåĚĹåřĘæŢřæ■őåd'ğåřŔåŔŠéĂĄè£ĞæÌěïijŇçĎűå

14 çňňå■ĄäžŇçńäïijŽåźűåŔŠçijŰçĺŃ

åŕźäžŐåźűåŔŠçijŰçĺŃ, PythonæIJL'åd'Žçğ■éŢ£æIJ§æŤŕæŇĄçŽĎæŰźæşŢ,
åŇĚæŃňåd'Žçž£çĺŃ, èřČçŤÍå■Řè£ŻçĺŃ, äžěåŔŁåŘĎçğ■åŘĎæăůçŽĎåĚşäžŐçŤ§æĹŘåŹĺåĞ¡æŢřçŽĎæŁĂåů
è£ŹäÿĂçńäářĘäijŽçżŹåĞžåźűåŔŠçijŰçĺŃåŘĎçğ■æŰźéÍćçŽĎæŁĂåůğ,
åŇĚæŃňéĂŽçŤÍçŽĎåd'Žçž£çĺŃæŁĂæIJŕäžěåŔŁåźűèąŇèőąçőŮçŽĎåőđçŐřæŰźæşŢ.

åČŔçżŔéłŇäÿřárŇçŽĎçĺŃåžŘåŚŸæL'Ăç§ěéĄŞçŽĎéĆčæăů,
åd'ğåőűæŃĚå£Čåźűåŕ§çŽĎçĺŃåžŔæIJL'æ¡IJåIJĺçŽĎå■śéŹl'. åŻăæ■d',
æIJńçńäçŽĎäÿżèęĄçŽőæăĞäźŇäÿĂæŸŕçżŹåĞžæŻt'åŁăåŔŕä£ąèţŰåŠŇæŸŞèřČèŕŢçŽĎäžčçăĄ.

Contents:

14.1 12.1 åŘŕåŁĺäÿŐåĄIJæ■ćçž£çĺŃ

éŮőécŸ

ä¡ăèęĄäÿžéIJĂèęĄåźűåŔŠæL'ğèąŇçŽĎäžčçăĄåĹŻåžž/éŤĂæŕĄçž£çĺŃ

èğčåĘşæŰźæąĹ

threading åžŞåŔŕäžěåIJĺå■ŢçŃňçŽĎçž£çĺŃäÿ■æL'ğèąŇäżżä¡ŢçŽĎåIJĺ
Python äÿ■åŔŕäžěèřČçŤĺçŽĎåŕźèśąãĂĆä¡ăåŔŕäžěåĹŻåžžäÿĂäÿł Thread
åŕźèśąåźűårĘä¡ăèęĄæL'ğèąŇçŽĎåŕźèśąäžě target åŔĆæŢřçŽĎå¡ćåijŔæŔŘä¿ŻçżŹèřěåŕźèśąãĂĆ
äÿŃéÍćæŸŕäÿĂäÿłçőĂå■ŢçŽĎä¿Ńå■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¡ääĹŻåżžåě¡äÿĂäÿłçž£çÍŃåŕźèśąåŘŐïijŇèřěåŕźèśąåźűäÿ■äijŽçńŇå■şæL’ğèąŇïijŇéŹd’éİđä¡ăèřČçŤĺ
`start()` æŰźæşŢïijĹå¡Şä¡ăèřČçŤÍ `start()` æŰźæşŢæŮűïijŇåőČäijŽèřČçŤĺä¡ăäijäéĂŠè£ŻæİěçŽĎåĞ¡æ
POSIX çž£çÍŃæĹŰèĂĚäÿĂäÿł Windows çž£çÍŃïijL’ïijŇè£ŹäžŻçž£çÍŃåřĘçŤśæŞ■ä¡IJçşżçż§æİěåĚÍæİČçőąçĹ

```
if t.is_alive():
    print('Still running')
else:
    print('Completed')
```

ä¡ääź§åŔŕäžěåřĘäÿĂäÿłçž£çÍŃåŁăåĚěåĹřå¡ŞåL’■çž£çÍŃïijŇåźűç■L’å¿ĚåőČçżĹæ■ćïijŽ

```
t.join()
```

PythonèğčéĞŁåŹÍçŻt’åĹřæL’ĂæIJL’çž£çÍŃéČ¡çżĹæ■ćåL’■äż■ä‡ÍæŇĄè£ŘèąŇãĂĆåŕźäžŐéIJĂèęĄéŢ£æ
ä¿ŃåęĆïijŽ

```
t = Thread(target=countdown, args=(10,), daemon=True)
t.start()
```

åŘŐåŔřçž£çÍŃæŮăæşTç■L’å¿ĚïijŇäÿ■è£ĞïijŇè£ŹäžŻçž£çÍŃäijŽåIJĺäÿżçž£çÍŃçżĹæ■ćæŮűèĞłåŁĺéŤĂa
éŹd’äžĘåęĆäÿŁæL’Ăçd’žçŽĎäÿd’äÿłæŞ■ä¡IJïijŇåźűæšąæIJL’åd’łåd’ŽåŔŕäžěåŕźçž£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)
```

åęĆædIJçž£çÍŃæL’ğèąŇäÿĂäžŻåČŔI/Oè£ŹæåůçŽĎéŸżåąđæŞ■ä¡IJïijŇéĆčäźĹéĂŽè£Ğè¡őèŕćæİěçźĹæ■
ä¿Ńå■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)

```
        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

çŤśäžŐåĚÍåśĂèğčéĞŁéŤĄŗijĹGILïijL'çŽĎåŐ§åŻăŗijŇPython çŽĎçž£çÍŃèćńéŹŘåĹűåĹřåŘŇäýĂæŮűåĹżåŘłåĚĄèőÿäýĂäýłçž£çÍŃæL'ğèąŇè£ŹæăůäýĂäýłæL'ğèąŇæÍąåđŇçŽĎçž£çÍŃæŻť'éĂĆçŤÍäžŐåď'ĎçŘĘI/OåŠŇåĚűäżŮéIJĂèęĄåźűåŘŚæL'ğèąŇçŽĎéŸżåąđæŞ■ä¡IJïijĹæŕŤåęĆ

æIJL'æŮűä¡ääijŽçIJŃåĹřäýŃè¿źè£ŹçğéĂŽè£ĞçżğæL'£ `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()
```

åř¡çőąè£ŹæăůäźšåŘŕäżěåůěä¡IJïijŇä¡Ęè£Źä¡£å¿Ůä¡ăçŽĎäżčçăĄä¿İèţŰäžŐ `threading` åžŞïijŇæL'Ăäżěä¡ăçŽĎè£ŹäžŻäżčçăĄåŘłèČ¡åIJÍçž£çÍŃäýŁäýŃæŰĞäý■ä¡£çŤÍãĂĆäýŁæŰĞæ `threading` åžŞæŮăåĚşçŽĎïijŇè£Źæăůåřśä¡£å¿Ůè£ŹäžŻäżčçăĄåŘŕäżěèćńçŤÍåIJÍåĚűäžŰçŽĎäýŁäýŃæŰČ `multiprocessing` æÍąåİŮåIJÍäýĂäýłå■ŢçŃňçŽĎè£ŻçÍŃäý■æL'ğèąŇä¡ăçŽĎäżčçăĄïijŽ

```
import multiprocessing
c = CountdownTask(5)
p = multiprocessing.Process(target=c.run)
p.start()
```

åĘ■æňąéĞ■çŤşïijŇè£Źæőţäżčçă Ąäž ĚéĂĆçŤÍäžŐ CountdownTask

çśżæŸŕäżěçŃňçńŃäžŐåőđéŹĚçŽĎåźűåŔŚæĽŃæőţïijĹåď'Žçž£çĺŃãĂĄåď'Žè£ŻçĺŃç■Ľ'ç■Ľ'ïijĽåőđçŐřçŽĎæ

14.2 12.2 åĹď'æŰ■çž£çĺŃæŸŕåŘęåůšçżŔåŘŕåŁĺ

éŮőécŸ

ä¡ăåůšçżŔåŘŕåŁĺäžĘäÿĂäÿłçž£çĺŃïijŃä¡ĘæŸŕä¡ăæČşç§ěéĄŞåőČæŸŕäÿ■æŸŕçIJ§çŽĎåůšçżŔåijĂåğŃè£

èğčåĘşæŰźæąĹ

çž£çĺŃçŽĎäÿĂäÿłåĚşéŤőçĽźæĂğæŸŕæŕŔäÿłçž£çĺŃéČ¡æŸŕçŃňçńŃè£ŘèąŇäÿŤçŁűæĂĄäÿ■åŔŕéćĎæţ
`threading` åžŞäÿ■çŽĎ `Event` åŕźèśąãĂĆ `Event` åŕźèśąåŇĚåŘńäÿĂäÿłåŔŕçŤśçž£çĺŃèő¿ç¡őçŽĎä£ąåŘůæ
åŕźèśąäÿ■çŽĎä£ąåŘůæăĞå£Ůèćńèő¿ç¡őäÿžåĄĞãĂĆåęĆæđIJæIJĽçž£çĺŃç■Ľ'å¿ĚäÿĂäÿł
event åŕźèśąïijŃèĂŇè£Źäÿł event åŕźèśąçŽĎæăĞå£ŮäÿžåĄĞïijŃéĆčäźĹè£Źäÿłçž£çĺŃåřĘäijŽèćńäÿĂçŻť'éŸź
event åŕźèśąçŽĎä£ąåŘůæăĞå£Ůèő¿ç¡őäÿžçIJ§ïijŃåőČåřĘåŤď'éĘŠæĽĂæIJĽç■Ľ'å¿Ěè£Źäÿł
event åŕźèśąçŽĎçž£çĺŃãĂĆåęĆæđIJäÿĂäÿłçž£çĺŃç■Ľ'å¿ĚäÿĂäÿłåůšçżŔèćńèő¿ç¡őäÿžçIJ§çŽĎ
event åŕźèśąïijŃéĆčäźĹåőČåřĘå£¡çŢèè£ŹäÿłäžŃäżűïijŃçžğçż■æĽ'ğèąŇãĂĆ
äÿŃè¿źçŽĎäżčçăĄåśŢçď'žäžĘåęĆä¡Ţä¡£çŤĺ `Event` æİěå■ŔèřČçž£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'ğèąŇè£Źæőţäžčçă ĄïijŃâĂIJcountdown is runningâĂİ æĂżæŸŕæŸ¿çď'žåIJĺ
âĂIJcountdown startingâĂİ äźŃåŘŐæŸ¿çď'žãĂĆè£ŹæŸŕçŤśäžŐä¡£çŤĺ
event æİěå■ŔèřČçž£çĺŃïijŃä¡£å¿Ůäÿžçž£çĺŃèęĄç■Ľ'åĹř `countdown()`
åĞ¡æŢřè¿ŞåĞžåŘŕåŁĺä£ąæĄŕåŘŐïijŃæĽ'■èČ¡çžğçż■æĽ'ğèąŇãĂĆ

èőÍèőž

event åŕźèśąæIJĂåě¡å■Ţæňąä¡£çŤĺïijŇåřśæŸŕèŕť'ïijŇä¡ăåĹŻåžžäÿĂäÿł event åŕźèśąïijŇèőľæ§Řäÿłçž£çÍŃç■L'å¿Ěè£ŹäÿłåŕźèśąïijŇäÿĂæŮęè£Źäÿłåŕźèśąèćńèő¿ç¡őäÿžçIJ§ïijŇä¡ăåřśåžŤèŕè clear() æŰźæşŢæÍěéĞ■ç¡ő event åŕźèśąïijŇä¡ĘæŸŕå¿ĹéŽ¿çąőä£ÍåőĽåĚĺåIJřæÿĚçŘĘ event åŕźèśåáźűåŕźåőČéĞ■æŰřętŃåĂijãĂĆå¿ĹåŔŕèČ¡äijŽåŘŚçŤ§éŤŹè£ĞäžŃäžűãĂĄæ■żéŤĄæĹŰèĂĚåĚűä event åŕźèśąçŽĎäžčçăĄäijŽåIJĺçž£çÍŃåĘ■æňąç■Ľ'å¿Ěè£ŹäÿłåŕźèśąäźŃåĽ'■æĽ'ğèąŇïijĽ'ãĂĆåęĆæđIJäÿĂäÿłçž£çÍŃéIJĂèęĄäÿ■åĄIJåIJřéĞ■åď'■ä¡£çŤÍ event åŕźèśąïijŇä¡ăæIJĂåě¡ä¡£çŤÍ Condition åŕźèśąæÍěäžčæŽ£ãĂĆäÿŃéÍćçŽĎäžčçăĄä¡£çŤÍ Condition åŕźèśąåődçŐřäžĘäÿĂäÿłåŚÍæIJ§åőŽæŮűåŹĺïijŇærŔå¡ŞåőŽæŮűåŹĺèűĚæŮűçŽĎæŮűåĂŹïijŇåĺ

```
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)

(çż■äÿŁéąţ)

```
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■èęĄçL’źçĆźæŸŕå¡ŞåőČèćńèő¿ç¡őäÿžçIJ§æŮűäijŽåŤd’éĘŠæL’ĂæIJL’ç■L’å¿Ě
Condition åŕźèśąæİěæŻ£äžčãĂĆèĂČèŹŚäÿĂäÿŃè£Źæőţä¡£çŤĺä£ąåŔůéĞŔåőđçŐřçŽĎäžčçăĄïijŽ

```
# 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¿źçŽĎäžčçăĄåřĘäijŽåŘŕåŁĺäÿĂäÿłçž£çĺŃæśäïijŇä¡ĘæŸŕåźűæšąæIJL’äżĂäźĹäžŃæČĚåŔŚ

```
>>> sema.release()
Working 0
>>> sema.release()
Working 1
>>>
```

çijŰåĘŹæűL’åŘŁåĹråď’ğéĞŔçŽĎçž£çĺŃéŮt’åŘŇæ■ěéŮőécŸçŽĎäžčçăĄäijŽèőľ’ä¡ăçŮŻäÿ■æňšçŤ§ãĂĆ

14.3 12.3 çž£çÍŃéŮťéĂŽä£ą

éŮőécŸ

ä¡ăçŽĎçÍŃåžŘäÿ■æIJL’åď’Žäÿłçž£çÍŃïijŇä¡ăéIJĂèęĄåIJÍè£ŹäžŻçž£çÍŃäźŃéŮť’åőL’åĚÍåIJřäžď’æ■ćä£ąa

èğčåĘşæŰźæąĹ

äžŐäÿĂäÿłçž£çÍŃåŘŠåŘęäÿĂäÿłçž£çÍŃåŘŠéĂĄæŢřæ■őæIJĂåőL’åĚÍçŽĎæŰźåijŔåŘŕèČ¡åřśæŸŕä¡£çŤÍ queue åžŞäÿ■çŽĎéŸ§åĹŮäžĘãĂĆåĹŻåžžäÿĂäÿłèćńåď’Žäÿłçž£çÍŃåĚśäžńçŽĎ Queue åŕźèśąïijŇè£ŹäžŻçž£çÍŃéĂŽè£Ğä¡£çŤÍ put() åŠŇ get() æŞ■ä¡IJæİěåŘŠéŸ§åĹŮäÿ■æůżåŁăæĹŰèĂĚåĹăéŹď’åĚČçť’ăãĂĆ 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’Ăäžěä¡ăåŔŕäžěéĂŽè£ĞåőČåIJÍåď’Žäÿłçž£çÍŃé å¡Şä¡£çŤĺéŸ§åĹŮæŮűïijŇå■ŔèřČçŤ§äžğèĂĚåŠŇæűĹèť’źèĂĚçŽĎåĚşéŮ■éŮőécŸåŔŕèČ¡äijŽæIJL’äÿĂäžŹé

```
from queue import Queue
from threading import Thread

# Object that signals shutdown
_sentinel = object()

# A thread that produces data
```

(continues on next page)

(çż■äÿŁéąţ)

```
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'źæőŁçŽĎåIJřæŰźïijŽæűÍèt'źèĂĚåIJÍèŕżåĹřè£ŹäÿłçL'źæőŁåĂijäźŃåŘŐçńŃ
åř¡çőąéŸ§åĹŮæŸŕæIJĂåÿÿèğĄçŽĎçž£çÍŃéŮt'éĂŽä£ąæIJžåĹűïijŇä¡ĘæŸŕäż■çĎűåŔŕäżèèĞłåůśéĂŽè£ĞåĹŽ
`Condition` åŘŸéĞŔæÌěåŇĚèčĚä¡ăçŽĎæŢřæ■őçżŞæđĎãĂĆäÿŃè¿źè£Źäÿłä¿Ńå■ŘæijŤçd'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§åĹŮæÌěè£ŻèąŇçž£çÍŃéŮt'éĂŽä£ąæŸŕäÿĂäÿłå■ŢåŘŚãĂĄäÿ■çąőåőŽçŽĎè£ĞçÍŃãĂĆéĂŽåÿ
`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()
```

åęĆædIJäÿĂäÿłçž£çÍŃéIJĂèęĄåIJÍäÿĂäÿłâĂIJæűĹèť źèĂĚâĂİçž£çÍŃåď ĎçŘĘåőŇçĽźåőŽçŽĎæŢřæ■ő
Event æŤ¿åĹřäÿĂèţůä¡£çŤĺïijŇè£ŹæăůâĂIJçŤ§ăžğèĂĚâĂİåřśåŔŕäżěéĂŽè£Ğè£ŹäÿłEventåŕźèśąæİěçŻŚæţ

```
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)

```
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'ŽçžŁçĺŃçĺŃåžŔåIJĺåď'ŽæŢřæČĚåĘţäýŃæŸŕäýĂäýłæŕŤè¿ČæŸŐæŹż
ä¡£çŤĺçž£çĺŃéŸ§åĹŮæIJL'äýĂäýłèęĄæşĺæĎŘçŽĎéŮőécŸæŸŕïijŇåŘŠéŸ§åĹŮäý■æůžåŁăæŢřæ■őéąźæŮűå

```
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 åŕźèśąæŔŘä¿ŻäýĂäžŻåIJĺå¡ŞåL'■äýŁäýŃæŰĞå¿ĹæIJL'çŤĺçŽĎéŹĎåŁăçL'źæĂğãĂĆæŕŤåęĆåIJ
Queue åŕźèśąæŮűæŔŘä¿ŻåŘŕéĂL'çŽĎ size åŔĆæŢřæİěéŹŘåĹűåŔŕäžěæůžåŁăåĹřéŸ§åĹŮäý■çŽĎåĚČçť'ă
âĂIJæűĹèť'źâĂİçŽĎéĂ§åžęå£ńïijŇéĆčäźĹä¡£çŤĺåŽžåőŽåď'ğåřŔçŽĎéŸ§åĹŮåřśåŔŕäžěåIJĺéŸ§åĹŮåůšæžąçZ
get() åŠŇ put() æŰźæşŢéČ¡æŤřæŇĄéİđéŸżåąđæŰźåijŔåŠŇèő¿åőŽèűĚæŮűïijŇä¿ŃåęĆïijŽ

```
import queue
q = queue.Queue()

try:
    data = q.get(block=False)
except queue.Empty:
    ...

try:
```

(continues on next page)

```
    q.put(item, block=False)
except queue.Full:
    ...

try:
    data = q.get(timeout=5.0)
except queue.Empty:
    ...
```

è£ŹäžŻæŞ■ä¡IJéČ¡åŔŕäžěçŤĺæİěéĄ£åĚ■å¡ŞæĽğèąŇæ§ŘäžŻçĽźåőŽéŸ§åĹŮæŞ■ä¡IJæŮűåŔŚçŤ§æŮăé
put() æŰźæşŢåŠŇäýĂäýłåŻžåőŽåď˙ğåřŔçŽĎéŸ§åĹŮäýĂèţůä¡£çŤĺïijŇè£Źæăůå¡ŞéŸ§åĹŮåůšæżąæŮűåřšá

```
def producer(q):
    ...
    try:
        q.put(item, block=False)
    except queue.Full:
        log.warning('queued item %r discarded!', item)
```

åęĆæđIJä¡ăèŕŢåŻ¿èőľæűĹèť´źèĂĚçž£çÍŃåIJĺæĽ˙ğèąŇåČŔ q.get() è£ŹæăůçŽĎæŞ■ä¡IJæŮűïijŇèűĚæŮűèĞłåŁĺçžĹæ■ćäžěä¿£æčĂæ§ěçžĹæ■ćæăĞå£ŮïijŇä¡ăåžŤèŕěä¡£çŤĺ q.get() çŽĎåŔŕéĂĽ˙åŔĆæŢř timeout ïijŇåęĆäýŃïijŽ

```
_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˙åőđçŤĺæŰźæşŢåŔŕäžěèŐůåŔŰäýĂäýłéŸ§åĹŮçŽĎå¡ŞåĽ■åď˙ğåřŔåŠŇçŁűæĂĄãĂĆä¡ĘèęĄæşĺæĎŔïijŇè
empty() åĹd˙æŰ■åĞžè£ŹäýłéŸ§åĹŮäýžçľ˙žïijŇä¡ĘåŘŇæŮűåŔęåď˙ŰäýĂäýłçž£çÍŃåŔŕèČ¡åůšçżŔåŘŚè£Źä

14.4 12.4 çžŹåĚşéŤőéČĺåĹĘåŁăéŤĄ

éŮőécŸ

ä¡ăéIJĂèęĄåŕźåď˙Žçž£çÍŃçÍŃåžŔäý■çŽĎäýt˙çŢŇåŇžåŁăéŤĄäżěéĄ£åĚ■çńđäžĽ˙æİąäżűãĂĆ

èğčåĘşæŰźæąĹ

èęĄåIJĺåď Žçž£çĺŃçĺŃåžŔäÿ■åőL’åĚĺä¡£çŤĺåŔŕåŔŸåŕźèśąïijŇä¡ăéIJĂèęĄä¡£çŤĺ threading åžŞäÿ■çŽĎ `Lock` åŕźèśąïijŇåřśåČŔäÿŃè¿źè£ŹäÿŁä¿Ńå■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■åŔěåİŮäÿĂèţůä¡£çŤĺåŔŕäžěä£ÌèŕĄäžŠæŰěæL’ğèąŇïijŇåřśæŸŕæŕŔæňąåŔŁæI
with èŕ■åŔěåŇĚåŘńçŽĎäżčçăĄåİŮãĂĆwith èŕ■åŔěäijŽåIJĺè£ŹäŷŁäżčçăĄåİŮæL’ğèąŇåL’■èĞŁåŁĺèŐůåŔŰéŤ

èőĺèőž

çž£çĺŃèřČåžęæIJňèťĺäÿŁæŸŕäÿ■çąőåőŽçŽĎïijŇåŻäæ■ď’ïijŇåIJĺåď Žçž£çĺŃçĺŃåžŔäÿ■éŤŹèŕŕåIJřä¡£çŤ
åIJĺäÿĂäžŻâĂIJèĂĄçŽĎâĂİ Python äżčçăĄäÿ■ïijŇæŸ¿åijŔèŐůåŔŰåŠŇéĞŁæŤ¿éŤĄæŸŕå¿Ĺåÿÿèğ 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)

```
        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£ĺèŕĄåIJĺè£Źäÿď çğ■æČĚåĘţäÿŃäż■èČ¡æ■čçąőéĞŁæŤ¿éŤĄïijĽãĂĆ äÿžäžĘéĄ£åĚ■åĞžçŐřæ■żéŤĄçŽĎæČĚåĘţïijŇä¡£çŤĺéŤĄæIJžåĹűçŽĎçĺŃåžŔåžŤèŕěèő¿åőŽäÿžæŕŔäÿłçž£çĺĺ åIJĺ `threading` åžŞäÿ■è£ŸæŔŘä¿ŻäžĘåĚűäżŰçŽĎåŘŃæ■ěåŐ§èŕ■ïijŇæŕŤåęĆ `RLock` åŠŇ `Semaphore` åŕźèśąãĂĆä¡ĘæŸŕæăźæ■őäžěå¿ĂçżŔéłŇïijŇè£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■ïijŇæšąæIJĽåŕźæŕŔäÿĂäÿłåőđä¿Ńäÿ■çŽĎåŔŕåŔŸåŕźèśąåŁäéŤĄïijŇåŔŰè decr æŰźæşŢãĂĆ è£Źçğ■åőđçŐřæŰźåijŔçŽĎäÿĂäÿłçĽźçĆźæŸŕïijŇæŮäèőžè£Źäÿłçśżæ IJĽåď ŽåŕŠäÿłåőđä¿

ä£ąåŔůéĞŔåŕźèśąæŸŕäÿĂäÿłåžžçńŇåIJĺåĚśäžńèőąæŢřåŹĺå§žçąĂäÿŁçŽĎåŘŇæ■ěåŐ§èŕ■ãĂĆåęĆæđIJèőąæŢ
èŕ■åŔěåřĘèőąæŢřåŹĺåĞŔ1ïijŇçž£çĺŃèćńåĚĄèőÿæĽğèąŇãĂĆwith
èŕ■åŔěæĽğèąŇçżŞæİ§åŘŐïijŇèőąæŢřåŹĺåŁäïijŚãĂĆåęĆæđIJèőąæŢřåŹĺäÿž0ïijŇçž£çĺŃåřĘèćńéŸżåąđïijŇçŽ

```
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■çŽĎåžŢåśĆçŘĘèőžåŠŇåðđçŐřæĎ§åĚť'èűčïijŇåŘŕäžěåŔĆèĂČæŞ

14.5 12.5 éŸšæ■ćæ■żéŤĄçŽĎåŁăéŤĄæIJžåĹű

éŮőécŸ

ä¡äæ■čåIJĺåĘŹäÿĂäÿłåď'Žçž£çĺŃçĺŃåžŔïijŇåĚűäÿ■çž£çĺŃéIJĂèęĄäÿĂæňąèŐůåŔŰåď'ŽäÿłéŤĄïijŇæ■d

èğčåĘşæŰźæąĹ

åIJĺåď'Žçž£çĺŃçĺŃåžŔäÿ■ïijŇæ■żéŤĄéŮőécŸå¿Ĺåď'ğäÿĂéČĺåĹĘæŸŕçŤšäžŐçž£çĺŃåŘŇæŮűèŐůåŔŰå
æŮűåĂŹåŔŚçŤ§éŸżåąđïijŇéĆčäźĹè£Źäÿłçž£çĺŃåřšåŔŕèČ¡éŸżåąđåĚűäžŰçž£çĺŃçŽĎæĽ'ğèąŇïijŇäžŐèĂŇã
èğčåĘşæ■żéŤĄéŮőécŸçŽĎäÿĂçğ■æŰźæąĹæŸŕäÿžçĺŃåžŔäÿ■çŽĎæŕŔäÿĂäÿłéŤĄåĹĘéĚ■äÿĂäÿłåŤŕäÿĂçŽ
æŸŕéİđåÿÿåőźæŸŞåðđçŐřçŽĎïijŇçď'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)

```
    # 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¡£çŤĺè£Źäýłäý Łäý ŃæŰĞçőąçŘĘåŹĺåŚćïij§ä¡ååŕŕäžěæŇL'çĚğæ■čåÿýéĂŤå¿ĎåĹŻåžžäýĂäýłéŤ
acquire() åĞ¡æŢřæİěçŤşèřůéŤĄïijŇ çd'žä¿ŃåęĆäýŃïijŽ

```
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()
```

åęĆædIJä¡ăæL'ğèąŇè£ŹæőţäžčçăĄïijŇä¡ăäijŽåŔŚçŐřåőČå■şä¡£åIJĺäý■åŘŇçŽĎåĞ¡æŢřäý■äžěäý■åŘŇç
åĚűåĚşéŤőåIJĺäžŐïijŇåIJĺçňňäýĂæőţäžčçăĄäý■ïijŇæĹŚäžňářźè£ŹäžŻéŤĄè£Żèą Ňäž ĘæŐŠåžŔãĂĆéĂŽè£ĞŻ
åęĆædIJæIJL'åď'Žäýł acquire() æŞ■ä¡IJècńåţŇåěŮèřČçŤĺïijŇåŕŕäžěéĂŽè£ĞçžçĺŃæIJňåIJřå■ŸåĆĺïijĹT
åĄĞèő¿ä¡ăçŽĎäžčçăĄæŸŕè£ŹæăůåĘŹçŽĎïijŽ

```
import threading
x_lock = threading.Lock()
y_lock = threading.Lock()

def thread_1():
```

(continues on next page)

(çż■äÿŁéąţ)

```
    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¡ãè£ŘèąŇè£ŹäÿłçL’ĹæIJňçŽĎäżčçăĄïijŇå£ĚåőŽäijŽæIJL’äÿĂäÿłçž£çÍŃåŔŠçŤ§åť’l’æžČïijŇåi

```
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
>>>
```

åŔŠçŤ§åť’l’æžČçŽĎåŐ§åŽăåIJÍäžŐïijŇærŔäÿłçž£çÍŃéČ¡èőřå¡ŢçİĂèĞłåůšåůšçżŔèŐůåŔŰåĹřçŽĎéŤĄã
`acquire()` åĞ¡æŢřäijŽæčĂæ§ëäźŃåL’■åůšçżŔèŐůåŔŰçŽĎéŤĄåĹŮèąĺïijŇ
çŤšäžŐéŤĄæŸŕæŇL’çĚğå■ĞåžŔæŐŠåĹŮèŐůåŔŰçŽĎïijŇæL’Ăäžěå Ğ¡æŢřäijŽèőd’äÿžäźŃåL’■åůšèŐůåŔŰç

èőÍèőž

æ■żéŤĄæŸŕæŕŔäÿĂäÿłåd’Žçž£çÍŃçÍŃåžŔéČ¡äijŽéİcäÿt’çŽĎäÿĂäÿłéŮőécŸïijĹåŕšåČŔåőČæŸŕæŕŔäÿĂ
çž£çÍŃåŔłèČ¡åŘŇæŮűä£ÍæŇĄäÿĂäÿłéŤĄïijŇè£ŹæăůçÍŃåžŔåŕšäÿ■äijŽèćńæ■żéŤĄéŮőécŸæL’ĂåŻřæL’řãĂ

æ■żéŤĄçŽĎæčĂæţŃäÿŐæĄćåd’■æŸŕäÿĂäÿłåĞääźŐæšąæIJL’äijŸéŻĚçŽĎèğčåĘşæŰźæąĹçŽĎæL’l’åśŢ

è£ŘèąŇçŽĎæŮűåĂŹäijŽæŕŔéŽŤäÿĂæőţæŮűéŮťéĞ■ç¡őèőąæŢřåŹĺïijŇåIJÍæšąæIJL’åŔŞçŤ§æ■žéŤĄçŽĎæČ
èűĚæŮűïijŇè£ŹæŮűçÍŇåžŘäijŽéĂŽè£ĞéĞ■åŘŕèĞłèžńæĄćåď■åĹřæ■čåÿÿçŁűæĂĄãĂĆ

éĄ£åĚ■æ■žéŤĄæŸŕåŘęåď ŰäÿĂçğ■èğčåĘşæ■žéŤĄéŮőécŸçŽĎæŰźåijŔïijŇåIJÍè£ŻçÍŇèŐůåŔŰéŤĄçŽ
æ■žéŤĄçŁűæĂĄãĂĆèŕĄæŸŐåřşçŢŹçżŹèŕžèĂĚä¡IJäÿžçżČäźääžĘãĂĆéĄ£åĚ■æ■žéŤĄçŽĎäÿżèęĄæĂİæČşa
æ■žéŤĄçŽĎäÿĂäÿłå£ĚèęĄæİąäžűïijŇäżŐèĂŇéĄ£åĚ■çÍŇåžŔè£ŻåĚěæ■žéŤĄçŁűæĂĄãĂĆ

äÿŇéİćäžěäÿĂäÿłåĚşäžŐçž£çÍŇæ■žéŤĄçŽĎçżŔåĚÿéŮőécŸïijŽâĂIJåŞšå■ęåőűåřšéď ŘéŮőécŸâĂİïijŇä
éİćåL’■æIJL’äÿĂçćŮéě■åŠŇäÿĂåŔłç■ůå■ŘãĂĆåIJÍè£ŹéĞŇæŕŔäÿłåŞšå■ęåőűåŔŕäžěçIJŇåĄŽæŸŕäÿĂäÿłçŃr
æĂİèĂČãĂĄåŘČéě■äÿL’çğ■çŁűæĂĄäÿ■çŽĎäÿĂäÿłãĂĆéIJĂèęĄæşÍæĎŔçŽĎæŸŕïijŇæŕŔäÿłåŞšå■ęåőűåŘČ
éĆčäźĹäżŰäžňäžŤäÿłéČ¡åŔłèČ¡æŃ£çİĂäÿĂåŔłç■ůå■ŘåİŘåIJÍéĆčåĎ£ïijŇçŻt’åĹřéě£æ■žãĂĆæ■ď æŮűäžŰä
äÿŇéİćæŸŕäÿĂäÿłçőĂå■ŢçŽĎä¡£çŤÍæ■žéŤĄéĄ£åĚ■æIJžåĹűèğčåĘşâĂIJåŞšå■ęåőűåřšéď 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’źåĹńæşÍæĎŔåĹřïijŇäÿžäžĘéĄ£åĚ■æ■žéŤĄïijŇæL’ĂæIJL’çŽĎåŁäéŤĄæŞ■ä¡IJå£Ěé
`acquire()` åĞ¡æŢřãĂĆåęĆæđIJäżčçăĄäÿ■çŽĎæ§ŘéČĺåĹĘçżŢè£Ğacquire
åĞ¡æŢřçŻt’æŐěçŤşèřůéŤĄïijŇéĆčäźĹæŢt’äÿłæ■žéŤĄéĄ£åĚ■æIJžåĹůåřšäÿ■ètůä¡IJçŤĺäžĘãĂĆ

14.6 12.6 ä£İå■Ÿçž£çÍŇçŽĎçŁűæĂĄä£ąæĄŕ

éŮőécŸ

ä¡ăéIJĂèęĄä£İå■Ÿæ■čåIJÍè£ŘèąŇçž£çÍŇçŽĎçŁűæĂĄïijŇè£ŹäÿłçŁűæĂĄåŕźäžŐåĚűäżŰçŽĎçž£çÍŇæŸ

èğčåĘşæŰźæąĹ

æIJL’æŮűåIJÍåď Žçž£çÍŇçijŰçÍŇäÿ■ïijŇä¡ăéIJĂèęĄåŘłä£İå■Ÿå¡ŞåL’■è£ŘèąŇçž£çÍŇçŽĎçŁűæĂĄãĂĆ
èęĄè£ŹäźĹåĄŽïijŇåŘŕä¡£çŤÍ `thread.local()` åĹŻåžžäÿĂäÿłæIJňåIJřçž£çÍŇå■ŸåĆĺåŕźèśąãĂĆ
åŕźè£ŹäÿłåŕźèśąçŽĎåśđæĂğçŽĎä£İå■ŸåŠŇèŕžåŘŰæŞ■ä¡IJéČ¡åŘłäijŽåŕźæL’ğèąŇçž£çÍŇåŘŕèğĄïijŇèĂŇåĚ

ä¡IJäÿžä¡£çŤÍæIJňåIJřå■ŸåĆÍçŽĎäÿĂäÿłæIJL’èűčçŽĎåőđéŹĚä¿Ńå■ŘïijŇ èĂČèŹŚåIJÍ8.3åřŔèŁĆåőŽäźL’è£ĞçŽĎ `LazyConnection` äÿŁäÿŇæŰĞçőąçŘĘåŹÍçśžãĂĆ äÿŇéÍćæĹŚäżňåŕźåőČè£Žèą̃ŇäÿĂäžŽåřŔçŽĎä£őæŤźä¡£å¿ŮåőČåŔŕäżěéĂĆçŤÍäžŐåd’Žçž£çÍŇïijŽ

```
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
```

äžčçăĄäÿ■ïijŇèĞłåůśèğĆår§åŕźäžŐ `self.local` åśđæĂğçŽĎä¡£çŤÍãĂĆ åőČèćńåĹİåğŃåŇŰäÿžäÿĂäÿł `threading.local()` åőđä¿ŃãĂĆ åĚűäžŰæŰźæşŢæŞ■ä¡IJèćńå■ŸåĆÍäÿž `self.local.sock` çŽĎåěŮæŐěå■Ůåŕźèśąã ĂĆ æIJL’äžĘè£ŹäžŽåřśåŘŕäžěåIJÍåd’Žçž£çÍŇäÿ■åőL’åĚÍçŽĎä¡£çŤÍ `LazyConnection` åőđä¿ŃäžĘãĂĆä¿ŃåęĆïijŽ

```
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()
```

åőČäźŃæĽĂäžèèąŇå¿ŮéĂŽçŽĎåŐ§åŻăæŸŕæŕŔäÿłçž£çĺŃäijŽåĹŻåžžäÿĂäÿłèĞłåůśäÿŞåśđçŽĎåěŮæŐ
åŻăæ■d'ïijŇå¡Şäÿ■åŘŇçŽĎçž£çĺŃæĽ'ğèąŇåěŮæŐěå■ŮæŞ■ä¡IJæŮűïijŇçŤśäžŐæŞ■ä¡IJçŽĎæŸŕäÿ■åŘŇçŽ

èőĺèőž

åIJĺåď'ğéČĺåĹEçĺŃåžŔäÿ■åĹŻåžžåŠŇæŞ■ä¡IJçž£çĺŃçL'źåőŽçŁűæĂĄåźűäÿ■äijŽæIJL'äžĂäźĹéŮőécŸãĄ
äÿ■è£ĞïijŇå¡ŞåĞžäžĘéŮőécŸçŽĎæŮűåĂŹïijŇéĂŽåÿÿæŸŕåŽääÿžæ§Řäÿłåŕźèśąèćńåď'Žäÿłçž£çĺŃä¡£çŤĺåĹ
æŕŤåęĆäÿĂäÿłåěŮæŐěå■ŮæĹŰæŰĞäžűãĂĆä¡ääÿ■èČ¡èől'æL'ĂæIJL'çž£çĺŃåĚśäžńäÿĂäÿłå■ŢçŃňåŕźèśąïijŇ
åŽääÿžåď'Žäÿłçž£çĺŃåŘŇæŮűèŕžåŠŇåĘźçŽĎæŮűåĂŹäijŽäžğçŤ§æůûäźśãĂĆ
æIJňåIJřçž£çĺŃå■ŸåĆĺéĂŽè£Ğèől'è£ŹäžŽèţĎæžŘåŔłèČ¡åIJĺèćńä¡£çŤĺçŽĎçž£çĺŃäÿ■åŔŕèğĄæİèèğčåĘşè£Ź

æIJňèŁĆäÿ■ïijŇä¡£çŤĺ `thread.local()` åŔŕäžèèől' `LazyConnection` çśżæŤŕæŇĄäÿĂäÿłçž£çĺŃäÿĂäÿłè£đæŐěïijŇ
èĂŇäÿ■æŸŕåŕźäžŐæL'ĂæIJL'çŽĎè£ŽçĺŃéČ¡åŔłæIJL'äÿĂäÿłè£đæŐěãĂĆ

åĚűåŐ§çŘĘæŸŕïijŇæŕŔäÿł `threading.local()` åőđä¿ŃäÿžæŕŔäÿłçž£çĺŃçżt'æŁd'çİĂäÿĂäÿłå■Ţç
æL'ĂæIJL'æŹőéĂŽåőđä¿ŃæŞ■ä¡IJæŕŤåęĆèŐůåŔŰãĂĄä£őæŤźåŠŇåĹăéŹd'åĂijäžĚäžĚæŞ■ä¡IJè£Źäÿłåůůåĺ
æŕŔäÿłçž£çĺŃä¡£çŤĺäÿĂäÿłçŃñçńŇçŽĎå■ŮåĚÿåŕśåŔŕäžěä£İèŕĄæŢřæ■őçŽĎéŽŤçęžäžĘãĂĆ

14.7 12.7 åĹŻåžžäÿĂäÿłçž£çĺŃæśă

éŮőécŸ

ä¡ăåĹŻåžžäÿĂäÿłåůěä¡IJèĂĚçž£çĺŃæśăïijŇçŤĺæİěåŞ■åžŤåőćæĹůçńŕèŕůæśĆæĹŰæL'ğèąŇåĚűäžŰçŽĎå

èğčåĘşæŰźæąĹ

`concurrent.futures` åĞ¡æŢřåžŞæIJL'äÿĂäÿł `ThreadPoolExecutor` çśżåŔŕäžěèćńçŤĺæİěåőŃæĹŘè£ŹäÿłäžžåŁąãĂĆ äÿŃéİćæŸŕäÿĂäÿłçőĂå■ŢçŽĎTCPæIJ■åŁąåŹĺijŇä¡£çŤĺäžŀ

```
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)

```
    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))
```

åęĆædIJä¡ăæČşæL'ŃåŁĺåĹŻåżžä¡äèĞłåůśçŽĎçž£çĺŃæśäïijŇ
éĂŽåÿÿåŘŕäžěä¡£çŤĺäÿĂäÿłQueueæİěè¡żæİ¿åőđçŐřãĂĆäÿŃéİćæŸŕäÿĂäÿłçĺ■å¿őäÿ■åŘŇä¡ĘæŸŕæL'ŃåŁĺåő

```
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` çŻÿåŕźäžŐæĽ'ŃåŁĺåőđçŐřçŽĎäÿĂäÿłåě¡åď'ĎåIJĺäžŐåőČä¡£å¿Ů äżżåŁąæŔŘäžď'èĂĚæŻť'æŰźä¿£çŽĎäżŐèćńèřČçŤĺåĞ¡æŢřäÿ■èŐůåŔŰè£ŤåŻđåĂijãĂĆä¿ŃåęĆïijŇä¡ääŔŕèČ

```
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■Ŕä¡IJïijŇçĎůåŘŐä çĽźåĹńçŽĎïijŇ`a.result()` æŞ■ä¡IJäijŽéŸżåąđè£ŻçÍŃçŻť'åĹřåŕźåžŤçŽĎåĞ¡æŢřæĽ'ğèąŇåőŇæĹŘåźűè£Ť

èőÍèőž

éĂŽåÿÿæİěèőšïijŇä¡ääžŤèŕěéĄ£åĚ■çijŰåĘŹçž£çÍŃæŢřéĞŘåŘŕäżěæŮäéŹŘåĹűäćđéŢ£çŽĎçÍŃåžŔãĂĆ

```
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)

```
        client_sock, client_addr = sock.accept()
        t = Thread(target=echo_client, args=(client_sock, client_
↪addr))
        t.daemon = True
        t.start()

echo_server(('',15000))
```

åř¡çőąè£Źäÿłáź§åŔŕäżěåůěä¡IJïijŇ ä¡ĘæŸŕåőČäÿ■èČ¡æŁţå¿ąæIJĽäžžèŕŢåŽ¿éĂŽè£ĞåĹŻåžžåď ğéĞŔçž
éĂŽè£Ğä¡£çŤĺéćĎåĔĹåĹİåğŃåŇŰçŽĎçž£çĺŃæśäïijŇä¡äåŔŕäżěèő¿ç¡őåŘŃæŮűè£Řèąŕçž£çĺŃçŽĎäÿŁéŹŘ

ä¡ääåŔŕèČ¡äijŽåĔşå£ČåĹŻåžžåď ğéĞŔçž£çĺŃäijŽæIJĽäżĂäźĹåŘŐæđIJãĂĆ
çŐřäżčæŞ■ä¡IJçşżçż§åŔŕäżěå¿Ĺè¡żæĬ¿çŽĎåĹŻåžžåĞää■Čäÿłçž£çĺŃçŽĎçž£çĺŃæśäãĂĆ
çŤŽèĞşïijŇåŘŃæŮűåĞää■Čäÿłçž£çĺŃç■Ľå¿Ĕåůěä¡IJåźűäÿ■äijŽåŕźåĚűäżŰäżčçăĄäžğçŤ§æĂğèČ¡å¡śåŞ■ãĂ
å¡ŞçĎűäžĘïijŇåęĆæđIJæĽĂæIJĽçž£çĺŃåŘŃæŮűèćńåŤď éĘŠåźűçńŃå■şåIJĺCPUäÿŁæĽğèąŇïijŇéĆčåřśäÿ■
éĂŽåÿÿïijŇä¡ääžŤèŕěåŔłåIJĺI/Oåď ĎçŘĘçŻÿåĔşäżčçăĄäÿ■ä¡£çŤĺçž£çĺŃæśäãĂĆ

åĹŻåžžåď ğçŽĎçž£çĺŃæśäçŽĎäÿĂäÿłåŔŕèČ¡éIJĂèęĄåĔşæşĺçŽĎéŮőéćŸæŸŕåĘĚå■ŸçŽĎä¡£çŤĺãĂĆ
ä¿ŃåęĆïijŇåęĆæđIJä¡ääåIJĺOS Xçşżçż§äÿŁéĺćåĹŻåžž2000äÿłçž£çĺŃïijŇçşżçż§æŸ¿çď žPythonè£ŻçĺŃä¡£çŤĺ
äÿ■è£ĞïijŇè£Źäÿłèőąçő ŮéĂŽåÿÿæŸŕæIJĽèŕŕåůőçŽĎãĂĆå¡ŞåĹŻåžžäÿĂäÿłçž£çĺŃæŮűïijŇæŞ■ä¡IJçşżçż§ä
æŤ¿ç¡őçž£çĺŃçŽĎæĽ ğèąŇæăĹïijĹéĂŽåÿÿæŸŕ8MBåď ğåřŔïijĽãĂĆä¡ĘæŸŕè£ŹäÿłåĘĚå■ŸåŔłæIJĽäÿĂåřŔ
åŽäæ■ď ïijŇPythonè£ŻçĺŃä¡£çŤĺåĹřçŽĎçIJ§åőđåĘĚå■ŸåĚűåőđå¿ĹåřŔ
ïijĹæŕŤåęĆïijŇåŕźäžŐ2000äÿłçž£çĺŃæİěèőšïijŇåŔłä¡£çŤĺåĹřäžĘ70MBçŽĎçIJ§åőđåĘĚå■ŸïijŇèĂŇäÿ■æŸŕ
åęĆæđIJä¡äæŃĚå£ČèŽŽæŃ§åĘĚå■Ÿåď ğåřŔïijŇåŔŕäżěä¡£çŤĺ `threading.stack_size()` åĞ¡æŢřæİěéŹ■ä¡ŐåőČãĂĆä¿ŃåęĆïijŽ

```
import threading
threading.stack_size(65536)
```

åęĆæđIJä¡ääåŁăäÿŁè£Źæİąèŕ■åŔěåźűåĘ■æňąè£ŘèąŇåĽ■éİćçŽĎåĹŻåžž2000äÿłçž£çĺŃèŕŢéłŇïijŇ
ä¡ääijŽåŔŚçŐřPythonè£ŻçĺŃåŔłä¡£çŤĺåĹřäžĘåď ğæęĆ210MBçŽĎèŹŽæŃ§åĘĚå■ŸïijŇèĂŇçIJ§åőđåĘĚå■Ÿ
æşĺæĎŔçž£çĺŃæăĹåď ğåřŔå£ĚéążèĞşåřŚäÿž32768å■ŮèŁĆïijŇéĂŽåÿÿæŸŕçşżçż§åĘĚå■Ÿéąţåď ğåřŔïijĹ409

14.8 12.8 çőĂå■ŢçŽĎåźűèąŇçijŰçĺŃ

éŮőéćŸ

ä¡ăæIJĽäÿłçĺŃåžŔèęĄæĽ ğèąŇCPUåŕĘéŻĘåđŃåůěä¡IJïijŇä¡ăæČşèőľäżŰåĹľçŤĺåď ŽæăÿCPUçŽĎäijŸå

èğčåĘşæŰźæąĹ

`concurrent.futures` åžŞæŔŘä¿ŻäžĘäÿĂäÿł `ProcessPoolExecutor` çśżïijŇ
åŔŕèćńçŤĺæİěåIJĺäÿĂäÿłå■ŢçŃňçŽĎPythonèğčéĞŁåŹĺäÿ■æĽ ğèąŇèőąçőŮåŕĘéŻĘåđŃåĞ¡æŢřãĂĆ
äÿ■è£ĞïijŇèęĄä¡£çŤĺåőČïijŇä¡ăéęŰåĚĹèęĄæIJĽäÿĂäžŻèőąçőŮåŕĘéŻĘåđŃçŽĎäżżåŁąãĂĆ
æĹŚäžňéĂŽè£ĞäÿĂäÿłçőĂå■ŢèĂŇåőđéŹĚçŽĎä¿Ńå■ŘæİěæijŤçď žåőČãĂĆåĄĞåőŽä¡ăæIJĽäÿłApache
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£ŮæŰĞäżűåĘĚåőźçśżäijijäÿŃéİćè£Źæăůïijž

```
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 -
...
```

äÿŃéİćæŸŕäÿĂäÿłèĎŽæIJňïijŇåIJÍè£ŹäžŻæŮěå£ŮæŰĞäżűäÿ■æ§ěæĽ¿åĞžæĽĂæIJL'èő£éŮőè£Ğrobot

```
# 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)

```
    return all_robots

if __name__ == '__main__':
    robots = find_all_robots('logs')
    for ipaddr in robots:
        print(ipaddr)
```

åL’■éİćçŽĎçÍŃåžŔä¡£çŤĺäžĘéĂŽåÿÿçŽĎmap-reduceéčŐæäijæİěçijŰåĘŹãĂĆ åĞ¡æŢř `find_robots()` åIJĺäÿĂäÿłæŰĞäżűåŘ■éŽĘåŘĹäÿŁåĄŽmapæŞ■ä¡IJïijŇåźűåřĘçżŞæđIJæśĞæĂżäÿžäÿĂä äź§åřśæŸŕ `find_all_robots()` åĞ¡æŢřäÿ■çŽĎ `all_robots` éŽĘåŘĹãĂĆ çŐřåIJĺïijŇåĄĞèő¿ä¡ăæČşèęĄä£őæŤźè£ŹäÿłçÍŃåžŔèőľ’åőČä¡£çŤĺåď’ŽæäÿCPUãĂĆ å¿ĹçőĂå■ŢâĂŤâĂŤåŘłéIJĂèęĄåřĘmap()æŞ■ä¡IJæŽ£æ■ćäÿžäÿĂäÿł `concurrent.futures` åžŞäÿ■çŤ§æĹŘçŽĎçśżäijijæŞ■ä¡IJå■şåŔřãĂĆ äÿŃéİćæŸŕäÿĂäÿłçőĂå■Ţä£őæŤźçL’ĹæIJñïijŽ

```
# 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)

```
    for ipaddr in robots:
        print(ipaddr)
```

éĂŽè£Ğè£ŹäÿŀäŁőæŤźåŘŐïijŇè£ŘèąŇè£ŹäÿŀèĎŽæIJňäžğçŤ§åŘŇæăůçŽĎçżŞæđIJïijŇä¡ĘæŸŕåIJÍåŻŻ
åőđéŹĚçŽĎæĂğèČ¡äijŸåŇŰæŢĹæđIJæăźæ■ő䡿äçŽĎæIJžåŹÍCPUæŢřéĞŔçŽĎäÿ■åŘŇèĂŇäÿ■åŘŇãĂĆ

èőÍèőž

ProcessPoolExecutor çŽĎåËÿåđŇçŤÍæşŢåęĆäÿŇïijŽ

```
from concurrent.futures import ProcessPoolExecutor

with ProcessPoolExecutor() as pool:
    ...
    do work in parallel using pool
    ...
```

åĚűåŐ§çŘĘæŸŕïijŇäÿĂäÿł ProcessPoolExecutor
åĹŻåžžNäÿłçŃňçńŇçŽĎPythonèğčéĞŁåŹÍïijŇ NæŸŕçşżçż§äÿŁéÍćåŔŕçŤÍCPUçŽĎäÿłæŢřãĂĆä¡ääŔŕäžěéĂŽ
ProcessPoolExecutor(N) æÍëä£őæŤź åď ĎçŘĘåŹÍæŢřéĞŔãĂĆè£Źäÿłåď ĎçŘĘæśääijŽäÿĂçŻť è£Řèą
çĎűåŘŐåď ĎçŘĘæśäèćńåĚşéŮ■ãĂĆäÿ■è£ĞïijŇçÍŃåžŔäijŽäÿĂçŻť ç■L’å¿ĚçŻť åĹřæL’ĂæIJL’æŔŘäžď çŽĎå

èćńæŔŘäžď åĹřæśääÿ■çŽĎåůěä¡IJå£ĚéążèćńáőŽäźL’äÿžäÿĂäÿłåĞ¡æŢřãĂĆæIJL’äÿď çğ■æŰźæşŢåŐżæ
åęĆæđIJä¡ăæČşèőľäÿĂäÿłåĹŮèąÍæŐÍåŕijæĹŰäÿĂäÿł map()
æŞ■ä¡IJåźűèąŇæL’ğèąŇçŽĎèŕİïijŇåŔŕä¡£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)
```

åŔęåď ŰïijŇä¡ääŔŕäžěä¡£çŤÍ pool.submit() æÍěæL’ŃåŁÍçŽĎæŔŘäžď å■ŢäÿłäżżåŀąïijŽ

```
# Some function
def work(x):
    ...
    return result

with ProcessPoolExecutor() as pool:
    ...
    # Example of submitting work to the pool
```

(continues on next page)

```
    future_result = pool.submit(work, arg)

    # Obtaining the result (blocks until done)
    r = future_result.result()
    ...
```

åęĆæđIJä¡ăæL'ŃåŁÍæŔŘäžd'äýĂäýłäżżåŁąïijŇçżŞæđIJæŸŕäýĂäýł Future åőđä¿ÑãĂĆ èęĄèÔůåŔŰæIJĂçżĹçżŞæđIJïijŇä¡ăéIJĂèęĄěŕĈçŤĺåőČçŽĎ result() æŰźæşŢãĂĆ åőČäijŽéŸżåąđè£ŽçÍÑçŻť åĽřçŻŞæđIJèćńè£ŤåŻđæĺěãĂĆ

åęĆæđIJäý■æČşéŸżåąđïijŇä¡ăè£ŸåŔŕäžěä¡£çŤÍäýĂäýłåŻđèřČåĞ¡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)
```

åŻđèřČåĞ¡æŢřæŐěåŔŮäýĂäýł Future åőđä¿ÑïijÑèćńçŤÍæĺěèÔůåŔŰæIJĂçżĹçŽĎçżŞæđIJïijĹæŕŤåęĆ åř¡çőąåd'ĎçŘĘæśăå¿Ĺåőźæ ŸŞä¡£çŤÍïijŇåIJÍèő¿èőąåd'ğçÍŃåžŔçŽĎæŮűåĂŹè£ŸæŸŕæIJL'å¿Ĺåd'ŽéIJĂèęĄæ

- è£Źçğ■ázűèąŇåd'ĎçŘĘæŁĂæIJŕåŔłéĂĆçŤÍäžŐéĆčäžŻåŔŕäžèèćńåĹĘèğčäýžäžŠçŻÿçŃňçńŃéČÍåĹĘçŽ
- èćńæŔŘäžd'çŽĎäżżåŁąå£Ěéążæ ŸŕçőĂå■ŢåĞ¡æŢřå¡ćåijŔãĂĆåŕźäžŐæŰźæşŢãĂĄéŮ■åŇĚåŠŇåĚűäżŮ
- åĞ¡æŢřåŔĆæŢřåŠŇè£ŤåŻđåĂijå£ĚéążåĕijåőźpickleïijŇåŻääýžèęĄä¡£çŤÍåĹřè£ŻçÍŃéŮt'çŽĎéĂŽä£ąï
- èćńæŔŘäžd'çŽĎäżżåŁąåĞ¡æŢřäý■ážŤä£ÍçŢŹçŁűæĂĄæĹŰæIJL'åL'ŕä¡IJçŤÍãĂĆéŹd'äžĘæL'Şå■řæŮěå

  äýĂæŮęåŘŕåŁÍä¡ääý■èČ¡æŐğåĹűå■Řè£ŻçÍŃçŽĎäżżä¡Ţèą Ñäýžïij ŇåŽäæ■d'æIJĂåě¡ä£ÍæŇĄçőĂå■ŢåŠ

- åIJÍUnixäýŁè£ŻçÍŃæśăéĂŽè£ĞèřČçŤÍ fork() çşżçż§èřČçŤÍèćńåĹŽåżžïijŇ

åőČäijŽåĚŃéŽĘPythonèğčéĞŁåŹÍïijŇåŇĚæŃňforkæŮűçŽĎæL'ĂæIJL'çÍŃåžŔçŁűæĂĄãĂĆ èĂŇåIJÍWindowsäýŁïijŇåĚŃéŽĘèğčéĞŁåŹÍæŮűäý■äijŽåĚŃéŽĘçŁűæĂĄãĂĆ åőđéŹĚçŽĎ-forkæŞ■ä¡IJäijŽåIJÍçňňäýĂæňąèřČçŤÍ pool.map() æĹŰ pool.submit() åŘŐåŔŚçŤ§ãĂĆ

- å¡Şä¡ăæůůåŘĹä¡£çŤÍè£ŻçÍŃæśăåŠŇåd'Žçž£çÍŃçŽĎæŮűåĂŹèęĄçL'źåĹńářŔå£ČãĂĆ

  ä¡ăåžŤèŕěåIJÍåĹŻåżžäżżä¡Ţçž£çÍŃäźŃåL'■åĚĹåĹŻåżžåźűæ£Äæť'è£ŻçÍŃæśăijÍæŕŤåęĆåIJÍçÍŃåžŔåŘŕ

14.9 12.9 PythonçŽĎåĚÍåśĂéŤĄéŮőécŸ

éŮőécŸ

ä¡ăåůšçżŔåŘňèŕť'è£ĞåĚÍåśĂèğčéĞŁåŹÍéŤĄGILïijŇæŃĚå£ČåőČäijŽå¡śåŞ■åĹřåd'Žçž£çÍŃçÍŃåžŔçŽĎæ

èğčåEşæŰźæąĹ

åř¡çőąPythonåőŇåĚĺæŤŕæŇĄåď Žçž£çĺŃçijŰçĺŃïijŇ ä¡EæŸŕèğčéĞŁåŹĺçŽĎCèŕ■èĺĂåőđçŐřéČĺåĹEåIJĺ
åőđéŹĚäÿŁïijŇèğčéĞŁåŹĺèćńäÿĂäÿłåĚĺåśĂèğčéĞŁåŹĺéŤĄä£İæŁď çİĂïijŇåőČçąőä£İäżżä¡ŢæŮűåĂŹéČ¡åŔ
GILæIJĂåď ğçŽĎéŰőécŸåřśæŸŕPythonçŽĎåď Žçž£çĺŃçĺŃåžŔåźűäÿ■èČ¡åĹl’çŤĺåď ŽæäÿCPUçŽĎäijŸåŁ£
ïijĹæŕŤåęĆäÿĂäÿłä¡£çŤĺäžĘåď Žäÿłçž£çĺŃçŽĎèőąçőŮåŕĘéŻĘåđŃçĺŃåžŔåŔłäijŽåIJĺäÿĂäÿłå■ŢCPUäÿŁéĺćë

åIJĺèőĺèőžæŹőéĂŽçŽĎGILäźŃåL’■ïijŇæIJL’äÿĂçĆźèęĄåijžèřČçŽĎæŸŕGILåŔłäijŽå¡śåŞ■åĹřéĆčäžŻäÿ
åęĆæđIJä¡ăçŽĎçĺŃåžŔåď ğéČĺåĹĘåŔłäijŽæűL’åŔŁåĹřI/OïijŇæŕŤåęĆç¡ŚçżIJäžď äžŠïijŇéĆčäźĹä¡£çŤĺåď Žç
åŻääÿžåőČäžňåď ğéČĺåĹĘæŮűéŮt’éČ¡åIJĺç■L’å¿ĚãĂĆåőđéŹĚäÿŁïijŇä¡ăåőŇåĚĺåŔŕäžěæŤ¿å£ČçŽĎåĹŻäžžå
çŐřäžčæŞ■ä¡IJçşżçż§è£ŘèąŇè£ŹäźĹåď Žçž£çĺŃæšąæIJL’äżżä¡ŢåÓŃåŁŻïijŇæšąåŢěåŔŕæŃĚå£ČçŽĎãĂĆ

èĂŇåŕźäžŐä¿İèţŰCPUçŽĎçĺŃåžŔïijŇä¡ăéIJĂèęĄåijĎæÿĚæěŽæL’ğèąŇçŽĎèőąçőŮçŽĎçL’źçĆźãĂĆ
ä¿ŃåęĆïijŇäijŸåŇŰåžŢåśĆçőŮæşŢèęĄæŕŤä¡£çŤĺåď Žçž£çĺŃè£ŘèąŇå£ńå¿Ůåď ŽãĂĆ
çśżäijijçŽĎïijŇçŤśäžŐPythonæŸŕèğčéĞŁæL’ğèąŇçŽĎïijŇåęĆæđIJä¡ăåřĘéĆčäžŻæĂğèČ¡çŞűécĹäžčçăĄçğżå
éĂ§åžęäź§äijŽæŔŘå■ĞçŽĎå¿Ĺå£ńãĂĆåęĆæđIJä¡ăèęĄæŞ■ä¡IJæŢřçžĎïijŇéĆčäźĹä¡£çŤĺNumPyè£ŹæăůçŽĎ
æIJĂåŘŐïijŇä¡ăè£ŸåŕäžěèĂČèŹŚäÿŇåĚűäžŰåŔŕéĂL’ăőđçŐřæŰźæąĹïijŇæŕŤåęĆPyPyïijŇåőČéĂŽè£ĞäÿĂ
ïijĹäÿ■è£ĞåIJĺåĘŹè£ŹæIJňäźęçŽĎæŮűåĂŹåőČè£Ÿäÿ■èČ¡æŤŕæŇĄPython 3ïijL’ãĂĆ

è£ŸæIJL’äÿĂçĆźèęĄæşĺæĐŔçŽĎæŸŕïijŇçž£çĺŃäÿ■æŸŕäÿŞéŮĺçŤĺæİěäijŸåŇŰæĂğèČ¡çŽĎãĂĆ
äÿĂäÿłCPUä¿İèţŰåđŇçĺŃåžŔåŔŕèČ¡äijŽä¡£çŤĺçž£çĺŃæİěçőąçŘĘäÿĂäÿłåŻ¿å¡ćçŤĺæĹůçŢŇéİćãĂĄäÿĂäÿłç¡
è£ŹæŮűåĂŹïijŇGILäijŽäžğçŤ§äÿĂäžŻéŮőécŸïijŇåŽääÿžåęĆæđIJäÿĂäÿłçž£çĺŃéŢ£æIJ§æŇĄæIJL’GILçŽĐ
äžŇåőđäÿŁïijŇäÿĂäÿłåĘźçŽĎäÿ■åě¡çŽĎCèŕ■èĺĂæL’l’åşŢäijŽåŕijèĞt’è£ŹäÿłéŮőécŸæŻt’åŁääÿěéĞ■ïijŇ
åř¡çőąäžčçăĄçŽĎèőąçőŮéČĺåĹĘäijŽæŕŤäźŃåL’■è£ŘèąŇçŽĎæŻt’å£ńäžŻãĂĆ

èŕt’äžĘè£ŹäźĹåď ŽïijŇçŐřåIJĺæČşèŕt’çŽĎæŸŕæĹŚäžňæIJL’äÿd’çğ■ç■ŰçŢěæİèèğčåĘşGILçŽĎçijžçĆźãĂ
éęŰåĚĹïijŇåęĆæđIJä¡ăåőŇåĚĺåůěä¡IJäžŐPythonçŐŕåćČäÿ■ïijŇä¡ăåŔŕäžěä¡£çŤĺ
multiprocessing æĺąåİŮæİěåĹŻåžžäÿĂäÿłè£ŽçĺŃæśäïijŇ
åźűåČŔå■ŔåŘŇåď ĐçŘĘåŹĺäÿĂæăůçŽĎä¡£çŤĺåőČãĂĆä¿ŃåęĆïijŇåĄĞåęĆä¡ăæIJL’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£ŽçĺŃæśäïijŽ

```
# Processing pool (see below for initiazation)
pool = None

# Performs a large calculation (CPU bound)
def some_work(args):
    ...
    return result
```

(continues on next 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£Źäýłé ĂŽè£Ğä¡£çŤĺäýĂäýłæŁĂåůğåĹ'çŤĺè£ŻçÍŃæśăèğčåĘşäžĘGILçŽĎéŮőécŸãĂĆ
å¡ŞäýĂäýłçž£çÍŃæČşèęĄæL'ğèąŇCPUårĘéŻĘåđŃåůëä¡IJæŮűïijŇäijŽårĘäżżåŁąåŔŞçżŹè£ŻçÍŃæśăĂĆ
çĎűåŘŐè£ŻçÍŃæśăäijŽåIJĺåŔęåď'ŰäýĂäýłè£ŻçÍŃäý■åŘŕåŁĺäýĂäýłå■ŢçŃňçŽĎPythonèğčéĞŁåŹĺæİěåůëä¡I
å¡Şçž£çÍŃç■L'å¿ĚçżŞæđIJçŽĎæŮűåĂŹäijŽéĞŁæŤ¿GILăĂĆ åźűäýŤïijŇçŤśäžŐèőąçőŮäżżåŁąåIJĺå■ŢçŃňè
åIJĺäýĂäýłåď'Žæäýçşżçż§äýŁéİćïijŇä¡ääijŽåŔŞçŐřè£Źäýłæ ĽĂæIJŕåŔŕäžèèőľ'ä¡åå¿Ĺåě¡çŽĎåĹ'çŤĺåď'ŽCPU

åŔęåď'ŰäýĂäýłèğčåĘşGILçŽĎç■ŰçŢěæŸŕä¡£çŤĺCæL'ľ'åśŢçijŰçÍŃæŁĂæIJŕãĂĆ
äýżèęĄæĂİæČşæŸŕåřĘèőąçőŮårĘéŻĘåđŃäżżåŁąè¡ňçğżçżŹCïijŇèů§PythonçŃňçńŃïijŇåIJĺåůëä¡IJçŽĎæŮű
è£ŹåŔŕäżěéĂŽè£ĞåIJĺCäžčçăĄäÿ■æŔŠåĚëäýŃéİćè£ŹæăůçŽ ĎçL'źæőŁåőŔæİěåőŇæĹŘïijŽ

```
#include "Python.h"
...

PyObject *pyfunc(PyObject *self, PyObject *args) {
   ...
   Py_BEGIN_ALLOW_THREADS
   // Threaded C code
   ...
   Py_END_ALLOW_THREADS
   ...
}
```

åęĆæđIJä¡ăä¡£çŤĺåĚűäžŰåůěåĚůèő£éŰőCèŕ■èĺĂïijŇærŤåęĆårźäžŐCythonçŽĎctypesåžŞïijŇä¡ăäÿ■éIJĂ
ä¿ŃåęĆïijŇctypesåIJĺèřČçŤĺCæŮűäijŽèĞłåŁéĞŁæŤ¿GILăĂĆ

èőĺèőž

èőÿåď'ŽçÍŃåžŔåŚŸåIJĺéİćårźçž£çÍŃæĂğèČ¡éŮőécŸçŽĎæŮűåĂŹïijŇél'ňäýŁårśäijŽæĂłç¡łGILïijŇäžĂ
åĚűåőđè£Źæăůå■Řåď'łäÿ■åŐŽéĄŞäź§åď'łåď'l'çIJ§äžĘçĆźãĂĆ
ä¡IJäÿžäýĂäýłçIJ§åőđçŽĎä¿Ńå■ŘïijŇåIJĺåď'Žçž£çÍŃçŽĎç¡ŚçżIJçijŰçÍŃäÿ■çěđçğŸçŽĎ
stalls åŔŕèČ¡æŸŕåŽääÿžåĚűäžŰåŐ§åŻăæŕŤåęĆäýĂäýłDNSæ§ěæL'¿åźűæŮűïijŇèĂŇèů§GILæŕńæŮăăĚş
æIJĂåŘŐä¡ăçIJ§çŽĎéIJĂèęĄåĚĹåŐżæŘđæĞĆä¡äçŽĎäžčçăĄæŸŕåŘęçIJ§çŽĎèćńGILå¡śåŞ■åĹřãĂĆ
åŘŇæŮűè£ŸèęĄæŸŐçŹ¡GILåď'ğéČĺåĹĘéČ¡åžŤèŕěåŔłåĚşæşÍCPUçŽĎåď'ĎçŘĘèĂŇäÿ■æŸŕI/O.

åęĆæđIJä¡ăåĞĘåď'Ğä¡£çŤĺäýĂäýłåď'ĎçŘĘåŹÍæśäïijŇæşÍæĎŔçŽĎæŸŕè£ŹæăůåĄŽæűĽåŘŁåĹřæŢřæ■ĉ
èćńæL'ğèąŇçŽĎæŞ■ä¡IJéIJĂèęĄæŤ¿åIJĺäýĂäýłéĂŽè£Ğdefèŕ■åŘěåőŽäźĽçŽĎPythonåĞ¡æŢřäý■ïijŇäÿ■èČ¡

åźűäÿŤåĞ¡æŢřåŔĆæŢřåŠŇè£ŤåŻđåĂijå£Ěéążèę Ąå Ěijåőźpickleã ĂĆ
åŘŇæăůïijŇèęĄæĽğèąŇçŽĎäżżåŁąéĞŔå£Ěéążèűşåď §åď ğäžěåijěèąěéćİåď ŰçŽĎéĂŽä£ąåijĂéŤĂãĂĆ

åŔęåď ŰäÿĂäÿłéŽ¿çĆźæŸŕå¡Şæůůå ŘĹä¡£çŤĺçž£çĺŃåŠŇè£ŻçĺŃæśăçŽĎæŮűůåĂŹäijŽèőľä¡ăå¿Łåď ť çŰi
åęĆæđIJä¡ăèęĄåŘŇæŮűä¡£çŤĺäÿď èĂĚïijŇæIJĂåě¡åIJĺçĺŃåžŔåŘŕåŁĺæŮűïijŇåĹŻåżžäżä¡Ţçž£çĺŃäźŃåĽ■å
çĎűåŘŐçž£çĺŃä¡£çŤĺåŘŇæăůçŽĎè£ŻçĺŃæśæİěè£ŻèąŇåőĆäžńçŽĎèőąçőŮåŕĘéŻĘąđŃåůěä¡IJãĂĆ

CæĽľåśŢæIJĂéĞ■èęĄçŽĎçĽźå¿ĄæŸŕåőĆäžňåŠŇPythonèğčéĞŁåŹĺæŸŕä£İæŇĄçŃňçńŃçŽĎãĂĆ
äź§åřśæŸŕèŕťïijŇåęĆæđIJä¡ăåĞĘđ ĞåřĘPythonäÿ■çŽĎäżżåŁąåĹĘéĚ■åĹřCäÿ■åŐżæĽğèąŇïijŇ
ä¡ăéIJĂèęĄçąőä£İĆäžčçăĄçŽĎæŞ■ä¡IJèů§Pythonä£İæŇĄçŃňçńŃïijŇ
è£ŹåřśæĎŘåŠşçİĂäÿ■èęĄä¡£çŤĺPythonæŢřæ■őçżŞæđĎäžěåŘŁäÿ■èęĄěřČçŤĺPythonçŽĎC
APIãĂĆ åŔęđ ŰäÿĂäÿłåřśæŸŕä¡ăèęĄçąőä£İĆæĽľåśŢæĽĂåĄŽçŽĎåůěä¡IJæŸŕèűşđ §çŽĎïijŇåĂijå¿Ůä¡ăè
äź§åřśæŸŕèŕťCæĽľåśŢæŃĚèť §ètůäžĘđ ğéĞŔçŽĎèőąçőŮäżżåŁąïijŇèĂŇäÿ■æŸŕåřŚæŢřåĞääÿłèőąçőŮãĂĆ

è£ŹäžŻèğčåĘşGILçŽĎæŰźæąĹåźűäÿ■èČ¡éĂĆçŤĺäžŐæĽĂæIJL éŰőécŸãĂĆ
ä¿ŇåęĆïijŇæ§ŘäžŻçśżåđŃçŽĎåžŢçŤĺçĺŃåžŔåęĆæđIJèćńåĹĘèğčäÿžåď Žäÿłè£ŻçĺŃåď ĎçŘĘçŽĎèŕİåźűäÿ■è
äź§äÿ■èČ¡åřĘåőČçŽĎéČĺåĹĘäžčçăĄæŤźæĹŘCèŕ■èĺĂæĽğèąŇãĂĆ
åŕźäžŐè£ŹäžŻåžŢçŤĺçĺŃåžŔïijŇä¡ăåřśèęĄèĞłåůśéIJĂæśĆèğčåĘşæŰźæąĹäžĘ
ïijĹæŕŤåęĆåď Žè£ŻçĺŃèő£éŰőåĚśäžńåĘĚå■ŸåŇžïijŇåď Žèğčæđŕåźĺè£ŘèąŇäžŐåŘŇäÿĂäÿłè£ŻçĺŃç■L ïijL
æĹŰèĂĚïijŇä¡ăè£ŸåŕäžěèĂČèŹŠäÿŃåĚűäžŰçŽĎèğčéĞŁåŹĺåőđçŐřïijŇæŕŤåęĆPyPyãĂĆ

äžĘèğčæŻť åď ŽåĚşäžŐåIJĺCæĽľåśŢäÿ■éĞŁæŤ¿GILïijŇèŕůåŔĆèĂČ15.7åŠŇ15.10åŕŘèŁĆãĂĆ

14.10 12.10 åőŽäźĽäÿĂäÿłActoräżżåŁą

éŮőécŸ

ä¡ăæČşåőŽäźĽèů§actoræĺąåijŕäÿ■çśżäijijâĂIJactorsâĂİèğŠèĽšçŽĎäżżåŁą

èğčåĘşæŰźæąĹ

actoræĺąåijŔæŸŕäÿĂçğ■æIJĂåŔď èĂĄçŽĎäź§æŸŕæIJĂçőĂå■ŢçŽĎåźűèąŇåŠŇåĹĘåÿČåijŔèőąçőŮèğčä
äžŇåőđäÿŁïijŇåőČåď ľçŤ§çŽĎçőĂå■ŢæĂğæŸŕåőČåęĆæ■ď åŔŮæňćè£ŐçŽĎéĞ■èęĄåŐ§åŽääźŇäÿĂãĂĆ
çőĂå■ŢæİèèőšïijŇäÿĂäÿłactoråřśæŸŕäÿĂäÿłåźűåŔŚæĽğèąŇçŽĎäżżåŁąïijŇåŔłæŸŕçőĂå■ŢçŽĎæĽğèąŇåŔ
åŞ■åžŤè£ŹäžŻæűĹæĄŕæŮűűïijŇåőČåŔŕèČ¡è£ŸäijŽçżŹåĚűäżŰactoråŔŚéĂĄæŻť è£ŻäÿĂæ■ěçŽĎæűĹæĄŕãĂ
actoräźŇéŮť çŽĎéĂŽä£ąæŸŕå■ŢåŘŚåŠŇåijĆæ■ěçŽĎãĂĆåŽäæ■ď ïijŇæűĹæĄŕåŘŚéĂĄèĂĚäÿ■ç§ěéĄŞæűĹ
äź§äÿ■äijŽæŐěæŤůåĹřäÿĂäÿłæűĹæĄŕåůšècńåď ĎçŘĘçŽĎåŽđåžŤæĹŰéĂŽç§ěãĂĆ

çżŞåŘĹä¡£çŤĺäÿĂäÿłçž£çĺŃåŠŇäÿĂäÿłéŸ§åĹŮåŔŕäžěå¿Łåőźæ ŸŞçŽĎåőŽäźĽactorïijŇä¿ŇåęĆïijŽ

```
from queue import Queue
from threading import Thread, Event

# Sentinel used for shutdown
class ActorExit(Exception):
    pass

class Actor:
```

(continues on next 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)

```
        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¡ää¡£çŤÍactoråőđä¿ŃçŽĎ send() æŰźæşŢåŔŠéĂĄæűĹæĄŕçżŹåőČäžñãĂĆ åĚűæIJžåĹűæŸŕïijŇè£ŹäÿłæŰźæşŢäijŽåřĘæűĹæĄŕæŤ¿åĚěäÿĂäÿçĎűåŘŐåřĘåĚűè¡ñäžd'çżŹåd'ĎçŘĘèćńæŐěåŔŮæűĹæĄŕçŽĎäÿĂäÿłåĘĚéČÍçž£çÍŃãĂĆ close() æŰźæşŢéĂŽè£ĞåIJÍéŸ§åĹŮäÿ■æŤ¿åĚěäÿĂäÿłçL'źæőŁçŽĎåŞÍåĚţåĂijïijĹActorExitïijL'æİěåĚşéŰçŤÍæĹůåŔŕäžěéĂŽè£Ğçżğæ L'£ActoråźűåőŽäźL'åőđçŐřèĞłåůšåd'ĎçŘĘéĂżè¿Śrun()æŰźæşŢæİěåőŽäźL'æŰřActorExit åijĆåÿÿçŽĎä¡£çŤÍåřśæŸŕçŤÍæĹůèĞłåőŽäźL'äžčçăĄåŔŕäžěåIJÍéIJĂèęĄçŽĎæŮűåĂŹæİěæ■ŢèĆïijĹåijĆåÿÿèćńget()æŰźæşŢæŁŻåĞžåźűäijăæŠ■åĞžåŐżïijL'ãĂĆ

åęĆædIJä¡ăæŤ¿åő¡åŕźäžŐåŘŇæ■ěåŠŇåijĆæ■ěæűĹæĄŕåŔŠéĂĄçŽĎèęĄæśĆïijŇ çśżactoråŕźèśąè£ŸåŔŕäžěéĂŽè£ĞçŤ§æĹŘåŹÍæİěçőĂåŇŰåőŽäźL'ãĂĆä¿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

actoræÍąåijŔçŽĎé■ĚåŁŻåřśåIJÍäžŐåőČçŽĎçőĂå■ŢæĂğãĂĆ
åőđéŹĚäÿŁïijÑè£ŹéĞÑäżĚäżĚåŔłæIJL'äÿĂäÿłæäÿå£ČæŞ■ä¡IJ `send()` .
çŤŽèĞşïijÑåŕźäžŐåIJÍå§žäžŐactorçşżçż§äÿ■çŽĎâĂIJæűĹæĄŕâĂİçŽĎæşŻåŇŰæęĆå£ţåŔŕäżěåůšåď'Žçğ■æŰ
ä¿ŇåęĆïijÑä¡ääŔŕäżěäżěåĚČçżĎå¡ćåijŔäijäéĂŠæăĞç■¿æűĹæĄŕïijÑèől'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¡IJäÿžåŔęåď'ŰäÿĂäÿłä¿Ńå■ŘïijÑäÿŃéİćçŽĎactoråĚĄèőÿåIJÍäÿĂäÿłåůěä¡IJèĂĚäÿ■è£ŘèąÑäżżæĎŔçŽ
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)

```
    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())
```

æIJĂåŘŐïijÑâĂIJåŔŚéĂĄâĂİäÿĂäÿłäżżåŁąæűĹæĄŕçŽĎæęĆå£ţåŔŕäżěèćńæL'l'åśŢåĹřåď'Žè£ŻçÍÑçŤŽ
ä¿ÑåęĆïijÑäÿĂäÿłçśżactorårźèśąçŽĎ send() æŰźæşŢåŔŕäżěèćńçijŰçÍÑèől'åőČèČ¡åIJÍäÿĂäÿłåëŮæŐëå■Ů
æĹŰéĂŽè£Ğæ§ŘäžŻæűĹæĄŕäÿ■éŮt'äżűïijĹæŕŤåęĆAMQPãĂĄZMQç■L'ïijL'æİěåŔŚéĂĄãĂĆ

14.11 12.11 åőđçŐřæűĹæĄŕåŔŚåÿČ/èőćéŸĚæÍąåđŃ

éŮőécŸ

ä¡ăæIJL'äÿĂäÿłå§žäžŐçž£çÍÑéĂŽä£ąçŽĎçÍÑåžŔïijÑæČşèől'åőČäżňåőđçŐřåŔŚåÿČ/èőćéŸĚæÍąåijŔçŽİ

èğčåĘşæŰźæąĹ

èęĄåőđçŐřåŔŚåÿČ/èőćéŸĚçŽĎæűĹæĄŕéĂŽä£ąæÍąåijŔïijÑ
ä¡ăéĂŽåÿÿèęĄåijŢåĚëäÿĂäÿłå■ŢçÑňçŽĎâĂIJäžď'æ■ćæIJžâĂİæĹŰâĂIJç¡ŚåĚşâĂİåŕźèśąä¡IJäÿžæL'ĂæIJL'æ
äź§åřśæŸŕèŕt'ïijÑäÿ■çŻt'æŐëåřĘæűĹæĄŕäżŐäÿĂäÿłäżżåŁąåŔŚéĂĄåĹřåŔęäÿĂäÿłïijÑèĂŇæŸŕåřĘåĚűåŔŚé
çĎűåŘŐçŤśäžď'æ■ćæIJžåřĘåőČåŔŚéĂĄçżŹäÿĂäÿłæĹŰåď'ŽäÿłèćńåĕşèĄŤäżżåŁąãĂĆäÿÑéİćæŸŕäÿĂäÿłéİđ

```
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)

```
# Return the Exchange instance associated with a given name
def get_exchange(name):
    return _exchanges[name]
```

äÿĂäÿłäžď'æ■ćæIJžåřśæŸŕäÿĂäÿłæŹőéĂŽåŕźèśąïijŇèt'§èt'čçžt'æŁď'äÿĂäÿłæt'żèůČçŽĎèőćéŸĚèĂĚéŻł
æŕŔäÿłäžď'æ■ćæIJžéĂŽè£ĞäÿĂäÿłåŘ■çğřåőŽä¡■ïijŇget_exchange()
éĂŽè£ĞçżŹåőŽäÿĂäÿłåŘ■çğřè£ŤåŻđçŻÿåžŤçŽĎ Exchange åőđä¿ŇãĂĆ

äÿŃéİćæŸŕäÿĂäÿłçőĂå■Ţä¿Ňå■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)
```

åř¡çőąåŕźäžŐè£ŹäÿłéŮőécŸæIJL'å¿Ĺåď'ŽçŽĎåŔŸçğ■ïijŇäÿ■è£ĞäÿĞåŔŸäÿ■çężåĚűåőŮãĂĆ
æűĹæĄŕäijŽècńåŔŚéĂĄçżŹäÿĂäÿłäžď'æ■ćæIJžïijŇçĎűåŘŐäžď'æ■ćæIJžäijŽåřĘåőČäžňåŔŚéĂĄçżŹècńçżŚå

èőĺèőž

éĂŽè£ĞéŸ§åĹŮåŔŚéĂĄæűĹæĄŕçŽĎäżżåŁąæĹŰçž£çÍŇçŽĎæĺąåijŔå¿Ĺåőźæ
äÿ■è£ĞïijŇä¡£çŤĺåŔŚåÿČ/èőćéŸĚæĺąåijŔçŽĎåě¡åď'ĐæŻť'åŁäæŸŐæŸ¿ãĂĆ

éęŰåĚĹïijŇä¡£çŤĺäÿĂäÿłäžď'æ■ćæIJžåŔŕäžěçőĂåŇŰåď'ğéČĺåĹĘæűL'åŔŁåĹřçž£çÍŇéĂŽä£ąçŽĎåůěä¡I
æŮäéIJĂåŐżåĘŹéĂŽè£Ğåď'Žè£ŻçÍŇæĺąåİŮæİěæŞ■ä¡IJåď'Žäÿłçž£çÍŇïijŇä¡ääŔłéIJĂèęĄä¡£çŤĺè£Źäÿłäžď'æ
æ§Řçğ■çÍŇåžęäÿŁïijŇè£Źäÿłåŕśèů§æŮěå£ŮæĺąåİŮçŽĎåůěä¡IJåŐ§çŘĘçśžäijijãĂĆ
åőđéŹĚäÿŁïijŇåőČåŔŕäžěè¡żæİ¿çŽĎèğčèĂęçÍŇåžŔäÿ■åď'ŽäÿłäżżåŀąãĂĆ

åĚűæňąïijŇäžď æ■ćæIJžåź£æŠ■æűĹæĄŕçżŹåď'Žäÿłèőćé ŸĚèĂĚçŽĎèČ¡åŁŻåÿęæİěäžĘäÿĂäÿłåĚĺæŰřçŽ
ä¿ŇåęĆïijŇä¡ääŔŕäżěä¡£çŤĺåď'Žäżżå Łąçşżçż§ãĂĄåź£æŠ■æĹŰæĽ'ĞåĞžãĂĆ
ä¡äè£ŸåŔŕäżěéĂŽè£ĞäżěæŹőéĂŽèőćé ŸĚèĂĚèžńäż¡çżŚåőŽæİěæđĎåżžèřČèŕŢåŠŇèŕŁæŰ■åůěåĚůãĂĆ
ä¿ŇåęĆïijŇäÿŇéİćæŸŕäÿĂäÿłçőĂå■ŢçŽĎèŕŁæŰ■çśžïijŇåŔŕäżěæŸ¿çď'žèćńåŔŚéĂĄçŽĎæűĹæĄŕïijŽ

```
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■èęĄçL'źçĆźæŸŕåőČèČ¡åĚijåőźåď'Žäÿłâ ĂIJtask-likeâĂİåŕźèśąãĂĆ ä¿ŇåęĆïijŇæűĹæĄŕæŐěåŔŮèĂĚåŔŕäżěæŸŕactorïijĹ12.10åřŔèŁĆäżŃçż■ïijL'ãĂĄå■ŔçĺŇ `send()` æŰźæşŢçŽĎäÿIJèě£ãĂĆ

åĚşäžŐäžď æ■ćæIJžçŽĎäÿĂäÿłåŔŕèČ¡éŮőécŸæŸŕåŕźäžŐèőćéŸĚèĂĚçŽĎæ■čçąőçżŚåőŽåŠŇèğčçżŚãĂ
äÿžäžĘæ■čçąőçŽĎçőąçŘĘèţĎæžŘïijŇæŕŘäÿĂäÿłçżŚåőŽçŽĎèőćéŸĚèĂĚå£ĚéążæIJĂçżĹèęĄèğčçżŚãĂĆ
åIJĺäžčçăĄäÿ■éĂŽåÿÿäijŽæŸŕåČŔäÿŇéİćè£ŹæăůçŽĎæĺąåijŔïijŽ

```
exc = get_exchange('name')
exc.attach(some_task)
try:
    ...
finally:
    exc.detach(some_task)
```

æ§Řçğ■æĎŔäźĽäÿŁïijŇè£ŹäÿłåŠŇä¡£çŤĺæŰĞäžűãĂĄéŤĄåŠŇçśžäijijåŕźèśąå¿ĹåČŔãĂĆ
éĂŽåÿÿå¿Ĺåőźæ ŸŞäijŽå£ŸèőřæIJĂåŘŐçŽĎ `detach()` æ■ěéłď'ãĂĆ
äÿžäžĘçőĂåŇŰè£ŹäÿłïijŇä¡ääŔŕäżěèĂČèŹŚä¡£çŤĺäÿŁäÿŇæŰĞçőąçŘĘåŹĺå■ŔèőőãĂĆ
ä¿ŇåęĆïijŇåIJĺäžď æ■ćæIJžåŕźèśąäÿŁåćđåŁääÿĂäÿł `subscribe()` æŰźæşŢïijŇåęĆäÿŇïijŽ

```
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)

(çż■äÿŁéąţ)

```
    @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£ŸåžŤèŕěæşĺæĎŔçŽĎæŸŕåĚşäžŐäžďæ■ćæIJžçŽĎæĂİæČşæIJL’å¿Ĺåď’Žçğ■çŽĎæL’l’åśŢåőđ
ä¿ŃåęĆïijŇäžď’æ■ćæIJžåŘŕäžěåőđçŐřäÿĂæŢť’äÿłæűĹæĄŕéĂŽéĄŞéŻĘåŘĹæĹŰæŔŘä¿Żäžď’æ■ćæIJžåŘ■çğ
äžď’æ■ćæIJžè£ŸåŔŕäžěèćńæL’l’åśŢåĹŕåĹĘåÿČåijŔèőąçőŮçÍŃåžŔäÿ■ïijĹæŕŤåęĆïijŇářĘæűĹæĄŕèůŕçŤśåĹřä

14.12 12.12 ä¡£çŤĺçŤ§æĹŘåŹĺäžčæŻ£çž£çÍŃ

éŮőécŸ

ä¡ăæČşä¡£çŤĺçŤ§æĹŘåŹĺïijĹå■ŔçÍŃïijL’æŻ£äžčçşżçż§çž£çÍŃæİěåőđçŐřåźűåŔŚãĂĆè£ŹäÿłæIJL’æŮűå

èğčåĘşæŰźæąĹ

èęĄä¡£çŤĺçŤ§æĹŘåŹĺåőđçŐřèĞłåůśçŽĎåźűåŔŚïijŇä¡ăéęŰåĚĹèęĄåŕźçŤ§æĹŘåŹĺåĞ¡æŢřåŠŇ
`yield` èŕ■åŘěæIJL’æůśåĹżçŘĘèğčãĂĆ `yield` èŕ■åŘěäijŽèőľäÿĂäÿłçŤ§æĹŘåŹĺæŇĆèţůåőČçŽĎæL’ğèąŇïi
åřĘçŤ§æĹŘåŹĺå¡ŞåĄŽæ§Řçğ■âĂIJäżżåŁąâĂİåźűä¡£çŤĺäżżåŁąå■Ŕä¡IJåĹĞæ■ćæİěæŻ£æ■ćåőČäžńçŽĎæL’ğ

èęĄæijŤçď žè£Źçğ■æĂİæČşïijŇèĂČèŹŚäÿŃéİćäÿď äÿłä¡£çŤÍçőĂå■ŢçŽĎ `yield` èŕ■åŔěçŽĎçŤ§æĹŘåŹĺåĞ¡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£ŹäžŽåĞ¡æŢřåIJĺåĘĚéČĺä¡£çŤÍyieldèŕ■åŔëïijŇäÿŃéİćæŸŕäÿĂäÿłåőđçŐřäžEçőĂå■ŢäżżåŁąèřČåžęåŹĺ

```
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)

```
sched.new_task(countup(15))
sched.run()
```

TaskScheduler çśžåIJÍäÿĂäÿłå¿łçŐŕäÿ■è£ŘèąŇçŢ§æĹŘåŹĺéŻĘåŘĹâĂŤâĂŤæŕŔäÿłéČ¡è£ŘèąŇåĹřçè£ŘèąŇè£ŹäÿłäĄŃå■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§âĂİçŽĎæIJĂåřŔæăÿçŢ§æĹŘåŹĺåĞ¡æŢřåřśæŸŕèőd'äÿžïijŇèĂŇyieldèŕ■åŘěæŸŕäżżåŁąæŇĆèţůçŽĎä£ąåŔůãĂĆèřČåžęåŹĺå¿łçŐŕæčĂæ§ěäżżåŁąåĹŮèąĺçŻt'åĹřæšąæIJL'äżżåŁąèęĄæL'ğèąŇäÿžæ■ćãĂĆ

åőđéŹĚäÿŁïijŇä¡ääŔŕèČ¡æČşèęĄä¡£çŢĺçŢ§æĹŘåŹĺæİěåőđçŐřçőĂå■ŢçŽĎåźűåŔŚãĂĆéĆčäźĹïijŇåIJĺåőđçŐřactoræĹŰç¡ŚçżIJæIJ■åŁąåŹĺçŽĎæŮűåĂŹä¡ääŔŕäžěä¡£çŢĺçŢ§æĹŘåŹĺæİěæŻ£äżčçž£ç

äÿŃéİćçŽĎäżčçăĄæijŢçd'žäžĘä¡£çŢĺçŢ§æĹŘåŹĺæİěåőđçŐřäÿĂäÿłäÿ■ä¿İèţŰçž£çĺŃçŽĎactorïijŽ

```
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)

```
    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()
```

åőŇåĚĺåijĎæĞĆè£ŹæőţäżčçăĄéIJĂèęĄæŻťæůśåĚěçŽĎå■ęäźăïijŇä¡ĘæŸŕåĚşéŤőçĆźåIJĺäžŐæŤűéŻĘæ
æIJňèťĺäÿŁïijŇèřČåžęåŹĺåIJĺæIJL'éIJĂèęĄåŔŠéĂĄçŽĎæűĹæĄŕæŮűäijŽäÿĂçŻťè£ŘèąŇçİĂãĂĆ
èőąæŢřçŤ§æĹŘåŹĺäijŽçżŹèĞłåůśåŔŠéĂĄæűĹæĄŕåźűåIJĺäÿĂäÿłéĂŠå¡Šå¿łçŐŕäÿ■çżŞæİ§ãĂĆ

äÿŃéİćæŸŕäÿĂäÿłæŻťåŁăéńŸçžğçŽĎä¿Ńå■ŘïijŇæijŤçď žäžĘä¡£çŤĺçŤ§æĹŘåŹĺæİěåőđçŐŕäÿĂäÿłåźűå

```
from collections import deque
from select import select
```

(continues on next 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)

```
    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)

```
# 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)

```
                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£ŹæőţäžčçăæIJL'çĆźåď■æİĆãĂĆäÿ■è£ĞïijŇåőČåőđçŐřäžĘäÿĂäÿłářŔåđŇçŽĎæŞ■ä¡IJçşżçż§ãĂĆ æIJL'äÿĂäÿłářšçżłçŽĎäżżåŁąéŸ§åĹŮïijŇåźűäÿŤè£ŸæIJL'åŽăI/OäijŚçIJăçŽĎäżżåŁąç■L'å¿ĚåŇžå§§ãĂĆ è£ŸæIJL'å¿Ĺåď'ŽèřČåžęåŹĺèť§èť čåIJĺářšçżłéŸ§åĹŮåŠŇI/Oç■L'å¿ĚåŇžå§§äźŇéŮť'çğżåŁĺäżżåŁąãĂĆ

èőĺèőž

åIJĺæđĎåżžå§žäžŐçŤ§æĹŘåŹĺçŽĎåźűåŔŚæąĘæđűæŮűïijŇéĂŽåÿÿäijŽä¡£çŤĺæŻť'åÿÿèğĄçŽĎyieldå¡ćå

```
def some_generator():
    ...
    result = yield data
    ...
```

ä¡£çŤĺè£Źçğ■å¡ćåijŔçŽĎyieldèŕ■åŔěçŽĎåĞ¡æŢřéĂŽåÿÿèćńçğřäÿžâĂIJå■ŔçĺŇâĂİãĂĆ éĂŽè£ĞěřČåžęåŹĺïijŇyieldèŕ■åŔěåIJĺäÿĂäÿłå¿łçŐŕäÿ■èćńåď'ĎçŘĘïijŇåęĆäÿŇïijŽ

```
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¿ŚçĺĺåżőæIJL'çĆźåď■æİĆãĂĆäÿ■è£ĞïijŇèćńäijăçżŹ `send()` çŽĎåĂijåőŽäźL'äžĘåIJĺyieldèŕ■åŔěéĘŠæİěæŮűçŽĎè£ŤåŻđåĂijãĂĆ åŻăæ■d'ïijŇåęĆæđIJäÿĂäÿłyieldåĞĘåď'ĞåIJĺářźäźŇåL'■yield-æŢřæ■őçŽĎåŽđåžŤäÿ■è£ŤåŽđçżŞæđIJæŮűïijŇäijŽåIJĺäÿŇäÿĂæňą `send()` æŞ■ä¡IJè£ŤåŽđãĂĆ åęĆæđIJäÿĂäÿłçŤ§æĹŘåŹĺåĞ¡æŢřåĹŽåijĂåğŇè£ŘèąŇïijŇåŔŚéĂĄäÿĂäÿłNoneåĂijäij

éŹď'äžĘåŔŚéĂĄåĂijåď'ŰïijŇè£ŸåŕřäžěåIJĺäÿĂäÿłçŤ§æĹŘåŹĺäÿŁéİćæL'ğèąŇäÿĂäÿł `close()` æŰźæşţãĂĆ åőČäijŽářijèĞť'åIJĺæL'ğèąŇyieldèŕ■åŔěæŮűæŁŻåĞžäÿĂäÿł `GeneratorExit` åijĆåÿÿïijŇäżŐèĂŇçżĹæ■ćæL'ğèąŇãĂĆ

åęĆæđIJè£ŻäÿĂæ■ěèő¿èőąïijÑäÿĂäÿłçŢ§æĹŘåŹĺåŔŕäžěæ■ŢèŐůè£ŹäÿłåijĆåÿÿåźűæL'ğèąŇæÿĚçŘĘæŞ■ä¡Ł
åŘŇæăůè£ŸåŔŕäžěä¡£çŤĺçŢ§æĹŘåŹĺçŽĎ `throw()` æŰźæşŢåIJÍyield-
èŕ■åŘěæL'ğèąŇæŮűçŢ§æĹŘäÿĂäÿłäżżæĎŔçŽĎæL'ğèąŇæŇĞäžd'ãĂĆ
äÿĂäÿłäżżåŁąèřČåžęåŹĺåŔŕåĹ'çŤĺåőČæİěåIJÍè£ŘèąŇçŽĎçŢ§æĹŘåŹĺäÿ■åd'ĎçŘĘéŤŹèŕŕãĂĆ

æIJĂåŘŐäÿĂäÿłä¿Ńå■Řäÿ■ä¡£çŤĺçŽĎ `yield from` èŕ■åŘèćńçŤĺæİěåőđçŐřå■ŔçÍŃïijÑåŔŕäžěèćńåĚ
æIJñèt'ĺäÿŁåřśæŸŕårĘæŐğåĹűæİČéĂŔæŸŐçŽĎäijàè¿ŞçžŹæŰřçŽĎåĞ¡æŢřãĂĆ
äÿ■åČŔæŹőéĂŽçŽĎçŢ§æĹŘåŹĺïijÑäÿĂäÿłä¡£çŤĺ `yield from`
èćńèřČçŤĺçŽĎåĞ¡æŢřåŔŕäžěè£ŤåŻđäÿĂäÿłä¡IJäÿž `yield from`
èŕ■åŘěçżŞæđIJçŽĎåĂijãĂĆ åĚşäžŐ `yield from` çŽĎæŻt'åd'Žä£ąæĄŕåŔŕäžěåIJÍ PEP
380 äÿ■æL'¿åĹřãĂĆ

æIJĂåŘŐïijÑåęĆæđIJä¡£çŤĺçŢ§æĹŘåŹĺçijŰçÍŃïijÑèęĄæŔŘéĘŠä¡ăçŽĎæŸŕåőČè£ŸæŸŕæIJL'å¿Ĺåd'Žç
çL'źåĹńæŸŕïijÑä¡ăå¿Ůäÿ■åĹřäžżä¡Ţçž£çÍŃåŔŕäžěæŔŘä¿ŻçŽĎåě¡åd'ĎãĂĆä¿ŃåęĆïijÑåęĆæđIJä¡ăæL'ğèąŇ
åőČäijŽářĘæŢt'äÿłäżżåŁąæŇĆèţůç§ěéĄŞæŞ■ä¡IJåőŇæĹŘãĂĆäÿžäžEèğčåĘşè£ŹäÿłéŮőécŸïijÑ
ä¡ăåŔłèČ¡éĂL'æŇl'ářĘæŞ■ä¡IJåğŤæt'¿çżŹåŘęåd'ŰäÿĂäÿłåŔŕäžěçŃñçńÑè£ŘèąŇçŽĎçž£çÍŃæĹŰè£ŻçÍŃãĂ
åŘęåd'ŰäÿĂäÿłéŹŘåĹűæŸŕåd'ğéČĺåĹĘPythonåžŞåźűäÿ■èČ¡å¿Ĺåě¡çŽĎåĚijåőźå§žäžŐçŢ§æĹŘåŹĺçŽĎçž£çĺ
åęĆæđIJä¡ăéĂL'æŇl'è£ŹäÿłæŰźæąĹïijÑä¡ăäijŽåŔŚçŐřä¡ăéIJĂèęĄèĞłåůśæŤźåĘŹå¿Ĺåd'ŽæăĞåĞĘåžŞåĞ¡æ
ä¡IJäÿžæIJñèŁĆæŘŘåĹřçŽĎå■ŔçÍŃåŠŇçŽÿåĚşæŁĂæIJŕçŽĎäÿĂäÿłå§žçąĂèČŇæŹřijÑåŔŕäžěæ§ěçIJŃ
PEP 342 åŠŇ âĂIJå■ŔçÍŃåŠŇåźűåŔŚçŽĎäÿĂéŮĺæIJL'èűčèŕ¿çÍŃâĂİ

PEP 3156 åŘŇæăůæIJL'äÿĂäÿłåĚşäžŐä¡£çŤĺå■ŔçÍŃçŽĎåijĆæ■ěI/OæĺąåđŃãĂĆ
çL'źåĹńçŽĎïijÑä¡ăäÿ■åŔŕèČ¡èĞłåůśåŐžåőđçŐřäÿĂäÿłåžŢåśĆçŽĎå■ŔçÍŃèřČåžęåŹĺãĂĆ
äÿ■è£ĞïijÑåĚşäžŐå■ŔçÍŃçŽĎæĂİæČşæŸŕå¿Ĺåd'ŽæţĄèąŇåžŞçŽĎå§žçąĂïijÑ åŇĚæŃñ
gevent, greenlet, Stackless Python äžěåŘŁåĚűäžŰçśžäijijåůěçÍŃãĂĆ

14.13 12.13 åd'Žäÿłçž£çÍŃéŸ§åĹÙè¡őèŕć

éŮőécŸ

ä¡ăæIJL'äÿĂäÿłçž£çÍŃéŸ§åĹÙéŻĘåŘĹïijÑæČşäÿžåĹřæİěçŽĎåĚČçt'ăè¡őèŕćåőČäżñïijÑ
åŕśèů§ä¡ăäÿžäÿĂäÿłåőćæĹůçńŕèřůæśĆåŐżè¡őèŕćäÿĂäÿłç¡ŚçžIJè£đæŐěéŻĘåŘĹçŽĎæŰźåijŔäÿĂæăůãĂĆ

èğčåĘşæŰźæąĹ

åŕźäžŐè¡őèŕćéŮőécŸçŽĎäÿĂäÿłåÿÿèğĄèğčåĘşæŰźæąĹäÿ■æIJL'äÿłå¿ĹåřŚæIJL'äžžç§ěéĄŞçŽĎæŁĂåů
æIJñèt'ĺäÿŁèőšåĚűæĂİæČşåŕśæŸŕïijŽåŕźäžŐæŕŔäÿłä¡ăæČşèęĄè¡őèŕćçŽĎéŸ§åĹŮïijÑä¡ăäĹŻåžžäÿĂåŕźè£đa
çĎűåŘŐä¡ăäIJĺåĚűäÿ■äÿĂäÿłåěŮæŐěå■ŮäÿŁéĺćçijŰåĘŹäžčçăĄæİěæăĞèŕĘå■ŸåIJÍçŽĎæŢřæ■őïijÑ
åŘęåd'ŰäÿĂäÿłåěŮæŐěå■ŮèćńäijăçžŹ `select()` æĹŰçśžäijijçŽĎäÿĂäÿłè¡őèŕćæŢřæ■őåĹřè¿¿çŽĎåĞ¡æŢř

```
import queue
import socket
import os

class PollableQueue(queue.Queue):
    def __init__(self):
        super().__init__()
```

(continues on next 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■çğżéŹd'äÿĂäÿłåĚČçt'ăæŮűäijŽäżŐåŔęåd'ŰäÿĂäÿłåěŮæŐěå■Ůäÿ■

`fileno()` æŰźæşŢä¡£çŤÍäÿĂäÿłåĞ¡æŢřæŕŤåęĆ `select()` æİěèől'è£ŹäÿłéŸ§åĹŮåŔŕäżěèćńè¡őèŕćãĂĆ åőČäżĚäżĚåŘłæŸŕæŽťéIJšäžĘåžŢåśĆèćń `get()` åĞ¡æŢřä¡£çŤÍłåĹřçŽĎsocketçŽĎæŰĞäżűæŔŔè£řçňęèĂŇåůšãĂĆ

äÿŃéİćæŸŕäÿĂäÿłä¿Ńå■ŘïijŇåőŽäźL'äžĘäÿĂäÿłäÿžåĹřæİěçŽĎåĚČçt'äçŽŚæŐğåd'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)

```
            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¡ăäijŽåŔŞçŐřè£Źäÿłæű Łèť źèĂĚäijŽæŐěåŔŮåŁřæL’ĂæIJL’çŽĎèćń

èőĺèőž

åŕźäžŐè¡õèŕćéİđçśżæŰĞäżűåŕźèśąïijŇæŕŤåęĆéŸ§åĹŮéĂŽåÿÿéČ¡æŸŕæŕŤè¿ČæčŸæL’ŃçŽĎéŮőécŸãĂ
ä¿ŃåęĆïijŇåęĆæđIJä¡ääÿ■ä¡£çŤĺäÿŁéİćçŽĎåěŮæŐěå■ŮæŁĂæIJŕïijŇ
ä¡ăåŤŕäÿĂçŽĎéĂL’æŃl’åřśæŸŕçijŰåĘŹäżčçăĄæİěå¿łçŐŕéĄ■åŐĘè£ŹäžŻéŸ§åĹŮåźűä¡£çŤĺäÿĂäÿłáőŽæŮűå

```
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£Źæăůå ĄŽåĚűåőđäÿ■åŘĹçŘĘïijŇè£ŸäijŽåijŢåĚěåĚűäżŰçŽĎæĂğèČ¡éŮőécŸãĂĆ
ä¿ŃåęĆïijŇåęĆæđIJæŰřçŽĎæŢřæ■õèćńåŁăåĚěåĹřäÿĂäÿłéŸ§åĹŮäÿ■ïijŇèĞşåřŚèęĄèŁś10æńçğŠæL’■èČ¡è
åęĆæđIJä¡ääźŃåL’■çŽĎè¡õèŕćè£ŸèęĄåŐżè¡õèŕćåĚűäżŰåŕźèśąïijŇæŕŤåęĆç¡ŚçżIJåěŮæŐěå■ŮéĆčè£ŸäijŽæ
ä¿ŃåęĆïijŇåęĆæđIJä¡ăæČşåŘŇæŮűè¡õèŕćåěŮæŐěå■ŮåŠŇéŸ§åĹŮïijŇä¡ăåŘŕèČ¡èęĄåČŔäÿŃéİćè£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)

```
            handle_read(r)
        for q in queues:
            if not q.empty():
                item = q.get()
                print('Got:', item)
```

è£ŹäÿłæŰźæąĹéĂŽè£ĞåřĘéŸ§åĹŮåŠŇåěŮæŐěå■Ůç■L’åŘŇåŕźå¿ĚæİèèğčåĘşäžĘåď’ğéČĺåĹĘçŽĎéŮőe
äÿĂäÿłå■ŢçŃňçŽĎ `select()` èřČçŤĺåŔŕèćńåŘŇæŮűçŤĺæİèè¡őèŕćãĂĆ
ä¡£çŤĺèűĚæŮűæĹŰåĚűäžŰå§žäžŐæŮűéŮt’çŽĎæIJžåĹűæİěæL’ğèąŇåŠĺæIJ§æĂğæčĂæ§ěåźűæšąæIJL’å£Ěèę
çŤŽèĞşïijŇåęĆæđIJæŢřæ■őèćńåŁääĚěåĹřäÿĂäÿłéŸ§åĹŮïijŇæűĹèt’źèĂĚåĞääźŐåŔŕäžěåőđæŮűçŽĎèćńéĂ
ař¡çőąäijŽæIJL’äÿĂçĆźçĆźåžŢåśĆçŽĎI/Oæ■§èĂŮïijŇä¡£çŤĺåőČéĂŽåÿÿäijŽèŐůå¿ŮæŻt’åě¡çŽĎåŞ■åžŤæŮ

14.14 12.14 åIJÍUnixçşżçż§äÿŁéİćåŘŕåŁĺåőĹæŁd’è£ŽçÍŃ

éŮőécŸ

ä¡ăæČşçijŰåĘŹäÿĂäÿłä¡IJäÿžäÿĂäÿłåIJÍUnixæĹŰçśżUnixçşżçż§äÿŁéİćè£ŘèąŇçŽĎåőĹæŁd’è£ŽçÍŃè£Ł

èğčåĘşæŰźæąĹ

åĹŻåžžäÿĂäÿłæ■čçąőçŽĎåőĹæŁd’è£ŽçÍŃéIJĂèęĄäÿĂäÿłçš¿çąőçŽĎçşżçż§èřČçŤĺåžŘåĹŮäžěåŘŁåŕźäž
äÿŃéİćçŽĎäžčçăĄåśŢçd’žäžĘæĂŐæăůåőŽäźL’äÿĂäÿłåőĹæŁd’è£ŽçÍŃïijŇåŘŕäžěåŘŕåŁĺåŘŐå¿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)

```
    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)

```
        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£ŻçÍŃïijŇçŤÍæĹůéIJĂèęĄä¡£çŤĺåęĆäÿŃçŽĎåŚ¡äžď'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£ŻçÍŃåŘŕäžěåőŇåĚĺåIJÍåŘŐåŘřè£ŘèąŇïijŇåŻäæ■ď'è£ŹäÿłåŚ¡äžď'äijŽçńŇå■şè£ŤåŻđãĂĆäÿ■è£ĞïijŇä¡ääŘŕäžěåČŔäÿŁéİćéĆčæăůæ§ěçIJŃäÿŐåőČçŻÿåĚşçŽĎpidæŰĞäžűåŠŇæŮěå£ŮãĂĆèęĄåĄIJæ

```
bash % daemon.py stop
bash %
```

èőÍèőž

æIJňèŁĆåőŽäźL'äžĘäÿĂäÿłåĞ¡æŢř `daemonize()` ïijŇåIJÍçÍŃåžŔåŘŕåŁÍæŮűèćńèřČçŤÍä¡£å¿ŮçÍŃåžŔ `daemonize()` åĞ¡æŢřåŔłæŐěåŔŮåĚşéŤőå■ŮåŔĆæŢřïijŇè£ŹæăůçŽĎèŕÍåŘŕéĂĽ'åŔĆæŢřåIJÍèćńä¡£çŤÍæŮåőČäijŽåijžåĹűçŤÍæĹůåČŔäÿŃéİćè£ŹæăůäĄ£çŤĺåőČïijŽ

```
daemonize('daemon.pid',
          stdin='/dev/null,
          stdout='/tmp/daemon.log',
          stderr='/tmp/daemon.log')
```

èĂÑäÿ■æŸŕåČŔäÿÑéÍćè£ŹæăůåŕńçşŁäÿ■æÿĚçŽĎèřČçŤĺïijŽ

```
# Illegal. Must use keyword arguments
daemonize('daemon.pid',
          '/dev/null', '/tmp/daemon.log','/tmp/daemon.log')
```

åĹŻåžžäÿĂäÿłåőĹæŁd'è£ŽçÍÑçŽĎæ■ěéłd'çIJÑäÿŁåŐžäÿ■æŸŕå¿ĹæŸŞæĞĆïijÑä¡ĘæŸŕåd'ğä¡ŞæĂİæČ
éęŰåĚĹïijÑäÿĂäÿłåőĹæŁd'è£ŽçÍÑå£ĚéąžèęĄäżŐçĹűè£ŽçÍÑäÿ■èĎśçężãĂĆ è£ŹæŸŕçŤś
`os.fork()` æŞ■ä¡IJæİěåőÑæĹŘçŽĎïijÑåźűçńÑå■şèćńçĹűè£ŽçÍÑçżĹæ■ćãĂĆ

åIJÍå■Řè£ŽçÍÑåŔŸæĹŘå■d'åĎ£åŘŐïijÑěřČçŤÍ `os.setsid()`
åĹŻåžžäžĘäÿĂäÿłåĚÍæŰřçŽĎè£ŽçÍÑäijŽèŕİïijÑåźűèő¿ç¡őå■Řè£ŽçÍÑäÿžéęŰéćĘãĂĆ
åőČäijŽèő¿ç¡őè£Źäÿłå■Řè£ŽçÍÑäÿžæŰřçŽĎè£ŽçÍÑçżĎçŽĎéęŰéćĘïijÑåźűçąőä£İäÿ■äijŽåĘ■æIJL'æŐğåĹűç
åęĆæđIJè£ŹäžŽåŘäÿŁåŐžåd'lé■Ťåźžïij ÑåŽääÿžåőČéIJÄèęĄåřĘåőĹæŁd'è£ŽçÍÑåŘÑçżĹçńŕåĹĘçężåijÄåźű
èřČçŤÍ `os.chdir()` åŠÑ `os.umask(0)` æŤźåŔŸäžĘą¡ŞåL'■åůëä¡IJçŽőå¡Ţåźűé Ğ■ç¡őæŰĞäžűæİČéŹŘæČ
ä£őæŤźçŽőå¡ŢéĂŽåÿÿæŸŕäÿłåě¡äÿżæĎŔïijÑåŽääÿžè£Źæăůåŕŕäžëä¡£å¿ŮåőČäÿ■åĘ■åůëä¡IJåIJÍèćńåŘŕåŁĺæ

åŔęåd'ŰäÿĂäÿłèřČçŤÍ `os.fork()` åIJÍè£ŹéĞŇæŽt'åŁăçěđçğŸçĆźãĂĆ
è£ŹäÿĂæ■ëä¡£å¿ŮåőĹæŁd'è£ŽçÍÑåd'śåŐžäžĘèŐůåŔŰæŰřçŽĎæŐğåĹűçżĹçńŕçŽĎèČ¡åŁŽåźűäÿŤèől'åőČæ
ïijĹæIJñèt'ĺäÿŁïijÑèŕědaemonæŤ¿åijČäžĘåőČçŽĎäijŽèŕİéęŰéćĘä¡Őä¡■ïijÑåŽăæ■d'åĘ■äź§æšąæIJL'æİČéŹŘ
år¡çőąä¡ăåŔŕäžëå£¡çŢèè£ŹäÿĂæ■ëïijÑä¡ĘæŸŕæIJĂåě¡äÿ■èęĄè£ŹäźĹåĄŽãĂĆ

äÿĂæŮęåőĹæŁd'è£ŽçÍÑèćńæ■čçąőçŽĎåĹĘçężïijÑåőČäijŽéĞ■æŰřåĹİåğÑåÑŰæăĞåĞĘI/OæţĄæÑĞåŘ
è£ŹäÿĂéČĺåĹĘæIJL'çĆźéŽ¿æĞĆãĂĆèů§æăĞåĞĘI/OæţĄçŽÿåĚşçŽĎæŰĞäžűåŕźèśąçŽĎåijŢçŤĺåIJÍèğčéĞŁå
ïijĹsys.stdout, sys.__stdout__ç■L'ïijL'ãĂĆ äźĚäźĚçőĂå■ŢçŽĎåĚşéŮ■
`sys.stdout` åźűéĞ■æŰřæŇĞåőŽåőČæŸŕèąÑäÿ■éĂŽçŽĎïijÑ
åŽääÿžæšąåŁđæşŢç§ééĄŞåőČæŸŕåŘęåĚĺéČĺéČ¡æŸŕçŤĺçŽĎæŸŕ `sys.stdout` ãĂĆ
è£ŹéĞÑïijÑæĹŚäžňæL'ŞåijĂäžĘäÿĂäÿłå■ŢçÑñçŽĎæŰĞäžűåŕźèśąïijÑåźűèřČçŤÍ `os.dup2()` ïijÑ çŤĺåőČæİěäžčæŽ£ècń `sys.stdout` ä¡£çŤĺçŽĎæŰĞäžűæŔŘè£řçňęãĂĆ
è£Źæăůïij Ñ`sys.stdout` ä¡£çŤĺçŽĎåŐ§åğÑæŰĞäžűäijŽècńåĚşéŮ■åźűçŤśæŰřçŽĎæİèæŽ£æ■ćãĂĆ
è£ŸèęĄåijžèřČçŽĎæŸŕäžżä¡ŢçŤĺäžŐæŰĞäžűçijÚçĄæĹŰæŰĞæIJñåd'ĎçŘĘçŽĎæăĞåĞĘI/OæţĄè£ŸäijŽä

åőĹæŁd'è£ŽçÍÑçŽĎäÿĂäÿłéĂŽåÿÿåőđèůţæŸŕåIJÍäÿĂäÿłæŰĞäžűäÿ■åĘźåĚèè£ŽçÍÑIDïijÑåŕäžěèćńåĚ
`daemonize()` åĞ¡æŢřçŽĎæIJĂåŘŐéČĺåĹĘåĘźäžĘè£ŹäÿłæŰĞäžűïijÑä¡ĘæŸŕåIJÍçÍÑåžŘçżĹæ■ćæŮűåĹăć
`atexit.register()` åĞ¡æŢřæşÍåĘÑäžĘäÿĂäÿłåĞ¡æŢřåIJÍPythonèğčéĞŁåŹÍçżĹæ■ćæŮűæL'ğèąÑãĂĆ
äÿĂäÿłårźäžŐSIGTERMçŽĎä£ąåŘůåd'ĎçŘĘåŹÍçŽĎåőŽäźL'åŘÑæăúéIJĂèęĄèćńäijŸéŽĚçŽĎåĚşéŮ■ãĂĆ
ä£ąåŘůåd'ĎçŘĘåŹÍçőĂå■ŢçŽĎæŁŽåĞžäžĘ `SystemExit()` åijĆåÿÿãĂĆ
æĹŰèőÿè£ŹäÿĂæ■ěçIJÑäÿŁåŐžæšąå£ĚèęĄïijÑä¡ĘæŸŕæšąæIJL'åőČïijÑ
çżĹæ■ćä£ąåŘůäijŽä¡£å¿ŮäÿæL'ğèąÑ `atexit.register()`
æşÍåĘÑçŽĎæÿĚçŘĘæŞ■ä¡IJçŽĎæŰůåĂŹårśæİĂæŐL'äžĘèğčéĞŁåŹÍãĂĆ
äÿĂäÿłæİĂæŐL'è£ŽçÍÑçŽĎä¿Ńå■Řäžčçă ĄåŘŕäžëåIJÍçÍÑåžŘæIJĂåŘŐçŽĎ `stop`
åŚ¡äžd'çŽĎæŞ■ä¡IJäÿ■çIJÑåĹřãĂĆ

æŻt'åd'ŽåĚşäžŐçijŰåĘŹåőĹæŁd'è£ŽçÍÑçŽĎä£ąæĄŕåŘŕäžëæ§ěçIJÑãĂŁUNIX
çŐŕáćČéńŸçžğçijŰçÍÑãĂŃ, çňňäžŇçL'Ĺ by W. Richard
Stevens and Stephen A. Rago (Addison-Wesley, 2005)ãĂĆ

åř¡çőąåőČæŸŕåĚşæşÍäÿŐCèŕ■èÍĂçijŰçÍŃïijŇä¡ĘæŸŕæĽĂæIJL’çŽĎåĘĚåőźéČ¡éĂĆçŤÍäžŐPythonïijŇ
åŻăäÿžæĽĂæIJL’éIJÄèęĄçŽĎPOSIXåĞ¡æŢřéČ¡åŔŕäžěåIJÍæăĞåĞĘåžŞäÿ■æĽ¿åĹřãĂĆ

15 çňňå■ĄäÿL’çńăïijŽèĎŽæIJňçijŰçÍŃäÿŐçşżçż§çőąçŘĘ

èőÿåď’Žäžžä¡£çŤÍPythonä¡IJäÿžäÿĂäÿłshellèĎŽæIJňçŽĎæŻ£äżčijŇçŤÍæİěåőđçŐřåÿÿçŤÍçşżçż§äżżåŁą

Contents:

15.1 13.1 éĂŽè£ĞéĞ■åőŽåŘŚ/çőąéĄŞ/æŰĞäžűæŐěåŘÙè¿ŞåĚě

éŮőécŸ

ä¡ăåÿŇæIJŻä¡ăçŽĎèĎŽæIJňæŐěåŘŮäżżä¡ŢçŤÍæĹůèőď’äÿžæIJĂçőĂå■ŢçŽĎè¿ŞåĚěæŰźåijŔãĂĆåŇĚæ
éĞ■åőŽåŘŚæŰĞäžűåĹřèŕěèĎŽæIJňïijŇæĹŰåIJÍåŚ¡äżď’èąŇäÿ■äijăéĂŠäÿĂäÿłæŰĞäžűåŘ■æĹŰæŰĞäžűåŘ■

èğčåĘşæŰźæąĹ

PythonåĘĚç¡őçŽĎ `fileinput` æÍąåİÙèőľ’è£ŹäÿłåŔŸå¿ŮçőĂå■ŢãĂĆåęĆæđIJä¡ăæIJL’äÿĂäÿłäÿŇéİćè

```
#!/usr/bin/env python3
import fileinput

with fileinput.input() as f_input:
    for line in f_input:
        print(line, end='')
```

éĆčäźĹä¡ăåřśèČ¡äżěåĽ■éİćæŔŘåĹřçŽĎæĽĂæIJL’æŰźåijŔæİěäÿžæ■d’èĎŽæIJňæŔŘä¿Żè¿ŞåĚěãĂĆåĄ
`filein.py` åźűåřĘåĚűåŔŸäÿžåŔŕæĽğèąŇæŰĞäžűïijŇ éĆčäźĹä¡ăåŔŕäžěåČŔäÿŇéİćè£ŹæăůèřČçŤÍåőČïijŇ

```
$ 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()` åĹŻåžžåźűè£ŤåŻđäÿĂäÿł `FileInput` çśżçŽĎåőđä¿ŇãĂĆ
èŕěåőđä¿ŇéŹd’äžĘæŇěæIJL’äÿĂäžŻæIJL’çŤÍçŽĎåÿőåŁľæŰźæşŢåď’ŰïijŇåőČè£ŸåŔŕèćńå¡ŞåĄŽäÿĂäÿłäÿŁ
åŻăæ■d’ïijŇæŢt’åŘĹèţůæİěïijŇåęĆæđIJæĹŚäžňèęĄåĘŹäÿĂäÿłæĽŞå■řåď’Žäÿłæ
ŰĞäžűè¿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)

(çż■äÿŁéąţ)

```
...
/etc/passwd 1 ##
/etc/passwd 2 # User Database
/etc/passwd 3 #

<other output omitted>
```

éĂŽè£ĞåřĘåőČä¡IJäÿžäÿĂäÿłäÿŁäÿŃæŰĞçőąçŘĘåŹÍä¡£çŤÍïijŇåŔŕäžěçąőä£ÍåőČäÿ■åĘ■ä¡£çŤÍæŮűæÚ èĂŇäÿŤæĹŚäžňåIJÍäźŃåŘŐè£ŸæijŤçď žäžĘ `FileInput` çŽĎäÿĂäžŻæIJL'çŤÍçŽĎåÿőåŁl'æŰźæşŢæİěèŐů

15.2 13.2 çżĹæ■ćçÍŃåžŔåźűçżŹåĞžéŤŹèŕŕä£ąææĄŕ

éŮőécŸ

ä¡ăæČşåŘŚæăĞåĞĘéŤŹèŕŕæL'Şå■řäÿĂæİąæűĹæĄŕåźűè£ŤåŻđæ§ŘäÿłéİđéŻűçŁűæĂAçăĄæİěçżĹæ■ćçÍ

èğčåĘşæŰźæąĹ

ä¡ăæIJL'äÿĂäÿłçÍŃåžŔåČŔäÿŃéİćè£ŹæăůçżĹæ■ćïijŇæŁŻåĞžäÿĂäÿł `SystemExit` åijCåÿÿïijŇä¡£çŤÍéŤŹèŕŕæűĹæĄŕä¡IJäÿžåŔĆæŢřãĂĆä¿ŃåęĆïijŽ

```
raise SystemExit('It failed!')
```

åőČäijŽåřĘæűĹæĄŕåIJÍ `sys.stderr` äÿ■æL'Şå■řïijŇçĎűåŘŐçÍŃåžŔäžěçŁűæĂAçăĄ1éĂĂåĞžãĂĆ

èőÍèőž

æIJňèŁĆèŹ¡çĎűå¿Łç§■åřŔïijŇä¡ĘæŸŕåőČèČ¡èğčåĘşåIJÍåĘŹèĎŽæIJňæŮűçŽĎäÿĂäÿłåÿÿèğĄéŮőécŸã äź§åřśæŸŕèŕťïijŇå¡Şä¡ăæČşèęAçżĹæ■ćæ§Řäÿłçİ Ńåžŕæ ŮűïijŇä¡ăåŔŕèČ¡äijŽåČŔäÿŃéİćè£ŹæăůåĘŹïijŽ

```
import sys
sys.stderr.write('It failed!\n')
raise SystemExit(1)
```

åęĆædIJä¡ăçŻť'æŐěåřĘæűĹæĄŕä¡IJäÿžåŔĆæŢřäijăçżŹ `SystemExit()` ïijŇéĆčäźĹä¡ăåŔŕäžěçIJAçŢěåĚűäžŰæ■ééłď'ïijŇ æŕŤåęĆimportèŕ■åŘěæĹŰåřĘéŤŹèŕŕæűĹæĄŕåĘŹåĚě `sys.stderr`

15.3 13.3 èğčædŘåŚ¡äžď'èąŇéĂĽéąź

éŮőécŸ

ä¡ăçŽĎçÍŃåžŔåęĆä¡ŢèČ¡åď §èğčædŘåŚ¡äžď'èąŇéĂĽéąźïijĹä¡■äžŐsys.argväÿ■ïijĽ

èğčåĘşæŰźæąĹ

argparse æĺąåİŮåŕŕèćńçŤĺæİěèğčæđŘåŠ¡äžd'èąŇéĂL'éąźãĂĆäÿŃéİćäÿĂäÿłçőĂå■Ţä¿Ńå■ŘæijŤçd'ż

```
# 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)
```

èŕěçÍŃåžŘåőŽäźL'äžĘäÿĂäÿłåęĆäÿŃä¡£çŤÍçŽĎåŠ¡äžd'èąŇèğčæđŘåŹÍïijŽ

```
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)

```
  -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 žäžĘçÍŃåžŔäÿ■çŽĎæŢřæ■őéČĺåĹĘãĂĆäżŤçżĘèğĆåŕ§print()èí■åŔěçŽĎæĽ'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
```

åŕźäžŐéĂL'éąźåĂijçŽĎè£ŻäÿĂæ■ěåď'ĎçŘĘçŤśçÍŃåžŔæÍěåĘşåőŽïijŇçŤÍä¡äèĞłåůśçŽĎéĂżè¿ŚæÍěæŻ£ `print()` åĞ¡æŢřãĂĆ

èőÍèőž

`argparse` æÍąåİŮæŸŕæăĞåĞĘåžŞäÿ■æIJĂåď'ğçŽĎæÍąåİŮäźŃäÿĂïijŇæŃěæIJL'åď'ğéĞŔçŽĎéĚ■ç¡őé æIJňèŁĆåŔłæŸŕæijŤçď žäžĘåĚűäÿ■æIJĂå§žçąĂçŽĎäÿĂäžŻçL'źæĂğïijŇåÿőåŁl'ä¡ääĚěéŮĺãĂĆ

äÿžäžĘèğčædŘåŚ¡äžď'èąŇéĂL'éąźïijŇä¡äéęŰåĚĹèęĄåĹŻåžžäÿĂäÿł `ArgumentParser` åőđä¿ŃïijŇ åźűä¡£çŤÍ `add_argument()`

æŰźæşŢåčřæŸŐä¡ăæČşèęĄæŤřæŇĄçŽĎéĂĽ′éąźãĂĆ åĲĺæŕŔäÿł add_argument() èřČçŤĺäÿ■ïijŇdest åŔĆæŢřæŇĞåőŽèğčæđŘçżŞæđĲèćńæŇĞæť¿çżŹåśđæĂğçŽĎåŘ■å■ŮãĂĆ metavar åŔĆæŢřèćńçŤĺæİěçŤ§æĹŘåÿőåĽľ䣿æĄŕãĂĆaction åŔĆæŢřæŇĞåőŽèů§åśđæĂğåŕźåžŤçŽĎåď′ĐçŘĘéĂżè¿ŚïijŇ éĂŽåÿÿçŽĎåĂijäÿž store ‚èćńçŤĺæİěå■ŸåĆĺæ§ŘäÿłåĂijæĹŰåřĘåď′Žäÿłå ŔĆæŢřåĂijæŤűéŻĘåĹřäÿĂäÿłåĹŮèąĺäÿ■ãĂĆ äÿŃéİćçŽĎåŔĆæŢřæŤűéŻĘæĽĂæIJĽåĽľä¡ŹçŽĎåŠ¡äżď′èąŇåŔĆæŢřåĹřäÿĂäÿłåĹŮèąĺäÿ■ãĂĆåĲĺæIJňä¿Ň

```python
parser.add_argument(dest='filenames',metavar='filename', nargs='*')
```

äÿŃéİćçŽĎåŔĆæŢřæăźæ■őåŔĆæŢřæŸŕåŘęå■ŸåĲĺæİěèő¿ç¡őäÿĂäÿł Boolean æăĞå£ŮïijŽ

```python
parser.add_argument('-v', dest='verbose', action='store_true',
                    help='verbose mode')
```

äÿŃéİćçŽĎåŔĆæŢřæŐěåŔŮäÿĂäÿłå■ŢçŃňåĂijåźűåřĘåĚűå■ŸåĆĺäÿžäÿĂäÿłå■Ůçňęäÿšïijž

```python
parser.add_argument('-o', dest='outfile', action='store',
                    help='output file')
```

äÿŃéİćçŽĎåŔĆæŢřèŕť æŸŐåĚĄèőÿæ§ŘäÿłåŔĆæŢřéĞ■åď′■åĞžçŐřåď′ŽæňąïijŇåźűåřĘåőČäżňè£¡åŁăåĹ required æăĞå£Ůèąĺçď′žèŕěåŔĆæŢřèĞşåřŚèęĄæIJĽäÿĂäÿłãĂĆ-p åŠŇ --pat èąĺçď′žäÿď′äÿłåŔĆæŢřåŘ■å¡ćåijŔéČ¡åŔŕä¡£çŤĺãĂĆ

```python
parser.add_argument('-p', '--pat',metavar='pattern', required=True,
                    dest='patterns', action='append',
                    help='text pattern to search for')
```

æIJĂåŘŐïijŇäÿŃéİćçŽĎåŔĆæŢřèŕť æŸŐæŐěåŔŮäÿĂäÿłåĂijïijŇä¡ĘæŸŕäijŽåřĘåĚűåŠŇåŔŕèČ¡çŽĎéĂ

```python
parser.add_argument('--speed', dest='speed', action='store',
                    choices={'slow','fast'}, default='slow',
                    help='search speed')
```

äÿĂæŮęåŔĆæŢřéĂĽ′éąźèćńæŇĞåőŽïijŇä¡ăåřśåŔŕäżěæL′ğèąŇ parser.parse() æŰźæşŢäžĘãĂĆ åőČäijŽåď′ĐçŘĘ sys.argv çŽĎåĂijåźűè£ŤåŽđäÿĂäÿłçżŞæđIJåőđä¿ŃãĂĆ æŕŔäÿłåŔĆæŢřåĂijäijŽèćńèő¿ç¡őæĹŘèŕěåőđä¿Ńäÿ■ add_argument() æŰźæşŢçŽĎ dest åŔĆæŢřæŇĞåőŽçŽĎåśđæĂğåĂijãĂĆ

è£Ÿå¿Ĺåď′Žçğ■åĚűäżŰæŰźæşŢèğčæđŘåŠ¡äżď′èąŇéĂĽ′éąźãĂĆ ä¿ŃåęĆïijŇä¡ăåŔŕèČ¡äijŽæL′ŃåŁĺçŽĎåď′ĐçŘĘ sys.argv æĹŰèĂĚä¡£çŤĺ getopt æląåİŮãĂĆ ä¡ĘæŸŕïijŇåęĆæđIJä¡ăéĞĞçŤĺæIJňèŁĆçŽĎæŰźåijŔïijŇåřĘäijŽå ĞŔåřŚå¿Ĺåď′ŽåĘŮä¡Źäżčçă argparse æląåİŮåůšçżŔäÿ¡äåď′ĐçŘĘäžĘãĂĆ ä¡ääåŔŕèČ¡è£ŸäijŽçćřåĹřä¡£çŤĺ optparse åž§èğčæđŘéĂĽ′éąźçŽĎäżčçăĄãĂĆ åř¡çőą optparse åŠŇ argparse å¿ĹåČŔïijŇä¡ĘæŸŕåŘŐèĂĚæŻť åĚĹè£ŻïijŇåŽăæ■ďåIJĺæŰřçŽĎçĺŃåžŔäÿ■ä¡ăåžŤèŕěä¡£çŤĺåőČãĂĆ

15.4 13.4 è£ŘèąŇæŮűåijźåĞžåŕĘçăĄè¿ŞåĚěæŔŘçď'ž

éŮőécŸ

ä¡ăåĘŹäžĘäÿłèĎŽæIJñïijŇè£ŘèąŇæŮűéIJĂèęĄäÿĂäÿłåŕĘçăĄãĂĆæ■d'èĎŽæIJñæŸŕäžd'äžŠåijŔçŽĎïij
èĂŇæŸŕéIJĂèęĄåijźåĞžäÿĂäÿłåŕĘçăĄè¿ŞåĚěæŔŘçd'žïijŇèől'çŤĺæĹůèĞłåůśè¿ŞåĚěãĂĆ

èğčåĘşæŰźæąĹ

è£ŹæŮűåĂŹPythonçŽĎ `getpass` æĺąåİŮæ■čæŸŕä¡ăæL'ĂéIJĂèęĄçŽĎãĂĆä¡ăåŔŕäžěèől'ä¡ăå¿Ĺè¡żæİ¿
åźűäÿŤäÿ■äijŽåIJĺçŤĺæĹůçżĹçńŕåŻđæŸ¿åŕĘçăĄãĂĆäÿŇéİćæŸŕåĚůä¡Şäžčçă Ąïijž

```
import getpass

user = getpass.getuser()
passwd = getpass.getpass()

if svc_login(user, passwd):     # You must write svc_login()
   print('Yay!')
else:
   print('Boo!')
```

åIJĺæ■d'äžčçăĄäÿ■ïijŇ`svc_login()` æŸŕä¡ăèęĄåőđçŐrçŽĎåd'ĎçŘĘåŕĘçăĄçŽĎåĞ¡æŢřïijŇåĚůä¡Şç

èőĺèőž

æşĺæĎŔåIJĺåL'■éİćäžčçăĄäÿ■ `getpass.getuser()`
äÿ■äijŽåijźåĞžçŤĺæĹůåŘ■çŽĎè¿ŞåĚěæŔŘçd'žãĂĆ åőČäijŽæăźæ■őèřęçŤĺæĹůçŽĎshel-
lçŐŕåćČæĹŰèĂĚäijŽä¿Ìæ■őæIJñåIJřçşżçż§çŽĎåŕĘçăĄåžŞïijĹæŤŕæŇĄ *pwd*
æĺąåİŮçŽĎåźşåŔřïijL'æİěä¡£çŤĺå¡ŞåL'■çŤĺæĹůçŽĎçŹżå¡ŢåŘ■ïijŇ

åęĆæđIJä¡ăæČşæŸ¿çd'žçŽĎåijźåĞžçŤĺæĹůåŘ■è¿ŞåĚěæŔŘçd'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£ŹäžŻéŮőécŸïi

15.5 13.5 èŐůåŔŰçżĹçńŕçŽĎåd'ğåřŔ

éŮőécŸ

ä¡ăéIJĂèęĄç§ěéĄŞå¡ŞåL'■çżĹçńŕçŽĎåd'ğåřŔäžěä¿£æ■čçąőçŽĎæăijåijŔåŇŰè¿ŞåĞžãĂĆ

èğčåĘşæŰźæąĹ

ä¡£çŤÍ `os.get_terminal_size()` åĞ¡æŢřæİěåĄŽåĹřè£ŹäÿĂçĆźãĂĆ äžčçăĄçđ'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'åđ'łåđ'ŽæŰźåijŔæİěå¿Ůç§ěçżĹçńŕåđ'ğářŔäžĘïijŇäżŐèŕżåŔŰçŐŕåćČåŔŸéĞŔåĹřæL'ğèąŇåžŢåśĆç `ioctl()` åĞ¡æŢřç■L'ç■L'ãĂĆ äÿ■è£ĞïijŇäÿžäżĂäźĹèęĄåŐżçăŤçl'űè£ŹäžŽåđ'■æİĆçŽĎåŁđæşŢèĂŇäÿ■æ

15.6 13.6 æL'ğèąŇåđ'ŰéČĺåŚ¡äžđ'åźűèŐůåŔŰåőČçŽĎè¿ŞåĞž

éŮőécŸ

ä¡ăæČşæL'ğèąŇäÿĂäÿłåđ'ŰéČĺåŚ¡äžđ'åźűäžěPythonå■ŮçňęäÿšçŽĎå¡ćåijŔèŐůåŔŰæL'ğèąŇçżŞæđIJãĂ

èğčåĘşæŰźæąĹ

ä¡£çŤÍ `subprocess.check_output()` åĞ¡æŢřãĆä¿ŇåęĆïijŽ

```
import subprocess
out_bytes = subprocess.check_output(['netstat','-a'])
```

è£ŹæőţäžčçăĄæL'ğèąŇäÿĂäÿłæŇĞåőŽçŽĎåŚ¡äžđ'åźűåřĘæL'ğèąŇçżŞæđIJäžěäÿĂäÿłå■ŮèŁĆå■Ůçňęäÿ åęĆæđIJä¡ăéIJĂèęĄæŰĞæIJñå¡ćåijŔè£ŤåŻđïijŇåŁääÿĂäÿłèğčçăĄæ■ěéłđ'å■şåŔŕãĆä¿ŇåęĆïijŽ

```
out_text = out_bytes.decode('utf-8')
```

åęĆæđIJèćńæL'ğèąŇçŽĎåŚ¡äžđ'äžěéİđéŻűçăĄè£ŤåŻđïijŇåřśäijŽæŁŻåĞžåijĆåÿÿãĂĆ äÿŃéİćçŽĎä¿Ňå■Řæ■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
```

éżŸèőďæČĚåĘţäÿŃïijŇcheck_output() äžĚäžĚè£ŤåŻđè¿ŞåĚěåĹřæăĞåĞĘè¿ŞåĞžçŽĎåĂijãĂĆ åęĆæđIJä¡ăéIJĂèęĄåŘŇæŮűæŤűéŻĘæăĞåĞĘè¿ŞåĞžåŠŇéŤŹèŕŕè¿ŞåĞžïijŇä¡£çŤÍ stderr åŔĆæŢřïijŽ

```
out_bytes = subprocess.check_output(['cmd','arg1','arg2'],
                                    stderr=subprocess.STDOUT)
```

åęĆæđIJä¡ăéIJĂèęĄçŤÍäÿĂäÿłèűĚæŮűæIJžåĹűæİěæL'ğèąŇåŚ¡äžď'ïijŇä¡£çŤÍ timeout åŔĆæŢřïijŽ

```
try:
    out_bytes = subprocess.check_output(['cmd','arg1','arg2'],␣
↪timeout=5)
except subprocess.TimeoutExpired as e:
    ...
```

éĂŽåÿÿæİěèőšïijŇåŚ¡äžď'çŽĎæL'ğèąŇäÿ■éIJĂèęĄä¡£çŤÍåĹřåžŢåśĆshellçŐŕåćČïijĹæŕŤåęĆshãĂĄbash äÿĂäÿłå■ŮçňęäÿšåĹŮèąĺäijŽèćńäijăéĂŠçżŹäÿĂäÿłä¡Őçžğçşżçż§åŚ¡äžď'ïijŇæŕŤåęĆ os.execve() ãĂĆ åęĆæđIJä¡ăæČşèőľ'åŚ¡äžď'èćńäÿĂäÿłshellæL'ğèąŇïijŇäijăéĂŠäÿĂäÿłå■Ůçňęäÿšå ŔĆæŢřïij shell=True. æIJL'æŮűåĂŹä¡ăæČşèęĄPythonåŐžæL'ğèąŇäÿĂäÿłåď'■æİĆçŽĎshellåŚ¡äžď'çŽĎæŮűåĂŹè

```
out_bytes = subprocess.check_output('grep python | wc > out',␣
↪shell=True)
```

éIJĂèęĄæşĺæĎŘçŽĎæŸŕåIJĺshelläÿ■æL'ğèąŇåŚ¡äžď'äijŽå■ŸåIJĺäÿĂåőŽçŽĎåőL'åĚĺéčŐéŹľïijŇçL'źåĹ è£ŹæŮűåĂŹåŘŕäžëä¡£çŤÍ shlex.quote() åĞ¡æŢřæİěåřĘåŔĆæŢřæ■čçąőçŽĎçŤĺåŘŇåijŢçŤĺåijŢèţůæİěã

èőĺèőž

ä¡£çŤÍ check_output() åĞ¡æŢřæŸŕæL'ğèąŇåď'ŰéČĺåŚ¡äžď'åźűèŐůåŔŰåĚűè£ŤåŻđåĂijçŽĎæIJĂçő ä¡ĘæŸŕïijŇåęĆæđIJä¡ăéIJĂèęĄåŕźå■Řè£ŽçĺŃåĄŽæŻť'åď'■æİĆçŽĎäžď'äžŠïijŇæŕŤåęĆçżŹåőČåŔŚéĂĄè¿Şå è£ŹæŮűåĂŹåŘŕçŻť'æŐëä¡£çŤÍ subprocess.Popen çśžãĂĆä¿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)

```
# To interpret as text, decode
out = stdout.decode('utf-8')
err = stderr.decode('utf-8')
```

subprocess æÍąåİŮåŕźäžŐä¿İèţŰTTYçŽĎåď ŰéČÍåŚ¡äżď äÿ■åŘĹéĂĆçŤÍãĂĆ ä¿ŇåęĆïijŇä¡ääÿ■èČ¡ä¡£çŤÍåőČæİěèĞłåŁÍåŇŰäÿĂäÿłçŤÍæĹůè¿ŞåĚěåŕĘçăĄçŽĎäżżåŁąïijĹæŕŤåęĆäÿĂäÿłss è£ŹæŮűåĂŹïijŇä¡ăéIJĂèęĄä¡£çŤÍåĹřçňňäÿL'æŰźæÍąåİŮäžĘïijŇæŕŤåęĆå§žäžŐèŚŮåŘ■çŽĎ expect åőűæŮŘçŽĎåůěåĚůïijĹpexpectæĹŰçśżäijijçŽĎïijL'

15.7 13.7 åď■åĹűæĹŰèĂĚçğżåŁÍæŰĞäżűåŠŇçŻőå¡Ţ

éŮőécŸ

ä¡ăæČşèęĄåď■åĹűæĹŰçğżåŁÍæŰĞäżűåŠŇçŻőå¡ŢïijŇä¡ĘæŸŕåŘĹäÿ■æČşèřČçŤÍshellåŚ¡äżď ãĂĆ

èğčåĘşæŰźæąĹ

shutil æÍąåİŮæIJL'å¿Ĺåď Žä¿£æ■ůçŽĎåĞ¡æŢřåŘŕäžěåď■åĹűæŰĞäżűåŠŇçŻőå¡ŢãĂĆä¡£çŤÍèţůæİěé

```
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¡ćåijŔçŽĎæŰĞäżűæĹŰçŻőå¡ŢåŘ■ãĂĆ åžŢåśĆèŕ■äźĽæÍąæŃ§äžĘçśżäijijçŽĎUnixåŚ¡äżď ïijŇåęĆäÿŁéİćçŽĎæşÍéĞŁéČÍåĹĘãĂĆ

éżŸèőď æČĚåĘţäÿŃïijŇåŕźäžŐçňęåŘůéŞ¿æŐěèĂŇåůšè£ŹäžŻåŚ¡äżď åď ĎçŘĘçŽĎæŸŕåőČæŇĞåŘŚçŽ ä¿ŇåęĆïijŇåęĆædIJæžŘæŰĞäżűæŸŕäÿĂäÿłçňęåŘůéŞ¿æŐěïijŇéĆčäźĹçŻőæăĞæŰĞäżűåřĘäijŽæŸŕçňęåŘůé åęĆædIJä¡ääŘłæČşåď■åĹűçňęåŘůéŞ¿æŐěæIJňèžńïijŇéĆčäźĹéIJĂèęĄæŇĞåőŽåĚşéŤőå■ŮåŘĆæŢř follow_symlinks ,åęĆäÿŃïijŽ

åęĆædIJä¡ăæČşä£ĺçŢŹèćńåď■åĹűçŻőå¡Ţäÿ■çŽĎçňęåŘůéŞ¿æŐěïijŇåČŔè£ŹæăůåĄŽïijŽ

```
shutil.copytree(src, dst, symlinks=True)
```

copytree() åŘŕäžěèőľä¡ăåIJĺåď■åĹűè£ĞçÍŃäÿ■éĂĽ'æŃľæĂğçŽĎå£¡çŢěæ§ŘäžŻæŰĞäžűæĹŰçŽőå
ä¡ăåŘŕäžěæŔŘä¿ŻäÿĂäÿłå£¡çŢěåĞ¡æŢřïijŇæŐěåŔŮäÿĂäÿłçŽőå¡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■æĺąåijŔçŽĎæŰĞäžűåŘ■æŸŕå¿Ĺåÿÿèğ Ąç ŽĎïijŇåŻäæ■ď'äÿĂäÿłä¿£æ■ůçŽĎåĞ¡æ
ignore_patterns() åůšçżŔåŇĚåŘńåIJĺéĞŇéÍćäžĘãĂĆä¿ŃåęĆïijŽ

```
shutil.copytree(src, dst, ignore=shutil.ignore_patterns('*~', '*.pyc
↪'))
```

èőÍèőž

ä¡£çŤÍ shutil åď■åĹűæŰĞäžűåŠŇçŻőå¡Ţäź§å£ŠçőĂå■ŢäžEçĆźåŘğãĂĆ
äÿ■è£ĞïijŇåŕźäžŐæŰĞäžűåĚČæŢřæ■őä£qæĄŕïijŇcopy2() è£ŹæăůçŽĎåĞ¡æŢřåŔłèČ¡år¡èĞłåůśæIJĂåď'ğè
èő£éŮőæŮűéŮt'ãĂĄåĹŻåžžæŮűéŮt'åŠŇæÍČéŹŘè£ŹäžŻå§žæIJñä£qæĄŕäijŽèćńä£ÍçŢŹïijŇ
ä¡ĘæŸŕåŕźäžŐæĽĂæIJL'èĂĚãĂĄACLsãĂĄèţĎæžŘforkåŠŇåĚűäžŰæŻt'æůśåśĆæňąçŽĎæŰĞäžűåĚČä£qæA
è£Źäÿłè£Ÿå¿Ůä¿ÌètŰäžŐåžŢåśĆæŞ■ä¡IJçşżçż§çśżåđŃåŠŇçŤÍæĹůæĽĂæŃěæIJL'çŽĎèő£éŮőæÍČéŹŘãĂĆ
ä¡ăéĂŽåÿÿäÿ■äijŽåŐżä¡£çŤÍ shutil.copytree() åĞ¡æŢřæÍěæĽ'ğèąŇçşżçż§åď'Ğäż¡ãĂĆ
å¡Şåď'ĎçŘĘæŰĞäžűåŘ■çŽĎæŮűåĂŹïijŇæIJĂåě¡ä¡£çŤÍ os.path
äÿ■çŽĎåĞ¡æŢřæÍěçąőä£ÍæIJĂåď'ğçŽĎåŔŕçğżæď■æĂğïijĹçL'źåĹńæŸŕåŘŇæŮűèĘéĂĆçŤÍäžŐUnixåŠŇWi
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¡£çŤÍ copytree() åď■åĹűæŰĞäžűåď'źçŽĎäÿĂäÿłæčŸæĽŃçŽĎéŮőécŸæŸŕåŕźäžŐéŤŹèŕŕçŽĎåď'Ď
ä¿ŃåęĆïijŇåIJĺåď■åĹűè£ĞçÍŃäÿ■ïijŇåĞ¡æŢřåŔŕèČ¡äijŽçćřåĹřæ■§åİŔçŽĎçňęåŘůéŞ¿æŐěïijŇåŻääÿžæİČéŹ
äÿžäžEèğčåEşè£ŹäÿłéŮőécŸïijŇæĽĂæIJL'çćřåĹřçŽĎéŮőécŸäijŽèćńæŤűéŻĘåĹřäÿĂäÿłåĹŮèąĺäÿ■åźűæĽ'Ş
äÿŃéÍćæŸŕäÿĂäÿłä¿Ńå■ŘïijŽ

```
try:
    shutil.copytree(src, dst)
except shutil.Error as e:
```

(continues on next 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¡ăæŔŘä¿ŻåĚşéŤőå■ŮåŔĆæŢř `ignore_dangling_symlinks=True` ïijŇè£ŹæŮűåĂŹ `copytree()` äijŽå£¡çŢěæŐĽ'æŮăæŢĹçňęåŘůéŞ¿æŐěãĂĆ

æIJňèŁĆæijŤçď'žçŽĎè£ŹäžŻåĞ¡æŢřéČ¡æŸŕæIJĂåÿÿèğĄçŽĎãĂĆäÿ■è£ĞïijŇshutil è£ŸæIJĽ'æŻť'åď'ŽçŽĎåŠŇåď'■åĹűæŢřæ■őçŻÿåĚşçŽĎæŞ■ä¡IJãĂĆ åőČçŽĎæŰĞæąčå¿ĹåĂijå¿ŮäÿĂçIJŃïijŇåŔĆèĂČ Python documentation

15.8 13.8 åĹŻåžžåŠŇèğčåŐŃå¡Šæąčæ ŰĞäžű

éŮőécŸ

ä¡ăéIJĂèęĄåĹŻåžžæĹŰèğčåŐŃåÿÿèğĄæăijåijŔçŽĎå¡Šæąčæ ŰĞäžűïijĹærŤåęĆ.tar, .tgzæĹŰ.zipïijL'

èğčåĘşæŰźæąĹ

shutil æĺąåİŮæŃěæIJL'äÿď'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()` çŽĎçňnäžŇäÿłåŔĆæŢřæŸŕæIJ§æIJŻçŽĎè¿ŞåĞžæăijåijŔãĂĆ åŘŕäžëä¡£çŤĺ `get_archive_formats()` èŐůåŘŰæL'ĂæIJL'æŤŕæŇAçŽĎå¡Šæąčæăijåij ŔåĹŮèąĺãĂĆä¿Ň

```
>>> shutil.get_archive_formats()
[('bztar', "bzip2'ed tar-file"), ('gztar', "gzip'ed tar-file"),
 ('tar', 'uncompressed tar file'), ('zip', 'ZIP file')]
>>>
```

èőÍèőž

Pythonè£ŸæIJL'åĚűäżŰçŽĎæĺąåİŮåŔŕçŤĺæİěåď'ĎçŘĘåď'Žçğ■å¡Šæąčæăijåij ŔïijĹærŤåęĆtarfile, zipfile, gzip, bz2ïijL'çŽĎåžŢåśĆçżĘèŁĆãĂĆ äÿ■è£ĞïijŇåęĆæđIJä¡ääžĚäžĚåŘłæŸŕèęĄåĹŻåžžæĹŰæŔŘåŔŰ åŘŕäžëçŻť'æŐëä¡£çŤĺ `shutil` äÿ■çŽĎè£ŹäžŻéńŸåśĆåĞ¡æŢřãĂĆ

è£ŹäžŽåĞ¡æŢřè£ŸæIJL'å¿Ĺåď'ŽåĚűäžŰéĂL'éąźïijŇçŢĺäžŐæŮěå£ŮæL'Şå■řãĂĄéćĎæčĂãĂĄæŰĞäžűa
åŔĆèĂČ shutilæŰĞæąč

15.9 13.9 éĂŽè£ĞæŰĞäżűåŘ■æ§ěæL'¿æŰĞäžű

éŮőécŸ

ä¡ăéIJĂèęĄåEŹäÿĂäÿłæűL'åŔŁåĹřæŰĞäžűæ§ěæL'¿æŞ■ä¡IJçŽĎèĎŽæIJñïijŇæŕŤåęĆåŕźæŮěå£Ůå¡Šæą
ä¡ăäÿ■æČşåIJĺPythonèĎŽæIJñäÿ■èřČçŢĺshellïijŇæĹŰèĂĚä¡ăèęĄåőđçŐřäÿĂäžŽshelläÿ■èČ¡åĄŽçŽĎåŁ§èČ

èğčåEşæŰźæąĹ

æ§ěæL'¿æŰĞäżűïijŇåŔŕä¡£çŢĺ `os.walk()` åĞ¡æŢřïijŇäijăäÿĂäÿłéąűçžğçŻőå¡ŢåŘ■çżŹåőČãĂĆ
äÿŇéIćæŸŕäÿĂäÿłä¿Ňå■ŘïijŇæ§ěæL'¿çL'źåőŽçŽĎæŰĞäżűåŘ■åźűç■ŤåžŤæL'ĂæIJL'çñęåŘĹæÌąäžűçŽĎæŰ

```
#!/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£Ìå■ŸèĎŽæIJñäÿžæŰĞäżűfindfile.pyïijŇçĎűåŘŐåIJĺåŚ¡äžd'èąŇäÿ■æL'ğèąŇåőČãĂĆ
æŇĞåőŽåĹÌåğŃæ§ěæL'¿çŻőå¡Ţäžěå ŔĹåŘ■å■Ůä¡IJäÿžä¡■ç¡ôåŔĆæŢřïijŇåęĆäÿŃïijŽ

èőÍèőž

`os.walk()` æŰźæşŢäÿžæĹŚäžňéĄ■åŐEçŻőå¡ŢæăŚïijŇ
æŕŔæňąè£ŻåĚëäÿĂäÿłçŻőå¡ŢïijŇåőČäijŽè£ŤåŻđäÿĂäÿłäÿL'åĚČçžĎïijŇåŇĚåŘńçŽÿåŕźäžŐæ§ěæL'¿çŻőå¡Ţ
äžěåŔĹéĆčäÿłçŻőå¡ŢäÿŃéÍćçŽĎæŰĞäżűåŘ■åĹŮèąÍãĂĆ

åŕźäžŐæŕŔäÿłåĚČçžĎïijŇåŔłéIJĂæčĂæţŃäÿĂäÿŃçŻőæăĞæŰĞäżűåŘ■æŸŕåŘęåIJÍæŰĞäżűåĹŮèąÍäÿ■
`os.path.join()` åŘĹåźűèůŕå¿ĎãĂĆ äÿžäžEéĄ£åĚ■åëĞæĂłçŽĎèůŕå¿ĎåŘ■æŕŤåęĆ `./`
`./foo//bar` ïijŇä¡£çŢĺäžEåŔęåď'Ûäÿď'äÿłåĞ¡æŢřæÌëä£őæ■čçżŞædIJãĂĆ çñňäÿĂäÿłæŸŕ
`os.path.abspath()` ,åőČæŐěåŔŮäÿĂäÿłèůŕå¿ĎïijŇåŔŕèČ¡æŸŕçŽÿåŕźèůŕå¿ĎïijŇæIJĂåŘŐè£ŤåŻđçžÌår
çñňäžŇäÿłæŸŕ `os.path.normpath()` ïijŇçŢĺæÌëè£ŤåŻđæ■čåÿÿèůŕå¿ĎïijŇåŔŕäžëèğčåEşåŔŇæŰIJæÌEã

år¡çőąè£ŹäÿłèĎŽæIJñçŽÿåŕźäžŐUNIXåźşåŘŕäÿŁéÍćçŽĎå¿Ĺåď'Žæ§ěæL'¿æÌëëőšèęĄçőĂå■Ţå¿Ĺåď'Žïij
åźűäÿŤïijŇè£ŸèČ¡å¿Ĺè¡żæÌ¿çŽĎåŁäåĚëåĚűäžŰçŽĎåŁ§èČ¡ãĂĆ
æĹŚäžňåE■æijŤçď'žäÿĂäÿłä¿Ňå■ŘïijŇäÿŃéÍćçŽĎåĞ¡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
åŘřåŔĆèĂČ5.11åřŔèŁĆåŠŇ5.13åřŔèŁĆç■L'çŻÿåĚşçńäèŁĆãĂĆ

15.10 13.10 èŕżåŔŰéĚ■ç¡őæŰĞäżű

éŮőécŸ

æĂŐæäůèŕżåŔŰæŹőéĂŽ.iniæäijåijŔçŽĎéĚ■ç¡őæŰĞäżűïij§

èğčåĘşæŰźæąĹ

configparser æÍąåİŮèČ¡èćńçŤÍæİěèŕżåŔŰéĚ■ç¡őæŰĞäżűãĂĆä¿ŇåęĆïijŇåĄĞèő¿ä¡ăæIJL'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)

```
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
    =================================
```

äýŃéİćæŸŕäýĂäýłèŕżåŔŮåŠŇæŔŘåŔŮåĚűäý■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
\=================================
>>>
```

åęĆædIJæIJL'éIJĂèęĄïijŇä¡ăè£ŸèČ¡ä£őæŤźéĚ■ç¡ő åźűä¡£çŤÍ `cfg.write()` æŰźæşŢåřĘåĚűåĘŹåŻđåĹřæŰĞäżűäý■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)

(çż■äÿŁéąţ)

```
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¡çŽĎæăijåijŔïijŇéİđåÿÿéĂĆçŤĺäžŐå■ŸåĆĺçÍŃåžŔäÿ■çŽ
åIJĺæŕŔäÿłéĚ■ç¡őæŰĞäžűäÿ■ïijŇéĚ■ç¡őæŢřæ■őäijŽèćńåĹĘçžĎïijĹæŕŤåęĆä¿Ńå■Řäÿ■çŽĎâĂIJinstallationâ
âĂIJdebugâĂİ åŠŇ âĂIJserverâĂİïijL'ãĂĆ æŕŔäÿłåĹĘçżĎåIJÍåĚűäÿ■æŇĞåőŽåŕźåžŤçŽĎåŘĎäÿłåŔŸéĞŔåĂ

åŕźäžŐåŔŕåőđçŐřåŘŇæăůåŁ§èČ¡çŽĎéĚ■ç¡őæŰĞäžűåŠŇPythonæžŘæŰĞäžűæŸŕæIJL'å¿Ĺåđ'ğçŽĎäÿ■
éęŰåĚĹïijŇéĚ■ç¡őæŰĞäžűçŽĎèŕ■æşŢèęĄæŻt'èĞłçŤśäžŻïijŇäÿŃéİćçŽĎèţŃåĂijèŕ■åŘěæŸŕç■L'æŢĹçŽĎïij

```
prefix=/usr/local
prefix: /usr/local
```

éĚ■ç¡őæŰĞäžűäÿ■çŽĎåŘ■å■ŮæŸŕäÿ■åŇžåĹĘåđ'ğářŔåĘŹçŽĎãĂĆä¿ŃåęĆïijŽ

```
>>> cfg.get('installation','PREFIX')
'/usr/local'
>>> cfg.get('installation','prefix')
'/usr/local'
>>>
```

åIJĺèğčæđŘåĂijçŽĎæŮűåĂŹïijŇgetboolean() æŰźæşŢæ§ěæL'¿äżżä¡ŢåŔŕèąŇçŽĎåĂijãĂĆä¿ŃåęĆ

```
log_errors = true
log_errors = TRUE
log_errors = Yes
log_errors = 1
```

æĹŰèőÿéĚ■ç¡őæŰĞäžűåŠŇPythonäżčçăĄæIJĂåđ'ğçŽĎäÿ■åŘŇåIJĺäžŐïijŇåőČåźűäÿ■æŸŕäżŐäÿŁèĂŇ
æŰĞäžűæŸŕåőL'èčĚäÿĂäÿłæŢt'ä¡Şèćńèŕżå ŔŰçŽĎãĂĆåęĆæđIJçćřåĹřäžĘåŔŸéĞŔæŻ£æ■ćïijŇåőČåőđéŹĚä
ä¿ŃåęĆïijŇåIJĺäÿŃéİćè£ŹäÿłéĚ■ç¡őäÿ■ïijŇprefix åŔŸéĞŔåIJĺä¡£çŤĺåőČçŽĎåŔŸéĞŔäźŃåL'■æĹŰäźŃåŘ

```
[installation]
library=%(prefix)s/lib
include=%(prefix)s/include
bin=%(prefix)s/bin
prefix=/usr/local
```

ConfigParser æIJL'äÿłåőźæŸŞèćńå£¡èğĘçŽĎçL'źæĂğæŸŕåőČèČ¡äÿĂæñąèŕżåŔŰåd'ŽäÿłéĚ■ç¡őæŰä¿ŃåęĆïijŇåĄĞèő¿äÿĂäÿłçŤĺæĹůåČŔäÿŃéIćè£ŹæăůædĎéĂăäžĘäżŰäżňçŽĎéĚ■ç¡őæŰĞäżűïijŽ

```
; ~/.config.ini
[installation]
prefix=/Users/beazley/test

[debug]
log_errors=False
```

èŕżåŔŰè£ŹäÿłæŰĞäżűïijŇåőČåřśèČ¡èů§äźŃåL'■çŽĎéĚ■ç¡őåŘĹåźűètůæIěãĂĆåęĆïijŽ

```
>>> # 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
>>>
```

äžŤçżĘèğĆåŕ§äÿŃ prefix åŔŸéĞŔæŸŕæĂŐæăůèęĘçŻŰåĚűäżŰçŻÿåĚşåŔŸéĞŔçŽĎïijŇæŕŤåęĆ library çŽĎèő¿åőŽåĂijãĂĆ äžğçŤ§è£Źçğ■çżŞæđIJçŽĎåŐ§åŻäæŸŕåŔŸéĞŔçŽĎæŤźåĘźéĞĞåŔŰçŽĎæ¡ääŔŕäžěåČŔäÿŃéIćè£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■èęĄäÿĂçĆźèęĄæşÍæĎŔçŽĎæŸŕPythonåźűäÿ■èČ¡æŤræŇĄ.iniæŰĞäżűåIJĺåçąőä£Ïä¡ääůšçżŔåŔĆéŸĚäžĘconfigparseræŰĞæąčäÿ■çŽĎèŕ■æşŢèŕęæČĚäžěåŘŁæŤræŇĄçL'źæĂğãĂĆ

15.11 13.11 çžŹçőĂå■ŢèĎŽæIJňáćđåŁăæŮěå£ŮåŁ§èČ¡

éŮőécŸ

ä¡ăåÿŇæIJŻåIJÍèĎŽæIJňåŠŇçÍŃåžŔäÿ■åřĘèŕŁæŰ■ä£ąæĄŕåĘŹåĚěæŮěå£ŮæŰĞäżűãĂĆ

èğčåĘşæŰźæąĹ

æĽŞå■řæŮěå£ŮæIJĂçőĂå■ŢæŰźåijŔæŸŕä¡£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()
```

äÿŁéIćäžŤäÿłæŮěå£ŮèřČçŤÍïijÍcritical(), error(), warning(), info(), debug()ïijĽäžěéŹ■åžŔæŰźåijŔèąÍçd'žäÿ■åŘŇçŽĎäÿěéĞ■çžğåĹńãĂĆ `basicConfig()` çŽĎ `level` åŔĆæŢřæŸŕäÿĂäÿłè£ĞæżďåŹÍãĂĆ æĽĂæIJĽçžğåĹńä¡ŐäžŐæ■ďçžğåĹńçŽĎæŮěå£ŮæűĹæĄŕéČ¡äijŽèćńå£¡çŢěæŐĽãĂĆ æŕŔäÿłloggingæŞ■ä¡IJçŽĎåŔĆæŢřæŸŕäÿĂäÿłæűĹæĄŕå■ŮçňęäÿšïijŇåŘŐéIćåĘ■èů§äÿĂäÿłæĹŰåd'ŽäÿłåŔĆ æđĎéĂăæIJĂçžĹçŽĎæŮěå£ŮæűĹæĄŕçŽĎæŮűåĂŹæĹŚäžňä¡£çŤÍäžĘ%æŞ■ä¡IJçňęæÍěæăijåijŔåŇŰæűĹæĄ

è£ŘèąŇè£ŹäÿłçÍŃåžŔåŘŐïijŇåIJÍæŰĞäżű `app.log` äÿ■çŽĎåĘĚåőźåžŤèŕěæŸŕäÿŃéIćè£ŹæăůïijŽ

```
CRITICAL:root:Host www.python.org unknown
ERROR:root:Could not find 'spam'
```

åęĆædIJä¡ăæČşæŤźåŔŸè¿ŞåĞžç■L'çžğïijŇä¡ăåŔŕäžěä£őæŤź `basicConfig()` èřČçŤÍäÿ■çŽĎåŔĆæŢřãĂĆä¿ŃåęĆïijŽ

```
logging.basicConfig(
     filename='app.log',
     level=logging.WARNING,
     format='%(levelname)s:%(asctime)s:%(message)s')
```

æIJĂåŘŐè¿ŞåĞžåŔŸæĹŘåęĆäÿŃïijŽ

```
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¡æŸŕçąñçijŰçăĄåĹřçÍŃåžŔäÿ■çŽĎãĂĆåęĆæđIJä¡ăæČşä¡£çŤĺéĚ■ç¡őæŰČåŔŕäżěåČŔäÿŃéİćè£Źæăůä£őæŤź `basicConfig()` èřČçŤĺïijŽ

```
import logging
import logging.config

def main():
    # Configure the logging system
    logging.config.fileConfig('logconfig.ini')
    ...
```

åĹŻåżžäÿĂäÿłäÿŃéİćè£ŹæăůçŽĎæŰĞäżűïijŇåŘ■å■ŮåŔń `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¡őïijŇåŔŕäżěçŻť æŐěçijŰè¿ŚæŰĞäżűlogconfig.iniå■şåŔŕãĂĆ

èőÍèőž

åř¡çőąåŕźäžŐ `logging` æÍąåİŮèĂŇåůšæIJL’å¿Ĺåď’ŽæŻť’énŸçžğçŽĎéĚ■ç¡őéĂL’éąźïijŇ
äÿ■è£Ğè£ŹéĞŇçŽĎæŰźæąĹåŕźäžŐçőĂå■ŢçŽĎçÍŃåžŔåŠŇèĎŽæIJňåůšçżŔèűşåď§äžĘãĂĆ
åŘłæČşåIJÍèřČçŤÍæŮěå£ŮæŞ■ä¡IJåL’■åĚĹæL’ğèąŇäÿŃbasicConfig()åĞ¡æŢřæŰźæşŢïijŇä¡äçŽĎçÍŃåžŔåřśè

åęĆæđIJä¡ăæČşèęĄä¡äçŽĎæŮěå£ŮæűĹæĄŕåĘŹåĹřæăĞåĞĘéŤŹèŕŕäÿ■ïijŇèĂŇäÿ■æŸŕæŮěå£ŮæŰĞäż
`basicConfig()` æŮűäÿ■äijäæŰĞäżűåŘ■åŔĆæŢřå■şåŔŕãĂĆä¿ŃåęĆïijŽ

```
logging.basicConfig(level=logging.INFO)
```

`basicConfig()` åIJÍçÍŃåžŔäÿ■åŘłèČ¡èćńæL’ğèąŇäÿĂæňąãĂĆåęĆæđIJä¡äçĺ■åŘŐæČşæŤźåŔŸæŮěå
åřśéIJĂèęĄåĚĹèŐůåŘŰ `root logger` ïijŇçĎűåŘŐçŻť’æŐëä£őæŤźåőČãĂĆä¿ŃåęĆïijŽ

```
logging.getLogger().level = logging.DEBUG
```

éIJĂèęĄåijžèřČçŽĎæŸŕæIJňèŁĆåŔłæŸŕæijŤçď’žäžĘ `logging`
æÍąåİŮçŽĎäÿĂäžŻå§žæIJňçŤÍæşŢãĂĆ åőČåŔŕäžěåĄŽæŻť’åď’ŽæŻť’énŸçžğçŽĎåőŽåĹűãĂĆ
åĚşäžŐæŮěå£ŮåőŽåĹűåŇŰäÿĂäÿłå¿Ĺåě¡çŽĎèţĎæžŘæŸŕ Logging Cookbook

15.12 13.12 çżŹåĞ¡æŢřåžŞåćđåŁäæŮěå£ŮåŁ§èČ¡

éŮőécŸ

ä¡ăæČşçżŹæ§Řäÿłå Ğ¡æŢřåžŞåćđåŁäæŮěå£ŮåŁ§èČ¡ïijŇä¡ĘæŸŕåŔĹäÿ■èČ¡å¡śåŞ■åĹřéĆčäžŻäÿ■ä¡£çŤĺ

èğčåĘşæŰźæąĹ

åŕźäžŐæČşèęĄæL’ğèąŇæŮěå£ŮæŞ■ä¡IJçŽĎåĞ¡æŢřåžŞèĂŇåůšïijŇä¡ääžŤèŕěåĹŻåżžäÿĂäÿłäÿŞåśđçŽĎ
`logger` åŕźèśąïijŇåźűäÿŤåČŔäÿŃéİćè£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¡őïijŇézŸèőď’æČĚåĘţäÿŃäÿ■äijŽæL’Şå■ŕæŮěå£ŮãĂĆä¿ŃåęĆïijŽ

```
>>> import somelib
>>> somelib.func()
>>>
```

äÿ■è£ĞïijŇåęĆæđIJéĚ■ç¡ő è£ĞæŮěå£Ůçşžçż§ïijŇéĆčäźĹæŮěå£ŮæűĹæĄŕæĽŞå■řářšåijĂåğŃçŤ§æŢĹï

```
>>> import logging
>>> logging.basicConfig()
>>> somelib.func()
CRITICAL:somelib:A Critical Error!
>>>
```

èőÍèőž

éĂŽåÿÿæİěèőšïijŇä¡ääÿ■åžŤèŕěåIJÍåĞ¡æŢřåžŞäžčçăĄäÿ■èĞłåůśéĚ■ç¡őæŮěå£Ůçşžçż§ïijŇæĹŰèĂĚæŸ

èřČçŤÍ `getLogger(__name__)` åĹŻåżžäÿĂäÿłåŠŇèřČçŤÍæĺąåİŮåŘŇåŘ■çŽĎlog-
geræĺąåİŮãĂĆ çŤšäžŐæĺąåİŮéČ¡æŸŕåŤŕäÿĂçŽĎïijŇåŻäæ■ď åĹŻåżžçŽĎloggeräź§åřĘæŸŕåŤŕäÿĂçŽĎãĂĆ

`log.addHandler(logging.NullHandler())` æŞ■ä¡IJåřĘäÿĂäÿłçľžåď ĎçŘĘåŹÍçżŚåőŽåĹřåĹ
äÿĂäÿłçľžåď ĎçŘĘåŹÍéżŸèőď äijŽå£¡çŢěěřČçŤÍæĽĂæIJĽçŽĎæŮěå£ŮæűĹæĄŕãĂĆ
åŻäæ■ď ïijŇåęĆæđIJä¡£çŤÍèŕěåĞ¡æŢřåžŞçŽĎæŮűåĂŹè£Ÿæšąæ IJĽéĚ■ç¡őæŮěå£ŮïijŇéĆčäźĹåřĘäÿ■äijŽæ

è£ŸæIJĽäÿĂçĆźåŕśæŸŕåŕźäžŐåŘĎäÿłåĞ¡æŢřåžŞçŽĎæŮěå£ŮéĚ■ç¡őåŔŕäžěæŸŕçŻÿäžŠçŃňçńŃçŽĎïij
ä¿ŇåęĆïijŇåŕźäžŐåęĆäÿŃçŽĎäžčçăĄïijŽ

```
>>> 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
>>>
```

åIJÍè£ŹéĞŇïijŇæăźæŮěå£ŮèćńéĚ■ç¡őæĹŘäžĚäžĚè¿ŞåĞžERRORæĹŰæŻť énŸçžğåĹńçŽĎæűĹæĄŕãĂ
äÿ■è£Ğ ïijŇsomelib çŽĎæŮěå£ŮçžğåĹńèćńå■ŢçŃňéĚ■ç¡őæĹŘåŔŕäžěè¿ŞåĞždebugçžğåĹńçŽĎæűĹæĄŕïi
åČŔè£ŹæăůæŻť æŤźå■ŢçŃňæĺąåİŮçŽĎæŮěå£ŮéĚ■ç¡őåŕźäžŐèřČèŕŢæİěèőšæŸŕå¿ĹæŰźä¿£çŽĎïijŇ
åŻääÿžä¡äæŮäéIJĂåŐżæŻť æŤźäżžä¡ŢçŽĎåĚĺåśĂæŮěå£ŮéĚ■ç¡őâĂŤâĂŤåŘłéIJĂèęĄä£őæŤźä¡äæČşèęĄæŻ

Logging HOWTO èŕęçżĘäžŃçż■äžĘåęĆä¡ŢéĚ■ç¡őæŮěå£ŮæĺąåİŮåŠŇåĚűäžŰæIJĽçŤÍæŁĂåůğïijŇåŔŕä

15.13 13.13 åőđçŐřäÿĂäÿłèőąæŮűåŹÍ

éŮőécŸ

ä¡äæČşèőřå¡ŢçÍŃåžŔæĽğèąŇåď ŽäÿłäżžåŀąæĽĂèŁśèť źçŽĎæŮűéŮť

èğčåĘşæŰźæąĹ

time æĺąåİŮåŇĚåŘńå¿Ĺåď˙ŽåĞ¡æŢřæİěæĽ˙ğèąŇèů§æŮűéŮť˙æIJĽ˙åĚşçŽĎåĞ¡æŢřãĂĆ
år¡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£ŹäÿłçśżåőŽäźĽ˙äžĘäÿĂäÿłåŘŕäżěèćńçŤĺæĹůæäźæ■őéIJĂèęĄåŘŕåŁĺãĂĄåĄIJæ■ćåŠŇéĞ■ç¡őçŽĎèőąą
åőČäijŽåIJĺ elapsed åśđæĂğäÿ■èőřå¡ŢæŢť˙äÿłæűĹèĂŮæŮűéŮť˙ãĂĆ
äÿŃéİćæŸŕäÿĂäÿłä¿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)

```
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¿ŻäžĘäÿĂäÿłçőĂå■ŢèĂŇåőđçŤÍçŽĎçśżæİěåőđçŐřæŮűéŮťèőřå¡ŢäžěåŔŁèĂŮæŮűèőąçć
åŘŇæŮűäź§æŸŕåŕźä¡£çŤÍwithèí■åŔěäžěåŔŁäÿŁäÿŃæŰĞçőąçŘĘåŹÍå■Ŕèőőç ŽĎäÿĂäÿłå¿Ĺåě¡çŽĎæijŤçď ž

åIJÍèőąæŮűäÿ■èęĄèĂČèŹŚäÿĂäÿłážŢåśĆçŽĎæŮűéŮťåĞ¡æŢřéŮőécŸãĂĆäÿĂèĹňæİěèŕťïijŇ
ä¡£çŤÍ `time.time()` æĹŰ `time.clock()` èőąçőŮçŽĎæŮűéŮťçš¿åžęåŻäæŞ■ä¡IJçşżçż§çŽĎäÿ■åŘŇäijŻ
èĂŇä¡£çŤÍ `time.perf_counter()` åĞ¡æŢřåŔŕäžěçąőä‡Ïä¡£çŤÍçşżçż§äÿŁéÍćæIJĂçš¿çąőçŽĎèőąæŮűåŹ

äÿŁè£řäžčçăĄäÿ■çŤś `Timer` çśżèőřå¡ŢçŽĎæŮűéŮťæŸŕéŠ§èąĺæŮűéŮťïijŇåźűåŇĚåŘńäžĘæL'ĂæIJL'ä
åęĆædIJä¡ääŔłæČşèőąçőŮèřěè£ZçÍŃæL'ĂèŁśèťźçŽĎCPUæŮűéŮťïijŇåžŤèřěä¡£çŤÍ `time.process_time()` æİěäžčæŻ£ïijŽ

```
t = Timer(time.process_time)
with t:
    countdown(1000000)
print(t.elapsed)
```

`time.perf_counter()` åŠŇ `time.process_time()` éČ¡äijŽè£ŤåŽđářŔæŢřå¡ćåijŔçŽĎçğŠæŢřæŮűéŮťãĂĆ ääđéŹĚçŽĎæŮűéŮťåĂijæšąæIJL'äżżä¡ŢæĎŔäźL'ïij

æŻťäď'ŽåĚşäžŐèőąæŮűäŠŇæĂğèČ¡åĹĘædŘçŽĎä¿Ńå■ŘèŕůäŔĆèĂČ14.13äřŔèŁĆãĂĆ

15.14 13.14 éŹŘäĹűäĘĚä■ŸäŠŇCPUçŽĎä¡£çŤÍéĞŔ

éŮőécŸ

ä¡ăæČşäŕźäIJÍUnixçşżçż§äÿŁéİćè£ŘèąŇçŽĎçÍŃäžŔèő¿ç¡őäĘĚä■ŸæĹŰCPUçŽĎä¡£çŤÍéŹŘäĹűãĂĆ

èğčåĘşæŰźæąĹ

`resource` æÍąäİŮèČ¡åŘŇæŮűæL'ğèąŇè£ŹäÿďäÿłäżżåŁąãĂĆä¿ŃåęĆïijŇèęĄéŹŘäĹűCPUæŮűéŮťïijI

```
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
```

çÍŃåžŔè£ŘèąŇæŮűïijŇSIGXCPU ä£ąåŔůåIJÍæŮűéŮť'è£ĞæIJ§æŮűèćńçŤ§æĹŘïijŇçĎűåŘŐæL'ğèąŇæý

èęĄéŹŘåĹűåĘĚå■Ÿä¡£çŤĺïijŇèő¿ç¡őåŔŕä¡£çŤĺçŽĎæĂżåĘĚå■ŸåĂijå■şåŔŕïijŇåęĆäÿŃïijŽ

```
import resource

def limit_memory(maxsize):
    soft, hard = resource.getrlimit(resource.RLIMIT_AS)
    resource.setrlimit(resource.RLIMIT_AS, (maxsize, hard))
```

åČŔè£Źæăůèő¿ç¡őäžĘåĘĚå■ŸéŹŘåĹűåŘŐïijŇçÍŃåžŔè£ŘèąŇåĹřæšąæIJL'åd'Žä¡ŹåĘĚå■ŸæŮűäijŽæŁŽ
MemoryError åijĆåÿÿãĂĆ

èőÍèőž

åIJÍæIJňèŁĆä¿Ńå■Řäÿ■ïijŇsetrlimit() åĞ¡æŢřèćńçŤĺæİěèő¿ç¡őçL'źåőŽètĎæžŘäÿŁéİćçŽĎè¡réŹŘ
è¡réŹŘåĹűæŸŕäÿĂäÿłåĂijïijŇå¡ŞèűĚè£ĞèŁŹäÿłåĂijçŽĎæŮűåĂŹæŞ■ä¡IJçşżçż§éĂŽåÿÿäijŽåŘŚéĂĄäÿĂäÿł
çąňéŹŘåĹűæŸŕçŤĺæİěæŇĞåőŽè¡réŹŘåĹűèČ¡èő¿åőŽçŽĎæIJĂåd'ğåĂijãĂĆéĂŽåÿÿæİěèőšïijŇè£ŹäÿłçŤśçşż
ař¡çőąçąňéŹŘåĹűåŘŕäžěæŤźåřŘäÿĂçĆźïijŇä¡ĘæŸŕæIJĂåě¡äÿ■èęĄä¡£çŤĺçŤĺæĹůè£ŽçĺŃåŐżä£őæŤźãĂĆ

setrlimit() åĞ¡æŢřè£ŸèČ¡èćńçŤĺæİěèő¿ç¡őå■Řè£ŽçĺŃæŢřéĞŘãĂĄæL'ŞåijĂæŰĞäžűæŢřäžěåŘŁç
æŻť'åd'Žèŕęæ ČĚèŕůåŘĆèĂČ resource æĺąåİŮçŽĎæŰĞæąčãĂĆ

éIJĂèęĄæşĺæĎŘçŽĎæŸŕæIJňèŁĆåĘĚåőźåŘłèČ¡éĂĆçŤĺäžŐUnixçşżçż§ïijŇåźűäÿŤäÿ■ä£İèŕĄæL'ĂæIJL
æŕŤåęĆæĹŚäżňåIJĺæţŃèŕŢçŽĎæŮűåĂŹïijŇåőĆèČ¡åIJĺLinuxäÿŁéİćæ■čåÿÿè£ŘèąŇïijŇä¡ĘæŸŕåIJĺOS
XäÿŁå■ť'äÿ■èČ¡ãĂĆ

15.15 13.15 åŘŕåŁĺäÿĂäÿłWEBæţŔèğĹåŹĺ

éŮőécŸ

ä¡ăæČşéĂŽè£ĞèĎŽæIJňåŘŕåŁĺæţŔèğĹåŹĺåźűæL'ŞåijĂæŇĞåőŽçŽĎURLç¡Śéąţ

èğčåĘşæŰźæąĹ

webbrowser æĺąåİŮèČ¡èćńçŤĺæİěåŘŕåŁĺäÿĂäÿłæţŔèğĹåŹĺïijŇåźűäÿŤäÿŐåźşåŘřæŮăåĚşãĂĆä¿ŃåęĆ

```
>>> import webbrowser
>>> webbrowser.open('http://www.python.org')
True
>>>
```

åőČäijŽä¡£çŤĺéžŸèőd'æţŔèğĹåŹĺæL'ŞåijĂæŇĞåőŽç¡ŚéąţãĂĆåęĆædIJä¡ăè£ŸæČşåŕźç¡ŚéąţæL'ŞåijĂæŰ

```
>>> # 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£ŹæăůåřśåŔŕäžěæL'ŞåijĂäÿĂäÿłæŰřçŽĎæţŔèğĹåŹĺçłŮåŔčæĹŰèĂĚæăĞç■¿ïijŇåŘłèęĄæţŔèğĹåŹĺæŤ

åęĆædIJä¡ăæČşæŇĞåőŽæţŔèğĹåŹĺçśžåđŃïijŇåŘŕäžěä¡£çŤĺ webbrowser.get() åĞ¡æŢřæİěæŇĞåőŽæ§Răÿłçĺ'źåőŽæţŔèğĹåŹĺãĂĆä¿ŃåęĆïijŽ

```
>>> c = webbrowser.get('firefox')
>>> c.open('http://www.python.org')
True
>>> c.open_new_tab('http://docs.python.org')
True
>>>
```

åŕźäžŐæŤŕæŇĄçŽĎæţŔèğĹåŹĺåŘ■çğřåĹŮèąĺåŔŕæ§ěéŸĚ'PythonæŰĞæąč <http://docs.python.org/3/library/webbrowser.html>'_

èőĺèőž

åIJĺèĎŽæIJňäÿ■æL'ŞåijĂæţŔèğĹåŹĺæIJL'æŮűåĂŹäijŽå¿ĹæIJL'çŤĺãĂĆä¿ŃåęĆïijŇæ§RăÿłèĎŽæIJňæL'ä¡ăæČşå£ńéĂ§æL'ŞåijĂäÿĂäÿłæţŔèğĹåŹĺæİěçąőä£ĺåőČåůšçżŔæ■čåÿÿè£ŘèąŇäžĘãĂĆ æĹŰèĂĚæŸŕæ§Răÿłçĺ'ŇåžŔäžěHTMLç¡ŚéąţæăijåijŔè¿ŞåĞžæŢřæ■őïijŇä¡ăæČşæL'ŞåijĂæţŔèğĹåŹĺæ§ěçIJŇäÿ■çőąæŸŕäÿŁéĺćåŞłçğ■æČĚåĘţïijŇä¡£çŤĺ webbrowser æĺąåİŮéČ¡æŸŕäÿĂäÿłçőĂå■Ţåðđç'ĺçŽĎèğčåĘşa

16 çňňå■ĄåŻŻçńäïijŽæţŃèŕŢãĂĄèřČèŕŢåŠŇåijĆåÿÿ

èŕŢéłŇè£ŸæŸŕå¿ĹæčŠçŽĎïijŇä¡ĘæŸŕèřČèŕŢïij§åřśæšąéĆčäźĹæIJL'èűčäžĘãĂĆäžŃåőđæŸŕïijŇåIJĺPyt

Contents:

16.1 14.1 æţŃèŕŢstdoutè¿ŞåĞž

éŮőécŸ

ä¡ăçŽĎçĺŃåžŔäÿ■æIJL'äÿłæŰźæşŢäijŽè¿ŞåĞžåĹřæăĞåĞĘè¿ŞåĞžäÿ■ïijĹsys.stdoutïijL'ãĂĆäź§åřśæŸŕè
ä¡ăæČşåĘźäÿłæţŃèŕŢæİèèŕĄæŸŐåőČïijŇçżźåőŽäÿĂäÿłè¿ŞåĚěïijŇçŻÿåžŢçŽĎè¿ŞåĞžèČ¡æ■čåÿÿæŸ¿çď'žå

èğčåĘşæŰźæąĹ

ä¡£çŤĺ `unittest.mock` æĺąåİŮäÿ■çŽĎ `patch()` åĞ¡æŢřïijŇ
ä¡£çŤĺèţůæİèéİđåÿÿçőĂå■ŢïijŇåŔřäžěäÿžå■ŢäÿłæţŃèŕŢæĺąæŃ§ `sys.stdout`
çĎűåŘŐåŽđæżŽïijŇ åźűäÿŤäÿ■äžğçŤ§åď'ğéĞŔçŽĎäÿť'æŮűåŔŸéĞŔæĹŰåIJĺæţŃèŕŢçŤĺä¿ŃçŻť'æŐěæŽť'éIJ

ä¡IJäÿžäÿĂäÿłä¿Ńå■ŘïijŇæĹŚäžňåIJĺ `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¿ŞåĞžåŔŚéĂĄåĹř `sys.stdout` ãĂĆ äÿžäžĘæţŃèŕŢè¿ŞåĞžçIJ§çŽĎåIJĺéĆčéĞŇïijŇä¡ăåŔřäžěä¡£çŤĺäÿĂäÿłæŽ£èžńåŕźèśąæİèæĺąæŃ
ä¡£çŤĺ `unittest.mock` æĺąåİŮçŽĎ `patch()` æŰźæşŢåŔřäžěå¿ĹæŰźä¿£çŽĎåIJĺæţŃèŕŢè£ŘèąŇçŽĎäÿŁ
åźűäÿŤå¡ŞæţŃèŕŢåőŇæĹŘæŮűåĂŹèĞłåŁĺè£ŤåŻđåőČäžňçŽĎåŐ§æIJL'çŁűæĂĄãĂĆäÿŃéĺćæŸŕåŕź
`mymodule` æĺąåİŮçŽĎæţŃèŕŢäžčçăĄïijŽ

```
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)

```
            mymodule.urlprint(protocol, host, domain)
            self.assertEqual(fake_out.getvalue(), expected_url)
```

èőÍèőž

urlprint() åĞ¡æŢřæŐěåŘŮäÿL'äÿłåŘĆæŢřïijŇæţŃèŕŢæŰźæşŢåijĂåğŇäijŽåĚĹèő¿ç¡őæŕŘäÿĂäÿłål
expected_url åŘŸéĞŘèćńèő¿ç¡őæĹŘåŇĚåŘńæIJ§æIJŻçŽĎè¿ŞåĞžçŽĎå■Ůçňęäÿšã ĂĆ

unittest.mock.patch() åĞ¡æŢřèćńçŤĺä¡IJäÿĂäÿłäÿŁäÿŃæŰĞçőąçŘĘåŹĺïijŇä¡£çŤĺ
StringIO åŕźèśąæİěäžčæŻ£ sys.stdout . fake_out
åŘŸéĞŘæŸŕåIJĺèřěè£ŻçĺŃäÿ■èćńåĹŻåžžçŽĎæĺąæŃ§åŕźèśąã ĂĆ åIJĺwith-
èŕ■åŘěäÿ■ä¡£çŤĺåőČåŘŕäžěæL'ğèąŇåŘĎçğ■æčĂæ§ěã ĂĆå¡Şwithèŕ■åŘěçżŞæİ§æŮűïijŇpatch
äijŽåřĘæL'ĂæIJL'äÿIJèě£æĄćåd'■åĹřæţŃèŕŢåijĂåğŃåL'■çŽĎçŁűæĂĄã ĂĆ
æIJL'äÿĂçĆźéIJĂèęĄæşĺæĎŘçŽĎæŸŕæ§ŘäžŻåŕźPythonçŽĎCæL'l'åśŢåŘŕèČ¡äijŽå£¡çŢěæŐL'
sys.stdout çŽĎéĚ■ç¡őèĂŇçŻť'æŐěåĘŹåĚěåĹřæăĞåĞĘè¿ŞåĞžäÿ■ã ĂĆ
éŹŘäžŐçŕĞåźĚïijŇæIJňèŁĆäÿ■äijŽæűL'åŘŁåĹřè£ŹæŰźéĺćçŽĎèőšèğčïijŇåőČéĂĆçŤĺäžŐçžŕPythonäżčçăĄă
åęĆæđIJä¡ăçIJ§çŽĎéIJĂèęĄåIJĺCæL'l'åśŢäÿ■æ■ŢèŐůI/OïijŇä¡ăåŘŕäžěåĚĹæL'ŞåijĂäÿĂäÿłäÿť'æŮűæŰĞäżű
æŻť'åd'ŽåĚşäžŐæ■ŢèŐůäžěå■Ůçňęäÿšå¡ćåijŔæ■ŢèŐůI/OåŠŇ StringIO
åŕźèśąèŕůåŘĆéŸĚ5.6åřŘèŁĆãĂĆ

16.2 14.2 åIJĺå■ŢåĚČæţŃèŕŢäÿ■çżŹåŕźèśąæL'Şèąěäÿą

éŮőécŸ

ä¡ăåĘŹçŽĎå■ŢåĚČæţŃèŕŢäÿ■éIJĂèęĄçżŹæŇĞåőŽçŽĎåŕźèśąæL'Şèąěäÿą¨ïijŇ
çŤĺæİěæŰ■èĺĂåőČäžňåIJĺæţŃèŕŢäÿ■çŽĎæIJ§æIJŻèąŇäÿžïijĹæŕŤåęĆïijŇæŰ■èĺĂèćńèřČçŤĺæŮűçŽĎåŘĆæŢ

èğčåĘşæŰźæąĹ

unittest.mock.patch() åĞ¡æŢřåŘŕèćńçŤĺæİěèğčåĘşè£ŹäÿłéŮőécŸãĂĆ
patch() è£ŸåŘŕèćńçŤĺä¡IJäÿĂäÿłèčĚéěřåŹĺãĂĄäÿŁäÿŃæŰĞçőąçŘĘåŹĺæĹŰå■ŢçŃňä¡£çŤĺïijŇář¡çőąåźűă
ä¿ŃåęĆïijŇäÿŃéİćæŸŕäÿĂäÿłářĘåőČå¡ŞåĄŽèčĚéěřåŹĺä¡£çŤĺçŽĎä¿Ńå■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£ŸåŘŕäžěèćńå¡ŞåĄŽäÿĂäÿłäÿŁäÿŃæŰĞçőąçŘĘåŹĺïijŽ

```
with patch('example.func') as mock_func:
    example.func(x)       # Uses patched example.func
    mock_func.assert_called_with(x)
```

æIJĂåŘŐïijÑä¡ăè£ŸåŔŕäžěæL'ŃåŁÍçŽĎä¡£çŤĺåőČæL'Şèąěäÿ Ąïij Ž

```
p = patch('example.func')
mock_func = p.start()
example.func(x)
mock_func.assert_called_with(x)
p.stop()
```

åęĆæđIJåŘŕèČ¡çŽĎèŕÍïijÑä¡ăèČ¡åd'§åŘäåŁäèčĚéěřåŹĺåŠÑäÿŁäÿŃæŰĞçőąçŘĘåŹÍæÍěçżŹåd'Žäÿłářžés

```
@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:
    ...
```

èőÍèőž

patch() æŐěåŘŮäÿĂäÿłåůšå■ŸåIJĺåŕźèśąçŽĎåĚÍèůŕå¿ĎåŘ■ïijÑåřĘåĚűæŻ£æ■ćäÿžäÿĂäÿłæŰřçŽĎåĂåŐ§æÍěçŽĎåĂijäijŽåIJÍèčĚéěřåŹĺåĞ¡æŢřæĹŰäÿŁäÿŃæŰĞçőąçŘĘåŹĺåőÑæĹŘåŘŐèĞłåŁÍæĄćåd'■åŻđæÍěãéżŸèőd'æČĚåĘţäÿŃïijÑæL'ĂæIJL'åĂijäijŽèćń MagicMock åőđä¿ŃæŻ£äžčãĂĆä¿ŃåęĆïijŽ

```
>>> x = 42
>>> with patch('__main__.x'):
...     print(x)
...
<MagicMock name='x' id='4314230032'>
>>> x
42
>>>
```

äy■è£ĞïijÑä¡ääŘŕäžěéĂŽè£ĞçżŹ patch() æŔŘä¿Żçňňäž Ñäÿłå ŔĆæŢřæÍěářĘåĂijæŻ£æ■ćæĹŘäżżä¡Ţ

```
>>> x
42
>>> with patch('__main__.x', 'patched_value'):
...     print(x)
...
```

(continues on next page)

```
patched_value
>>> x
42
>>>
```

èćńçŤĺæİěä¡IJäÿžæŻ£æ■ćåĂijçŽĎ MagicMock åőđä¿ŃèČ¡åđ'§æĺąæŃ§åŘŕèřČçŤĺåŕźèśąåŠŇåőđä¿ŃãĂ
äžŰäžňèőřå¡ŢåŕźèśąçŽĎä¡£çŤĺä£ąæĄŕåźűåĚĄèőÿä¡ăæL'ğèąŇæŰ■èĺĂæčĂæ§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')
>>>
```

äÿĂèĹňæİěèőšïijŇè£ŹäžŻæŞ■ä¡IJäijŽåIJĺäÿĂäÿłå■ŢåĚČæţŃèŕŢäÿ■åőŇæĹŘãĂĆä¿ŃåęĆïijŇåĄĞèő¿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)

```
    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■čåÿÿæÍěëőšïijŇè£ŹäýłåĞ¡æŢřäijŽä¡£çŤÍ urlopen() äžŐWe-bäýŁéÍćèŐůåŔŰæŢřæ■őåźűèğčæđŘåőČãĂĆ åIJÍå■ŢåĚČæţŃèŕŢäÿ■ïijŇä¡ăåŔŕäžěçżŹåőČäÿĂäýłécĎåĚĹåőŽ

```
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 æĺąåİŮäÿ■çŽĎ urlopen() åĞ¡æŢřèćńäÿĂäýłæĺąæŃ§åŕźèśąæŻ£äžčïijŇ èŕěåŕźèśąäijŽè£ŤåŻđäÿĂäýłåŇĚåŘńæţŃèŕŢæŢřæ■őçŽĎ ByteIO().

è£ŸæIJL’äÿĂçĆźïijŇåIJĺæL’Şèąěäÿ ĄæŮűæĹŚäžňä¡£çŤĺäžĘ example.urlopen æÍěäžčæŻ£ urllib.request.urlopen ãĂĆ å¡Şä¡ăåĹŻåžžèąěäÿĄçŽĎæŮűåĂŹïijŇä¡ăå£Ěéążä¡£çŤĺåőČäžňåIJĺæţŃèŕŢäžčçăĄäÿ■çŽĎåŘ■çğřãĂĆ çŤśäžŐæţŃèŕŢäžčçăĄä¡£çŤĺäžĘ from urllib.request import urlopen ,éĆčäźĹ dowprices() åĞ¡æŢř äÿ■ä¡£çŤĺçŽĎ urlopen() åĞ¡æŢřåőđéŹĚäÿŁářśä¡■äžŐ example æĺąåİŮäžĘãĂĆ

æIJňèŁĆåőđéŹĚäÿŁåŘłæŸřářź unittest.mock æĺąåİŮçŽĎäÿĂæňąæţĚářÍè¿Ďæ■ćãĂĆ æŻť åď ŽæŻť énŸçžğçŽĎçL’źæĂğïijŇèŕůåŔĆèĂČ åőŸæŰźæŰĞæąč

16.3 14.3 åIJÍå■ŢåĚČæţŃèŕŢäÿ■æţŃèŕŢåijĆåÿÿæČĚåEţ

éŮőécŸ

ä¡äæČşåĘŹäÿłæţŃèŕŢçŤÍä¿ŃæÌěåĞEçąőçŽĎåĹd'æŰ■æ§ŘäÿłåijĆåÿÿæŸŕåŘęèćńæŁŻåĞžãĂĆ

èğčåEşæŰźæąĹ

åŕźäžŐåijĆåÿÿçŽĎæţŃèŕŢåŔŕä¡£çŤÍ `assertRaises()` æŰźæşŢãĂĆ ä¿ŃåęĆïijŇåęĆæđIJä¡äæČşæţŃèŕŢæ§ŘäÿłåĞ¡æŢřæŁŻåĞžäžĘ `ValueError` åijĆåÿÿïijŇåČŔäÿŃéÌćè£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¡äæČşæţŃèŕŢåijĆåÿÿçŽĎåĚůä¡ŞåĂijïijŇéIJĂèęĄçŤÍåĹřåŔęåd'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■ŸåIJÍæĂğæŔŘä¿ŻäžĘäÿĂäÿłçőĂä¿£æŰźæşŢãĂĆ äÿĂäÿłåÿÿèğĄçŽĎéŹůéŸśæŸŕæL'ŃåŁÍåŐžè£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■æŰźæşŢçŽĎéŮőécŸåIJĺäžŐåőČå¿Ĺåőźæ ŸŞéĄŮæijŔåĚűäżŰæČĚåĘţïijŇæŕŤåęĆæšąæIJL'äżżä¡T
éĆčäźĹä¡ãè£Ÿå¿ŮéIJĂèęĄåćđåŁãåŘęåď'ŰçŽĎæčĂæţŃè£ĞçĺŃïijŇåęĆäÿŃéİćè£ŹæăůïijŽ

```
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'ĎçŘĘæL'ĂæIJL'çżĘèŁĆïijŇåŻãæ■d'ä¡ãåžŤèŕëä¡£çŤĺåőČãĂĆ

assertRaises() çŽĎäÿĂäÿłçijžçĆźæŸŕåőČæţŃäÿ■äžĘåijĆåÿÿåĚůä¡ŞçŽĎåĂijæŸŕåď'ŽåřŚãĂĆ
äÿžäžĘæţŃèŕŢåijĆåÿÿåĂijïijŇåŔŕäžëä¡£çŤĺ assertRaisesRegex() æŰźæşŢïijŇ
åőČåŔŕåŘŇæŮűæţŃèŕŢåijĆåÿÿçŽĎå■ŸåIJĺäžëåŘŁéĂŽè£Ğæ■čåĹŽåijŔåŇźéĚ■åijĆåÿÿçŽĎå■Ůçňęäÿšèąĺçd

```
class TestConversion(unittest.TestCase):
    def test_bad_int(self):
        self.assertRaisesRegex(ValueError, 'invalid literal .*',
                               parse_int, 'N/A')
```

assertRaises() åŠŇ assertRaisesRegex()
è£ŸæIJL'äÿĂäÿłåőźæŸŞå£¡çŢěçŽĎåIJřæŰźåřśæŸŕåőČäžňè£ŸèČ¡èćńå¡ŞåĄŽäÿŁäÿŃæŰĞçőąçŘĘåŹĺä¡£çŤĺ

```
class TestConversion(unittest.TestCase):
    def test_bad_int(self):
        with self.assertRaisesRegex(ValueError, 'invalid literal .*
↪'):
            r = parse_int('N/A')
```

ä¡Eä¡ãçŽĎæţŃèŕŢæűL'åŘŁåĹřåď'ŽäÿłæL'ğèąŇæ■ěéłď'çŽĎæŮűåĂŹè£Źçğ■æŰźæşŢåřšå¿ĹæIJL'çŤĺäžĘ

16.4 14.4 åřĘæţŃèŕŢè¿ŞåĞžçŤĺæŮėå£Ůèőřå¡ŢåĹřæŰĞäżűäÿ■

éŮőécŸ

ä¡ãåÿŇæIJŻåřĘå■ŢåĚČæţŃèŕŢçŽĎè¿ŞåĞžåĘŹåĹřåĹřæ§Řäÿłæ ŰĞäżűäÿ■åŐżïijŇèĂŇäÿ■æŸŕæL'Şå■řå

èğčåĘşæŰźæąĹ

è£ŘèąŇå■ŢåĚČæţŃèŕŢäÿĂäÿłåÿÿèğĄæŁĂæIJŕåřśæŸŕåIJĺæţŃèŕŢæŰĞäżűåžŢéČĺåŁãåĚëäÿŃéİćè£Źæőţ

```
import unittest

class MyTest(unittest.TestCase):
    pass
```

(continues on next page)

```
if __name__ == '__main__':
    unittest.main()
```

è£ŹæăůçŽĎèŕÍæţŃèŕŢæŰĞäžűůářśæŸŕůŔŕæĽ'ğèąŇçŽĎïijŇůźűäÿŤäijŽářĘè£ŘèąŇæţŃèŕŢçŽĎçżŞæđIJæ
åęĆæđIJä¡ăæČşéĞ■åőŽůŘŚè¿ŞůĞžïijŇářśéIJĂèęĄåČŔäÿŃéÍćè£ŹæăůŭŁőæŤź `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§åĚt'èűčçŽĎéČĺåĹĘåźűäÿ■æŸŕůřĘæţŃèŕŢçżŞæđIJéĞ■åőŽůŘŚůĹřäÿĂäÿłæŰĞäžűäÿ■ïijŇ
èĂŇæŸŕéĂŽè£Ğè£ŹæăůůĄŽůŘŚä¡ăåśŢçd'žäžĘ `unittest`
æĺąåİŮäÿ■äÿĂäžŽůĂijå¿ŮůĚşæşĺçŽĎůĘĚéČĺåůěä¡IJåŐ§çŘĘãĂĆ

`unittest` æĺąåİŮéęŰůĚĹäijŽçżĎèčĚäÿĂäÿłæţŃèŕŢåěŮäžűãĂĆ
è£ŹäÿłæţŃèŕŢåěŮäžűůŇĚåŘńäžĘä¡ăåőŽäźL'çŽĎůŘĎçğ■æŰźæşŢãĂĆäÿĂæŮęåěŮäžűçżĎèčĚåőŇæĹŘïijŇů

è£Źäÿd'æ■ěæŸŕůĹĘåijĂçŽĎïijŇ`unittest.TestLoader`
åőđä¿Ńèćńç̀ŤĺæİěçżĎèčĚæţŃèŕŢåěŮäžűãĂĆ `loadTestsFromModule()`
æŸŕůőČůőŽäźL'çŽĎæŰźæşŢäźŃäÿĂïijŇçŤĺæİěæŤűéŽĘæţŃèŕŢçŤĺä¿ŃãĂĆ åőČäijŽäÿž
`TestCase` çśżæL'ńæŘŕæ§ŘäÿłæĺąåİŮůźűářĘůĚüäÿ■çŽĎæţŃèŕŢæŰźæşŢæŘŕůřŰůĞžæİěãĂĆ
åęĆæđIJä¡ăæČşè£ŻèąŇçżĘçšŠůžęçŽĎæŐğůĹűïijŇ ůŘŕäżěä¡£çŤĺ
`loadTestsFromTestCase()` æŰźæşŢæİěäžŐæ§Řäÿłçżğæ L'£TestCaseçŽĎçśżäÿ■æŘŕůřŰæţŃèŕŢæŰźæ
`TextTestRunner` çśżæŸŕäÿĂäÿłæţŃèŕŢè£ŘèąŇçśżçŽĎä¿Ńů■ŘïijŇ
è£ŹäÿłçśżçŽĎäÿżèęĄçŤĺéĂŤæŸŕæL'ğèąŇæ§ŘäÿłæţŃèŕŢåěŮäžűäÿ■åŇĚåŘńçŽĎæţŃèŕŢæŰźæşŢãĂĆ
è£Źäÿłçśżèů§æL'ğèąŇ `unittest.main()` åĞ¡æŢřæL'Ăä¡£çŤĺçŽĎæţŃèŕŢè£ŘèąŇźĺæŸŕäÿĂæăůçŽĎãĂĆ
äÿ■è£ĞïijŇæĹŚäżňåIJĺè£ŹéĞŇårźåőČè£ŻèąŇäžĘäÿĂäžŽůĹŮůžŢåśĆéĚ■ç¡ôïijŇůŇĚæŃňè¿ŞůĞžæŰĞäžűůŠ
år¡çőąæIJňèŁĆä¿Ńů■Řäžčçăąå¿ĹårŚïijŇä¡ĘæŸŕèČ¡æŇĞårijä¡ăåęĆä¡Ţårź
`unittest` æąĘæđűè£ŻèąŇæŻt'è£ŻäÿĂæ■ěçŽĎèĞłåőŽäźL'ãĂĆ
èęĄæČşèĞłåőŽäźL'æţŃèŕŢåěŮäžűçŽĎèčĚéĚ■æŰźåijŔïijŇä¡ăåŘŕäżěåŕź `TestLoader`
çśżæL'ğèąŇæŻt'åd'ŽçŽĎæŞ■ä¡IJãĂĆ äÿžäžĘèĞłåőŽäźL'æţŃèŕŢè£ŘèąŇïijŇä¡ăåŘŕäżěædĎéĂääÿĂäÿłèĞłåůś
`TextTestRunner` çŽĎůŁ§èČ¡ãĂĆ èĂŇè£ŹäžŽůůšçżŔèűĚůĞžäžĘæIJňèŁĆçŽĎèŇČůŻt'ãĂĆ`unittest`
æĺąåİŮçŽĎæŰĞæąčåŕźåžŢåśĆåőđçŐřůŐ§çŘĘæIJL'æŻt'æůśåĚěçŽĎèőšèğčïijŇůŘŕäżěåŐžçIJŃçIJŃãĂĆ

16.5 14.5 å£¡çŢěæĹŰæIJ§æIJŻæţŃèŕŢåď śèťě

éŮőécŸ

ä¡ăæČşåIJĺå■ŢåĚČæţŃèŕŢäÿ■å£¡çŢěæĹŰæăĞèőřæ§ŘäžŻæţŃèŕŢäijŽæŇL'çĚğécĎæIJ§è£ŘèąŇåď śèť'ěã

èğčåĘşæŰźæąĹ

unittest æĺąåİŮæIJL'èčĚéěřåŹĺåŔŕçŤĺæİěæŐğåĹűåŕźæŇĞåőŽæţŃèŕŢæŰźæşŢçŽĎåď'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()
```

åęĆædIJä¡ăåIJĺMacäÿŁè£ŘèąŇè£Źæőţäžčçă ĄïijŇä¡ääijŽå¿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)

(çż■äÿŁéąţ)

```
Ran 5 tests in 0.002s

OK (skipped=2, expected failures=1)
```

èőÍèőž

skip() èčĚéěřåŹÍèČ¡èćńçŤÍæÍěå£¡çŢěæ§Řäÿłä¡ääÿ■æČşè£ŘèąŇçŽĎæţŃèŕŢãĂĆ skipIf() åŠŇ skipUnless() åŕźäžŐä¡ääŘłæČşåIJÍæ§ŘäÿłçĽźåőŽåźşåŘřæĹŰPythonçĽĹæIJňæĹŰåĚű ä¡£çŤÍ @expected çŽĎåď śèť ěèčĚéěřåŹÍæÍěæăĞèőřéĆčäžŻçąőåőŽäijŽåď śèť ěçŽĎæţŃèŕŢïijŇåźűäÿŤåŕźè£

å£¡çŢěæŰźæşŢçŽĎèčĚéěřåŹÍè£ŸåŘŕäžěèćńçŤÍæÍěčĚéěřæŢť äÿłæţŃèŕŢçśžïijŇæŕŤåęĆïijŽ

```
@unittest.skipUnless(platform.system() == 'Darwin', 'Mac specific␣
↪tests')
class DarwinTests(unittest.TestCase):
    pass
```

16.6 14.6 åď ĎçŘĘåď ŽäÿłåijĆåÿÿ

éŮőéćŸ

ä¡ăæIJL'äÿĂäÿłäžčçăĄçL'Ğæőţå ŔŕèČ¡äijŽæŁŻåĞžåď Žäÿłäÿ■åŘŇçŽĎåijĆåÿÿïijŇæĂŐæăůæL'■èČ¡äÿ■

èğčåĘşæŰźæąĹ

åęĆædIJä¡ääŔŕäžěçŤĺå■Ţäÿłäžčçă ĄåİŮåď ĎçŘĘäÿ■åŘŇçŽĎåijĆåÿÿïijŇåŔŕäžěårĘåőČäžňæŤ¿åĚěäÿĂä

```
try:
    client_obj.get_url(url)
except (URLError, ValueError, SocketTimeout):
    client_obj.remove_url(url)
```

åIJÍè£Źäÿłä¿Ńå■Řäÿ■ïijŇåĚČçěŰäÿ■äžżä¡ŢäÿĂäÿłåijĆåÿÿåŔŠçŤ§æŮűéČ¡äijŽæL'ğèąŇ remove_url() æŰźæşŢãĂĆ åęĆædIJä¡ääČşårźåĚűäÿ■æ§Řäÿłåij Ćåÿÿè£ŻèąŇäÿ■åŘŇçŽĎåď ĎçŘĘïijŇåŕ except èŕ■åŔěäÿ■ïijŽ

```
try:
    client_obj.get_url(url)
except (URLError, ValueError):
    client_obj.remove_url(url)
except SocketTimeout:
    client_obj.handle_url_timeout(url)
```

å¿Ĺåď ŽçŽĎåijĆåÿÿäijŽæIJL'åśĆçžğåĚşçşżïijŇårźäžŐè£Źçğ■æČĚåĘţïijŇä¡ääŔŕèČ¡ä¡£çŤĺåőČäžňçŽĎäÿ

```
try:
    f = open(filename)
except (FileNotFoundError, PermissionError):
    pass
```

åŔŕäżěèćńéĞ■åĘŹäÿžïijŽ

```
try:
    f = open(filename)
except OSError:
    pass
```

OSError æŸŕ FileNotFoundError åŠŇ PermissionError åijĆåÿÿçŽĎå§žçśżãĂĆ

èőĺèőž

åř¡çőąåď ĎçŘĘåď Žäÿłåij Ćåÿÿæ IJňěžńåźűæšąäżĂäźĹçL’źæőŁçŽĎïijŇäÿ■è£Ğä¡ăåŔŕäżěä¡£çŤĺ as åĚşéŤőå■ŮæÍěèŐůå¿Ůèćńæ ŁŻåĞžåijĆåÿÿçŽĎåijŢçŤĺïijŽ

```
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¿Ńå■Řäÿ■ïijŇ e åŔŸéĞŔæŇĞåŘŚäÿĂäÿłèćńæŁŻåĞžçŽĎ OSError åijĆåÿÿåðđä¿ŃãĂĆ è£Źäÿłå IJĺä¡ăæČşæŻt’è£ŻäÿĂæ■ěåĹĘæđŘè£Źäÿłåij Ćåÿÿç ŽĎæŮűåĂŹäijŽå¿ĹæIJL’çŤĺï

åŘŇæŮűè£ŸèęĄæşĺæĎŔçŽĎæŮűåĂŹ except èŕ■åŔěæŸŕéąžåžŔæčĂæ§ěçŽĎïijŇçňňäÿĂäÿłåŇźéĚ■ç ä¡ăåŔŕäżěå¿Ĺåőźæ ŸŞçŽĎæđĎéĂăåď 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■åŘěåźűæšąæIJL'æL'ğèąŇçŽĎåŐ§åŻăæŸŕ `OSError` æŽťäÿĂèĹňïijŇåőČåŔŕåŇźéĚ■ `FileNotFoundError` åijĆåÿÿïijŇ äžŐæŸŕåřśæŸŕçňňäÿĂäÿłåŇźéĚ■çŽĎãĂĆ åIJĺèřČèŕŢçŽĎæŮűåĂŹïijŇåęĆæđIJä¡ăåŕźæ§ŘäÿłçL'źåőŽåijĆåÿÿä¡ăåŘŕäžěéĂŽè£Ğæ§ěçIJŃèŕěåijĆåÿÿçŽĎ `__mro__` åśđæĂğæİěå£ńéĂ§æţŔèğĹãĂĆæŕŤåęĆïijŽ

```
>>> FileNotFoundError.__mro__
(<class 'FileNotFoundError'>, <class 'OSError'>, <class 'Exception'>
↪,
 <class 'BaseException'>, <class 'object'>)
>>>
```

äÿŁéİćåĹŮèąĺäÿ■äžżä¡ŢäÿĂäÿłçŽťåĹř `BaseException` çŽĎçśžéČ¡èČ¡èćńçŤĺäžŐ `except` èŕ■åŘěãĂĆ

16.7 14.7 æ■ŢèŐůæL'ĂæIJL'åijĆåÿÿ

éŮőécŸ

æĂŐæăůæ■ŢèŐůäžčçăĄäÿ■çŽĎæL'ĂæIJL'åijĆåÿÿïij§

èğčåĘşæŰźæąĹ

æČşèęĄæ■ŢèŐůæL'ĂæIJL'çŽĎåijĆåÿÿïijŇåŘŕäžěçŽť'æŐěæ■ŢèŐů `Exception` å■şåŘŕïijŽ

```
try:
    ...
except Exception as e:
    ...
    log('Reason:', e)        # Important!
```

è£ŹäÿłåřĘäijŽæ■ŢèŐůéŹď'äžĘ `SystemExit` ãĂĄ `KeyboardInterrupt` åŠŇ `GeneratorExit` äźŇåď'ŰçŽĎæL'ĂæIJL'åijĆåÿÿãĂĆ åęĆæđIJä¡ăè£ŸæČşæ■ŢèŐůè£ŹäÿL'äÿłåijĆåÿÿïijŇåřĘ `Exception` æŤźæĹŘ `BaseException` å■şåŘŕãĂĆ

èőÍèőž

æ■ŢèŐůæL'ĂæIJL'åijĆåÿÿéĂŽåÿÿæŸŕçŤšäžŐçÍŇåžŔåŚŸåIJÍæ§ŘäžŽåď'■æİĆæŞ■ä¡IJäÿ■åźűäÿ■èČ¡èőř åęĆæđIJä¡ăäÿ■æŸŕå¿ĹçżĘå£ČçŽĎäžžïijŇè£Źäź§æŸŕçijŰåĘŹäÿ■æŸŞèřČèŕŢäžčçăĄçŽĎäÿĂäÿłçőÅå■ŢæŰź

æ■čåŽăåęĆæ■d'ïijŇåęĆæđIJä¡ăéĂL'æŃl'æ■ŢèŐůæL'ĂæIJL'åijĆåÿÿïijŇéĆčäźĹåIJÍæ§ŘäÿłåIJřæŰźïijĹæ åęĆæđIJä¡ăæšąæIJL'è£Źæăůå ĄŽïijŇæIJL'æŮűåĂŹä¡ăçIJŃåĹřåijĆåÿÿæL'Şå■řæŮűåŘŕèČ¡æŚÿäÿ■çİĂåď'ťèĎ

```
def parse_int(s):
    try:
```

(continues on next page)

```
        n = int(v)
    except Exception:
        print("Couldn't parse")
```

èŕŢçİĂè£Řèą Ňè£ŹäÿłåĞ¡æŢřïijŇçżŞæđIJåęĆäÿŃïijŽ

```
>>> parse_int('n/a')
Couldn't parse
>>> parse_int('42')
Couldn't parse
>>>
```

è£ŹæŮűåĂŹä¡ăářśäijŽæŇăåď'ť'æČşïijŽâĂIJè£ŹåŠŃåŻđäžŃåŢŁïij§âĂİ åĄĞåęĆä¡ăåČŘäÿŃéİćè£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¡èŐůåŘŮåęĆäÿŃè¿ŞåĞžïijŇæŇĞæŸŐäžĘæIJL'äÿłçijŰçÍŃéŤŹèŕřïijŽ

```
>>> parse_int('42')
Couldn't parse
Reason: global name 'v' is not defined
>>>
```

å¿ĹæŸŐæŸ¿ïijŇä¡ăåžŤèŕěår¡åŔŕèČ¡ařĘåijĆåÿÿåď'ĎçŘĘåŹĺåőŽäźL'çŽĎçš¿åĞĘäÿĂäžŽãĂĆ äÿ■è£ĞïijŇèęĄæŸŕä¡ăå£Ěéążæ■ŢèŐůæL'ĂæIJL'åijĆåÿÿïijŇçąőä£İæL'Şå■řæ■čçąőçŽĎèŕŁæŰ■ä£ąæĄŕæĹŰå

16.8 14.8 åĹŻåżžèĞłåőŽäźL'åijĆåÿÿ

éŮőécŸ

åIJĺä¡ăæđĎåżžçŽĎåžŤçŤĺçÍŃåžŔäÿ■ïijŇä¡ăæČşařĘåžŢåśĆåijĆåÿÿåŇĚèčĚæĹŘèĞłåőŽäźL'çŽĎåijĆåÿÿ

èğčåĘşæŰźæąĹ

åĹŻåżžæŰřçŽĎåijĆåÿÿå¿ĹçőĂå■ŢâĂŤâĂŤåőŽäźL'æŰřçŽĎçśżïijŇèől'åőČçżğæL'£èĞł Exception ïijĹæĹŰèĂĚæŸŕäżżä¡ŢäÿĂäÿłåůšå■ŸåIJĺçŽĎåijĆåÿÿçśżådŇïijL'ãĂĆ ä¿ŃåęĆïijŇåęĆædIJä¡ăçijŰåĘŹç¡ŚçżIJçŻÿåĚşçŽĎçÍŃåžŔïijŇä¡ăåŔŕèČ¡äijŽåőŽäźL'äÿĂäžŽçśżäijijåęĆäÿŃçŽ

```
class NetworkError(Exception):
    pass
```

(continues on next page)

```
class HostnameError(NetworkError):
    pass

class TimeoutError(NetworkError):
    pass

class ProtocolError(NetworkError):
    pass
```

çĎűåŘŐçŤĺæĹůářšåŔŕäžěåČŔéĂŽåÿÿéĆčæăůä¡£çŤĺè£ŹäžŽåijĆåÿÿäžĘïijŇä¿ŃåęĆïijŽ

```
try:
    msg = s.recv()
except TimeoutError as e:
    ...
except ProtocolError as e:
    ...
```

èŐĺèŐž

èĞłåőŽäźL’åijĆåÿÿçśżåžŤèŕěæĂżæŸŕçżğæL’£èĞłåĘĚç¡őçŽĎ Exception çśżïijŇ æĹŰèĂĚæŸŕçżğæL’£èĞłéĆčäžŽæIJňèžńářšæŸŕäżŐ Exception çżğæL’£èĂŇæİěçŽĎçśżãĂĆ ář¡çőąæL’ĂæIJL’çśżåŘŇæŮűäź§çżğæL’£èĞł BaseException ïijŇä¡Eä¡ääÿ■åžŤèŕěä¡£çŤĺè£ŹäÿłåšžçśżæİěåőŽäźL’æŰřçŽĎåijĆåÿÿãĂĆ BaseException æŸŕäÿžçşżçż§éĂĂåĞžåijĆåÿÿèĂŇä£İçŢŹçŽĎïijŇæŕŤåęĆ KeyboardInterrupt æĹŰ SystemExit äžěåŘŁåĚűäžŰéĆčäžŽäijŽçżŹåžŤçŤĺåŔŚéĂĄä£ąåŘůèĂŇéĂĂåĞžçŽĎåijĆåÿÿãĂĆ åŽăæ■d’ïijŇæ■ŢèŐůè£ŹäžŽåijĆåÿÿæIJňèžńæšąäžĂäźĹæĎŔäźL’ãĂĆ è£ŹæăůçŽĎèŕİïijŇåĄĞåęĆä¡ăçżğæL’£ BaseException åŘŕèČ¡äijŽåŕijèĞt’ä¡ăçŽĎèĞłåőŽäźL’åijĆåÿÿäÿ■ä

åIJÍçÍŃåžŔäÿ■åijŢåĚěèĞłåőŽäźL’åijĆåÿÿåŘŕäžěä¡£å¿Ůä¡ăçŽĎäžčçăĄæŽt’åĚůåŘŕèŕżæĂğïijŇèČ¡æÿĚæŽ è£ŸæIJL’äÿĂçğ■èő¿èőąæŸŕåřĘèĞłåőŽäźL’åijĆåÿÿéĂŽè£ĞçżğæL’£çżĎåŘĹèţůæİěãĂĆåIJĺåď■æİĆåžŤçŤĺçÍŃ ä¡£çŤĺå§žçśżæİěåĹĘçżĎåŘĎçğ■åijĆåÿÿçśżäź§æŸŕå¿ĹæIJL’çŤĺçŽĎãĂĆåőČåŘŕäžěèőľ’çŤĺæĹůæ■ŢèŐůäÿĂä

```
try:
    s.send(msg)
except ProtocolError:
    ...
```

ä¡äè£ŸèČ¡æ■ŢèŐůæŻt’åď’ğèŇČåŻt’çŽĎåijĆåÿÿïijŇářšåČŔäÿŃéİćè£ŹæăůïijŽ

```
try:
    s.send(msg)
except NetworkError:
    ...
```

åęĆædIJä¡äæČşåőŽäźL’çŽĎæŰřåijĆåÿÿéĞ■åĘŹäžĘ __init__() æŰźæşŢïijŇ çąőä£İä¡ăä¡£çŤĺæL’ĂæIJL’åŔĆæŢřèřČçŤĺ Exception.__init__() ïijŇä¿ŃåęĆïijŽ

```
class CustomError(Exception):
    def __init__(self, message, status):
        super().__init__(message, status)
        self.message = message
        self.status = status
```

çIJŃäÿŁåŐżæIJL'çĆźåěĞæĂłïijŃäÿ■è£ĞExceptionçŽĎéżŸèőď'èąŃäÿžæŸŕæŐěåŔŮæL'ĂæIJL'äijăéĂŠç
.args åśđæĂğäÿ■. å¿Ĺåď'ŽåĚűäżŮåĞ¡æŢřåžŞåŠŃéČĺåĹĘPythonåžŞéżŸèőď'æL'ĂæIJL'åijĆåÿÿéČ¡å£Ěéąża
.args åśđæĂğïijŃ åŽăæ■ď'åęĆæđIJä¡ăå£¡çŢěäžĘè£ŹäÿĂæ■ěïijŃä¡ăäijŽåŔŠçŐřæIJL'äžŻæŮűåĂŹä¡ååőŽäź
äÿžäžĘæijŤçď'ž .args çŽĎä¡£çŤĺïijŃèĂČèŹŠäÿŃäÿŃéİćè£ŹäÿłäĄ£çŤĺåĘĚç¡őçŽĎ *Run-*
timeError‘ åijĆåÿÿçŽĎäžď'äžŠäijŽèŕĺïijŃ æşłæĎŔçIJŃraiseèŕ■åŔěäÿ■ä¡£çŤĺçŽĎåŔĆæŢřäÿłæŢřæŸŕæĂŐæă

```
>>> 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')
>>>
```

åĚşäžŐåĹŻåżžèĞłåőŽäźL'åijĆåÿÿçŽĎæŻt'åď'Žä£ąæĄŕïijŃèŕůåŔĆèĂČ‘Pythonåő˘ŸæŰźæŰĞæąč
<https://docs.python.org/3/tutorial/errors.html>‘_

16.9 14.9 æ■ŢèŐůåijĆåÿÿåŘŐæŁŻåĞžåŔęåď'ŰçŽĎåijĆåÿÿ

éŮőécŸ

ä¡ăæČşæ■ŢèŐůäÿĂäÿłåijĆåÿÿåŘŐæŁŻåĞžåŔęåď'ŰäÿĂäÿłäÿ■åŘŃçŽĎåijĆåÿÿïijŃåŘŃæŮűè£Ÿå¿ŮåIJ

èğčåĘşæŰźæąĹ

äÿžäžĘéŞ¿æŐěåijĆåÿÿïijŃä¡£çŤĺ `raise from` èŕ■åŔěæİěäžčæŻ£çőĂå■ŢçŽĎ `raise` èŕ■åŔěãĂĆ åőČäijŽèőľ'ä¡ååŘŃæŮűä£İçŢŹäÿd'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)

```
...
>>> example()
Traceback (most recent call last):
  File "<stdin>", line 3, in example
ValueError: invalid literal for int() with base 10: 'N/A'
```

äÿŁéİćçŽĎåijĆåÿÿæŸŕäÿŃéİćçŽĎåijĆåÿÿäžğçŤ§çŽĎçŻtˇæŐěåŐ§åŻäïijŽ

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "<stdin>", line 5, in example
RuntimeError: A parsing error occurred
>>>
```

åIJÍåŻđæžŕäÿ■åŔŕäžěçIJŃåĹřïijŇäÿďäÿłåijĆåÿÿéČ¡èćńæ■ŢèŐůãĂĆ èęĄæČşæ■ŢèŐůè£Źæăůç ŽĎåijĆåÿïijŇä¡ăåŔŕäžěä¡£çŤĺäÿĂäÿłçőĂå■ŢçŽĎ èŕ■åŔěãĂĆ äÿ■è£ĞïijŇä¡ăè£ŸåŔŕäžěéĂŽè£Ğæ§ěçIJŃåijĆåÿÿåŕźè śąçŽĎ åśđæĂğæİěèů§èÿłåijĆåÿÿéŞ¿ãĂĆä¿ŃåęĆïijŽ

except __cause__

```
try:
    example()
except RuntimeError as e:
    print("It didn't work:", e)

    if e.__cause__:
        print('Cause:', e.__cause__)
```

å¡ŞåIJÍ except åİŮäÿ■åŔĹæIJL’åŔęåď 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'
```

åIJÍåď ĎçŘĘäÿŁè£řåijĆåÿÿçŽĎæŮűåĂŹïijŇåŔęåď ŰäÿĂäÿłåijĆåÿÿåŔŠçŤ§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£ŽæŮűåĂŹïijŇNameError åijĆåÿÿèćńä¡IJäÿžçÍŇåžŘæIJĂçŻÍåijĆåÿÿèćńæŁŻåĞžïijŇèĂŇäÿ■æŸŕä¡■äžŐ

åęĆædIJïijŇä¡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¿èőąäżčçăĄæŮűïijŇåIJÍåŔęåd'ŰäÿĂäÿł except äżčçăĄåİŮäÿ■ä¡£çŤÍ raise èŕ■åŘěçŽĎæŮűåĂŹä¡äèęĄçĽźåĹńářŘå£ČäžĘãĂĆ åď'ğåď'ŽæŢřæČĚåĘţäÿŃïijŇè£Źçğ■ raise èŕ■åŘěéČ¡åžŤèŕěèćńæŤźæĹŘ raise from èŕ■åŘěãĂĆäź§åřśæŸŕèŕť'ä¡ääžŤèŕěä¡£çŤÍäÿŃéİćè£ŹçğĄ

```
try:
    ...
except SomeException as e:
    raise DifferentException() from e
```

è£ŹæăůåĄŽçŽĎåŐ§åŻăæŸŕä¡ääžŤèŕěæŸ¿çd'žçŽĎåřĘåŐ§åŻăéŞ¿æŐěţůæİěãĂĆ äź§åřśæŸŕèŕť'ïijŇDifferentException æŸŕçŻť'æŐěäžŐ SomeException èą■çŤ§èĂŇæİěãĂĆ è£Źçğ■åĚşçşżåŔŕäžěäžŐåŻđæžŕçżŞædIJäÿ■çIJŃåĞžæİěãĂĆ

åęĆædIJä¡ääČŔäÿŃéİćè£ŹæăůåĘŹäżčçăĄïijŇä¡ääż■çĎűäijŽå¿ŮåĹřäÿĂäÿłéŞ¿æŐěåijĆåÿÿïijŇ äÿ■è£Ğè£ŹäÿłåźűæšąæIJL'å¿ĹæÿĚæŹřçŽĎèŕť'æŸŐè£ŹäÿłåijĆåÿÿéŞ¿åĹřåžŢæŸŕåĘĚéČĺåijĆåÿÿè£ŸæŸŕæ§

```
try:
    ...
except SomeException:
    raise DifferentException()
```

å¡Şä¡ää¡£çŤÍ raise from èŕ■åŘěçŽĎèŕİïijŇåŕšå¿ĹæÿĚæěŽçŽĎèąÍæŸŐæŁŻåĞžçŽĎæŸŕçňňäžŇäÿłå

æIJĂåŘŐäÿĂäÿłä¿Ńå■Řäÿ■éŽŘèŮŔåijĆåÿÿéŞ¿ä£ąæĄŕãĂĆ åř¡çőąéŽŘèŮŔåijĆåÿÿéŞ¿ä£ąæĄŕäÿ■åĹl'äžŐåŻđæžŕïijŇåŘŇæŮűåőČäź§äÿćåd'śäžĘå¿Ĺåd'ŽæIJL'çŤÍçŽĎèŕČ äÿ■è£ĞäÿĞäžŃçŽĘåźşç■L'ïijŇæIJL'æŮűåĂŹåŔłä£İçŢŹéĂĆå¡ŞçŽĎä£ąæĄŕäź§æŸŕå¿ĹæIJL'çŤÍçŽĎãĂĆ

16.10 14.10 éĞ■æŰřæŁŻåĞžèćńæ■ŢèŐůçŽĎåijĆåÿÿ

éŮőécŸ

ä¡ăåIJĺäÿĂäÿł `except` åİŮäÿ■æ■ŢèŐůäžĘäÿĂäÿłåijĆåÿÿïijŇçŐřåIJĺæČşéĞ■æŰřæŁŻåĞžåőČãĂĆ

èğčåĘşæŰźæąĹ

çőĂå■ŢçŽĎä¡£çŤĺäÿĂäÿłå■ŢçŃňçŽĎ `rasie` èŕ■åŔěå■şåŔŕïijŇä¿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£ŹäÿłéŮőécŸéĂŽåÿÿæŸřå¡Şä¡ăéIJĂèęĄåIJĺæ■ŢèŐůåijĆåÿÿåŘŐæĽğèąŇæ§ŘäÿłæŞ■ä¡IJïijĹæŕŤåęĆèč
äÿĂäÿłå¿Ĺåÿÿèğ ĄçŽĎçŤĺæşŢæŸřåIJĺæ■ŢèŐůæĽĂæIJĽåijĆåÿÿçŽĎåď ĎçŘĘåŹĺäÿ■ïijŽ

```
try:
   ...
except Exception as e:
   # Process exception information in some way
   ...

   # Propagate the exception
   raise
```

16.11 14.11 è¿ŞåĞžè■ęåŚŁä£ąæĄŕ

éŮőécŸ

ä¡ăåÿŇæIJŻèĞłåůśçŽĎçÍŃåžŔèČ¡çŤ§æĹŘè■ęåŚŁä£ąæĄŕïijĹæŕŤåęĆåž§åijĆçL'źæĂğæĹŰä¡£çŤĺéŮőéć

èğčåĘşæŰźæąĹ

èęĄè¿ŞåĞžäÿĂäÿłè■ęåŚŁæűĹæĄŕïijŇåŔŕä¡£çŤĺ 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() çŽĎåŔĆæŢřæŸŕäÿĂäÿłè■ęåŚŁæűĹæĄŕåŠŇäÿĂäÿłè■ęåŚŁçśżïijŇè■ęåŚŁçśżæIJL'åęĆäÿŃåĞă DeprecationWarning, SyntaxWarning, RuntimeWarning, ResourceWarning, æĹŰ FutureWarning.

åŕźè■ęåŚçŽĎåď'ĎçŘĘåŔŰåĘşäžŐä¡ăåęĆä¡Ţè£ŘèąŇèğčéĞŁåŹĺäžěåŔŁäÿĂäžŻåĚűäžŰéĚ■ç¡őãĂĆ ä¿ŃåęĆïijŇåęĆæđIJä¡ää¡£çŤĺ -W all éĂL'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)
```

éĂŽåÿÿæİěèőšïijŇè■ęåŚŁäijŽè¿ŞåĞžåĹřæăĞåĞĘéŤŹèŕŕäÿŁãĂĆåęĆæđIJä¡ăæČşèőšè■ęåŚŁè¡ñæ■ćäÿžå -W error éĂL'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

åIJĺä¡ăçżť'æŁd'è¡ŕäžűïijŇæŔŘçd'žçŤĺæĹůæ§ŘäžŻä£ąæĄŕïijŇä¡ĘæŸŕåŔĹäÿ■éIJĂèęĄåřĘåĚűäÿŁå■ĞäÿŁ ä¿ŃåęĆïijŇåĄĞèő¿ä¡ăåĞĘåd'Ğä£őæŤźæ§ŘäÿłåĞ¡æŢřåžŞæĹŰæąĘæđűçŽĎåŁ§èČ¡ïijŇä¡ăåŔŕäžèåĚĹäÿžä¡ăè ä¡ăè£ŸåŔŕäžěè■ęåŚŁçŤĺæĹůäÿĂäžŻåŕźäžčçăĄæIJL'éŮőécŸçŽĎä¡£çŤĺæŰźåijŔãĂĆ

ä¡IJäÿžåŘęåd'ŰäÿĂäÿłåĘĚç¡őåĞ¡æŢřåžŞçŽĎè■ęåŚä¡£çŤĺä¿Ńå■ŘïijŇäÿŃéİćæijŤçd'žäžĘäÿĂäÿłæšąæ

```
>>> import warnings
>>> warnings.simplefilter('always')
>>> f = open('/etc/passwd')
```

(continues on next page)

```
>>> del f
__main__:1: ResourceWarning: unclosed file <_io.TextIOWrapper name=
↪'/etc/passwd'
 mode='r' encoding='UTF-8'>
>>>
```

éżŸèőď'æČĚåĘţäÿŃïijŇåźűäÿ■æŸŕæL'ĂæIJL'è■ęåŚŁæűĹæĄŕéČ¡äijŽåĞžçŐřãĂĆ-W éĂL'éąźèČ¡æŐğåĹűè■ęåŚŁæűĹæĄŕçŽĎè¿ŞåĞžãĂĆ -W all äijŽè¿ŞåĞžæL'ĂæIJL'è■ęåŚŁæűĹæĄŕïijŇ-W ignore å£¡çŢěæŐL'æL'ĂæIJL'è■ęåŚŁïijŇ-W error åřĘè■ęåŚŁè¡ňæ■ćæĹŘåijĆåÿÿãĂĆ åŔęåď'ŰäÿĂçğ■éĂL'æŃľ'ïijŇä¡ăè£ŸåŔŕäżěä¡£çŤĺ warnings.simplefilter() åĞ¡æŢřæŐğåĹűè¿ŞåĞžãĂĆ always åŔĆæŢřäijŽèőľ'æL'ĂæIJL'è■ęåŚŁæűĹæĄŕåĞžçŐřïijŇ` ignore å£¡çŢěèřČæL'ĂæIJL'çŽĎè■ęåŚŁïijŇerror åřĘè■ęåŚŁè¡ňæ■ćæĹŘåijĆåÿÿãĂĆ

åŕźäžŐçőĂå■ŢçŽĎçŤ§æĹŘè■ęåŚŁæűĹæĄŕçŽĎæČĚåĘţè£ŹäžŻåůšçżŔèűşåď'§äžĘãĂĆ warnings æĺąåİŮåŕźè£ĞæżďˇåŠŇè■ęåŚŁæűĹæĄŕåď'ĎçŘĘæŔŘä¿ŻäžĘåď'ğéĞŔçŽĎæŻť'énŸçžğçŽĎéĚ■ç¡æŻť'åď'Žä£ąæĄŕèŕůåŔĆèĂČ PythonæŰĞæąč

16.12 14.12 èřČèŕŢå§žæIJńçŽĎçÍŃåžŔåťľæžČéŤŹèŕŕ

éŮőécŸ

ä¡ăçŽĎçÍŃåžŔåťľæžČåŘŐèŕěæĂŐæăůåŐżèřČèŕŢåőČïij§

èğčåĘşæŰźæąĹ

åęĆæđIJä¡ăçŽĎçÍŃåžŔåŻääÿžæ§Řäÿłåij Cåÿÿè ĂŇåť'ľæžČïijŇè£ŘèąŇ python3 -i someprogram.py åŔŕæL'ğèąŇçőĂå■ŢçŽĎèřČèŕŢãĂĆ -i éĂL'éąźåŔŕèőľçÍŃåžŔçżŞæİ§åŘŐæL'ŞåijĂäÿĂäÿłäžď'äžŠåijŔshellãĂĆ çĎűåŘŐä¡ăářśèČ¡æ§ěçIJŃçŐŕáćČïijŇä¿ŃåęĆïijŇåĄĞèő¿ä¡ăæIJL'äÿŃéİćçŽĎäžčçăĄïijŽ

```
# sample.py

def func(n):
    return n + 10

func('Hello')
```

è£ŘèąŇ python3 -i sample.py äijŽæIJL'çśžäijijåęĆäÿŃçŽĎè¿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)

```
TypeError: Can't convert 'int' object to str implicitly
>>> func(10)
20
>>>
```

åęĆædIJä¡ăçIJŃäÿ■åĹřäÿŁéİćè£ŹæăůçŽĎïijŇåŔŕäżěåIJÍçÍŃåžŔåť'l'æžČåŘŐæL'ŞåijĂPythonçŽĎèřČèŕţ

```
>>> 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
>>>
```

åęĆædIJä¡ăçŽĎäżčçăĄæL'ĂåIJÍçŽĎçŐŕåćČå¿ĹéŽ¿èŐůåŔŰäžd'äžŠshellïijĹæŕŤåęĆåIJÍæ§Řäÿłæ IJ■åŁą
éĂŽåÿÿåŔŕäżěæ■ŢèŐůåijĆåÿÿåŘŐèĞłåůśæL'Şå■řèů§èÿłä£ąæĄŕãĂĆä¿ŃåęĆïijŽ

```
import traceback
import sys

try:
    func(arg)
except:
    print('**** AN ERROR OCCURRED ****')
    traceback.print_exc(file=sys.stderr)
```

èęĄæŸŕä¡ăçŽĎçÍŃåžŔæšąæIJL'åť'l'æžČïijŇèĂŇåŔłæŸŕäžğçŤ§äžĘäÿĂäžŻä¡ăçIJŃäÿ■æĞĆçŽĎçżŞædI
ä¡ăåIJÍæĎ§åĚť'èűčçŽĎåIJřæŰźæŔŠåĚěäÿĂäÿŃ `print()` èí■åŔěäź§æŸŕäÿłäÿ■éŤŹçŽĎéĂL'æŃl'ãĂĆ
äÿ■è£ĞïijŇèęĄæŸŕä¡ăæL'ŞçőŮè£ŹæăůåĄŽïijŇæIJL'äÿĂäžŻářŔæŁĂåůğåŔŕäżěåÿőåŁl'ä¡ããĂĆ
éęŰåĚĹïijŇ`traceback.print_stack()` åĞ¡æŢřäijŽä¡ăçÍŃåžŔè£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)

```
  File "<stdin>", line 3, in sample
  File "<stdin>", line 3, in sample
  File "<stdin>", line 5, in sample
>>>
```

åŘęåď Ű ïijÑä¡ãè£ŸåŔŕäžěåČŔäÿÑéÍćè£ŹæăůäĄ£çŤÍ pdb.set_trace() åIJÍäżżä¡ŢåIJřæŰźæĽŃåŁÍçŽĎåŘŕåŁÍèřČèŕŢåŹÍïijŽ

```
import pdb

def func(arg):
    ...
    pdb.set_trace()
    ...
```

å¡ŞçÍŃåžŔæŕŤè¿Čåď ğèĂŇä¡ăæČşèřČèŕŢæŐğåĹűæţĄçÍŃäžěåŔŁåĞ¡æŢřåŔĆæŢřçŽĎæŮűåĂŹè£Źäÿłåi ä¿ŃåęĆïijÑäÿĂæŮęèřČèŕŢåŹÍåijĂåğÑè£ŘèąÑïijÑä¡ăářśèČ¡åď §ä¡£çŤÍ print æÍěèğĆæţŃåŔŸéĞŔåĂijæĹŰæŢšåĞżæ§ŘäÿłåŚ¡äżď æŕŤåęĆ w æÍěèŐůåŔŰè£¡èÿłä£ąæĄŕãĂĆ

èőÍèőž

äÿ■èęĄåřĘèřČèŕŢåijĎçŽĎè£ĞäžŐåď■æİĆåŇŰãĂĆäÿĂäžŻçőĂå■ŢçŽĎéŤŹèŕŕåŔłéIJĂèęĄèğĆår§çÍŃåž åőđéŹĚçŽĎéŤŹèŕŕäÿĂèĹňæŸŕåăĘæăĹçŽĎæIJĂåŘŐäÿĂèąÑãĂĆ ä¡ăåIJÍåijĂåŘŚçŽĎæŮűåĂŹïijÑäź§åŔŕäžěåIJÍä¡ăéIJĂèęĄèřČèŕŢçŽĎåIJřæŰźæŔŠåĚěäÿĂäÿŃ print() åĞ¡æŢřæÍèèŕŁæŰ■ä£ąæĄŕïijĹåŔłéIJĂèęĄæIJĂåŘŐåŔŚåÿČçŽĎæŮűåĂŹåĹăéŹď è£ŹäžŻæĽŞå■ř

èřČèŕŢåŹÍçŽĎäÿĂäÿłåÿÿèğĄçŤÍæşŢæŸŕèğĆæţŃæ§ŘäÿłåůšçżŔåťľæžČçŽĎåĞ¡æŢřäÿ■çŽĎåŔŸéĞŔãĂ ç§ěéĄŞæĂŐæăůåIJÍåĞ¡æŢřåťľæžČåŘŐè£ŻåĚèèřČèŕŢåŹÍæŸŕäÿĂäÿłå¿ĹæIJĽçŤÍçŽĎæŁĂèČ¡ãĂĆ

å¡Şä¡ăæČşèğčåĽŰäÿĂäÿłéÍđåÿÿåď■æİĆçŽĎçÍŃåžŔïijÑåžŢåśĆçŽĎæŐğåĹűéĂżè¿Śä¡ăäÿ■æŸŕå¿ĹæÿĚ æŔŠåĚě pdb.set_trace() è£ŹæăůçŽĎèŕ■åŘěářšå¿ĹæIJĽçŤÍäžĘãĂĆ

åőđéŹĚäÿŁïijÑçÍŃåžŔäijŽäÿĂçŻťè£ŘèąÑåĹřçćřåĹř set_trace() èŕ■åŘëä¡■ç¡őïijÑçĎűåŘŐçńÑéľňè£ŻåĚěèřČèŕŢåŹÍãĂĆ çĎűåŘŐä¡ăářśåŔŕäžěåĄŽæŻťåď ŽçŽĎäžÑäžĘãĂĆ

åęĆæđIJä¡ăä¡£çŤÍIDEæÍěåĄŽPythonåijĂåŔŚïijÑéĂŽåÿÿIDEéČ¡äijŽæŔŘä¿ŻèĞłåůšçŽĎèřČèŕŢåŹÍæÍěæŻťåď Žè£ŹæŰźéÍćçŽĎä£ąæĄŕåŔŕäžěåŔĆèĂČä¡ăä¡£çŤÍçŽĎIDEæĽŃåĘÑãĂĆ

16.13 14.13 çżŹä¡ăçŽĎçÍŃåžŔåĄŽæĂğèČ¡æţŃèŕŢ

éŮőécŸ

ä¡ăæČşæţŃèŕŢä¡ăçŽĎçÍŃåžŔè£ŘèąÑæĽĂèŁśèťźçŽĎæŮűéŮťåźůåĄŽæĂğèČ¡æţŃèŕŢãĂĆ

èğčåĘşæŰźæąĹ

åęĆædIJä¡ăåŔłæŸŕçőÄå■ŢçŽĎæČşætŃèŕŢäÿŃä¡ăçŽĎçÍŃåžŔæŢt'ä¡ŞèŁśèt'źçŽĎæŮűéŮt'ïijŇéĂŽåÿÿä¡£çŤÍUnixæŮűéŮt'åĞ¡æŢřåřśèąŇäžĘïijŇæŕŤåęĆïijŽ

```
bash % time python3 someprogram.py
real 0m13.937s
user 0m12.162s
sys  0m0.098s
bash %
```

åęĆædIJä¡ăè£ŸéIJĂèęĄäÿĂäÿłçÍŃåžŔåŘĎäÿłçżĘèŁĆçŽĎèŕęçżĘæŁěåŚŁïijŇåŔŕäżěä¡£çŤÍ cProfile æÍąåÍŮïijŽ

```
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£Źäÿd'äÿłædĄçńŕäźŃéŮt'ãĂĆæŕŤåęĆä¡ăåůšçżŔç§ěéĄŞäžčçăĄè£Ì åŕźäžŐè£ŹäžŽåĞ¡æŢřçŽĎæĂğèČ¡æţŃèŕŢïijŇåŔŕäżěä¡£çŤÍäÿĂäÿłçőÄå■ŢçŽĎèčĚéěřåŹÍïijŽ

```
# timethis.py

import time
from functools import wraps

def timethis(func):
    @wraps(func)
    def wrapper(*args, **kwargs):
```

(continues on next 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¡£çŤĺè£ŹäýłèčĚéěřåŹĺïijŇåŔłéIJĂèęĄåřĘåĚűæŤ¿ç¡őåIJĺä¡ăèęĄè£ŻèąŇæĂğèČ¡æţŃèŕŢçŽĎåĞ¡æŢ

```
>>> @timethis
... def countdown(n):
...     while n > 0:
...             n -= 1
...
>>> countdown(10000000)
__main__.countdown : 0.803001880645752
>>>
```

èęĄæţŃèŕŢæ§Řäýłäżčçă ĄåİŮè£ŘèąŇæŮűéŮt'ïijŇä¡ăåŔŕäżěåőŽäźL'äýĂäýłäýŁäýŃæŰĞçőąçŘĘåŹĺïijŇ

```
from contextlib import contextmanager

@contextmanager
def timeblock(label):
    start = time.perf_counter()
    try:
        yield
    finally:
        end = time.perf_counter()
        print('{} : {}'.format(label, end - start))
```

äýŃéİćæŸŕä¡£çŤĺè£ŹäýłäýŁäýŃæŰĞçőąçŘĘåŹĺçŽĎä¿Ńå■ŘïijŽ

```
>>> with timeblock('counting'):
...     n = 10000000
...     while n > 0:
...             n -= 1
...
counting : 1.5551159381866455
>>>
```

åŕźäžŐæţŃèŕŢå¿ĹåřŔçŽĎäżčçăĄçL'Ğæőţè£ŘèąŇæĂğèČ¡ïijŇä¡£çŤĺ timeit æĺąåİŮäijŽå¿ĹæŰźä¿£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)

```
0.10836604500218527
>>>
```

timeit äijŽæL'ğèąŇçňňäýĂäýłåŔĆæŢřäý■èŕ■åŔě100äýĞæňąåźűèőąçőŮè£ŘèąŇæŮűéŮť'ãĂĆ çňňäžŇäýłåŔĆæŢřæŸŕè£ŘèąŇæţŃèŕŢäźŃåL'■éĚ■ç¡ő çŐŕåćČãĂĆåęĆæđIJä¡ăæČşæŤźåŔŸå¿łçŐŕæL'ğèąŇæň åŔŕäžěåČŔäýŃéİćè£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
>>>
```

èőÍèőž

å¡ŞæL'ğèąŇæĂğèČ¡æţŃèŕŢçŽĎæŮűåĂŹïijŇéIJĂèęĄæşÍæĎŔçŽĎæŸŕä¡ăèŐůåŔŰçŽĎçżŞæđIJéČ¡æŸŕè time.perf_counter() åĞ¡æŢřäijŽåIJÍçżŹåőŽåźşåŔřäýŁèŐůåŔŰæIJĂéńŸçš¿ážęçŽĎèőąæŮűåĂijãĂĆ äý■è£ĞïijŇåőČäż■çĎűè£ŸæŸŕå§žäžŐæŮűéŠ§æŮűéŮť'ïijŇå¿Ĺåď'ŽåŻăçť'ääijŽå¡śåŞ■åĹřåőČçŽĎçš¿çąőážęï åęĆæđIJä¡ăářźäžŐæL'ğèąŇæŮűéŮť'æŻť'æĎ§åĚť'èűčïijŇä¡£çŤÍ 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¡ăæČşè£ŻèąŇæŻť'æůśåĚěçŽĎæĂğèČ¡åĹĘæđŘïijŇéĆčäźĹä¡äéIJĂèęĄèŕęçżĘéŸİ time ãĂĄtimeit åŠŇåĚűäżŰçŻÿåĚşæĺąåİŮçŽĎæŰĞæąčãĂĆ è£Źæăůä¡ăåŕäžěçŘĘèğčåŠŇåźşåŔřçŻÿåĚşçŽĎåůőåijĆäžěåŔĽäÿĂäžŻåĚűäžŰéŹůéŸśãĂĆ è£ŸåŕäžěåŔĆèĂČ13.13ařŔèŁĆäý■çŻÿåĚşçŽĎäÿĂäýłåĹŻåžžèőąæŮűåŹĺçśżçŽĎä¿Ńå■ŘãĂĆ

16.14 14.14 åŁäéĂ§çÍŃåžŕè£ŘèąŇ

éŮőécŸ

ä¡ăçŽĎçÍŃåžŔè£ŘèąŇåd'łæĚćïijŇä¡ăæČşåIJĺäÿ■ä¡£çŤĺåd'■æİĆæŁĂæIJŕæŕŤåęĆCæL'l'åśŢæĹŰJITçijŰ

èğčåĘşæŰźæąĹ

åĚşäžŐçÍŃåžŔäijŸåŇŰçŽĎçňňäÿĂäÿłåĞĘåĹŹæŸŕâĂIJäÿ■èęĄäijŸåŇŰâĂİïijŇçňňäžŇäÿłåĞĘåĹŹæŸŕ
åęĆæđIJä¡ăçŽĎçÍŃåžŔè£ŘèąŇçijŞæĚćïijŇéęŰåĚĹä¡ăå¿Ůä¡£çŤÍ14.13åřŔèŁĆçŽĎæŁĂæIJŕåĚĹåŕźåőČè£Żèă

éĂŽåÿÿæİěèőšä¡ăäijŽåŔŞçŐřä¡ăå¿ŮçÍŃåžŔåIJÍåřŚæŢřåĞääÿłçČ■çĆźåIJřæŰźèŁśèť´źäžĘåď´ğéĞŔæŮűé
æŕŤåęĆåĘĚå■ŸçŽĎæŢřæ■őåď´ĎçŘĘå¿łçŐŕãĂĆäÿĂæŮęä¡ăåőŽä¡■åĹřè£ŹäžŻçĆźïijŇä¡ăåřśåŔŕäžěä¡£çŤĺäÿŇ

ä¡£çŤĺåĞ¡æŢř

å¿ĹåďŽçÍŃåžŔåŚŸåĹŽåijĂåğŃäijŽä¡£çŤÍPythonèŕ■èÍĂåĘŹäÿĂäžŻçőĂå■ŢèĎŽæIJñãĂĆ
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§ěéĄŞïijŇåČŔè£ŹæăůåőŽäźL’åIJĺåĚĺåśĂèŇČåŽť´çŽĎäžčçăĄè£ŘèąŇèţůæİěèęĄæŕŤåőŽ
è£Źçğ■éĂ§åžęåůőåijĆæŸŕçŤśäžŐåśĂéČĺåŔŸéĞŔåŠŇåĚĺåśĂåŔŸéĞŔçŽĎåőđçŐřæŰźåijŔïijĹä¡£çŤĺåśĂéČĺ
åŽăæ■d’ïijŇåęĆæđIJä¡ăæČşèől’çÍŃåžŔè£ŘèąŇæŽť´å£ńäžŽïijŇåŔłéIJĂèęĄåřĘèĎŽæIJňèŕ■åŔěæŤ¿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§åžęçŽĎåůőåijĆåŔŰåĘşäžŐåőđéŹĚè£ŘèąŇçŽĎçÍŃåžŔïijŇäÿ■è£Ğæăźæ■őçżŔéłŇïijŇä¡£çŤĺåĞ¡æŢï
30%çŽĎæĂğèČ¡æŔŘå■ĞæŸŕå¿ĹåÿÿèğĄçŽĎãĂĆ

åř¡åŔŕèČ¡åŐżæŐL’åśđæĂğèő£éŰő

æŕŔäÿĂæňąä¡£çŤĺçĆź(.)æŞ■ä¡IJçňęæİěèő£éŰőåśđæĂğçŽĎæŮűåĂŹäijŽåÿęæİěéćİåď´ŰçŽĎåijĂéŤĂãĂ
åőČäijŽèğęåŔŠçL’źåőŽçŽĎæŰźæşŢïijŇæŕŤåęĆ `__getattribute__()` åŠŇ
`__getattr__()` ïijŇè£ŹäžŽæŰźæşŢäijŽè£ŻèąŇå■ŮåĚÿæŞ■ä¡IJæŞ■ä¡IJãĂĆ

éĂŽåÿÿä¡ăåŔŕäžěä¡£çŤĺ `from module import name`
è£ŹæăůçŽĎåŕijåĚěå¡ćåijŔïijŇäžěåŔŁä¡£çŤĺçżŠåőŽçŽĎæŰźæşŢãĂĆ
åĄĞèő¿ä¡ăæIJL’åęĆäÿŃçŽĎäžčçăĄçL’Ğæőţïijž

```python
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)
```

åIJÍæĹŚäžňæIJžåŹÍäýŁéİćæţŃèŕŢçŽĎæŮűåĂŹïijŇè£ŹäýłçíŃåžŔèŁśèť źäžĘåď ğæęĆ40çğŠãĂĆçŐřåIJÍa
`compute_roots()` åĞ¡æŢřåęĆäýŃïijŽ

```python
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 ğæęĆæŸŕ29çğŠãĂĆåŤŕäýĂäý■åŘŇäźŃåď ĎåřśæŸŕæűĹéŹ
çŤÍ `sqrt()` äžčæŻ£äžĘ `math.sqrt()` ãĂĆ `The result.append()` æŰźæşŢèćńèţŃçżŹäýĂäýłåśĂéČĺåŔŸéĞŔ `result_append` ïijŇçĎűåŘŐåIJÍåĘĚéČĺå¿łçŐŕäý■ä¡£çŤĺåőČãĂĆ

äý■è£ĞïijŇè£ŹäžŻæŤźåŔŸåŘłæIJL'åIJÍåď ğéĞŔéĞ■åď ■äžčçăĄäý■æL'■æIJL'æĎŔäźL'ïijŇærŤåęĆå¿łçĆ
åŻăæ■ď ïijŇè£ŹäžŻäijŸåŇŰäź§åŘłæŸŕåIJÍæ§ŘäžŻçL'źåőŽåIJřæŰźæL'■åžŤèŕěèćńä¡£çŤĺãĂĆ

çŘĘèğčåśĂéČĺåŔŸéĞŔ

äźŃåL'■æŔŘè£ĞïijŇåśĂéČĺåŔŸéĞŔäijŽærŤåĚĺåśĂåŔŸéĞŔè£ŘèąŇéĂ§åžęå£ńãĂĆ
åŕźäžŐéćŚçźĄèő£éŮőçŽĎåŘ■çğřïijŇéĂŽè£ĞårĘè£ŹäžŻåŘ■çğřåŔŸæĹŘåśĂéČĺåŔŸéĞŔåŔŕäžěåŁăéĂ§çÍŃä
ä¿ŃåęĆïijŇçIJŃäýŃäźŃåL'■åŕźäžŐ `compute_roots()` åĞ¡æŢřè£Żèą Ňä£őæŤźåŘŐçŽĎçL'ĹæIJňïijŽ

```python
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£ŹäýłçL'ĹæIJňäý■ïijŇ`sqrt` äžŐ `match` æÍąåİŮèćńæŃ£åĞžåźűæŤ¿åĚěäžĘäýĂäýłåśĂéČĺåŔŸéĞŔ

åęĆædIJä¡äè£ŘèąÑè£Źäÿłäżčçă Ąïij Ñåď ğæęĆèŁśèť ź25çğŠïijĹåŕźäžŐäźŃåL’■29çğŠåŔĹæŸŕäÿĂäÿłæŤźè£Żź
è£Źäÿłéćİåď ŰçŽĎåŁäéĂ§åŐ§åŻäæŸŕåŻääÿžåŕźäžŐåśĂéČĺåŔŸéĞŔ `sqrt`
çŽĎæ§ěæL’¿èęĄå£ńäžŐåĚĺåśĂåŔŸéĞŔ `sqrt`

åŕźäžŐçśżäÿ■çŽĎåśđæĂğèő£éŮőäź§åŘŇæăůéĂĆçŤĺäžŐè£Źäÿłå Ő§çŘĘãĂĆ
éĂŽåÿÿæİěèőšïijŇæ§ěæL’¿æ§Řäÿłå Äijæŕ Ťåę Ć `self.name`
äijŽæŕŤèő£éŮőäÿĂäÿłåśĂéČĺåŔŸéĞŔèęĄæĚćäÿĂäžŻãĂĆ åIJĺåĘĚéČĺå¿łçŐŕäÿ■ïijŇåŔŕäžěårĘæ§Řäÿłé IJĂè

```
# 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£éŮőãĂĄæŔ
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£Żèą Ñäÿ Ääÿłçő Åå■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
>>>
```

åŘŕäžěçIJŃåĹřïijŇèő£éŮőåśđæĂğyçŻÿæŕŤåśđæĂğxèĂŇèĺĂæĚćçŽĎäÿ■æ■ćäÿĂçĆźçĆźïijŇåď ğæęĆæ
åęĆædIJä¡ääIJĺæĎŘæĂğèČ¡çŽĎèŕİïijŇéĆčäźĹåřśéIJĂèęĄéĞ■æŰřåőąèğĘäÿŃåŕźäžŐyçŽĎåśđæĂğèő£éŮőåŽ

åęĆæđIJæšąæIJL’å£ĚèęĄïijŇářšä¡£çŤĺçőĂå■Ţåśđæ˘ĂğåŘğãĂĆ
åęĆæđIJäžĚäžĚæŸŕåŽääÿžåĚűäžŰçijŰçĺŃèŕ■èĺĂéIJĂèęĄä¡£çŤĺgetter/setteråĞ¡æŢřářšåŐżä£őæŤźäžčçăĄéč

ä¡£çŤĺåĘĚç¡őçŽĎåőźåŹĺ

åĘĚç¡őçŽĎæŢřæ■őçśžåđŃæŕŤåęĆå■Ůçňęäÿšã˘ĂĄåĚČçżĎãĂĄåĹŰèąĺãĂĄéŽĘåŘĹåŠŇå■ŮåĚÿéČ¡æŸŕ
åęĆæđIJä¡ăæČşèĞłåůśåőđçŐřæŰřçŽĎæŢřæ■őçżŞæđĎïijĹæŕŤåęĆéŞ¿æŐěåĹŰèąĺãĂĄåźşèąąæăŚç■L’ïijL’ïijŇ
éĆčäźĹèęĄæČşåIJĺæĂğèČ¡äÿŁè¿¿åĹřåĘĚç¡őçŽĎéĂ§åžęåĞääźŐäÿ■åŘŕèČ¡ïijŇåŽäæ■d’ïijŇè£ŸæŸŕäźŰäźŰ

éĄ£åĚ■åĹŻåžžäÿ■å£ĚèęĄçŽĎæŢřæ■őçżŞæđĎæĹŰåd’■åĹű

æIJL’æŮűåĂŹçĺŃåžŘåŚŸæČşæŸ¿æŚĘäÿŃïijŇæđĎéĂääÿĂäžŻåźűæšąæIJL’å£ĚèęĄçŽĎæŢřæ■őçżŞæđ

```
values = [x for x in sequence]
squares = [x*x for x in values]
```

äź§èőÿè£ŹéĞŇçŽĎæČşæşŢæŸŕéęŰåĚĹåřĘäÿĂäžŻåĂijæŤűéŽĘåĹřäÿĂäÿłåĹŰèąĺäÿ■ïijŇçĎűåŘŐä¡£çŤ
äÿ■è£ĞïijŇçňňäÿĂäÿłåĹŰèąĺåőŇåĚĺæšąæIJL’å£ĚèęĄïijŇåŘŕäžěçőĂå■ŢçŽĎåČŘäÿŃéÍćè£ŹæăůåĘŹïijŽ

```
squares = [x*x for x in sequence]
```

äÿŐæ■d’çŻÿåĚşïijŇè£ŸèęĄæşĺæĎŔäÿŃéĆčäžŻåŕźPythonçŽĎåĚśäžńæŢřæ■őæIJžåĹűè£ĞäžŐåĄŔæL’ğ
æIJL’äžŻäžžåźűæšąæIJL’å¿Ĺåě¡çŽĎçŘĘèğčæĹŰä£ąäżżPythonçŽĎåĘĚå■ŸæĺąåđŃïijŇæżěçŤĺ
`copy.deepcopy()` äźŃçśżçŽĎåĞ¡æŢřăĂĆ éĂŽåÿÿåIJĺè£ŹäžŻäżčçăĄäÿ■æŸŕåŔŕäžěåŐżæŐL’åd’■åĹűæŞ

èőÍèőž

åIJĺäijŸåŇŰäźŃåL’■ïijŇæIJL’å£ĚèęĄåĚĹçăŤçl’űäÿŃä¡£çŤĺçŽĎçőŮæşŢăĂĆ
éĂL’æŃl’äÿĂäÿłåd’■æİĆåžęäÿž O(n log n) çŽĎçőŮæşŢèęĄæŕŤä¡ăåŐżèŕČæŢt’äÿĂäÿłåd’■æİĆåžęäÿž
O(n**2) çŽĎçőŮæşŢæL’ĂåÿęæİěçŽĎæĂğèČ¡æŘŘå■ĞèęĄåd’ğå¿Ůåd’ŽãĂĆ

åęĆæđIJä¡ăèğL’å¿Ůä¡ăè£ŸæŸŕå¿Ůè£ŻèąŇäijŸåŇŰïijŇéĆčäźĹèŕůäžŐæŢt’ä¡ŞèĂČèŹŚãĂĆ
ä¡IJäÿžäÿĂèĹńåĞĘåĹŹïijŇäÿ■èęĄåŕźçĺŃåžŘçŽĎæŕŘäÿĂäÿłéČĺåĹĘéČ¡åŐżäijŸåŇŰ,åŻăäÿžè£ŹäžŻä£őæŤź
ä¡ăåžŤèŕěäÿŞæşĺäžŐäijŸåŇŰäžğçŤ§æĂğèČ¡çŞűécĹçŽĎåIJŕæŰźïijŇæŕŤåęĆåĘĚéČĺå¿łçŐŕãĂĆ

ä¡ăè£ŸèęĄæşĺæĎŔå¿ő åřŘäijŸåŇŰçŽĎçżŞæđIJãĂĆä¿ŃåęĆèĂČèŹŚäÿŃéÍćåĹŻåžžäÿĂäÿłå■ŮåĚÿçŽĎ

```
a = {
    'name' : 'AAPL',
    'shares' : 100,
    'price' : 534.22
}

b = dict(name='AAPL', shares=100, price=534.22)
```

åŘŐéÍćäÿĂçğ■åĘŹæşŢæŻť’çőĂæť’ĄäÿĂäžŻïijĹä¡ăäÿ■éIJĂèęĄåIJĺåĚşéŤőå■ŮäÿŁè¿ŞåĚěåijŢåŕůïijL’ãĂ
äÿ■è£ĞïijŇåęĆæđIJä¡ăårĘè£ŹäÿďäÿłäžčçăĄçL’Ğæőţè£ŻèąŇæĂğèČ¡æţŃèŕŢåŕźæŕŤæŮűïijŇäijŽåŔŚçŐŕä¡£ç
`dict()` çŽĎæŰźåijŔäijŽæĚćäžĘ3åĂ■ãĂĆ çIJŃåĹřè£ŹäÿłïijŇä¡ăæŸŕäÿ■æŸŕæIJL’åĘšåŁĺæŁŁæL’ĂæIJL’ä¡ă
`dict()` çŽĎäžčçăĄéČ¡æŻ£æ■ćæĹŘçňňäÿĂçğ■ãĂĆ äÿ■åd’§ïijŇèĄłæŸŐçŽĎçĺŃåžŘåŚŸåŔłäijŽåĘşæşĺäžŰ

åęĆæđIJä¡ăçŽĎäijŸåŇŰèęĄæśĆæŕŤè¿ČéńŸïijŇæIJňèŁĆçŽĎè£ŹäžŻçőĂå■ŢæŁĂæIJŕæżąęűşäÿ■äžĘïij
ä¿ŃåęĆïijŇPyPyåůěçĺŃæŸŕPythonèğčéĞŁåŹĺçŽĎåŘęåd’ŰäÿĂçğ■åődçŐřïijŇåőČäijŽåĹĘæđŘä¡ăçŽĎçĺŃåžł

åőČæIJL’æŮűåĂŹèČ¡æđĄåd’ğçŽĎæŔŘå■ĞæĂğèČ¡ïijŇéĂŽåÿÿåŔŕäžěæŐěè£ŚCäžčçăĄçŽĎéĂ§åžęãĂĆ
äÿ■è£ĞåŔŕæČIJçŽĎæŸŕŕijŇåĹřåĘŹè£ŹæIJñäźęä¡■ç¡őïijŇPyPyè£Ÿäÿ■èČ¡åőŇåĚíæŤŕæŇĄPython3.
åŻăæ■d’ïijŇè£ŹäÿłæŸŕä¡ăářĘæİěéIJĂèęĄåŐžçăŤçľűçŽĎãĂĆä¡ăè£ŸåŔŕäžěèĂČèŹŚäÿŃNumbaåůěçÍŃïijŇ
NumbaæŸŕäÿĂäÿłåIJÍä¡ăä¡£çŤÍèČĚéěřåŹÍæİěéĂL’æŃl’PythonåĞ¡æŢřè£ŻèąŇäijŸåŇŰæŮűçŽĎåŁÍæĂĄçijŰ
è£ŹäžŻåĞ¡æŢřäijŽä¡£çŤÍLLVMèćńçijŰèŕŚæĹŘæIJňåIJřæIJžåŹÍçăĄãĂĆåőČåŘŇæăůåŔŕäžěædĄåd’ğçŽĎæŔ
ä¡ĘæŸŕŕijŇèů§PyPyäÿĂæăůïijŇåőČåŕźäžŐPython 3çŽĎæŤŕæŇĄÇŐřåIJÍè£ŸåĄIJçŢŹåIJÍåőđéłŇéŸűæőţãĂĆ

æIJĂåŘŐæĹŚåijŢçŤÍJohn Ousterhoutèŕť’è£ĞçŽĎèŕÍä¡IJäÿžçżŞář¿ïijŽâĂIJæIJĂåě¡çŽĎæĂğèČ¡äijŸåŇŰ
çŻť’åĹřä¡ăçIJ§çŽĎéIJĂèęĄäijŸåŇŰçŽĎæŮűåĂŹåĘ■åŐżèĂČèŹŚåőČãĂĆçąőä£Íä¡ăçÍŃåžŔæ■čçąőçŽĎè£Řè

17 çňňå■ĄäžŤçńäïijŽCèŕ■èÍĂæĽ’åśŢ

æIJňçńăçİĂçIJijäžŐäżŐPythonèő£éŮőCäžčçăĄçŽĎéŮőécŸãĂĆèőÿåď’ŽPythonåĘĚç¡őåžŞæŸŕçŤÍCåĘŹ
èő£éŮőCæŸŕèől’PythonçŽĎåŕźçŐřæIJL’åžŞè£ŻèąŇäžd’äžŠäÿĂäÿłéĞ■èęĄçŽĎçżĎæĹŘéČÍåĹĘãĂĆ
è£Źäź§æŸŕäÿĂäÿłå¡Şä¡äéÍćäÿt’äżŐPython 2 åĹř Python 3æĽ’l’åśŢäžčçăĄçŽĎéŮőécŸãĂĆ
èŹ¡çĎűPythonæŔŘä¿ŻäžĘäÿĂäÿłáź£æşŻçŽĎçijŰçÍŃAPIïijŇåőđéŹĚäÿŁæIJL’å¿Ĺåd’ŽæŰźæşŢæİěåd’ĎçŘĘ
çŻÿæŕŤèŕŢåŻ¿çżŹåĞžåŕźäžŐæŕŔäÿĂäÿłåŔŕèČ¡çŽĎåůěåĚůæĹŰæŁĂæIJŕçŽĎèŕęçżĘåŔĆèĂČïijŇ
æĹŚäźĹéĞĞçŤÍçŽĎæŸŕæŸŕéŻĘäÿ■åIJÍäÿĂäÿłåŕŘçĽ’ĞæőţçŽĎC++äžčçăĄïijŇäžěåŘŁäÿĂäžŻæIJL’äžčèąÍæ
è£ŹäÿłçŻőæăĞæŸŕæŔŘä¿ŻäÿĂçşžåĹŰçŽĎçijŰçÍŃæÍąæİ£ïijŇæIJL’çżŔéłŇçŽĎçÍŃåžŘåŚŸåŕźäžěæĽ’l’åśŢè

è£ŹéĞŇæŸŕæĹŚäžňåřĘåIJÍåd’ğéČÍåĹEçğŸçś■äÿ■åůěä¡IJçŽĎäžčçăĄïijŽ

```c
/* 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)

```
/* 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£ŹæőţäžčçăĄåÑĚåŘńäžĘåď Žçğ■äÿ■åŘÑçŽĎCèŕ■èÍĂçijŰçÍÑçL’źæĂğãĂĆ éęŰåĚĹïijÑè£ŹéĞŇæIJL’å¿Ĺåď ŽåĞ¡æŢřæŕŤåęĆ `gcd()` åŠŇ `is_mandel()` ãĂĆ `divide()` åĞ¡æŢřæŸŕäÿĂäÿłè£ŤåŽđåď Žäÿłå Äijç ŽĎCåĞ¡æŢřä¿Ńå■ŘïijÑåĚűäÿ■æIJL’äÿĂäÿłæŸŕéĂŽè£Č `avg()` åĞ¡æŢřéĂŽè£ĞäÿĂäÿłCæŢřçżĎæL’ğèąŇæŢřæ■őèĄŽéŻĘæŞ■ä¡IJãĂĆ`Point` åŠŇ `distance()` åĞ¡æŢřæűL’åŔŁåĹřäžĘCçżŞædĎä¡ŞãĂĆ

åŕźäžŐæŐěäÿŃæİěçŽĎæL’ĂæIJL’åřŔèŁĆïijÑåĚĹåĄĞåőŽäÿŁéİćçŽĎäžčçăĄåůšçżŔèćńåĘŹåĚěäžĘąäÿĂ çĎűåŘŐåőČäžňçŽĎåőŽäźL’èćńåĘŹåĚěäÿĂäÿłåŘ■åŔńâĂIJsample.hâĂİçŽĎåď ť’æŰĞäžűäÿ■ïijÑ åźűäÿŤèćńçijŰèŕŚäÿžäÿĂäÿłåžŞåŔńâĂIJlibsampleâĂİïijÑèČ¡èćńéŞ¿æŐěåĹřåĚűäžŰCèŕ■èÍĂäžčçăĄäÿ■ãĂĆ çijŰèŕŚåŠŇéŞ¿æŐěçŽĎçżĘèŁĆä¿İæ■őçşżçż§çŽĎäÿ■åŘÑèĂŇäÿ■åŘÑïijÑä¡ĘæŸŕè£ŹäÿłäÿæŸŕæĹŚäžňåĚåęĆæđIJä¡äèęĄåď ĎçŘĘCäžčçăĄïijÑæĹŚäžňåĄĞåőŽè£ŹäžŽå§žçąĂçŽĎäÿIJèě£ä¡äéČ¡æŐÑæŔąäžĘãĂĆ

Contents:

17.1 15.1 ä¡£çŤÍctypesèő£éŮőCäžčçăĄ

éŮőécŸ

ä¡ăæIJL'äÿĂäžŻCåĞ¡æŢřåůšçżŔèćńçijŰèŕŠåĹřåĚśäžńåžŞæĹŰDLLäÿ■ãĂĆä¡ååÿŇæIJŻåŔŕäžěä¡£çŤÍçžíèĂŇäÿ■çŤÍçijŰåĘŹéćÍåd'ŰçŽĎCäžčçăĄæĹŰä¡£çŤÍçňňäÿL'æŰźæL'l'åśŢåůěåĚůãĂĆ

èğčåĘşæŰźæąĹ

åŕźäžŐéIJĂèęĄèřČçŤÍCäžčçăĄçŽĎäÿĂäžŻåřŔçŽĎéŮőécŸïijŇéĂŽåÿÿä¡£çŤÍPythonæăĞåĞĘåžŞäÿ■çŽctypes æÍąåİŮåřśèűşåd'§äžĘãĂĆ èęĄä¡£çŤÍ ctypes ïijŇä¡ăéęŰåĚĹèęĄçąőä£Ía¡ăèęĄèő£éŮőçŽĎCäžčçăĄïijĹåŘŇæăůçŽĎædűædĎãĂĄå■Ůåd'ğåřŔãĂĄçijŰèŕŠåŹĺç■L'ïijL'çŽĎæ§ŘäÿłåĚśäžńåžŞäÿ■äžĘãĂĆäÿžäžĘè£ŻèąŇæIJńèŁĆçŽĎæijŢçd'žïijŇåĄĞèő¿ä¡ăæIJL'äÿĂäÿłåĚśäžńåžŞåŘ■å■ŮåŔńlibsample.so ïijŇéĞŇéİćçŽĎåĘĚåőźåřśæŸŕ15çńääżŇçż■éČĺåĹĘéĆčæăůãĂĆåŔęåd'Űè£ŸåĄĞèő¿è£Źäÿłlibsample.so æŰĞäžűècńæŢ¿ç¡őåĹřä¡■äžŐ sample.py æŰĞäžűçŻÿåŘŇçŽĎçŻőå¡Ţäÿ■äžĘãĂĆ

èęĄèő£éŮőè£ŹäÿłåĞ¡æŢřåžŞïijŇä¡ăèęĄåĚĹæđĎåžžäÿĂäÿłåŇĚèčĚåőČçŽĎPythonæÍąåİŮïijŇåęĆäÿŇè

```
# 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)

```
    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)

```
distance.restype = ctypes.c_double
```

åęĆædIJäÿĂåĹĞæ■čåÿÿïijŇä¡ăářśåŔŕäžěåŁăè¡¡åźűä¡£çŤĺéĞŇéİćåőŽäźL’çŽĎCåĞ¡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

æIJňářŔèŁĆæIJL’å¿Ĺåď’ŽåĂijå¿ŮæĹŚäžňèŕęçżĘèőĺèőžçŽĎåIJřæŰźãĂĆ éęŰåĚĹæŸŕářźäžŐCåŠŇPythonäžčçăĄäÿĂèţůæL’ŞåŇĚçŽĎéŮőécŸïijŇåęĆædIJä¡ăåIJĺä¡£çŤĺ `ctypes` æİěëő£éŰőçijŰèŕŚåŘŐçŽĎCäžčçăĄïijŇ éĆčäźĹéIJĂèęĄçąőä£Ìè£ŹäÿłåĚśäžńåžŞæŤ¿åIJĺ `sample.py` æĺąåİŮåŘŇäÿĂäÿłåIJřæŰźãĂĆ äÿĂçğ■åŕŕèČ¡æŸŕářĘçŤ§æĹŘçŽĎ . `so` æŰĞäżűæŤ¿ç¡őåIJĺèęĄä¡£çŤĺåőČçŽĎPythonäžčçăĄåŘŇäÿĂäÿłçŽőå¡ŢäÿŇãĂĆ æĹŚäžňåIJĺ `recipeâĂĭsample.py` äÿ■ä¡£çŤĺ `__file__` åŔŸéĞŔæİěæ§ěçIJŃåőČèćńåőL’èčĚçŽĎä¡■ç¡őïijŇ çĎűåŘŐædĎéĂääÿĂäÿłæŇĞåŘŚåŘŇäÿĂäÿłçŽőå¡Ţäÿ■ç `libsample.so` æŰĞäžűçŽĎèůŕå¿ĎãĂĆ

åęĆædIJCåĞ¡æŢřäžŞèćńåőL’èčĚåĹřåĚűäžŰåIJřæŰźïijŇéĆčäźĹä¡ăářśèęĄä£őæŤźçŻÿäžŤçŽĎèůŕå¿ĎãĂ åęĆædIJCåĞ¡æŢřäžŞåIJĺä¡ăæIJžåŹĺäÿŁèćńåőL’èčĚäÿžäÿĂäÿłæăĞåĞĘåžŞäžĘïijŇ éĆčäźĹåŔŕäžěä¡£çŤĺ `ctypes.util.find_library()` åĞ¡æŢřæİěæ§ěæL’¿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¡ăç§ěéĄŞäžĘCåĞ¡æŢřäžŞçŽĎä¡■ç¡őïijŇéĆčäźĹářśåŔŕäžěåČŔäÿŃéİcè£Źæăůä¡£çŤĺ `ctypes.cdll.LoadLibrary()` æİěåĽäè¡¡åőČïijŇ åĚűäÿ■ `_path` æŸŕæăĞåĞĘåžŞçŽĎåĚĺèůŕå¿ĎïijŽ

```
_mod = ctypes.cdll.LoadLibrary(_path)
```

åĞ¡æŢřåžŞèćńåŁäè¡¡åŘŐïijÑä¡ăéIJĂèęĄçijŰåĘŹåĞääÿłèŕ■åŔěæİěæŔŘåŔŰçL'źåőŽçŽĎçňęåŔůåźűæŇĞ
åřśåČŔäÿŃéİćè£Źäÿłäžčçă Ąç L'Ğæőţäÿ Ăæăůïijž

```
# 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£Źæőţäžčçă Ąäÿ■ïijÑ .argtypes åśđæĂğæŸŕäÿĂäÿłåĚČçžĎïijÑåŇĚåŘńäžĘæ§ŘäÿłåĞ¡æŢřçŽĎè
èĂŇ .restype åřśæŸŕçŽÿåžŤçŽĎè£ŤåŽđçśžåđŃãĂĆ ctypes
åőŽäźL'äžĘåď ğéĞŔçŽĎçśžåđŃåŕźèśąïijĹærŤåęĆc_double, c_int, c_short, c_floatç■L'ïijL'ïijÑ
äžčèąĺäžĘåŕźåžŤçŽĎCæŢřæ■őçśžåđŃãĂĆåęĆæđIJä¡ăæČşèől' PythonèČ¡åď §äijăéĂŠæ■čçąőçŽĎåŔĆæŢřçśžå
éĆčäźĹè£ŹäžŽçśžåđŃç■¿åŘ■çŽĎçżŠåőŽæŸŕå¿ĹéĞ■èęĄçŽĎäÿĂæ■ěãĂĆåęĆæđIJä¡ăæšąæIJL'è£ŹäźĹåĄŽïi
è£ŸåŔŕèČ¡äijŽåŕijèĞt' æŢt' äÿłèğčéĞŁåŹĺè£ŽçĺŃæŇĆæŐL'ãĂĆ
ä¡£çŤĺctypesæIJL'äÿĂäÿłéžžçČęçĆźçŽĎåIJřæŰźæŸŕåŐ§çŤ§çŽĎCäžčçăĄä¡£çŤĺçŽĎæIJŕèŕ■åŔŕèČ¡èů§Pytho
divide() åĞ¡æŢřæŸŕäÿĂäÿłå¿Ĺåě¡çŽĎä¿Ńå■ŘïijÑåőĆéĂŽè£ĞäÿĂäÿłåŔĆæŢřéŹd' äžěåŔęäÿĂäÿłåŔĆæŢ
ař¡çőąè£ŹæŸŕäÿĂäÿłå¿Ĺåÿÿèğ ĄçŽĎCæŁĂæIJŕïijÑä¡ĘæŸŕåIJĺPythonäÿ■å■t' äÿ■ç§ěéĄŞæĂŐæăůæÿĚæŹřçŽ
ä¿ŃåęĆïijÑä¡ääÿ■èČ¡åČŔäÿŃéİćè£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¡IJïijÑåőČäijŽè£ĺåŔ■PythonåŕźäžŐæŢt' æŢřçŽĎäÿ■åŘŕæŻt' æŤźåŐ§å
åŕźäžŐæűL'åŘŁåĹřæŇĞéŠĹçŽĎåŔĆæŢřïijÑä¡ăéĂŽåÿÿéIJĂèęĄåĚĹædĎåžžäÿĂäÿłçŻÿåžŤçŽĎctypesåŕźèśą

```
>>> x = ctypes.c_int()
>>> divide(10, 3, x)
3
>>> x.value
1
>>>
```

åIJÍè£ŹéĞŇïijÑäÿĂäÿł ctypes.c_int åőđä¿ŃèćńåĹŻåžžåźűä¡IJäÿžäÿĂäÿłæŇĞéŠĹèćńäijăè£ŽåŐžã
èů§æŹőéĂŽPythonæŢt' å¡ćäÿ■åŘŇçŽĎæŸŕïijÑäÿĂäÿł c_int
åŕźèśąæŸŕåŔŕäžěèćńä£őæŤźçŽĎãĂĆ .value åśđæĂğåŔŕèćńçŤĺæİěèŐůåŘŰæĹŰæŻt' æŤźè£ŹäÿłåĂijãĂĆ

åŕźäžŐéĆčäžŽäÿ■åČŔPythonçŽĎCèřČçŤĺïijÑéĂŽåÿÿåŔŕäžěåĘŹäÿĂäÿłåřŘçŽĎåŇĚèčĚåĞ¡æŢřãĂĆ
è£ŹéĞŇïijÑæĹŚäžňèől' divide() åĞ¡æŢréĂŽè£ĞåĚČçżĎæİěè£ŤåŽđäÿd' 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¡æŢřåŔĹæŸŕäÿĂäÿłæŰřçŽĎæŇŠæĹŸãĂĆCäžčçăĄæIJ§æIJŻæŐěåŔŮåĹřäÿĂäÿłæŇĞéŠĹåŠ
ä¡ĘæŸŕïijŇåIJĺPythonäÿ■ïijŇæĹŚäžňå£Ěéąžè ĂČèŹŚè£ŹäÿłéŮőécŸïijŽæŢřçžĎæŸŕåŢěïij§åőČæŸŕäÿĂäÿłåĹ
è£ŸæŸŕ array æĺąåĺŮäÿ■çŽĎäÿĂäÿłæŢřçžĎïij§è£ŸæŸŕäÿĂäÿł numpy
æŢřçžĎïij§è£ŸæŸŕèŕťæĽĂæIJĽéČ¡æŸŕïij§ åőđéŹĚäÿŁïijŇäÿĂäÿłPythonâĂIJæŢřçžĎâĂİæIJĽåđŽçğ■å¡ćå

DoubleArrayType æijŤçđžäžĘæĂŐæăůåđĎçŘĘè£Źçğ■æČĚåĘţãĂĆ
åIJĺè£Źäÿłçśžäÿ■åőŽäźĽäžĘäÿĂäÿłå■ŢäÿłæŰźæşŢ from_param() ãĂĆ
è£ŹäÿłæŰźæşŢçŽĎèğŠèĽšæŸŕæŐěåŮŮäÿĂäÿłå■ŢäÿłåŔĆæŢřçĎűåŘŐářĘåĚűåŘŠäÿŃè¡ňæ■ćäÿžäÿĂäÿłåŘĺ
ïijĹæIJňä¿Ńäÿ■æŸŕäÿĂäÿł ctypes.c_double çŽĎæŇĞéŠĹïijĽãĂĆ
åIJĺ from_param() äÿ■ïijŇä¡ăåŔŕäžěåĄŽäżżä¡Ţä¡ăæČşåĄŽçŽĎäžŃãĂĆ
åŔĆæŢřçŽĎçśžåđŃåŘ■èćńæŔŘåŔŰåĞžæİěåźűèćńçŤĺäžŐåĹĘåŘŠåĹřäÿĂäÿłæŽť åĚůä¡ŞçŽĎæŰźæşŢäÿ■åŐ
ä¿ŃåęĆïijŇåęĆæđIJäÿĂäÿłåĹŮèąĺèćńäijăéĂŠè£ĞæİěïijŇéĆčäźĹ typename åřśæŸŕ list
ïijŇ çĎűåŘŐ from_list æŰźæşŢèćńèřČçŤĺãĂĆ

åŕźäžŐåĹŮèąĺåŠŇåĚČçżĎïijŇfrom_list æŰźæşŢářĘåĚűè¡ňæ■ćäÿžäÿĂäÿł ctypes
çŽĎæŢřçžĎåŕźèśąãĂĆ è£Źäÿłçİ JŃäÿŁåŐżæIJĽçĆźåěĞæĂłïijŇäÿŃéİćæĹŚäžňä¡£çŤĺäÿĂäÿłäžď äžŠåijŔä¿Ń
ctypes æŢřçžĎïijŽ

```
>>> 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
>>>
```

åŕźäžŐæŢřçžĎåŕźèśąïijŇfrom_array() æŔŘåŔŰåžŢåśĆçŽĎåĘĚå■ŸæŇĞéŠĹåźűåřĘåĚűè¡ňæ■ćäÿž
ctypes æŇĞéŠĹåŕźèśąãĂĆä¿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)

```
4298687200
>>> ctypes.cast(ptr, ctypes.POINTER(ctypes.c_double))
<__main__.LP_c_double object at 0x10069cd40>
>>>
```

from_ndarray() æijŤçd'žäžĘåŕźäžŐ numpy æŢřçżĎçŽĎè¡ňæ■ćæŞ■ä¡IJãĂĆ éĂŽè£ĞåőŽäźL' DoubleArrayType çśżåźűåIJÍ avg() çśżåđŃç■¿åŘ■äÿ■ä¡£çŤÍåőČïijŇ éĆčäźĹè£ŹäÿłåĞ¡æŢřåřśèČ¡æŐěåŔŮåd'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¡ăæijŤçd'žäžĘæĂŐæăůåd'ĎçŘĘäÿĂäÿłçőĂå■ŢçŽĎCçżŞæđĎãĂĆ åŕźäžŐçżŞæđĎä¡ŞïijŇä¡ăåŔłéIJĂèęĄåČŔäÿŃéİćè£ŹæăůçőĂå■ŢçŽĎåőŽäźL'äÿĂäÿłçśżïijŇåŇĚåŘńçŻÿåžŢç

```
class Point(ctypes.Structure):
    _fields_ = [('x', ctypes.c_double),
                ('y', ctypes.c_double)]
```

äÿĂæŮęçśżècńåőŽäźL'åŘŐïijŇä¡ăåřśåŔŕäžěåIJÍçśżåđŃç■¿åŘ■äÿ■æĹŰèĂĚæŸŕéIJĂèęĄåőđä¿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
>>>
```

æIJĂåŘŐäÿĂäžŽåřŔçŽĎæŔŘçd'žïijŽåęĆæđIJä¡ăæČşåIJÍPythonäÿ■èő£éŮőäÿĂäžŽåřŔçŽĎCåĞ¡æŢřïijŇ ctypes æŸŕäÿĂäÿłå¿ĹæIJL'çŤÍçŽĎåĞ¡æŢřåžŞãĂĆ år¡çőąåęĆæ■d'ïijŇåęĆæđIJä¡ăæČşèęĄåŐżèő£éŮőäÿĂä Swig (15.9èŁĆäijŽèőšåĹř) æĹŰ CythonïijĹ15.10èŁĆïijL'ãĂĆ

åŕźäžŐåd'ğåđŃåžŞçŽĎèő£éŮőæIJL'äÿłäÿżèęĄéŮőécŸïijŇçŤśäžŐctypesåźűäÿ■æŸŕåőŇåĚÍèĞłåŁÍåŇŰï éĆčäźĹä¡ăåřśå£ĚéążèŁśèt'žåd'ğéĞŘæŮűéŮt'æÍěçijŰåĘŹæL'ĂæIJL'çŽĎçśżåđŃç■¿åŘ■ïijŇåřśåČŔä¿Ńå■Řäÿ åęĆæđIJåĞ¡æŢřåžŞåd'§åd'■æÍĆïijŇä¡ăè£Ÿå¿ŮåŐżçijŰåĘŹå¿Ĺåd'ŽåřŔçŽĎåŇĚèčĚåĞ¡æŢřåŠŇæŤŕæŇĄçśż åŘęåd'ŰïijŇéŹd'éIđä¡ăåůšçżŔåőŇåĚÍçš¿éĂŽäžĘæL'ĂæIJL'åžŢåśĆçŽĎCæŐěåŔčçżĘèŁĆïijŇåŇĚæŃňåĘĚå■ éĂŽåÿÿäÿĂäÿłå¿ĹåřŔçŽĎäžčçăĄçijžéŹůãĂĄèő£éŮőèűŁçŢŇæĹŰåĚűäžŰçśżäijijéŤŹèŕŕåřśèČ¡èőľ PythonçÍŃ

ä¡IJäÿž ctypes çŽĎäÿĂäÿłæŻ£äżčïijŇä¡ăè£ŸåŔŕäżěèĂČèŹŚäÿŃCFFIãĂĆCFFIæŔŘä¿ŻäžĘå¿Ĺåď'Žç
ä¡ĘæŸŕä¡£çŤĺCèŕ■æşŢåźűæŤŕæŇĄæŻt'åď'ŽéńŸçžğçŽĎCäžčçăĄçśžåđŃãĂĆ
åĽřåĘŹè£ŹæIJňäźęäÿžæ■ćïijŇCFFIè£ŸæŸŕäÿĂäÿłçŻÿåŕźè¿ČæŰřçŽĎåůěçĺŃïijŇ
ä¡ĘæŸŕåőČçŽĎæţĄèąŇåžęæ■čåIJĺå£néĂ§äÿŁå■ĞãĂĆ çŤŽèĞşè£ŸæIJL'åIJĺèőĺèőžåIJĺPythonářĘæİěçŽĎçL'İ

17.2 15.2 çőĂå■ŢçŽĎCæĽľåśŢæĺąåİŮ

éŮőécŸ

ä¡ăæČşäÿ■ä¿İéİăåĚűäżŰåůěåĚůïijŇçŻt'æŐëä¡£çŤĺPythonçŽĎæL'ľåśŢAPIæİěçijŰåĘŹäÿĂäžŽçőĂå■ŢçŹ

èğčåĘşæŰźæąĹ

åŕźäžŐçőĂå■ŢçŽĎCäžčçăĄïijŇæđĎåżžäÿĂäÿłèĞłåőŽäźĽæL'ľåśŢæĺąåİŮæŸŕå¿Ĺåőźæ ŸŞçŽĎãĂĆ
ä¡IJäÿžçňňäÿĂæ■ěïijŇä¡ăéIJĂèęĄçąőä£İä¡ăçŽĎCäžčçăĄæIJL'äÿĂäÿłæ■čçąőçŽĎåď't'æŰĞäžűãĂĆä¿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);
```

éĂŽåÿÿæİěèőšïijŇè£Źäÿłåď't'æŰĞäžűęęĄåŕźåžŤäÿĂäÿłåůšçżŔèćńå■ŢçŃňçijŰèŕŚè£ĞçŽĎåžŞãĂĆ
æIJL'äžĘè£ŹäžŻïijŇäÿŃéİćæĹŚäžňæijŤçď'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)

```
/* 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’ľåśŢæĺąåİŮïijŇåČŔäÿŃéİćè£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■şåŔřïijŽ

```
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'Ăçđ'žïijŇåőČäijŽåĹŻåžžäÿĂäÿłåŘ■å■ŮåŔń `sample.so` çŽĎåĚśäžńåžŞãĂĆå¡Şèćńçij Űèŕ ŚåŘŐïijŇä¡ăářśèČ¡ářĘåőČä¡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¡ăæŸŕåIJĺWindowsæIJžåŹĺäÿŁéĺćåřİèŕŢè£ŹäžŻæ■ěéłď'ïijŇåŔŕèČ¡äijŽéĄĞåĹřåŘĎçğ■çŐŕåćČ PythonçŽĎäžŇè£ŻåĹűåĹĘåŔŚéĂŽåÿÿä¡£çŤĺäžĘMicrosoft Visual StudioæİěæđĎåžžãĂĆ äÿžäžĘèől'è£ŹäžŻæL'l'åśŢèČ¡æ■čåÿÿåůěä¡IJïijŇä¡ăéIJĂèęĄä¡£çŤĺåŘŇæăůæĹŰåĚijåőźçŽĎåůěåĚůæİěçijŰè åŔĆèĂČçŻÿäžŤçŽĎ PythonæŰĞæąč

èőÍèőž

åIJÍåřÌèŕŢäžžä¡ŢæĽŃåĘŹæĽ'ľåśŢäźŃåĽ■ïijŇæIJĂåě¡èČ¡åĚĹåŔĆèĂČäÿŃPythonæŰĞæąčäÿ■çŽĎ
æĽ'ľåśŢåŠŇåţŇåĚěPythonèğčéĞŁåŹÍ . PythonçŽĎCæĽ'ľåśŢAPIå¿Ĺåď'ğïijŇåIJÍè£ŹéĞŇæŢť'äÿłåŐżèőšè£řå
äÿ■è£Ğåŕźäž0æIJĂæäÿå£ČçŽĎéČĺåĹĘè£ŸæŸŕåŔŕäžěèőÍèőžäÿŃçŽĎãĂĆ

éęŰåĚĹïijŇåIJÍæĽ'ľåśŢæĺąåİŮäÿ■ïijŇä¡ăåĘŹçŽĎåĞ¡æŢřéČ¡æŸŕåČŔäÿŃéİćè£ŹæăůçŽĎäÿĂäÿłæŹőéĂ

```
static PyObject *py_func(PyObject *self, PyObject *args) {
  ...
}
```

`PyObject` æŸŕäÿĂäÿłèČ¡èąĺçď'žäžžä¡ŢPythonåŕźèśąçŽĎCæŢřæ■őçśžåđŃãĂĆ
åIJÍäÿĂäÿłéńŸçžğåśĆéİćïijŇäÿĂäÿłæĽ'ľåśŢåĞ¡æŢřåřśæŸŕäÿĂäÿłæŐěåŔŮäÿĂäÿłPythonåŕźèśą
ïijĹåIJÍ PyObject *argsäÿ■ïijĽåĚČçžĎåźűè£ŤåŽđäÿĂäÿłæŰřPythonåŕźèśąçŽĎCåĞ¡æŢřãĂĆ
åĞ¡æŢřçŽĎ `self` åŔĆæŢřåŕźäžŐçőĂå■ŢçŽĎæĽ'ľåśŢåĞ¡æŢræšąæIJL'èćńä¡£çŤĺåĹřïijŇ
äÿ■è£ĞåęĆæđIJä¡ăæČşåőŽäźĽæŰřçŽĎçśžæĹŰèĂĚæŸŕĆäÿ■çŽĎåŕźèśąçśžåđŃçŽĎèŕĺåřśèČ¡æť¿äÿŁçŤĺåIJž
éĆčäźĹ `self` åřśèČ¡åijŢçŤĺéĆčäÿłåőđä¿ŃäžĘãĂĆ

`PyArg_ParseTuple()` åĞ¡æŢřèćńçŤĺæİěåřĘPythonäÿ■çŽĎåĂijè¡ñæ■ćæĹŘCäÿ■åŕźåžŤèąĺçď'žãĂĆ
åőČæŐěåŔŮäÿĂäÿłæŇĞåőŽè¿ŞåĚěæăijåijŔçŽĎæăijåijŔåŇŰå■Ůçňęäÿšä¡IJäÿžè¿ŞåĚěïijŇæŕŤåęĆâĂIJiâĂİ
åŘŇæăůè£ŸæIJL'å■ŸæŤ¿è¡ñæ■ćåŘŐçżŞæđIJçŽĎCåŔŸéĞŔçŽĎåIJřåİĂãĂĆ
åęĆæđIJè¿ŞåĚěçŽĎåĂijäÿ■åŇźéĚ■è£ŹäÿłæăijåijŔåŇŰå■ŮçňęäÿšïijŇåřśäijŽæŁŻåĞžäÿĂäÿłåijĆåÿÿåźűè£Ťá
éĂŽè£ĞæčĂæ§ěåźűè£ŤåŽđNULLïijŇäÿĂäÿłåŘĹéĂĆçŽĎåijĆåÿÿäijŽåIJÍèŕĈçŤĺäžčçăĄäÿ■èćńæŁŻåĞžãĂĆ

`Py_BuildValue()` åĞ¡æŢřèćńçŤĺæİěæăźæ■őCæŢřæ■őçśžåđŃåĹŻåžžPythonåŕźèśąãĂĆ
åőČåŘŇæăůæŐěåŔŮäÿĂäÿłæăijåijŔåŇŰå■ŮçňęäÿšæİěæŇĞåőŽæIJ§æIJŻçśžåđŃãĂĆ
åIJÍæĽ'ľåśŢåĞ¡æŢřäÿ■ïijŃåőČèćńçŤĺæİěè£ŤåŽđçżŞæđIJçżŹPythonãĂĆ
`Py_BuildValue()` çŽĎäÿĂäÿłçĽźæĂğæŸŕåőČèČ¡æđĎåžžæŻť'åŁăåď'■æİĆçŽĎåŕźèśąçśžåđŃïijŇæŕŤåęĆ
åIJÍ `py_divide()` äžčçăĄäÿ■ïijŇäÿĂäÿłä¿Ńå■ŘæijŤçď'žäžĘæĂŐæăůè£ŤåŽđäÿĂäÿłåĚČçżĎãĂĆäÿ■è£Ğï

```
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)
```

åIJÍæĽ'ľåśŢæĺąåİŮåžŢéČĺïijŇä¡ăäijŽåŔŚçŐřäÿĂäÿłåĞ¡æŢřèąĺïijŇæŕŤåęĆæIJňèŁĆäÿ■çŽĎ
`SampleMethods` èąĺãĂĆ è£ŹäÿłèąĺåŔŕäžěåĹŮåĞžCåĞ¡æŢřãĂĄPythonäÿ■ä¡£çŤĺçŽĎåŘ■å■ŮãĂĄæŰĞæa
æĽĂæIJL'æĺąåİŮéČ¡éIJĂèęĄæŇĞåőŽè£ŹäÿłèąĺïijŇåŽăäÿžåőČåIJÍæĺąåİŮåĹİåğŇåŇŰæŮűèęĄèćńä¡£çŤĺåĹř

æIJĂåŘŐçŽĎåĞ¡æŢř `PyInit_sample()` æŸŕæĺąåİŮåĹİåğŇåŇŰåĞ¡æŢřïijŇä¡ĘèŕěæĺąåİŮçňňäÿĂæì
è£ŹäÿłåĞ¡æŢřçŽĎäÿžèęĄåůěä¡IJæŸŕåIJÍèğčéĞŁåŹĺäÿ■æşĺåĘŇæĺąåİŮåŕźèśąãĂĆ

æIJĂåŘŐäÿĂäÿłèęĄçĆźéIJĂèęĄæŔŘåĞžæİěïijŇä¡£çŤĺCåĞ¡æŢřæİěæĽ'ľåśŢPythonèęĄèĂČèŹŚçŽĎäžl
ïijĹåőđéŹĚäÿŁïijŃC APIåŇĚåŘńäžĘèűĚè£Ğ500äÿłåĞ¡æŢřïijĽãĂĆä¡ăåžŤèŕěårĘæIJňèŁĆå¡ŞåĄŽæŸŕäÿĂäÿ
æŻť'åď'ŽéńŸçžğåĘĚåőźïijŇåŔŕäžěçIJŃçIJŃ `PyArg_ParseTuple()` åŠŇ
`Py_BuildValue()` åĞ¡æŢřçŽĎæŰĞæąčïijŃ çĎűåŘŐè£ŻäÿĂæ■ěæĽ'ľåśŢåijĂãĂĆ

17.3 15.3 çijŰåĘŹæĽľåśŢåĞ¡æŢřæŞ■ä¡IJæŢřçżĎ

éŮőécŸ

ä¡ăæČşçijŰåĘŹäÿĂäÿłCæĽ'ľåśŢåĞ¡æŢřæİěæŞ■ä¡IJæŢřçżĎïijŇåŔřèČ¡æŸřèćńarrayæĺąåİŮæĹŰçśžäijijN
äÿ■è£ĞïijŇä¡ăæČşèőľä¡ăçŽĎåĞ¡æŢřæŻť'åŁăéĂŽçŤĺïijŇèĂŇäÿ■æŸřéŠĹåŕźæ§Řäÿłç L'źåőŽçŽĎåžŞæĽĂçŤ

èğčåĘşæŰźæąĹ

äÿžäžĘèČ¡èőľæŐěåŔŮåŠŇåď'ĎçŘĘæŢřçżĎåĚůæIJL'åŔřçğżæď'■æĂğïijŇä¡ăéIJĂèęĄä¡£çŤĺåĹř
Buffer Protocol . äÿŃéİćæŸřäÿĂäÿłæĽ'ŃåĘŹçŽĎCæĽ'ľåśŢåĞ¡æŢřä¿Ńå■ŘïijŇ
çŤĺæİěæŐěåŔŮæŢřçżĎæŢřæ■őåźűèřČçŤĺæIJňçńăåijĂçŕĞéČĺåĹĘçŽĎ `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)

```
  /* Indicate we're done working with the buffer */
  PyBuffer_Release(&view);
  return Py_BuildValue("d", result);
}
```

äÿŃéİćæĹŚäżňæijŤçd'žäÿŃè£ŹäÿłæĽl'åśŢåĞ¡æŢřæŸŕåęĆä¡Ţåůěä¡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

åřĘäÿĂäÿłæŢřçżĎåŕźèśąäijąçżŹCåĞ¡æŢřåŔŕèČ¡æŸŕäÿĂäÿłæĽl'åśŢåĞ¡æŢřåĄŽçŽĎæIJĂåÿÿèğĄçŽĎäž
å¿Ĺåď'ŽPythonåžŤçŤĺçĺŃåžŔïijŇäżŐåŻ¿åČŔåď'ĎçŘĘåĹřçğŚå■ęèőąçőŮïijŇéČ¡æŸŕå§žäžŐéńŸæĂğèČ¡çŽĎ
éĂŽè£ĞçijŰåĘŹèČ¡æŐěåŔŮåźűæŞ■ä¡IJæŢřçżĎçŽĎäžčçăĄïijŇä¡ăåŔŕäžěçijŰåĘŹå¿Ĺåě¡çŽĎåĚijåőźè£ŹäžŽ
èĂŇäÿ■æŸŕåŔŕłèČ¡åĚijåőźä¡äèĞłåůśçŽĎäžčçăĄãĂĆ

äžčçăĄçŽĎåĚşéŤőçĆźåIJĺäžŐ `PyBuffer_GetBuffer()` åĞ¡æŢřãĂĆ
çżŹåőŽäÿĂäÿłäżżæĎŔçŽĎPythonåŕźèśąïijŇåőČäijŽèŕŢçİĂåŐżèŐůåŔŰåžŢåśĆåĘĚå■Ÿä£ąæĄŕïijŇåőČçőĂå
1. äijăçżŹ `PyBuffer_GetBuffer()` çŽĎçL'źæőŁæăĞå£ŮçżŹåĞžäžĘæL'ĂéIJĂçŽĎåĘĚå■ŸçijŞåĘśçśžåđ
ä¿ŃåęĆïijŇ`PyBUF_ANY_CONTIGUOUS` èąĺçd'žæŸŕäÿĂäÿłè£đçż■çŽĎåĘĚå■ŸåŇžå§§ãĂĆ

åŕźäžŐæŢřçżĎãĂĄå■ŮèŁĆå■ŮçňęäÿšåŇåĚűäžŰçśžäijijåŕźèśąèĂŇèĺĂïijŇäÿĂäÿł

Py_buffer çżŞæđĎä¡ŞåŇĚåŘńäžĘæĽĂæIJL'åžŢåśĆåĘĚå■ŸçŽĎä£ąæĄŕãĂĆ
åőČåŇĚåŘńäÿĂäÿłæŇĞåŘŚåĘĚå■ŸåIJŕåİĂãĂĄåđ'ğářŔãĂĄåĚČçť'ăåđ'ğářŔãĂĄæăijåijŔåŠŇåĚűäžŰçżĘèŁ

```
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■ïijŇæĹŚäžňåŔłåĚşæşĺæŐěåŔŮäÿĂäÿłåŔŇçš¿åžęæţőçĆźæŢřæŢřçżĎä¡IJäÿžåŔĆæŢřãĂĆ
èęĄæčĂæ§ěåĚČçť'ăæŸŕåŘęæŸŕäÿĂäÿłåŔŇçš¿åžęæţőçĆźæŢřïijŇåŔłéIJĂéłŇèŕĄ
format åśđæĂğæŸŕäÿ■æŸŕå■ŮçňęäÿšâĂİdâĂİ. è£Źäÿłäź§æŸŕ struct
æĺąåİŮçŤĺæİěçijŰçăĄäžŇè£ŻåĹűæŢřæ■őçŽĎãĂĆ éĂŽåÿÿæİèèőšïijŇformat
åŔŕäžěæŸŕäżżä¡ŢåĚijåőź struct æĺąåİŮçŽĎæăijåijŔåŇŮå■Ůçňęäÿšïij Ň
åźűäÿŤåęĆæđIJæŢřçżĎåŇĚåŘńäžĘCçżŞæđĎçŽĎèŕİåőČåŔŕäžěåŇĚåŘńåď'ŽäÿłåĂijãĂĆ
äÿĂæŮęæĹŚäžňåůšçżŔçąőőŽäžĘåžŢåśĆçŽĎçijŞå■ŸåŇžä£ąæĄŕïijŇéĆčåŔłéIJĂèęĄçőĂå■ŢçŽĎářĘåőČäij
åőđéŹĚäÿŁïijŇæĹŚäžňäÿ■å£ĚæŇĚå£ČæŸŕæĂŐæăůçŽĎæŢřçżĎçśżåđŇæĹŮèĂĚåőČæŸŕèćńäžĂäźĹåžŞåĹŽ
è£Źäź§æŸŕäÿžäżĂäźĹè£Źäÿłåĝ¡æŢřèČ¡åĚijåőź array æĺąåİŮäź§èČ¡åĚijåőź numpy
æĺąåİŮäÿ■çŽĎæŢřçżĎäžĘãĂĆ

åIJĺè£ŤåŻđæIJĂçżĹçżŞæđIJäźŇåĽ■ïijŇåžŢåśĆçŽĎçijŞåĘšåŇžèğĘåŻ¿å£Ěéążä¡£çŤĺ
PyBuffer_Release() éĞŁæŤ¿æŐĽ'ãĂĆ äźŇæĽ'Ăäžěèęą£ŹäÿĂæ■ěæŸŕäÿžäžĘèČ¡æ■čçąőçŽĎçőąçŘĚ

åŘŇæăůïijŇæIJňèŁĆäź§äžĚäžĚåŘłæŸŕæijŤçd'žäžĘæŐěåŔŮæŢřçżĎçŽĎäÿĂäÿłåřŔçŽĎäžčçăĄçĽĞæő
åęĆæđIJä¡ăçIJ§çŽĎèęĄåd'ĎçŘĘæŢřçżĎïijŇä¡ăåŔŕèČ¡äijŽçćřåĹřåd'Žçžť'æŢřæ■őãĂĄåd'ğæŢřæ■őãĂĄäÿ■å
éĆčäźĹåřśå¿ŮåŐżå■ęæŽť'énŸçžğçŽĎäÿIJèě£äžĘãĂĆä¡áéIJĂèęĄåŔĆèĂČåőŸæŰźæŰĞæąčæİěèŐůåŔŮæŽť

åęĆæđIJä¡ăéIJĂèęĄçijŰåĘŹæűĽ'åŔŁåĹřæŢřçżĎåd'ĎçŘĘçŽĎåd'ŽäÿłæĽ'l'åśŢïijŇéĆčäźĹéĂŽè£ĞCytho

17.4 15.4 åIJĺCæĽ'l'åśŢæĺąåİŮäÿ■æŞ■ä¡IJéŽŘå¡ćæŇĞéŠĹ

éŮőécŸ

ä¡ăæIJL'äÿĂäÿłæĽ'l'åśŢæĺąåİŮéIJĂèęĄåd'ĎçŘĘCçżŞæđĎä¡Şäÿ■çŽĎæŇĞéŠĹïijŇ
ä¡ĘæŸŕä¡ăåŔĹäÿ■æČşæŽť'éIJšçżŞæđĎä¡Şäÿ■äżżä¡ŢåĘĚéČĺçżĘèŁĆçżŹPythonãĂĆ

èğčåĘşæŰźæąĹ

éŽŘå¡ćçżŞæđĎä¡ŞåŔŕäžěå¿ĹåőźæŸ§çŽĎéĂŽè£ĞårĘåőČäžňåŇĚèčĚåIJĺèČűåŻŁårźèśąäÿ■æİěåd'ĎçŘĘ
èĂČèŹŚæĹŚäžňä¿Ňå■ŘäžčçăĄäÿ■çŽĎäÿŇåĹŮCäžčçăĄçĽĞæőţïijŽ

```
typedef struct Point {
    double x,y;
} Point;

extern double distance(Point *p1, Point *p2);
```

äÿŃéİćæŸŕäÿĂäÿłä¡£çŤĺèČűåŻŁåŇĚèčĚPointçżŞæđĎä¡ŞåŠŇ `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)

```
  }
  result = distance(p1,p2);
  return Py_BuildValue("d", result);
}
```

åIJÍPythonäÿ■åŔŕäżěåČŔäÿŃéÍćè£Źæăůæİěä¡£çŤÍè£ŹäžŻåĞ¡æŢřïijŽ

```
>>> 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

èČűåŻŁåŠŇCæŇĞéŠĹçśżäijijãĂĆåIJÍåĘĚéČÍïijŇåőČäžňèŐůåŔŰäÿĂäÿłéĂŽçŤÍæŇĞéŠĹåŠŇäÿĂäÿłåŔ
PyCapsule_New() åĞ¡æŢřå¿Ĺåőźæ ŸŞçŽĎèćńåĹŻåżžãĂĆ
åŔęåď ŰïijŇäÿĂäÿłåŔŕéĂĽ'çŽĎæđŘæđĎåĞ¡æŢřèČ¡èćńçżŚåőŽåĹřèČűåŻŁäÿŁïijŇçŤÍæÍěåIJÍèČűåŻŁåŕźèśąè

èęĄæŔŘåŔŰèČűåŻŁäÿ■çŽĎæŇĞéŠĹïijŇåŔŕä¡£çŤÍ PyCapsule_GetPointer()
åĞ¡æŢřåźűæŇĞåőŽåŘ■çğřãĂĆ åęĆæđIJæŔŘä¿ŻçŽĎåŘ■çğřåŠŇèČűåŻŁäÿ■åŇźéĚ■æĹŰåĚűäżŰéŤŹèŕŕåĞž

æIJňèŁĆäÿ■ïijŇäÿĂåŕźåůěåĚůåĞ¡æŢřâĂŤâĂŤ PyPoint_FromPoint()
åŠŇ PyPoint_AsPoint() èćńçŤÍæÍěåĹŻåżžåŠŇäžŐèČűåŻŁåŕźèśąäÿ■æŔŘåŔŰ-
Pointåőđä¿ŃãĂĆ åIJÍäżżä¡ŢæĽ'åśŢåĞ¡æŢřäÿ■ïijŇæĹŚäžňäijŽä¡£çŤÍè£ŹäžŻåĞ¡æŢřèĂŇäÿ■æŸŕçŻt'æŐěä¡
è£Źçğ■èő¿èőąä¡£å¿ŮæĹŚäžňåŔŕäžěå¿Ĺåőźæ ŸŞçŽĎåžŤåŕźåřĘæÍěåŕźPointåžŢäÿŇçŽĎåŇĚèčĚçŽĎæŻt'æŤź
ä¿ŃåęĆïijŇåęĆæđIJä¡ääEşåőŽä¡£çŤÍåŘęåď ŰäÿĂäÿłèČűåŻŁäžĘïijŇéĆčäźĹåŘłéIJÀèęĄæŻt'æŤźè£Źäÿd'äÿłå

åŕźäžŐèČűåŻŁåŕźèśąäÿĂäÿłéŽ¿çĆźåIJÍäžŐåđČåIJ¿åŻđæŤűåŠŇåĘĚå■ŸçőąçŘĘãĂĆ
PyPoint_FromPoint() åĞ¡æŢřæŐěåŘŮäÿĂäÿł must_free
åŘĆæŢřïijŇ çŤÍæÍěæŇĞåőŽå¡ŞèČűåŻŁèćńéŤĂæŕĄæŮűåžŢåśĆPoint *
çżŞæđĎä¡ŞæŸŕåŘęåžŤèŕěèćńåŻđæŤűãĂĆ åIJÍæ§ŘäžŻCäžčçăĄäÿ■ïijŇå¡ŠåśđéŭőécŸéĂŽåÿÿå¿ĹéŽ¿èćńåď'Ì
çÍŃåžŘåŚŸåŔŕäžěä¡£çŤÍ extra åŘĆæŢřæÍěæŐğåĹűïijŇèĂŇäÿ■æŸŕå■ŢæŰźéÍćçŽĎåĘşåőŽåđČåIJ¿åŻđæŤ
èęĄæşÍæĎŘçŽĎæŸŕåŠŇçŐřæIJL'èČűåŻŁæIJL'åĚşçŽĎæđŘæđĎåŹÍèČ¡ä¡£çŤÍ
PyCapsule_SetDestructor() åĞ¡æŢřæÍěæŻt'æŤźãĂĆ

åŕźäžŐæűL'åŘŁåĹřçżŞæđĎä¡ŞçŽĎCäžčçăĄèĂŇèÍĂïijŇä¡£çŤÍèČűåŻŁæŸŕäÿĂäÿłæŕŤè¿ČåŘĹçŘĘçŽĎè
ä¿ŃåęĆïijŇæIJL'æŮűåĂŹä¡ăåźűäÿ■åĚşå£ČæŽt'éIJšçżŞæđĎä¡ŞçŽĎåĘĚéČÍä£qæĄŕæĹŰèĂĚårĘåĚűè¡ñæ■ćæ
éĂŽè£Ğä¡£çŤÍèČűåŻŁïijŇä¡ääŔŕäžěåIJÍåőČäÿŁéÍćæŤ¿äÿĂäÿłè¡žéĞŔçžğçŽĎåŇĚèčĚåŹÍïijŇçĎűåŘŐářĘåőČ

17.5 15.5 äžŐæĽľåśŢæÍąåİŮäÿ■åőŽäźĽåŠŇåŕijåĞžCçŽĎAPI

éŮőécŸ

ä¡ăæIJL'äÿĂäÿłCæĽľåśŢæÍąåİŮïijŇåIJÍåĘĚéČÍåőŽäźĽäžĘå¿Ĺåď'ŽæIJL'çŤÍçŽĎåĞ¡æŢřïijŇä¡ăæČşåřĘå APIä¿ŻåĚűäžŮåIJřæŰźä¡£çŤÍãĂĆ ä¡ăæČşåIJÍåĚűäžŮæĽľåśŢæÍąåİŮäÿ■ä¡£çŤÍè£ŹäžŻåĞ¡æŢřïijŇä¡ĘæŸŕäÿ åźűäÿŤéĂŽè£ĞCçijŰèŕŚåŹÍ/éŞ¿æŐěåŹÍæİěåĄŽçIJŃäÿŁåŐżçL'źåĹńåď'■æİĆïijĹæĹŰèĂĚäÿ■åŔŕèČ¡åĄŽåĹř

èğčåĘşæŰźæąĹ

æIJňèŁĆäÿżèęĄéŮőécŸæŸŕåęĆä¡Ţåď'ĎçŘĘ15.4åřŔèŁĆäÿ■æŔŘåĹřçŽĎPointåŕźèśąãĂĆäżŤçżĘåŻđäÿ

```
/* 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);
}
```

çŐřåIJÍçŽĎéŮőécŸæŸŕæĂŐæăůåřĘ `PyPoint_AsPoint()` åŠŇ `Point_FromPoint()` åĞ¡æŢřä¡IJäÿžAPIåŕijåĞžïijŇè£ŹæăůåĚűäžŮæĽľåśŢæÍąåİŮèČ¡ä¡£çŤÍåźűéŞ¿æŐěåőČäžňïijŇæŕŤåęĆåęĆædIJä¡ăæIJL'åĚűäžŮæĽľåśŢäź§

èęĄèğčåĘşè£ŹäÿłéŮőécŸïijŇéęŰåĚÍèęĄäÿž `sample` æĽľåśŢåĘŹäÿłæŰřçŽĎåď'ť'æŰĞäžűåŘ■åŔń `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)

```
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'åśŢæĺąåİŮæİěåąńåĚĚèąĺæäijåźűåřĘåőČåČŔäÿŃéİćè£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)

```
/* 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;
}
```

æIJĂåŘŐïijŇäýŃéİćæŸŕäýĂäýłæŰřçŽĎæĽľåśŢæĺąåİŮä¿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)

```
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£ŹäýłæŰřæĺąåİŮæŮűïijÑä¡ăçŤŽèĞşäý■éIJĂèęĄåŐżèĂČèŹŚæĂŐæăůåřĘåĞ¡æŢřåžŞæĹŰäżčçă
ä¿ÑåęĆïijÑä¡ăåŔŕäżěåČŔäýÑéĺćè£ŹæăůåĹŻåžžäýĂäýłçőĂå■ŢçŽĎ `setup.py` æŰĞäžűïijŽ

```
# setup.py
from distutils.core import setup, Extension

setup(name='ptexample',
      ext_modules=[
        Extension('ptexample',
                  ['ptexample.c'],
                  include_dirs = [],  # May need pysample.h␣
↪directory
                  )
        ]
)
```

åęĆædIJäýĂåĹĞæ■čåýÿïijÑä¡ääijŽåŔŚçŐřä¡ăçŽĎæŰřæL'l'åśŢåĞ¡æŢřèČ¡åŠÑåőŽäźL'åIJĺåĚűäżŰæĺąåİ
APIåĞ¡æŢřäýĂèţůè£ŘèąÑçŽĎå¿Ĺåě¡ãĂĆ

```
>>> import sample
>>> p1 = sample.Point(2,3)
>>> p1
<capsule object "Point *" at 0x1004ea330>
>>> import ptexample
```

(continues on next page)

```
>>> ptexample.print_point(p1)
2.000000 3.000000
>>>
```

èőÍèőž

æIJňèŁĆå§žäžŐäÿĂäÿłåL’■æŔŘåřśæŸŕřijÑèČűåŻŁåŕźèśąèČ¡èŐůåŔŰäżżä¡Ţä¡ăæČşèęĄçŽĎåŕźèśąçŽĎ
è£Źæăůç ŽĎèŕİřijÑåőŽäźL’æĺąåİŮäijŽåąńåĚĚäÿĂäÿłåĞ¡æŢřæŇĞéŠĹçŽĎçżŞæđĎä¡ŞřijÑåĹŻåżžäÿĂäÿłæŇČ
ä¿ŃåęĆ `sample._point_api`.

åĚűäžŰæĺąåİŮèČ¡åď§åIJĺářijåĚěæŮűèŐůåŔŰåĹřè£ŹäÿłåśđæĂğåźűæŔŘåŔŰåžŢåśĆçŽĎæŇĞéŠĹãĂĆ
äžŃåőđäÿŁřijÑPythonæŔŘä¿ŻäžĘ `PyCapsule_Import()`
åůěåĚůåĞ¡æŢřřijÑäÿžäžĘåőŇæĹŘæL’ĂæIJL’çŽĎæ■ěéłď’ãĂĆ
ä¡ăåŘłéIJĂæŔŘä¿Żåśđæ ĂğçŽĎåŘ■å■Ůå■şåŔřřijĹæŕŤåęĆsample._point_apiïijL’ïijÑçĎűåŘŐäžŰåřśäijŽäÿĂa

åIJĺářĘèćńářijåĞžåĞ¡æŢřåŔŸäÿžåĚűäžŰæĺąåİŮäÿ■æŹőéĂŽåĞ¡æŢræŮűïijÑæIJL’äÿĂäžŻCçijŰçíÑéŹůc
åIJĺ `pysample.h` æŰĞäžűäÿ■ïijÑäÿĂäÿł `_point_api`
æŇĞéŠĹèćńçŤĺæİèæŇĞåŘŚåIJĺářijåĞžæĺąåİŮäÿ■èćńåĹİåğŃåŇŰçŽĎæŰźæşŢèąĺãĂĆ
äÿĂäÿłçŻÿåĚşçŽĎåĞ¡æŢř `import_sample()` èćńçŤĺæİèæŇĞåŘŚèČűåŽŁářijåĚěźűåĹİåğŃåŇŰè£Źäÿłæ
è£ŹäÿłåĞ¡æŢřå£Ěéążåİĺäżżä¡ŢåĞ¡æŢřèćńä¡£çŤĺäźŃåL’■èćńèřČçŤĺãĂĆéĂŽåÿÿæİèèőšïijÑåőČäijŽåIJĺæĺąåİ
æIJĂåŘŐïijŇCçŽĎéćĎåď’ĎçŘĘåőŘèćńåőŽäźL’ïijÑèćńçŤĺæİèéĂŽè£ĞæŰźæşŢèąĺåŐżåĹĘåŘŚè£Źäžżapiåğ
çŤĺæĹůåŘłéIJĂèęĄä¡£çŤĺè£ŹäžŻåŐ§åğŃåĞ¡æŢřåŘ■çğřå■şåŔřïijÑäÿ■éIJĂèęĄéĂŽè£ĞåőŘåŐżäžĘèğčåĚűä

æIJĂåŘŐïijÑè£ŸæIJL’äÿĂäÿłéĞ■èęĄçŽĎåŐ§åŻäèől’ä¡ăåŐżä¡£çŤĺè£ŹäÿłæŁĂæIJŕæİèéŞ¿æŐěæĺąåİŮâ
åęĆædIJä¡ăäÿ■æČşä¡£çŤĺæIJňæIJžçŽĎæŁĂæIJŕïijÑéĆčä¡ăåŕšå£Ěéążä¡£çŤĺåĚśäžńåžŞçŽĎéńŸçžğçL’źæĂğå
ä¿ŃåęĆïijÑåřĘäÿĂäÿłæŹőéĂŽçŽĎAPIåĞ¡æŢræŤ¿åĚěäÿĂäÿłåĚśäžńåžŞåźűçąőä£ÍæL’ĂæIJL’æL’l’åśŢæĺąåİŮ
è£Źçğ■æŰźæşŢçąőåőđåŘŕèąÑïijÑä¡ĘæŸŕåőČçŽÿåŕźçźĄçŘŘïijÑçL’źåĹńæŸŕåIJĺåď’ğđĎńçşżçż§äÿ■ãĂĆ
æIJňèŁĆæijŤçď’žäžĘåęĆä¡ŢéĂŽè£ĞPythonçŽĎæŹőéĂŽåŕijåĚěæIJžåĹűåŠÑäżĚäżĚåĞääÿłèČűåŽŁèřČçŤĺæİ
åŕźäžŐæĺąåİŮçŽĎçijŰèŕŚïijÑä¡ăåŘłéIJĂèęĄåőŽäźL’åď’t’æŰĞäžűïijÑèĂÑäÿ■éIJĂèęĄèĂČèŹŚåĞ¡æŢřåžŞçŽ

æŻt’åď’ŽåĚşäžŐåĹl’çŤĺC APIæİèædĎéĂäæL’l’åśŢæĺąåİŮçŽĎä£ąæĄŕåŘŕäžěåŘĆèĂČ
PythonçŽĎæŰĞæąč

17.6 15.6 äżŐCèŕ■èÍĂäÿ■èřČçŤÍPythonäžčçăĄ

éŮőécŸ

ä¡ăæČşåIJÍCäÿ■åőL’åĚÍçŽĎæL’ğèąÑæ§ŘäÿłPythonèřČçŤĺåźűè£ŤåŻđçżŞæđIJçżŹCãĂĆ
ä¿ŃåęĆïijÑä¡ăæČşåIJÍCèŕ■èÍĂäÿ■ä¡£çŤĺæ§ŘäÿłPythonåĞ¡æŢřä¡IJäÿžäÿĂäÿłåŻđèřČãĂĆ

èğčåĘşæŰźæąĹ

åIJÍCèŕ■èÍĂäÿ■èřČçŤÍPythonéİđåÿÿçőĂå■ŢïijÑäÿ■è£Ğèő¿èőąåĹřäÿĂäžŻåřŘçł■éŮÍãĂĆ
äÿŃéİćçŽĎCäžčçăĄåŚŁèŕL’ä¡ăæĂŐæăůåőL’åĚÍçŽĎèřČçŤÍïijŽ

```
#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)

```
  Py_XDECREF(result);
  PyGILState_Release(state);
  abort();   // Change to something more appropriate
}
```

èęĄä¡£çŤĺè£ŹäÿłåĞ¡æŢřïijŇä¡ăéIJĂèęĄèŐůåŘŰäijăéĂŠè£ĞæİěçŽĎæ§Řäÿłåůšå■ŸåIJÍPythonèřČçŤĺçŽ
æIJL'å¿Ĺåď'ŽçğæŰźæşŢåŘŕäżěèől'ä¡ăè£ŹæăůåĄŽïijŇ æŕŤåęĆářĘäÿĂäÿłåŘŕèřČçŤĺåŕźèśąäijăçżŹäÿĂäÿłæL'

äÿŃéİćæŸŕäÿĂäÿłçőĂå■Ţä¿Ńå■ŘçŤĺæİěæŐl'éěřäžŐäÿĂäÿłåţŇåĚěçŽĎPythonèğčéĞŁåŹĺäÿ■èřČçŤĺäÿ

```
#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;
}
```

èęĄædĎåżžä¿Ńå■ŘäżčçăĄïijŇä¡ăéIJĂèęĄçijŰèŕŚCåźűårĘåőČéŞ¿æŐěåĹřPythonèğčéĞŁåŹĺãĂĆ
äÿŃéİćçŽĎMakefileåŘŕäżěæŢŹä¡äæĂŐæăůåĄŽïijĹäÿ■è£ĞåIJĺä¡äæIJžåŹĺäÿŁéİćéIJĂèęĄäÿĂäžŽéĚ■ç¡őïijL'

```
all::
        cc -g embed.c -I/usr/local/include/python3.3m \
          -L/usr/local/lib/python3.3/config-3.3m -lpython3.3m
```

çijŰèŕŠåźűè£ŘèąŇäijŽäžğçŤ§çśżäijijäÿŃéİćçŽĎè¿ŞåĞžïijŽ

```
0.00 0.00
0.10 0.01
0.20 0.04
0.30 0.09
0.40 0.16
...
```

äÿŃéİćæŸŕäÿĂäÿłçĺ■å¿őäÿ■åŘŇçŽĎä¿Ńå■ŘïijŇåśŢçd'žäžĘäÿĂäÿłæL'l'åśŢåĞ¡æŢřïijŇ åőČæŐěåŔŮäÿĂäÿłåŔŕèřČçŤĺåŕźèśąåŠŇåĚűäžŰåŔĆæŢřïijŇåźűårĘåőČäžňäijäéĂŠçżŹ `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£ŹäÿłæL'l'åśŢåĞ¡æŢřïijŇä¡ăèęĄåČŔäÿŃéİćè£ŹæăůæţŃèŕŢåőČïijŽ

```
>>> import sample
>>> def add(x,y):
...     return x+y
...
>>> sample.call_func(add,3,4)
7.0
>>>
```

èőĺèőž

åęĆæđIJä¡ăåIJĺCèŕ■èĺĂäÿ■èřČçŤĺPythonïijŇèęĄèőřä¡ŔæIJĂéĞ■èęĄçŽĎæŸŕCèŕ■èĺĂäijŽæŸŕäÿżä¡ŞãĂ äźş§åřśæŸŕèŕťïijŇCèŕ■èĺĂèť§èť'čæđĎéĂăåŔĆæŢřãĂĄèřČçŤĺPythonåĞ¡æŢřãĂĄæčĂæ§ěåijĆåÿÿãĂĄæčĂæ

ä¡IJäÿžçňňäÿĂæ■ěïijŇä¡ăå£ĚéąžåĚĹæIJL'äÿĂäÿłèąĺçd'žä¡ăåřĘèęĄèřČçŤĺçŽĎPythonåŔŕèřČçŤĺåŕźèśąãĂ è£ŹåŔŕäžěæŸŕäÿĂäÿłåĞ¡æŢřãĂĄçśžãĂĄæŰźæşŢãĂĄåĘĚç¡őæŰźæşŢæĹŰåĚűäžŰäżżæĎŔåðđçŐřäžĘ `__call__()` æŞ■ä¡IJçŽĎäÿIJèě£ãĂĆ äÿžäžĘçąőä£ÍæŸŕåŔŕèřČçŤĺçŽĎïijŇåŔŕäžěåČŔäÿŃéİćçŽĎäžčçăĄè `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)

```
    fprintf(stderr,"call_func: expected a callable\n");
    goto fail;
  }
  ...
```

åIJÍCäžčçăĄéĞŇåď ĎçŘĘéŤŹèŕŕä¡ăéIJĂèęĄæăijåď ŰçŽĎåřŔå£ČãĂĆäÿĂèĹňæİěèőšïijŇä¡ăäÿ■èČ¡äžĚä
éŤŹèŕŕåžŤèŕěä¡£çŤÍCäžčçăĄæŰźåijŔæİěèćńåď ĎçŘĘãĂĆåIJÍè£ŹéĞŇïijŇæĹŚäžňæĽŞçőŮåřĘåŕźéŤŹèŕŕçŽĎ
`abort()` çŽĎéŤŹèŕŕåď ĎçŘĘåŹÍãĂĆ åőČäijŽçżŞæİ§æŐĽæŢť äÿłçÍŇåžŔïijŇåIJÍçIJ§åőđçŐŕåćČäÿŇéİćä¡ăă
ä¡ăèęĄèőřä¡ŔçŽĎæŸŕåIJÍè£ŹéĞŇCæŸŕäÿżèğŠïijŇåŻăæ■ď åźűæšąæIJL'èů§æŁŻåĞžåijĆåÿÿçŻÿåŕźåžŤçŽĎæ
éŤŹèŕŕåď ĎçŘĘæŸŕä¡ääIJÍçijŰçÍŇæŮűå£ĚéążèęĄèĂČèŹŚçŽĎäžŇæČĚãĂĆ

èřČçŤÍäÿĂäÿłåĞ¡æŢřçŻÿåŕźæİěèőšå¿ĹçőĂå■ŢâĂŤâĂŤåŔłéIJĂèęĄä¡£çŤÍ
`PyObject_Call()` ïijŇ äijăäÿĂäÿłåŔŕèřČçŤÍåŕźèśąçżŹåőČãĂĄäÿĂäÿłåŔĆæŢřåĚČçżĎåŠŇäÿĂäÿłåŔŕéĂ
èęĄæđĎåżžåŔĆæŢřåĚČçżĎæĹŰå■ŮåĚÿïijŇä¡ääŔŕäżěä¡£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■ŮåŔĆæŢřïijŇä¡ääŔŕäżěäijăéĂŠNULLãĂĆå¡Şä¡ăèęĄèřČçŤÍåĞ¡æŢřæŮűïijŇ
éIJĂèęĄçąőä£Ìä¡£çŤÍäžĘ `Py_DECREF()` æĹŰèĂĚ `Py_XDECREF()` æÿĚçŘĘåŔĆæŢřãĂĆ
çňňäžŇäÿłåĞ¡æŢřçŻÿåŕźåőL'åĚÍçĆźïijŇåŻääÿžåőČåĚĄèőÿäijăéĂŠNULLæŇĞéŠĹïijĹçŻt'æŐěå£¡çŢěåőČïijĹ
è£Źäź§æŸŕäÿžäžĂäźĹæĹŚäžňä¡£çŤÍåőČæÌěæÿĚçŘĘåŔŕéĂL'çŽĎåĚşéŤőå■ŮåŔĆæŢřãĂĆ

èřČçŤÍäÿĞPythonåĞ¡æŢřäźŇåŘŐïijŇä¡ää£ĚéążæčĂæ§ěæŸŕåŘęæIJL'åijĆåÿÿåŔŚçŤ§ãĂĆ
`PyErr_Occurred()` åĞ¡æŢřåŔŕèćńçŤÍæÌěåĄŽè£Źäžűäž ŇãĂĆ
åŕźåŕźäžŐåijĆåÿÿçŽĎåď ĎçŘĘåŕśæIJL'çĆźéžżçČęäžĘïijŇçŤśäžŐæŸŕçŤÍCèŕ■èĺĂåĘŹçŽĎïijŇä¡ăæšąæIJL'åČ
åŻăæ■ď ïijŇä¡ää£ĚéążèęĄèő¿ç¡őäÿĂäÿłåijĆåÿÿçŁűæĂĄçĂąïijŇæL'Şå■ŕåijĆåÿÿä£ąæĄŕæĹŰåĚűäžŰçŻÿåžŤ
åIJÍè£ŹéĞŇïijŇæĹŚäžňéĂL'æŃľäžĘçőĂå■ŢçŽĎ `abort()`
æİěåď ĎçŘĘãĂĆåŘęåď ŰïijŇäijăçż§CçÍŇåžŔåŚŸåŔŕèČ¡äijŽçŻt'æŐěěľ'çÍŇåžŔåěŤæžČãĂĆ

```
...
/* Check for Python exceptions (if any) */
if (PyErr_Occurred()) {
  PyErr_Print();
  goto fail;
}
```

(continues on next page)

(çż■äÿŁéąţ)

```
...
fail:
  PyGILState_Release(state);
  abort();
```

äżŐèřČçŤÍPythonåĞ¡æŢřçŽĎè£ŤåŻđåĂijäÿ■æŔŘåŔŰä£ąæĄŕéĂŽåÿÿèęĄè£ŻèąŇçśżåđŃæčĂæ§ěåŠŇæ
èęĄè£ŹæăůåĄŽçŽĎèŕÍïijŇä¡ää£Ěéążä¡£çŤÍPythonåŕźèśąåśĆäÿ■çŽĎåĞ¡æŢřãĂĆ
åIJÍè£ŹéĞŇæĹŚäżňä¡£çŤÍäžĘ PyFloat_Check() åŠŇ PyFloat_AsDouble()
æİěæčĂæ§ěåŠŇæŔŘåŔŰPythonæţőçĆźæŢřãĂĆ

æIJĂåŘŐäÿĂäÿłéÛőéćŸæŸŕåŕźäžŐPythonåĚĺåśĂéŤĄçŽĎçőąçŘĘãĂĆ
åIJÍCèŕ■èĺĂäÿ■èőéÛőPythonçŽĎæÛűåĂŹïijŇä¡ăéIJĂèęĄçąőä£İGILèćńæ■čçąőçŽĎèŐůåŔŰåŠŇéĞŁæŤ¿ä
äÿ■çĎűçŽĎèŕÍïijŇåŔŕèČ¡äijŽåŕijèĞť èğčéĞŁåŹĺè£ŤåŻđéŤŹèŕŕæŢřæ■őæĹŰèĂĚçŻť æŐěåĚťæžČãĂĆ
èřČçŤÍ PyGILState_Ensure() åŠŇ PyGILState_Release()
åŔŕäżĕçąőä£İäÿĂåĹĞéČ¡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();
}
```

äÿĂæÜęè£ŤåŻđïijŇPyGILState_Ensure() åŔŕäżěçąőä£ÍèřČçŤÍçž£çÍŃçŃňå■ăPythonèğčéĞŁåŹĺã
åřśçőŮCäżčçăĄè£ŘèąŇäžŐåŔęåď ŰäÿĂäÿłèğčéĞŁåŹĺäÿ■ç§ěéĄŞçŽĎçž£çÍŃäź§æšąäžŃãĂĆ
è£ŹæÛűåĂŹïijŇCäżčçăĄåŔŕäżěèĞłçŤśçŽĎä¡£çŤÍäżżä¡ŢåőČæČşèęĄçŽĎPython C-API
åĞ¡æŢřãĂĆ èřČçŤÍæĹŘåŁ§åŘŐïijŇPyGILState_Release()èćńçŤÍæİèèőšèğčéĞŁåŹĺæĄćåď■åĹřåŐ§åğŃçŁű

èęĄæşĺæĎŔçŽĎæŸŕæŕŔäÿĂäÿł PyGILState_Ensure()
èřČçŤÍå£Ěéążèů§çİĂäÿĂäÿłåŇźéĚ■çŽĎ PyGILState_Release()
èřČçŤÍâĂŤâĂŤå■şä¿£æIJĽéŤŹèŕŕåŔŚçŤ§ãĂĆ åIJÍè£ŹéĞŇïijŇæĹŚäżňä¡£çŤÍäÿĂäÿł goto
èŕ■åŔěçIJŃäÿŁåŐżæŸŕäÿłåŔŕæĂŢçŽĎèő¿èőąïijŇ ä¡ĘæŸŕåőđéŹĚäÿŁæĹŚäżňä¡£çŤÍåőČæİèèőšæŐğåĹűæİČ
åIJÍ fail: æăĞç■¿åŘŐéİćçŽĎäżčçăĄåŠŇPythonçŽĎ fianl:
åİÛçŽĎçŤÍéĂŤæŸŕäÿĂæăůçŽĎãĂĆ

åęĆæđIJä¡ăä¡£çŤÍæĽĂæIJĽè£ŹäžŻçžęåőŽæİěçijŰåĘŹCäżčçăĄïijŇåŇĚæŃňåŕźGILçŽĎçőąçŘĘãĂĄåij
ä¡ăäijŽåŔŚçŐřäżŐCèŕ■èĺĂäÿ■èřČçŤÍPythonèğčéĞŁåŹĺæŸŕåŔŕéİąçŽĎâĂŤâĂŤåŕźçőÛåĘ■åď■æİĆçŽĎçÍŃåž

17.7 15.7 äžŐCæĽľåśŢäÿ■éĞŁæŤ¿åĚĺåśĂéŤĄ

éŮőécŸ

ä¡ăæČşèőľCæĽľåśŢäžčçăĄåŠŇPythonèğčéĞŁåŹĺäÿ■çŽĎåĚűäžŰè£ŻçÍŃäÿĂèţůæ■čçąőçŽĎæĽğèąŇïij
éĆčäźĹä¡ăåřšéIJĂèęĄåŐžéĞŁæŤ¿åźűéĞ■æŰřèŐůåŔŰåĚĺåśĂèğčéĞŁåŹĺéŤĄïijĹGILïijĽãĂĆ

èğčåĘşæŰźæąĹ

åIJÍCæĽľåśŢäžčçăĄäÿ■ïijŇGILåŔŕäžěéĂŽè£ĞåIJĺäžčçăĄäÿ■æŔŠåĚëäÿŃéIćè£ŹæăůçŽĎåőŔæİěéĞŁæ

```
#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;
}
```

èőÍèőž

åŔłæIJL'å¡Şä¡ăçąőä£İæšąæIJL'Python C APIåĞ¡æŢřåIJÍCäÿ■æĽğèąŇçŽĎæŮűåĂŹä¡ăæĽ■èČ¡åőĽ'åĚĺçŽ
GILéIJĂèęĄèćńéĞŁæŤ¿çŽĎåÿÿèğĄçŽĎåIJžæŹŕæŸŕåIJÍèőąçőŮåŕĘéŻĘåđŃäžčçăĄäÿ■éIJĂèęĄåIJÍCæŢřçżĎ
æĹŰèĂĚæŸŕèęĄæĽğèąŇéŸżåąđçŽĎI/OæŞ■ä¡IJæŮűïijĹærŤåęĆåIJÍäÿĂäÿłæŰĞäžűæŔŔè£řçňęäÿŁèŕżåŔŰa

å¡ŞGILèćńéĞŁæŤ¿åŘŐïijŇåĚűäžŰPythonçž£çÍŃæĽ■èćńåĚĄèőÿåIJÍèğčéĞŁåŹĺäÿ■æĽğèąŇãĂĆ
`Py_END_ALLOW_THREADS` åőŔäijŽéŸżåąđæĽğèąŇçŻť'åĹřèřČçŤÍçž£çÍŃéĞ■æŰřèŐůåŔŰäžĘGILãĂĆ

17.8 15.8 CåŠŇPythonäÿ■çŽĎçž£çÍŃæůůçŤÍ

éŮőécŸ

ä¡ăæIJL'äÿĂäÿłçÍŃåžŔéIJĂèęĄæůůåŘĹä¡£çŤÍCãĂĄPythonåŠŇçž£çÍŃïijŇ
æIJL'äžŻçž£çÍŃæŸŕåIJÍCäÿ■åĹŻåžžçŽĎïijŇèűĚåĞžäžĘPythonèğčéĞŁåŹÍçŽĎæŐğåĹűèŇČåŻť'ãĂĆ
åźűäÿŤäÿĂäžŻçž£çÍŃè£Ÿä¡£çŤÍäžĘPython C APIäÿ■çŽĎåĞ¡æŢřãĂĆ

èğčåĘşæŰźæąĹ

åęĆædIJä¡ăæČşåřĘCãĂĄPythonåŠŇçž£çÍŃæůůåŘĹåIJÍäÿĂèţůïijŇä¡ăéIJĂèęĄçąőä£İæ■čçąőçŽĎåĹİåğŇ
èęĄæČşè£ŹæăůåĄŽïijŇåŔŕäžěåřĘäÿŃåĹŮäžčçăĄæŤ¿åĹřä¡ăçŽĎCäžčçăĄäÿ■åźűçąőä£İåőČåIJÍäżżä¡Ţçž£çÍŃ

```
#include <Python.h>
  ...
  if (!PyEval_ThreadsInitialized()) {
    PyEval_InitThreads();
  }
  ...
```

åŕźäžŐäżżä¡ŢèřČçŤÍPythonåŕźèśąæĹŰPython C APIçŽĎCäžčçăĄïijŇçąőä£Ìä¡ăéęŰåĚĹåůšçżŔæ■čçąőåI
è£ŹåŔŕäżěçŤÍ `PyGILState_Ensure()` åŠŇ `PyGILState_Release()`
æÌěåĄŽåĹřïijŇåęĆäÿŇæĽĂçđ'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);
...
```

æŕŔæňąèřČçŤÍ `PyGILState_Ensure()` éČ¡èęAçŻÿåžŢçŽĎèřČçŤÍ
`PyGILState_Release()`.

èőÍèőž

åIJÍæűĽ'åŔŁåĹřCåŠŇPythonçŽĎéńŸçžğçÍŃåžŔäÿ■ïijŇå¿Ĺåđ'ŽäžŃæČĚäÿĂètůåĄŽæŸŕå¿Ĺåÿÿè§ĄçŽĎ
åŔŕèČ¡æŸŕåŕźCãĂĄPythonãĂĄCçž£çÍŃãĂĄPythonçž£çÍŃçŽĎæůůåŘĹä¡£çŤÍãĂĆ
åŔłèęÄä¡ăçąőä£Ìè§čéĞŁåŹÍèćńæ■čçąőçŽĎåĹİåğŃåŇŰïijŇåźűäÿŢæűĽ'åŔŁåĹřè§čéĞŁåŹÍçŽĎCäžčçăĄæĽ'ğ

èęĄæşÍæĎŘçŽĎæŸŕèřČçŤÍ `PyGILState_Ensure()`
åźűäÿ■äijŽçńŃåĹżæŁćå■ăæĹŰäÿ■æŰ■è§čéĞŁåŹÍãĂĆ åęĆæđIJæIJL'åĚűäžŰäžčçăĄæ■čåIJÍæL'ğèąŇïijŇè£Ź
åIJÍåĘĚéČÍïijŇè§čéĞŁåŹÍäijŽæL'ğèąŇåŠÍæIJ§æĂğçŽĎçž£çÍŃåĹĞæ■ćïijŇåŻăæ■d'åęĆæđIJåĚűäžŰçž£çÍŃål
èřČçŤÍèĂĚæIJĂçżĹè£ŸæŸŕåŔŕäžèè£ŘèąŇçŽĎïijÍår¡çőąåŔŕèČ¡èęĄåĚĹç■L'äÿĂäijŽïijL'ãĂĆ

17.9 15.9 çŤÍSWIGåŇĚèčĚCäžčçăĄ

éŮőécŸ

ä¡ăæČşèől'ä¡ăåĘŹçŽĎCäžčçăĄä¡IJäÿžäÿĂäÿłCæL'l'åśŢæÍąåİŮæÌěèő£éŰőïijŇæČşéĂŽè£Ğä¡£çŤÍ
SwigåŇĚèčĚçŤ§æĹŘåŹÍ æÌěåőŇæĹŘãĂĆ

èğčåĘşæŰźæąĹ

SwigéĂŽè£Ğèğčæđ̌RCåđ'ť'æŰĞäžűåźűèĞłåŁÍåĹŻåžžæL'l'åśŢäžčçăĄæÌěæŞ■ä¡IJãĂĆ
èęĄä¡£çŤÍåőČïijŇä¡ăåĚĹèęĄæIJL'äÿĂäÿłCåđ'ť'æŰĞäžűãĂĆä¿ŃåęĆïijŇæĹŚäžñçđ'žä¿ŃçŽĎåđ'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£Źäÿłåđ'ť'æŰĞäžűïijŇäÿŃäÿĂæ■ěåřśæŸŕçijŰåĘŹäÿĂäÿłSwigâĂİæŐěåŔčâĂİæŰ
æŇL'çĚğçžęåőŽïijŇè£ŹäžŻæŰĞäžűäžěâĂİ.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)

```
  $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¡ăåĘŹåě¡äžĘæŐěåŘčæŰĞäżűĭijŇåřśåŔŕäžěåIJÍåŚ¡äžd'èąŇåůěåĚůäÿ■èřČçŤÍSwigäžĘĭijŽ

```
bash % swig -python -py3 sample.i
bash %
```

swigçŽĎè¿ŞåĞžåřśæŸŕäÿd'äÿłæŰĞäżűĭijŇsample_wrap.cåŠŇsample.pyãĂĆ åŘŐéÍćçŽĎæŰĞäżűåřśæŸŕçŤÍæĹůéIJĂèęĄåŕijåĚěçŽĎãĂĆ èĂŇsample_wrap.cæŰĞäżűæŸŕéIJĂèęĄèćńçijŰèŕŚåĹřåŘ■åŔń _sample çŽĎæŤŕæŇĄæĺąåİŮçŽĎCäžčçăĄãĂĆ è£ŹäÿłåŔŕäžěéĂŽè£Ğèů§æŹőéĂŽæĽ'åśŢæĺąåİŮäÿĂæăůçŽĎæŁĂæä¿ŇåęĆĭijŇä¡ăåĹŻåžžäžĘäÿĂäÿłåęĆäÿŇæĽĂçd'žçŽĎ `setup.py` æŰĞäżűĭijŽ

```
# 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)

```
        ]
)
```

èęĄçijŰèŕŚåŠŇæţŃèŕŢïijŇåIJĺsetup.pyäýŁæL'ğèąŇpython3ïijŇåęĆäýŃïijŽ

```
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 %
```

åęĆæđIJäýĂåĹĞæ■čåÿÿçŽĎèŕİïijŇä¡ăäijŽåŔŚçŐřä¡ăåřšåŔŕäžěå¿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

SwigæŸŕPythonåŐĘåŘšäÿ■æđĎåžžæL'l'åśŢæĺąåİŮçŽĎæIJĂåŔd'èĂĄçŽĎåůěåĚůäźŃäÿĂãĂĆ SwigèČ¡èĞłåŁĺåŇŰå¿Ĺåd'ŽåŇĚèčĚçŤ§æĹŘåŹĺçŽĎåd'ĎçŘĘãĂĆ

æL'ĂæIJL'SwigæŐěåŘčéČ¡äžěçśżäijijäÿŃéİćè£Źæăůç ŽĎäÿžåijĂåd't'ïijŽ

```
%module sample
%{
#include "sample.h"
%}
```

è£Źäÿłäż Ëäż ËåŘłæŸŕåčřæŸŐäžĘæĽľ'åśŢæĺąåİŮçŽĎåŘ■çğřåźűæŇĞåőŽäžĘCåď'ť'æŰĞäžűïijŇ
äÿžäžĘèČ¡èőľçijŰèŕŚéĂŽè£Gå£Ëéążèę Ąå ŇËåŘńè£ŹäžŽåď'ť'æŰĞäžűïijĹä¡■äžŐ %{ åŠŇ %}
çŽĎäžčçăĄïijĽ'ïijŇ åřĘåőČäžňäźŇéŮť'åď'■åĹűçšŸèť'ť'åĹřè¿ŞåĞžäžčçăĄäÿ■ïijŇè£Źäź§æŸŕä¡äèę ĄæŤ¿ç¡őæ

SwigæŐěåŘčçŽĎåžŢäÿŇéČĺåĹĘæŸŕäÿĂäÿłCåčřæŸŐåĹŮèąĺïijŇä¡äéIJĂèę ĄåIJĺæĽ'ľ'åśŢäÿ■åŇĚåŘńåő
è£ŹéĂŽåÿÿäżŐåď'ť'æŰĞäžűäÿ■èćńåď'■åĹűãĂĆåIJĺæĹŚäžňçŽĎä¿Ňå■Řäÿ■ïijŇæĹŚäžňäžĚäżĚåČŔäÿŇéİćè£

```
%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);
```

æIJL'äÿĂçĆźéIJĂèę ĄåijžěřČçŽĎæŸŕè£ŹäžŽåčřæŸŐäijŽåŚŁèŕĽSwigä¡ăæČşèę ĄåIJĺPythonæĺąåİŮäÿ■å
éĂŽåÿÿä¡ăéIJĂèę AçijŰè¿Śè£ŹäÿłåčřæŸŐåĹŮèąĺæĹŰçŻÿåžŢçŽĎä£őæŤźäÿŇåőČãĂĆ
ä¿ŇåęĆïijŇåęĆæđIJä¡ääÿ■æČşæ§ŘäžŽåčřæŸŐèćńåŇĚåŘńè£ŻæİěïijŇä¡äèę ĄåřĘåőČäžŐåčřæŸŐåĹŮèąĺäÿ■

ä¡£çŤĺSwigæIJĂåď'■æİĆçŽĎåIJřæŰźæŸŕåőČèČ¡çżŹCäžčçăĄæŔŘä¿Žåď'ğéĞŔçŽĎèĞłåőŽäźL'æŞ■ä¡IJ
è£ŹäÿłäÿżéćŸåď'łåď'ğïijŇè£ŹéĞŇæŮăæşŢåśŢåijĂïijŇä¡ĘæŸŕæĹŚäžňåIJĺæIJňèŁĆè£ŸåL'ľ'åśŢçď'žäžĘäÿĂäž

çňňäÿĂäÿłèĞłåőŽäźL'æŸŕ `%extend` æŇĞäžď'åĚĄèőÿæŰźæşŢèćńéŹĎåŁääĹřåůšå■ŸåIJĺçŽĎçżŞædĎä
æĹŚä¿Ňå■Řäÿ■ïijŇè£ŹäÿłèćńçŤĺæİěæůżåŁääÿĂäÿłPointçżŞædĎä¡ŞçŽĎædĎéĂääŹĺæŰźæşŢãĂĆ
åőČåŘŕäžěèőľä¡ääČŔäÿŇéİćè£Źæăůä¡£çŤĺè£Źäÿłçż ŞædĎä¡Ş ïijŽ

```
>>> p1 = sample.Point(2,3)
>>>
```

åęĆædIJçŢěè£ĞçŽĎèŕİïijŇPointåŕźèśąåřśå£ËéążäžěæŻť'åŁääď'■æİĆçŽĎæŰźåijŔæİěèćńåĹŻåžžïijŽ

```
>>> # Usage if %extend Point is omitted
>>> p1 = sample.Point()
>>> p1.x = 2.0
>>> p1.y = 3
```

çňňäžŇäÿłèĞłåőŽäźL'æűL'åŔŁåĹřåŕź `typemaps.i` åžŞçŽĎåijŢåĚěåŠŇ
`%apply` æŇĞäžď'ïijŇ åőČäijŽæŇĞçď'žSwigåŔĆæŢŕç■¿åŘ■ `int *remainder`

èęĄèćńå¡ŞåĄŽæŸŕè¿ŞåĞžåĂijãĂĆ è£ŹäÿłåőđéŹĚäÿŁæŸŕäÿĂäÿłæĺąåijŔåŇźéĚ■èğĎåĹŹãĂĆ åIJĺæŐĕäÿŇæİĕçŽĎæĽĂæIJL'åčŕæŸŐäÿ■ïijŇäżżä¡ŢæŮűåĂŹåŔłèęĄçćŕäÿŁ `int *remainder` ïijŇäżŰåřśäijŽèćńä¡IJäÿžè¿ŞåĞžãĂĆ è£ŹäÿłèĞłåőŽäźĽæŰźæşŢåŔŕäžĕèől' `divide()` åĞ¡æŢřè£ŤåŻđäÿd'äÿłåĂijãĂĆ

```
>>> sample.divide(42,8)
[5, 2]
>>>
```

æIJĂåŘŐäÿĂäÿłæűL'åŔŁåĹř `%typemap` æŇĞäżd'çŽĎèĞłåőŽäźL'åŔŕèČ¡æŸŕè£ŹéĞŇåśŢçd'žçŽĎæIJĂ äÿĂäÿłtypemapåřśæŸŕäÿĂäÿłåIJĺè¿ŞåĚĕäÿ■çL'źåőŽåŔĆæŢřæĺąåijŔçŽĎèğĎåĹŹãĂĆ åIJĺæIJňèŁĆäÿ■ïijŇäÿĂäÿłtypemapèćńåőŽäźL'äÿžåŇźéĚ■åŔĆæŢřæĺąåijŔ `(double *a, int n)`. åIJĺtypemapåĘĚéČĺæŸŕäÿĂäÿłCäžčçăĄçL'Ğæőţïij ŇåőČåŚłèŕL'SwigæĂŐæăůåřĘäÿĂäÿłPythonåŕ æIJňèŁĆäžčçăĄä¡£çŤĺäžĘPythonçŽĎçijŞå■Ÿå■ŔèőőåőżåŇźéĚ■äżżä¡ŢçIJŇäÿŁåŐżçśżäijijåŘŇçš¿åžęæŢřçż ïijĹæŕŤåęĆNumPyæŢřçżĎãĂĄarrayæĺąåİŮåĹŻåžžçŽĎæŢřçżĎç■L'ïijL'ïijŇæŻť'åd'ŽèŕůåŔĆèĂČ15.3åřŔèŁĆ

åIJĺtypemapäžčçăĄåĘĚéČĺïijŇ$1åŠŇ$2è£ŹæăůçŽĎåŔŸéĞŔæŻ£æ■ćäijŽèŐůåŔŰtypemapæĺąåijŔçŽĎC ïijĹæŕŤåęĆ$1æŸăåřĎäÿž `double *a` ïijL'ãĂĆ$inputæŇĞåŘśäÿĂäÿłä¡IJäÿžè¿ŞåĚĕçŽĎ `PyObject *` åŔĆæŢřïijŇ èĂŇ `$argnum` åřśäžčèąĺåŔĆæŢřçŽĎäÿłæŢřãĂĆ

çijŰåĘŹåŠŇçŘĘèğčtypemapsæŸŕä¡£çŤĺSwigæIJĂå§žæIJňçŽĎåL'■æŔŘãĂĆ äÿ■äžĚæŸŕèŕť'äžčçăĄæŻť'çědçğŸïijŇèĂŇäÿŤä¡ăéIJĂèęĄçŘĘèğčPython C APIåŠŇSwigåŠŇåőĆäžd'äžŠçŽĎæŰźåijŔãĂĆ SwigæŰĞæąćæIJL'æŻť'åd'Žè£ŹæŰźéİćçŽĎçżĘèŁĆïijŇåŔŕäž

äÿ■è£ĞïijŇåęĆædIJä¡ăæIJL'åd'ğéĞŔçŽĎCäžčçăĄéIJĂèęĄèćńæŽť'éIJšäÿžæL'l'åśŢæĺąåİŮãĂĆ SwigæŸŕäÿĂäÿłéİđåÿÿåijžåd'ğçŽĎåůěåĚůãĂĆåĚşéŤőçĆźåIJĺäžŐSwigæŸŕäÿĂäÿłåd'ĎçŘĘCåčŕæŸŐçŽĎçij éĂŽè£Ğåijžåd'ğçŽĎæĺąåijŔåŇźéĚ■åŠŇèĞłåőŽäźL'çżĎäżűïijŇåŔŕäžèèől'ä¡ăæŻť'æŤźåčŕæŸŐæŇĞåőŽåŠŇçş æŻť'åd'Žä£ąæĄŕèŕůåŐżæ§ěéŸĚ Swigç¡ŚçńŹ ïijŇ è£ŸæIJL' çL'źåőŽäžŐPythonçŽĎçŻÿåĚşæŰĞæąč

17.10 15.10 çŤĺCythonåŇĚèčĚCäžčçăĄ

éŮőécŸ

ä¡ăæČşä¡£çŤĺCythonæİěåĹŻåžžäÿĂäÿłPythonæL'l'åśŢæĺąåİŮïijŇçŤĺæİěåŇĚèčĚæ§Řäÿłåůšå■ŸåIJĺçŽĎ

èğčåĘşæŰźæąĹ

ä¡£çŤĺCythonæđĎåžžäÿĂäÿłæL'l'åśŢæĺąåİŮçIJŃäÿŁåŐżå¿ĹæL'ŃåĘŹæL'l'åśŢæIJL'äžŽçśżäijijïijŇ åŻăäÿžä¡ăéIJĂèęĄåĹŻåžžå¿Ĺåd'ŽåŇĚèčĚåĞ¡æŢřãĂĆäÿ■è£ĞïijŇèů§åL'■éİćäÿ■åŘŇçŽĎæŸŕijŇä¡ääÿ■éIJĂ

ä¡IJäÿžåĞĘåd'ĞïijŇåĄĞèő¿æIJňçńääżŃçż■éČĺåĹĘçŽĎçd'žä¿Ńäžčçă ĄåůšçżŕèćńçijŰèŕŚåĹřæ§Řäÿłåŕń `libsample` çŽĎCåĞ¡æŢřåžŞäÿ■äžĘãĂĆ éęŰåĚĹåĹŻåžžäÿĂäÿłåŘ■åŕń `csample.pxd` çŽĎæŰĞäžűïijŇåęĆäÿŃæL'Ăçd'žïijŽ

```
# csample.pxd
#
# Declarations of "external" C functions and structures
```

(continues on next 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£Źäÿ łæŰĞäżűåIJÍCythonäÿ■çŽĎä¡IJçŤĺářśèů§CçŽĎåď'ť'æŰĞäżűäÿĂæăůãĂĆ åĹİåğŃåčřæŸŐ `cdef extern from "sample.h"` æŇĞåőŽäžĘæĽ'Ăå■ęçŽĎCåď'ť'æŰĞäżűãĂĆ æŐěäÿŃæİěçŽĎåčřæŸŐéČ¡æŸŕæİěèĞłäžŐéĆčäÿłåď'ť'æŰĞäżűãĂĆæŰĞäżűåŘ■æŸŕ `csample.pxd` ïijŇèĂŇäÿ■æŸŕ `sample.pxd` âĂŤâĂŤè£ŹçĆźå¿ĹéĞ■èęĄãĂĆ

äÿŃäÿĂæ■ěïijŇåĹŻåžžäÿĂäÿłåŘ■äÿž `sample.pyx` çŽĎéŮőécŸãĂĆ èŕěæŰĞäżűäijŽåőŽäźĽ'åŇĚèčĚåŹĺïijŇçŤĺæİěæąěæŐěPythonèğčéĞŁåŹĺåĹř `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)

```
    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)
```

èŕěæŰĞäżűæŻť 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
)
```

èęĄæđĎåżžæĹŚäžňæţŃèŕŢçŽĎçŻőæăĞæĺąåİŮïijŇåČŔäÿŃéİćè£ŹæăůåĄŽïijŽ

```
bash % python3 setup.py build_ext --inplace
```

(continues on next 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 %
```

åęĆæđIJäÿĂåĹĞéąžåĹl’çŽĎèŕĬïijŇä¡ăåžŤèŕěæIJL’äžĘäÿĂäÿłæL’l’åśŢæĺąåİŮ `sample.so` ïijŇåŔŕåIJĺäÿŃéİćä¿Ńå■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

æIJňèŁĆåŇĚåŘŕäžĘå¿Ĺåd’ŽåL’■éİćæL’Ăèőšç ŽĎéńŸçžğçL’źæĂğïijŇåŇĚæŃňæŢřçżĎæŞ■ä¡IJãĂĄåŇĚæŕŔäÿĂéČĺåĹĘéČ¡äijŽéĂŘäÿłèćńèőšè£řåĹřïijŇä¡ĘæŸŕæĹŚäžňæIJĂåě¡èČ¡åd’■äźääÿĂäÿŃåL’■éİćåĞäářŔèŁåIJĺéąűåśĆïijŇä¡£çŤĺCythonæŸŕå§žäžŐCäźŃäÿŁãĂĆ.pxdæŰĞäżűäžĚäżĚåŔłåŇĚåŘńCáőŽäźL’ïijĹçśżäijij.hä.pyxæŰĞäżűåŇĚåŘŕäžĘåőđçŐřïijĹçśżäijij.cæŰĞäżűïijL’ãĂĆcimport èŕ■åŘèèćńCythonçŤĺæĺěåŕijåĚě.pxdæŰĞäżűäÿ■çŽĎåőŽäźL’ãĂĆ åőČèů§ä¡£çŤĺæŹőéĂŽçŽĎåŁäè¡¡PythonæĺąåİŮçŽĎåŕijåĚèŕ■åŘěæŸŕäÿ■åŘŇçŽĎãĂĆ

åř¡çőą *.pxd* æŰĞäżűåŇĚåŘŕäžĘåőŽäźL’ïijŇä¡ĘåőČäżňåźűäÿ■æŸŕçŤĺæĺèĞłåŁĺåĹŻåžžæL’l’åśŢäžčçăĄç

åŻžæ■ď’ïijŇä¡ăè£ŸæŸŕèęĄåĘŹåŇĚèčĚåĞ¡æŢřãĂĆä¿ŃåęĆïijŇåřśçőŮ `csample.pxd` æŰĞäżűåčřæŸŐäžĘ `int gcd(int, int)` åĞ¡æŢřïijŇ ä¡ääž■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■éIJĂèęĄåŐżåĄŽåd’łåd’ŽçŽĎæŮűãĂĆ Cythonäij Žç Ť§æĹŘåŇĚèčĚäžčçăĄæİěæ■čçąőçŽĎè¡ñæ■ćåŔĆæŢřåŠŇè£ŤåŻđåĂijãĂĆ çżŠåőŽåĹřåśđæĂğäÿŁçŽĎCæŢřæ■őçśžåđŃæŸŕåŔŕéĂL’çŽĎãĂĆäÿ■è£ĞïijŇåęĆæđIJä¡ääŇĚåŘńäžĘåőČäžñ ä¿ŃåęĆïijŇåęĆæđIJæIJL’äžžä¡£çŤĺèt’§æŢřæİěèřČçŤĺè£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¡æŢřåĄŽåŘęåd’ŰçŽĎæčĂæ§ěïijŇåŘłéIJĂèęĄä¡£çŤĺåŘęåd’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()‘‘ åčřæŸŐæIJL’äÿłå¿ĹæIJL’èűčä¡ĘæŸŕæŕŤè¿ČéŽ¿çŘĘèğ åIJÍè£ŹäÿłæŰĞäżűäÿ■ïijŇåĞ¡æŢřèćńåčřæŸŐäÿžçĎűåŘŐäÿĂäÿłbintèĂŇäÿ■æŸŕäÿĂäÿłintãĂĆ åőČäijŽèőľåĞ¡æŢřåĹŻåžžäÿĂäÿłæ■čçąőçŽĎBooleanåĂijèĂŇäÿ■æŸŕçőĂå■ŢçŽĎæŢt’æŢřãĂĆ åŻžæ■ď’ïijŇè£ŤåŻđåĂij0èąĺçd’žFalseèĂŇ1èąĺçd’žTrueãĂĆ

åIJÍCythonåŇĚèčĚåŹĺäÿ■ïijŇä¡ääŘŕäžěéĂL’æŃľåčřæŸŐCæŢřæ■őçśžåđŃïijŇäź§åŘŕäžěä¡£çŤĺæL’ĂæIJ åŕźäžŐ `divide()` çŽĎåŇĚèčĚåŹĺåśŢçd’žäžĘè£ŹæăůäÿĂäÿłä¿Ńå■ŘïijŇåŘŇæŮűè£ŸæIJL’åęĆä¡ŢåŐżåd’Ď

```
def divide(x,y):
    cdef int rem
    quot = csample.divide(x,y,&rem)
    return quot, rem
```

åIJÍè£ŹéĞŇïijŇ`rem` åŘŸéĞŘèćńæŸ¿çd’žçŽĎåčřæŸŐäÿžäÿĂäÿłCæŢt’ådŃåŘŸéĞŘãĂĆ å¡ŞåőČèćńäijăåĚě `divide()` åĞ¡æŢřçŽĎæŮűåĂŹïijŇ`&rem` åĹŻåžžäÿĂäÿłèů§CäÿĂæăůçŽĎæŇĞåŘŠåőČçŽĎæŇĞéŠĹãĂĆ `avg()` åĞ¡æŢřçŽĎäżčçăĄæijŤçd’žäžĘCythonæŻt’éńŸçžğçŽĎçL’źæĂğãĂĆ éęŰåĚĹ `def avg(double[:] a)` åčřæŸŐäžĘ `avg()` æŐěåŘŮäÿĂäÿłäÿĂçźt’çŽĎåŘŇçš¿åžęåĘĚå■ŸèğĘåŻ¿ãĂĆ æIJĂæČŁåěĞçŽĎéČĺåĹĘæŸŕè£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■d'åŇĚèčĚåŹÍäÿ■ïijŇa.size0 åŠŇ &a[0] åĹĘåĹńåijŢçŤÍæŢřçżĎåĚČçt'ääÿłæŢřåŠŇåžŢåśĆæŇ
èŕ■æşŢ <double *> &a[0] æŢŹä¡äæĂŐæăůårĘæŇĞéŠĹè¡ňæ■ćäÿžäÿ■åŘŇçŽĎçśžåđŃžĎæŇĞéŠĹãĂĆ
åL'■æŔŘæŸŕCäÿ■çŽĎ avg() æŐěåŔŮäÿĂäÿłæ■čçąőçśžåđŃçŽĎæŇĞéŠĹãĂĆ
åŔĆèĂČäÿŃäÿĂèŁĆåĚşäžŐCythonåĘĚå■ŸègĘåŻ¿çŽĎæŻt'énŸçžğèőšè£řãĂĆ

éŹd'äžĘåd'ĎçŘĘéĂŽåÿÿçŽĎæŢřçżĎåd'ŰïijŇavg() çŽĎè£Źäÿłä¿Ńå■Řè£ŸåśŢçd'žäžĘåęĆä¡Ţåd'ĎçŘ
èŕ■åŔě with nogil: åčřæŸŐäžĘäÿĂäÿłäÿ■éIJĂèęĄGILåřśèČ¡æL'ğèąŇçŽĎäżčçăĄåİŮãĂĆ
åIJÍè£ŹäÿłåİŮäÿ■ïijŇäÿ■èČ¡æIJL'äżżä¡ŢçŽĎæŹőéĂŽPythonåŕźèśąâĂŤâĂŤåŔłèČ¡ä¡£çŤÍèćńåčřæŸŐäÿž
cdef çŽĎåŕźèśąåŠŇåĞ¡æŢřãĂĆ åŘęåd'ŰïijŇåd'ŰéČÍåĞ¡æŢřå£Ěéążç Őřåőđç ŽĎåčřæŸŐåőČäžňèČ¡äÿ■ä¿İèţ
åŻăæ■d'ïijŇåIJÍcsample.pxdæŰĞäżűäÿ■ïijŇavg() èćńåčřæŸŐäÿž double avg(double
*, int) nogil.

åŕźPointçżŞæđĎä¡ŞçŽĎåd'ĎçŘĘæŸŕäÿĂäÿłæŇŚæĹŸãĂĆæIJňèŁĆä¡£çŤÍèČűåŻŁåŕźèśąåŕĘPointåŕźèśąå
èęAè£ŹæăůåĄŽçŽĎèŕĬïijŇäžŢåśĆCythonäżčçăĄçÍ■å¿őæIJL'çĆźåd'■æİĆãĂĆ
éęŰåĚĹïijŇäÿŃéİćçŽĎåŕijåĚèèćńçŤÍæİěåijŢåĚěCåĞ¡æŢřåžŞåŠŇPython C
APIäÿ■åőŽäźL'çŽĎåĞ¡æŢřïijŽ

```
from cpython.pycapsule cimport *
from libc.stdlib cimport malloc, free
```

åĞ¡æŢř del_Point() åŠŇ Point() ä¡£çŤÍè£ŹäÿłåŁ§èČ¡æİěåĹŽåžžäÿĂäÿłèČűåŻŁåŕźèśąïijŇ
åőČäijŽåŇĚèčĚäÿĂäÿł Point * æŇĞéŠĹãĂĆcdef del_Point() åřĘ del_Point()
åčřæŸŐäÿžäÿĂäÿłåĞ¡æŢřïijŇ åŘłèČ¡éĂŽè£ĞCythonèő£éŰőïijŇèĂŇäÿ■èČ¡äžŐPythonäÿ■èő£éŰőãĂĆ
åŻăæ■d'ïijŇè£ŹäÿłåĞ¡æŢřåŕźåd'ŰéČÍæŸŕäÿ■åŘŕèğĄçŽĎâĂŤâĂŤåőČèćńçŤÍæİěå¡ŞåĄŽäÿĂäÿłåŻđěřČåĞ¡æ
åĞ¡æŢřèŘČçŤÍæŕŤåęĆ PyCapsule_New() ãĂĄPyCapsule_GetPointer()
çŻt'æŐěæİěèĞłPython C APIåźűäÿŤäžěåŘŇæăůçŽĎæŰźåijŔèćńä¡£çŤÍãĂĆ

distance åĞ¡æŢřäžŐ Point() åĹŽåžžçŽĎèČűåŻŁåŕźèśąäÿ■æŔŘåŔŰæŇĞéŠĹãĂĆ
è£ŹéĞŇèęĄæşÍæĎŔçŽĎæŸŕä¡ăäÿ■éIJĂèęĄæŇĚå£ČâijĆåÿÿåd'ĎçŘĘãĂĆ
åęĆædIJäÿĂäÿłéŤŹèŕŕçŽĎåŕźèśąèćńäijăè£ŻæİěïijŇPyCapsule_GetPointer()
äijŽæŁŽåĞžäÿĂäÿłåijĆåÿÿïijŇ ä¡ĘæŸŕCythonåůšçżŔç§ěéĄŞæĂŐäźĹæ§ěæL'¿åĹřåőČïijŇåźűårĘåőČäžŐ
distance() äijăéĂŠåĞžåŐżãĂĆ

åd'ĎçŘĘPointçżŞæđĎä¡ŞäÿĂäÿłçijžçĆźæŸŕåőČçŽĎåőđçŐřæŸŕäÿ■åŔŕèğĄçŽĎãĂĆ
ä¡ăäÿ■èČ¡èő£éŰőäżżä¡Ţåśđæ Ăğæİěæ§ěçIJŃåőČçŽĎåĘĚéČÍãĂĆ
è£ŹéĞŇæIJL'åŘęåd'ŰäÿĂçğ■æŰźæşŢåŐżåŇĚèčĚåőČïijŇårśæŸŕåőŽäźL'äÿĂäÿłæL'l'åśŢçśžåđŃïijŇåęĆäÿŃ

```
# sample.pyx
```

(continues on next 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¿ŃåŔŸéĞŔïijŇæŃěæIJĽäÿĂäÿłæŇĞåŘŠåžŢåśĆPointçżŞæđĎä¡ŞçŽĎæŇĞéŠĹãĂĆ `__cinit__()` åŠŇ `__dealloc__()` æŰźæşŢéĂŽè£Ğ `malloc()` åŠŇ `free()` åĹŻåżžåźűéŤĂæŕĄåžŢåśĆCçżŞæđĎä¡ŞãĂĆxåŠŇyåśđæĂğçŽĎåčřæŸŐèőľä¡ăèŐůåŔŰåŠŇèő¿ç¡őåžŢåśĆçżŞæđĎä¡ŞçŽĎåśđæĂğåĂijãĂĆ `distance()` çŽĎåŇĚèčĚåŹÍè£ŸåŔŕäžěèćńä£őæŤźïijŇä¡£å¿ŮåőČèČ¡æŐěåŔŮ `Point` æĽľåśŢçśżåđŃåőđä¿Ńä¡IJäÿžåŔĆæŢřïijŇ èĂŇäijăéĂŠåžŢåśĆæŇĞéŠĹçżŹĆåĞ¡æŢřãĂĆ

åĄŽäžĘè£ŹäÿłæŤźåŔŸåŘŐïijŇä¡ääijŽåŔŞçŐřæŞ■ä¡IJPointåŕźèśąåřśæŸ¿å¿ŮæŻť'åŁäèĞłçĎűäžĘïijŽ

```
>>> 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)

```
2.0
>>> p1.y
3.0
>>> sample.distance(p1,p2)
2.8284271247461903
>>>
```

æIJňèŁĆåůšçżŕæijŤçd'žäžĘå¿Ĺåd'ŽCythonçŽĎæäÿå£ČçL'źæĂğïijŇä¡ååŕŕäžěäżěæ■d'äÿžå§žåĞĘæİěæäÿ■è£ĞïijŇä¡ăæIJĂåě¡åĚĹåŐžéŸĚèŕžäÿŃåőŸæŰźæŰĞæąčæİěäžĘèğčæŻt'åd'Ža£ąæĄŕãĂĆ

æŐěäÿŃæİěåĞäèŁĆè£ŸäijŽçżğçż■æijŤçd'žäÿĂäžŻCythonçŽĎåĚűäżŰçL'źæĂğãĂĆ

17.11 15.11 çŤÍCythonåĘŹéńŸæĂğèČ¡çŽĎæŢřçżĎæŞ■ä¡IJ

éŮőécŸ

ä¡ăèęĄåĘŹéńŸæĂğèČ¡çŽĎæŞ■ä¡IJæİěèĞłNumPyäźŃçśżçŽĎæŢřçżĎèőąçőŮåĞ¡æŢřãĂĆ ä¡ăåůšçżŕç§ěéĄŞäžĘCythonè£Źæăůç ŽĎåůěåĚůäijŽèőľ'åőČåŔŸå¿ŮçőĂå■ŢïijŇä¡ĘæŸŕåźűäÿ■çąőåőŽèřěæĂ

èğčåĘşæŰźæąĹ

ä¡IJäÿžäÿĂäÿłä¿Ńå■ŘïijŇäÿŃéİćçŽĎäžčçăĄæijŤçd'žäžĘäÿĂäÿłCythonåĞ¡æŢřïijŇçŤÍæİěä£őæŢt'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]
```

èęAçijŰèŕŚåŠŇæđĎåżžè£ŹäÿłæĽľåśŢïijŇä¡ăéIJĂèęĄäÿĂäÿłåČŔäÿŃéİćè£ŹæăůçŽĎ `setup.py` æŰĞäžű 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■čïijŇåźűäÿŤåŔŕäžěéĂĆçŤĺäžŐåď'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¡ãè£Ÿäij ŽåŔŠçŐřè£ŘèąÑçŤ§æĹŘçżŞæđIJéİđåÿÿçŽĎå£ńãĂĆ äÿÑéIćæĹŚäżñåřĘæIJñä¿ŃåŠÑnumpyäÿ■çŽĎåůšå■ŸåIJÍçŽĎ 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¡ăçIJŃåĹřçŽĎïijŇåőČèęĄå£ńå¿Ĺåd'ŽâĂŤâĂŤè£ŹæŸŕäÿĂäÿłå¿ĹæIJL'èűčçŽĎçżŞæđIJïijŇåŻă

èőÍèőž

æIJňèŁĆåĹl'çŤÍäžĘCythonçśžåđŃçŽĎåĘĚå■ŸègĘåŻ¿ïijŇæđĄåd'ğçŽĎçőĂåŇŰäžĘæŢřçżĎçŽĎæŞ■ä¡I cpdef clip() åčřæŸŐäžĘ clip() åŘŇæŮűäÿžCçžğåĹńåĞ¡æŢřäžěåŔŁPythonçžğåĹńåĞ¡æŢřãĂĆ åIJÍCythonäÿ■ïijŇè£ŹäÿłæŸŕå¿ĹéĞ■èęĄçŽĎïijŇåŻăäÿžåőČèąÍçd'žæ■d'åĞ¡æŢřèřČçŤÍèęĄæŕŤåĚűäžŰCytho ïijĹæŕŤåęĆä¡ăæČşåIJÍåŘęåd'ŰäÿĂäÿłäÿ■åŘŇçŽĎCythonåĞ¡æŢřäÿ■èřČçŤÍclip()ïijL'ãĂĆ

çśžåđŃåŔĆæŢř double[:] a åŠŇ double[:] out åčřæŸŐè£ŹäžŽåŔĆæŢřäÿžäÿĂçżt'çŽĎåŘŇçš¿åžęæŢřçżĎãĂĆ ä¡IJäÿžè¿ŞåĚëïijŇåőČäżñäijŽèő£éŮőäżžä¡Ţåőđç0řäžĘåĘĚå■ŸègĘåŻ¿æŐëåŘčçŽĎæŢřçżĎåŕźèśąïijŇè£ŹäÿŁ 3118æIJL'èŕęçżĘåőŽäźL'ãĂĆ åŇĚæŃñäžĘNumPyäÿ■çŽĎæŢřçżĎåŠŇåĘĚç¡őçŽĎarrayåžŞãĂĆ

å¡Şä¡ăçijŰåĘŹçŤ§æĹŘçżŞæđIJäÿžæŢřçżĎçŽĎäżčçăĄæŮűïijŇä¡ăåžŤèŕëéĄţå¿łäÿŁéİćçd'žä¿ŃéĆčæăůè åőČäijŽåřĘåĹŻåżžè¿ŞåĞžæŢřçżĎçŽĎèť'čäżžçżŹèřČçŤÍèĂĚïijŇäÿ■éIJĂèęĄç§ëéĄŞä¡äæŞ■ä¡IJçŽĎæŢřçżĎç ïijĹåőČäżĚäżĚåĄĞèő¿æŢřçżĎåůšçżŔåĞĘåd'Ğåě¡äžĘïijŇåŔłéIJĂèęĄåĄŽäÿĂäżŽåřŘçŽĎæčÄæ§ëæŕŤåęĆçą åIJÍåČŔNumPyäźŃçśžçŽĎåžŞäÿ■ïijŇä¡£çŤÍ numpy.zeros() æĹŰ numpy.zeros_like() åĽŻåżžè¿ŞåĞžæŢřçżĎçŻÿåŕźèĂŇèÍĂæŕŤè¿Čåőźæ ŸŞãĂĆåŘęåd'ŰïijŇèęĄåĽŻåżžæIJłåĽİ ä¡ååŔŕäžěä¡£çŤÍ numpy.empty() æĹŰ numpy.empty_like() . åęĆædIJä¡ăæČşèęĘçŻŰæŢřçżĎåĘĚåőźä¡IJäÿžçżŞæđIJçŽĎèŕÍéĂĽ'æŃľè£ŹäÿďäÿłäijŽæŕŤè¿Čå£ńçĆźãĂĆ

åIJÍä¡ăçŽĎåĞ¡æŢřåődçŐřäÿ■ïijŇä¡ååŔłéIJĂèęĄçőĂå■ŢçŽĎéĂŽè£ĞäÿŇæăĞè£ŘçőŮåŠŇæŢřçżĎæ§ěæ CythonäijŽèť'§èť'čäÿžä¡ăçŤ§æĹŘéńŸæŢĹçŽĎäžčçăĄãĂĆ

clip() åőŽäźL'äźŃåL'■çŽĎäÿď'äÿłèčĚéěřåŹĺåŔŕäžěäijŸåŇŰäÿŇæĂğèČ¡ãĂĆ @cython.boundscheck(False) çIJĄåŐżäžĘæL'ĂæIJL'çŽĎæŢřçżĎèűŁçŢŇæčĂæ§ěïijŇ å¡Şä¡ăç§ëéĄŞäÿŇæăĞèő£éŮőäÿ■äijŽèűŁçŢŇçŽĎæŮűåĂŹåŔŕäžěä¡£çŤĺåőČãĂĆ @cython.wraparound(False) æűĹéŹď'äžĘçŽÿåŕźæŢřçżĎår¿éČĺçŽĎèť'§æŢřäÿŇæăĞçŽĎåď'ĎçŘĘïijĺ åijŢåĚèè£Źäÿď'äÿłèČĚéěřåŹĺåŔŕäžěædĄåď'ğçŽĎæŘŘå■ĞæĂğèČ¡ïijĹæţŇèŕŢè£Źäÿłä¿Ńå■ŘçŽĎæŮűåĂŹåď

äżžä¡ŢæŮűåĂŹåď'ĎçŘĘæŢřçżĎæŮűïijŇçăŤçl'ůåźűæŤźåŰĎåžŢåśĆçőŮæşŢåŘŇæăůåŔŕäžěædĄåď'ğçŽ ä¿ŃåęĆïijŇèĂČèŹŠåŕź clip() åĞ¡æŢřçŽĎåęĆäÿŇä£őæ■čïijŇä¡£çŤĺæİąäžűèąĺè¿¿åijŔïijŽ

```
@cython.boundscheck(False)
@cython.wraparound(False)
cpdef clip(double[:] a, double min, double max, double[:] out):
```

(continues on next 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£Źäÿłçl'ĹæIJñçŽĎäžčçăĄè£ŘèąŇéĂ§åžęèęĄå£ń50%äžëäÿŁïijĹ2.44çğ
`timeit()` æţŃèŕŢçŽĎ3.76çğŠïijL'ãĂĆ

åĹřè£ŹéĞŇäÿžæ■ćïijŇä¡ääŔřèČ¡æČşç§ěéĄŞè£Źçğ■äžčçăĄæĂŐäźĹèČ¡èů§æL'ŃåĘŹCèŕ■èĺĂPKåŚćïij§
ä¿ŃåęĆïijŇä¡ääŔřèČ¡åĘŹäžĘåęĆäÿŃçŽĎCåĞ¡æŢřåźűä¡£çŤĺåL'■éİćåĞäèŁĆçŽĎæŁĂæIJŕæİěæL'ŃåĘŹæL'l'

```
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);
  }
}
```

æĹŚäžňæšąæIJL'åśŢçd'žè£ŹäÿłçŽĎæL'l'åśŢäžčçăĄïijŇä¡ĘæŸŕèŕŢéłŇäźŃåŘŐïijŇæĹŚäžňåŔŚçŐřäÿĂä
æIJĂåžŢäÿŃçŽĎäÿĂèąŇæŕŤä¡ăæČşèśąçŽĎè£ŘèąŇçŽĎå£ńå¿Låd'ŽãĂĆ

ä¡ääŔŕäžěåŕźåőđä¿ŃäžčçăĄæđĎåžžåd'Žäÿłæl'l'åśŢãĂĆ åŕźäžŐæ§ŘäžŻæŢřçżĎæŞ■ä¡IJïijŇæIJĂåě¡èęĄ
èęĄè£ŹæăůåĄçŽĎèŕĺïijŇéIJĂèęĄä£őæŤźäžčçăĄïijŇä¡£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¡ăæČşåĘŹäÿĂäÿłæŞ■ä¡IJäžŇçžť'æŢřçżĎçŽĎçL'ĹæIJñïijŇäÿŃéİćæŸŕåŘŕäžěåŔĆèĂČäÿŃïijŽ

```
@cython.boundscheck(False)
@cython.wraparound(False)
cpdef clip2d(double[:,:] a, double min, double max, double[:,:]␣
↪out):
    if min > max:
```

(continues on next 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]
```

åÿŇæIJŽèŕżèĂĚäÿ■èęĄå£ŸäžĘæIJñèŁĆæL’ĂæIJL’äžčçăĄéČ¡äÿ■äijŽçżŚåőŽåĹřæ§ŘäÿłçL’źåőŽæŢřçżĎ è£ŹæăůäžčçăĄåřśæŻť’æIJL’çĄţæť’żæĂğãĂĆ äÿ■è£ĞïijŇèęĄæşĺæĎŔçŽĎæŸŕåęĆædIJåď’ĎçŘĘæŢřçżĎèęĄæ è£ŹäžŽåĘĚåőźåůšçżŔèűĚåĞžæIJñèŁĆèŇČåŻť’ïijŇæŻť’åď’Žä£ąæĄfèŕůåŔĆèĂČ PEP 3118 ïijŇ åŘŇæŮű CythonæŰĞæąčäÿ■åĚşäžŐâĂIJçśžåđŃåĘĚå■ŸèğĘåŻ¿âĂİ çŕĞäź§åĂijå¿ŮäÿĂèŕżãĂĆ

17.12 15.12 åřĘåĞ¡æŢřæŇĞéŠĹè¡ñæ■ćäÿžåŔŕèřČçŤĺåŕźèśą

éŮőécŸ

ä¡ăåůšçżŔèŐůå¿ŮäžĘäÿĂäÿłèćńçijŰèŕŚåĞ¡æŢřçŽĎåĘĚå■ŸåIJŕåİĂïijŇæČşåřĘåőČè¡ñæ■ćæĹŘäÿĂäÿłP è£ŹæăůçŽĎèŕİä¡ăåřśåŔŕäžěåřĘåőČä¡IJäÿžäÿĂäÿłæL’ľåśŢåĞ¡æŢřä¡£çŤĺäžĘãĂĆ

èğčåĘşæŰźæąĹ

ctypes æĺąåİŮåŔŕèćńçŤĺæİěåĹŻåżžåŇĚèčĚäżżæĎŔåĘĚå■ŸåIJŕåİĂçŽĎPythonåŔŕèřČçŤĺåŕźèśąãĂĆ äÿŇéİćçŽĎä¿Ńå■ŘæijŤçď’žäžĘæĂŐæăůèŐůåŔŰCåĞ¡æŢřçŽĎåŐ§ågŇãĂĄåžŢåśĆåIJŕåİĂïijŇäžěåŔŁąęĆä¡

```
>>> 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)

```
0.9092974268256817
>>> func(0)
0.0
>>>
```

èőÍèőž

èęĄæđĎåżžäÿĂäÿłåŔŕèřČçŤĺåŕźèśąïijŇä¡ăéęŰåĚĹéIJĂèęĄåĹŻåżžäÿĂäÿł
CFUNCTYPE åőđä¿ŃãĂĆ CFUNCTYPE() çŽĎçňňäÿĂäÿłåŔĆæŢřæŸŕè£ŤåŻđçśżåđŃãĂĆ
æŐëäÿŃæİěçŽĎåŔĆæŢřæŸŕåŔĆæŢřçśżåđŃãĂĆäÿĂæŮęä¡ăåőŽäźL’äžĘåĞ¡æŢřçśżåđŃïijŇä¡ăåřśèČ¡åřĘåőČ
çŤ§æĹŘçŽĎåŕźèśąèćńå¡ŞåĄŽæŹőéĂŽçŽĎåŔŕéĂŽè£Ğ ctypes
èő£éŮőçŽĎåĞ¡æŢřæİëä¡£çŤĺãĂĆ

æIJňèŁĆçIJŃäÿŁåŐżåŔŕèČ¡æIJL’çĆźçěđçğŸïijŇåĄŔåžŢåśĆäÿĂçĆźãĂĆ
ä¡ĘæŸŕïijŇä¡ĘæŸŕåőČèćńáź£æşŻä¡£çŤĺäžŐåŘĎçğ■énŸçžğäżčçăĄçŤ§æĹŘæŁĂæIJŕæŕŤåęĆå■şæŮűçijŰèŕŠ

ä¿ŃåęĆïijŇäÿŃéİćæŸŕäÿĂäÿłä¡£çŤĺ `llvmpy` æL’l’åśŢçŽĎçőĂå■Ţä¿Ńå■ŘïijŇçŤĺæİěædĎåżžäÿĂäÿłåřI
åźűåřĘåĚűè¡ńæ■ćäÿžäÿĂäÿłPythonåŔŕèřČçŤĺåŕźèśąãĂĆ

```
>>> 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■æŸŕèŕťåIJĺè£ŹäÿłåśĆéİćçŁŕäžĘäżżä¡ŢéŤŹèŕŕåřśäijŽåŕijèĞťPythonèğčéĞŁåŹĺæŇĆæŐL’ãĂĆ
èęĄèőřå¿ŮçŽĎæŸŕä¡ăæŸŕåIJĺçŻťæÔěèů§æIJžåŹĺçžğåĹńçŽĎåĘĚå■ŸåIJřåİĂåŠŇæIJňåIJřæIJžåŹĺçăĄæL’Şä

17.13 15.13 äijäéĂŠNULLçżŞář¿çŽĎå■ŮçňęäÿšçżŹCåĞ¡æŢřážŞ

éŮőécŸ

ä¡ăèęĄåĘŹäÿĂäÿłæĽľåśŢæĺąåİŮïijŇéIJĂèęĄäijăéĂŠäÿĂäÿłNULLçżŞář¿çŽĎå■ŮçňęäÿšçżŹCåĞ¡æŢřääÿ■è£ĞïijŇä¡ăäÿ■æŸŕå¿ĹçąőáőŽæĂŐæăůä¡£çŤÍPythonçŽĎUnicodeå■Ůçňęäÿšå Őżåőđç ŐřåőČãĂĆ

èğčåĘşæŰźæąĹ

èőÿåď ŽCåĞ¡æŢřážŞåŇĚåŘńäÿĂäžŻæŞ■ä¡IJNULLçżŞář¿çŽĎå■ŮçňęäÿšïijŇèćńåčřæŸŐçśżåđŃäÿž `char *`. èĂČèŹŚåęĆäÿŃçŽĎCåĞ¡æŢřïijŇæĹŚäžňçŤĺæİěåĄŽæijŤçď žåŠŇætŃèŕŢçŤĺçŽĎïijŽ

```
void print_chars(char *s) {
    while (*s) {
        printf("%2x ", (unsigned char) *s);

        s++;
    }
    printf("\n");
}
```

æ■ď åĞ¡æŢřäijŽæĽŞå■řèćńäijăè£Żæİěå■ŮçňęäÿšçŽĎæŕŔäÿłå■ŮçňęçŽĎå■ĄåĚ■è£ŻåĹűèąÍçď žïijŇè£Ź

```
print_chars("Hello");   // Outputs: 48 65 6c 6c 6f
```

åŕźäžŐåIJÍPythonäÿ■èřČçŤĺè£ŹæăůçŽĎCåĞ¡æŢřïijŇä¡ăæIJL'åĞăçğ■éĂĽ'æŃľãĂĆ éęŰåĚĹïijŇä¡ăåŔŕäżěéĂŽè£ĞèřČçŤÍ `PyArg_ParseTuple()` åźűæŇĞåőŽâĂİyâĂIJè¡ňæ■ćçăĄæİěéŹŘåĹűåőČåŔłèČ¡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¡£çŤĺæŰźæşŢåęĆäÿŃãĂĆäžŤçżĘèğĆår§åţŇåĚëäžĘNULLå■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)

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'str' does not support the buffer interface
>>>
```

åęĆædIJä¡ăæČşäijăéĂŠUnicodeå■ŮçňęäýšïijŇåIJÍ PyArg_ParseTuple() äý■ä¡£çŤÍâĂİsâĂIJæăijåijŔçăĄïijŇåęĆäýŃïijŽ

```
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¡Sèćńä¡£çŤÍçŽĎæŮűåĂŹïijŇåőČäijŽèĞłåŁÍåřĘæĽĂæIJL'å■Ůçňęäýšè¡ňæ■ćäýžäżěNULLçżŞář¿çŽĎU8çijŰçăĄãĂĆä¿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
>>>
```

åęĆædIJåŽääýžæ§ŘäžŽåŐ§åŽäïijŇä¡ăèęĄçŻt'æŐěä¡£çŤÍ PyObject * èĂŇäý■èČ¡ä¡£çŤÍ PyArg_ParseTuple() ïijŇäýŃéIćçŽĎä¿Ńå■ŘåŘŚä¡ååśŢçd'žäžĘæĂŐæăůäżŐå■ŮèŁĆåŠŇå■Ůçňęäýšářźèśąäý■æčĂæ§ěåŠŇæŔŘåŔŰäýchar * 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)

```
    }
    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’■éÍćäÿď’çğ■è¡ňæ■ćéČ¡åŔŕäžěçąőä£ÍæŸŕNULLçżŞår¿çŽĎæŢřæ■őïijŇ
ä¡ĘæŸŕåőČäżňåźűäÿ■ačĂæ§ěå■Ůçňęäÿšäÿ■éŮt’æŸŕåŘęåţŇåĚěäžĘNULLå■ŮèŁĆãĂĆ
åŻăæ■ď’ïijŇåęĆæđIJè£Źäÿłå¿ĹéĞ■èęĄçŽĎèŕÍïijŇéĆčä¡ăéIJĂèęĄèĞłåůśåŐżåĄŽæčĂæ§ěäžĘãĂĆ

èőÍèőž

åęĆæđIJåŔŕèČ¡çŽĎèŕÍïijŇä¡ăåžŤèŕěéĄ£åĚ■åŐżåĘŹäÿĂäžŻä¿ÌèţŰäžŐNULLçżŞår¿çŽĎå■ŮçňęäÿšïijŇ
æIJĂåě¡çżŞåŘĹä¡£çŤÍäÿĂäÿłæŇĞéŠĹåŠŇéŢ£åžęåĂijæÌěåď’ĎçŘĘå■Ůçňęäÿšã ĂĆ
äÿ■è£ĞïijŇæIJL’æŮűåĂŹä¡ăå£ĚéążåŐżåď’ĎçŘĘCèŕ■èÍĂéĄŮçŢŹäżčçăĄæŮűåřśæšąå¿ŮéĂL’æŃl’äžĘãĂĆ

åř¡çőąå¿Ĺåőźæ ŸŞä¡£çŤÍïijŇä¡ĘæŸŕå¿Ĺåőźæ ŸŞå£¡ègĘçŽĎäÿĂäÿłéŮőécŸæŸŕåIJÍ
`PyArg_ParseTuple()` äÿ■ä¡£çŤÍâĂIJsâĂÌæăijåijŔåŇŰçăĄäijŽæIJL’åĘĚå■Ÿæ■§èĂŮãĂĆ
ä¡Ęä¡ăéIJĂèęĄä¡£çŤÍè£Źçğ■è¡ňæ■ćçŽĎæŮűåĂŹïijŇäÿĂäÿłUTF-
8å■ŮçňęäÿšècńåĹŻåžžźűæřÿäźĚéŹĎåŁăåIJÍåŐ§ăğŇå■ŮçňęäÿšåŕźèśąäÿŁéÍćãĂĆ
åęĆæđIJåŐ§ăğŇå■Ůçňęäÿšå ŇĚåŘńéÌđASCIIå■Ůçňę ŽĎèŕÍïijŇåřšäijŽåŕijèĞt’å■Ůçňęäÿš ŽĎåřžåŕÿåćđåĹřäÿ

```
>>> 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¡ăåIJÍäźŐè£ŹäÿłåĘĚå■ŸçŽĎæ■§èĂŮïijŇä¡æIJĂåě¡éĞ■åĘŹä¡ăçŽĎCæL’l’åśŢäžčçăĄïijŇèől’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£ĞĝĖ£ŹäÿŁä£ŏæŤźïijŇäÿĂäÿŁUTF-8çijŰçăAçŽĎå■Ůçňęäÿšæăźæ■őéIJĂèęĄèćńåĹŻåžžïijŇçĎűåŘĆ

```
>>> 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
>>>
```

åęĆædIJä¡ăèŕŢçİĂäijăéĂŠNULLçżŞåř¿å■ŮçňęäÿšçżŹctypesåŇĚèčĚè£ĞçŽĎåĞ¡æŢřïijŇ
èęĄæşÍæĎŔçŽĎæŸŕctypesåŔŁèČ¡åĚĄèőÿäijăéĂŠå■ŮèŁĆïijŇåźűäÿŤåőČäÿ■äijŽæčĂæ§ĕäÿ■éŮť åţŇåĚěçŽĎ

```
>>> 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
>>>
```

åęĆædIJä¡ăæČşäijăéĂŠå■ŮçňęäÿšèĂŇäÿ■æŸŕå■ŮèŁĆïijŇä¡ăéIJĂèęĄåĚĹæĽğèąŇæĽŃåŁĺçŽĎUTF-8çijŰçăĄãĂĆä¿ŃåęĆïijŽ

```
>>> print_chars('Hello World'.encode('utf-8'))
48 65 6c 6c 6f 20 57 6f 72 6c 64
>>>
```

åŕźäžŐåĚűäżŰæĽľåśŢåůěåĚůïijĹæŕŤåęĆSwigãĂĄCythonïijĽïijŇ

åIJÍä¡ăä¡£çŤĺåőČäžňäijăéĂŠå■ŮçňęäÿšçżŹCäżčçăĄæŮűèęĄåĚĹåě¡åě¡å■ęäźăçŻÿåžŤçŽĎäÿIJèě£äžEąĂĆ

17.14 15.14 äijăéĂŠUnicodeå■ŮçňęäÿšçżŹCåĞ¡æŢřåžŞ

éŮőécŸ

ä¡ăèęĄåĘŹäÿĂäÿłæL'ľåśŢæĺąåİŮïijŇéIJĂèęĄåřĘäÿĂäÿłPythonå■ŮçňęäÿšäijăéĂŠçżŹCçŽĎæ§Řäÿłåž§

èğčåĘşæŰźæąĹ

è£ŹéĞŇæĹŚäžňéIJĂèęĄèĂČèŹŚå¿Ĺåd'ŽçŽĎéŮőécŸïijŇä¡ĘæŸŕæIJĂäÿżèęĄçŽĎéŮőécŸæŸŕçŐřå■Ÿç
åŻăæ■d'ïijŇä¡ăçŽĎæŇŚæĹŸæŸŕåřĘPythonå■Ůçňęäÿšè¡ňæ■ćäÿžäÿĂäÿłèČ¡èćńCçŘĘèğčçŽĎå¡ćåijŔãĂĆ

äÿžäžĘæijŤçd'žçŽĎçŻőçŽĎïijŇäÿŃéIćæIJL'äÿd'äÿłCåĞ¡æŢřïijŇçŤĺæİěæŞ■ä¡IJå■ŮçňęäÿšæŢřæ■őåźűè
äÿĂäÿłä¡£çŤĺå¡ćåijŔäÿž `char *, int` å¡ćåijŔçŽĎå■ŮèŁĆïijŇ
èĂŇåŘęäÿĂäÿłä¡£çŤĺå¡ćåijŔäÿž `wchar_t *, int` çŽĎåő¡å■Ůçňęå¡ćåijŔïijŽ

```
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");
}
```

åŕźäžŐéIćåŘŚå■ŮèŁĆçŽĎåĞ¡æŢř `print_chars()` ïijŇä¡äéIJĂèęĄåřĘPythonå■Ůçňęäÿšè¡ňæ■ćäÿžä
8. äÿŃéIćæŸŕäÿĂäÿłè£ŹæăůçŽĎæL'ľåśŢåĞ¡æŢřä¿Ńå■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)

```
  Py_RETURN_NONE;
}
```

åŕźäžŐéĆčäžŻéIJĂèęĄåď ĎçŘĘæIJžåŹĺæIJňåIJř wchar_t çśżåđŃçŽĎåžŞåĞ¡æŢřiijŇä¡ăåŔŕäżěåČŔäÿŃéİćè£ŹæăůçijŰåĘŹæĽľ'åśŢäżčçăĄïijŽ

```
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;
}
```

äÿŃéİćæŸŕäÿĂäÿłäžď'äžŠäijŽèŕİæİěæijŤçď'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£ŹäÿłéİćåŘŚå■ŮèŁĆçŽĎåĞ¡æŢř print_chars() æŸŕæĂŐæăůæŐěåŔŮUTF-8çijŰçăĄæŢřæ■őçŽĎïijŇ äžěåŔŁ print_wchars() æŸŕæĂŐæăůæŐěåŔŮUnicodeçijŰçăĄåĂijçŽĎ

èőÍèőž

åIJÍçżğçż■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;
}
```

åęĆædIJä¡ăäż■çĎűè£ŸæŸŕæČşèęĄäijăéĂŠå■ŮçňęäÿšïijŇ ä¡ăéIJĂèęĄç§ěéĄŞPython 3åŔŕä¡£çŤĺäÿĂäÿłåŘĹéĂĆçŽĎå■Ůçňęäÿšèąĺçď'žïijŇ

åőČåźűäÿ■çŽťæŐěæŸäåřĎåĹŕä¡£çŤĺæăĞåĞĘçśžåđŃ char * æĹŰ wchar_t * ïijĹæŽťåďŽçżĘèŁĆåŔĆèĂČPEP 393ïijĽçŽĎCåĞ¡æŢřåžŞãĂĆ åŽäæ■ďïijŇèęĄåIJÍCäÿ■èąĺçďžèŁŹäÿłå■Ůçňęäÿšæ Ţřæ■őïijŇäÿĂäžŽè¡ňæ■ćè£Ÿæ Ÿŕå£ĚéąžèęĄçŽĎãĂĆ åIJÍ PyArg_ParseTuple() äÿ■ä¡£çŤĺâĂİs#âĂİ åŠŇâĂİu#âĂİæäijåijŔåŇŰçăĄåŔřäżěåőĽåĚĺçŽĎæĽğèą

äÿ■è£Ğè£Źçğ■è¡ňæ■ćæIJĽäÿłçijžçĆźåřśæŸŕåőČåŔŕèČ¡äijŽåŕijèĞť åŐ§åğŃå■ŮçňęäÿšåŕźèśąçŽĎåřžåŕ äÿĂæŮçè¡ňæ■ćè£ĞåŘŐïijŇäijŽæIJĽäÿĂäÿłè¡ňæ■ćæŢřæ■őçŽĎåď■åĹűéŽĎåŁäåĹřåŐ§åğŃå■Ůçňęäÿšåŕźèś ä¡ăåŔřäżěèğĆåŕ§äÿŃè£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
>>>
```

åŕźäžŐåřŚéĞŔçŽĎå■ŮçňęäÿšåŕźèśąïijŇåŔŕèČ¡æšąäžĂäźĹå¡śåŞ■ïijŇ ä¡ĘæŸŕåęĆæđIJä¡ăéIJĂèęĄåIJÍæĽľåśŢäÿ■åď ĎçŘĘåď ğéĞŔçŽĎæŰĞæIJňïijŇä¡ăåŔŕèČ¡æČşéĄ£åĚ■è£Źäÿł äÿŃéİćæŸŕäÿĂäÿłä£őèőćçĽĹæIJňåŔřäżěéĄ£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 çŽĎåďĎçŘĘæŮűæČşèęĄéĄ£åĚ■åĘĚå■Ÿæ■§èĂŮåřśæŻť åŁăéŽ¿åŁđäžĘãĂĆ åIJÍåĘĚéČĺïijŇPythonä¡£çŤĺæIJĂéńŸæŢĹçŽĎèąĺçďžæİěå■ŸåĆĺå■Ůçňęäÿšã ĂĆ ä¿ŃåęĆïijŇåŔłåŇĚåŘńASCIIçŽĎå■Ůçňęäÿšèćńå■ŸåĆĺäÿžå■ŮèŁĆæŢřçżĎïijŇ èĂŇåŇĚåŘńèŇČåŻťäżŐU+0000åĹřU+FFFFçŽĎå■Ůçňęç ŽĎå■Ůçňęäÿšä¡£çŤĺåŔŇå■ŮèŁĆèąĺçďžãĂĆ çŢśäžŐåŕźäžŐæŢřæ■őçŽĎèąĺçďžä¡ćåijŔäÿ■æŸŕå■ŢäÿĂçŽĎïijŇä¡ääÿ■èČ¡åřĘåĘĚéČĺæŢřçżĎè¡ňæ■ćäÿž wchar_t * çĎűåŘŐæIJ§æIJŻåőČèČ¡æ■čçąőçŽĎåůěä¡IJãĂĆ ä¡ăåžŤèŕěåĹŻåżžäÿĂäÿł wchar_t æŢřçżĎåźűåŘŠåĚűäÿ■åď■åĹűæŰĞæIJňãĂĆ PyArg_ParseTuple() çŽĎâĂİu#âĂİæäijåijŔçăĄåŔřäżěåÿőåŁľä¡ăéńŸæŢĹçŽĎåőŇæĹŘåőČïijĹåőČåřĘåď■åĹűçŽşæđIJéŽĎåŁäåĹ

åęĆæđIJä¡ăæČşéĄ£åĚ■éŢ£æŮűéÚťåĘĚå■Ÿæ■§èĂŮïijŇä¡ăåŤŕäÿĂçŽĎéĂĽ'æŃľåřśæŸŕåď■åĹűUnicod
åřĘåőČäijăéĂŠçżŹCåĞ¡æŢřïijŇçĎűåŘŐåŻđæŤűè£ŹäÿłæŗřçżĎçŽĎåĘĚå■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()
åĹŻåżžäÿĂäÿłäÿťæŮűçŽĎwchar_tçijŞåĘšåźűåď■åĹűæŢřæ■őè£ŻåŐżãĂĆ
è£ŹäÿłçijŞåĘšèćńäijăéĂŠçżŹCçĎűåŘŐèćńéĞŁæŤ¿æŐĽ'ãĂĆ ä¡ĘæŸŕæĹŚåĘŹè£ŹæIJňäźęçŽĎæŮűåĂŹïijŇè

åęĆæđIJä¡ăç§ěéĄŞCåĞ¡æŢřåžŞéIJĂèęĄçŽĎå■ŮèŁĆçijŰçăĄåźűäÿ■æŸŕUTF-8ïijŇ
ä¡ååŔŕäžěåijžåĹűPythonä¡£çŤĺæĽ'ľåśŢçăĄæİěæĽ'ğèąŇæ■čçąőçŽĎè¡ňæ■ćïijŇåřśåČŔäÿŃéİćè£ŹæăůïijŽ

```
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)

```
  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£ŹäÿłäżčçăĄäÿ■ïijŇ`PyUnicode_KIND()` åŠŇ `PyUnicode_DATA()` è£Źäÿď äÿłåőŔåŠŇUnicodeçŽĎåŔŕåŔŸåő¡åžęå■ŸåĆĺæIJĽåĚşïijŇè£ŹäÿłåIJÍPEP 393äÿ■æIJĽæŔŔè£řãĂĆ `kind` åŔŸéĞŔçijŰçăĄåžŢåśĆå■ŸåĆĺïijĹ8ä¡■ãĂĄ16ä¡■æĹŰ32ä¡■ïijĽäžĕåŔŁæŇ åIJÍåőđéŹĚæČĚåĘţäÿ■ïijŇä¡ääźűäÿ■éIJĂèęĄç§ěéĄŞäżżä¡Ţèů§è£ŹäžŻåĂijæIJĽåĚşçŽĎäÿIJèě£ïijŇ åŔłéIJĂèęĄåIJíæŔŘåŔŰå■ŮçňęçŽĎæŮűåĂŹåřĘåőČäžňäijăçżŹ `PyUnicode_READ()` åőŔãĂĆ

è£ŸæIJĽæIJĂåŘŐåĞäåŔěïijŽå¡ŞäżŐPythonäijăéĂŠUnicodeå■ŮçňęäÿšçżŹCçŽĎæŮűåĂŹïijŇä¡äåžŤè åęĆæđIJæIJĽUTF-8åŠŇåő¡å■Ůçňęäÿď çğ■éĂĽæŃľïijŇèŕůéĂĽæŃľUTF-8. åŕźUTF-8çŽĎæŤŕæŇĄæŻť åŁăæŹőéĄ■äÿĂäžŻïijŇäź§äÿ■åőźæŸŞçŁŕéŤŹïijŇèğčéĞŁåŹĺäź§èČ¡æŤŕæŇĄçŽĎæŻť åě æIJĂåŘŐïijŇçąőä£Ĭä¡ääžŤçżĘéŸĚèŕżäžĘ åĚşäžŐåď ĐçŘĘUnicodeçŽĎçŻÿåĚşæŰĞæąč

17.15 15.15 Cå■Ůçňęäÿšè¡ňæ■ćäÿžPythonå■Ůçňęäÿš

éŮőéćŸ

æĂŐæäůåřĘCäÿ■çŽĎå■Ůçňęäÿšè¡ňæ■ćäÿžPythonå■ŮèŁĆæĹŰäÿĂäÿłå■Ůçňęäÿšåŕźèśąïij§

èğčåĘşæŰźæąĹ

Cå■Ůçňęäÿšä¡£çŤĺäÿĂåŕź `char *` åŠŇ `int` æİěèąĺçď žïijŇä¡ăéIJĂèęĄåĘşåőŽå■ŮçňęäÿšåĹŕåžŢæŸŕçŤĺäÿĂäÿłåŐ§åğŃå■ŮèŁĆå■Ůçňęäÿšè£ŸæŸŕäÿĂäÿłUnicodeå■Ůç å■ŮèŁĆåŕźèśąåŔŕäžĕåČŔäÿŃéİćè£Źæăůä¡£çŤĺ `Py_BuildValue()` æİěæđĐåžžïijŽ

```
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■ŮçňęäÿšïijŇäźűäÿŤä¡ăç§ěéĄŞ `s` æŇĞåŘŠäžĘUTF-8çijŰçăĄçŽĎæŢřæ■őïijŇåŔŕäžĕä¡£çŤĺäÿŃéİćçŽĎæŰźåijŔïijŽ

```
PyObject *obj = Py_BuildValue("s#", s, len);
```

åęĆædIJ s ä¡£çŤĺåĚűäžŰçijŰçăĄæŰźåijŔïijŇéĆčäźĹåŔŕäžěåČŔäÿŃéIćä¡£çŤĺ PyUnicode_Decode() æİěædĎåžžäÿĂäÿłå■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");
```

åęĆædIJä¡ăæĄřåě¡æIJL'äÿĂäÿłçŤĺ wchar_t *, len åŕźèąĺçd'žçŽĎåő¡å■ŮçňęäÿšïijŇæIJL'åĞäçg■éĂL'æŃl'æĂğãĂĆéęŰåĚĹä¡ääŔŕäžěä¡£çŤĺ Py_BuildValue() ïijž

```
wchar_t *w;     /* Wide character string */
int len;        /* Length */

PyObject *obj = Py_BuildValue("u#", w, len);
```

åŔęåd'ŰïijŇä¡ăè£ŸåŔŕäžěä¡£çŤĺ PyUnicode_FromWideChar() :

```
PyObject *obj = PyUnicode_FromWideChar(w, len);
```

åŕźäžŐåő¡å■ŮçňęäÿšïijŇåźűæšąæIJL'åŕźå■Ůçňęæţŕæ■őè£ŻèąŇèğčædŘâĂŤâĂŤåőČèćńåĄĞåőŽæŸŕåŐ§

èőÍèőž

åřĘCäÿ■çŽĎå■Ůçňęäÿšè¡ňæ■ćäÿžPythonå■ŮçňęäÿšéĄţå¿łåŠŇI/OåŘŇæăůçŽĎåŐ§åĹŹãĂĆ äźş§åřśæŸŕèřťïijŇæIěèĞłCäÿ■çŽĎæŢŕæ■őå£Ěéążæăźæ■őäÿĂäžŻèğčçăĄåŹÍèćńæŸ¿åijŔçŽĎèğčçăĄäÿžäÿĂä éĂŽåÿÿçijŰçăĄæäijåijŔåŇĚæŃňASCIIãĂĄLatin-1åŠŇUTF-8. åęĆædIJä¡ääźűäÿ■çąőåőŽçijŰçăĄæŰźåijŔæĹŰèĂĚæŢŕæ■őæŸŕäžŇè£ŻåĹűçŽĎïijŇä¡ăæIJĂåě¡åřĘå■Ůçňęäÿ å¡ŞædĎéĂääÿĂäÿłåŕźèśąçŽĎæŮűåĂŹïijŇPythonéĂŽåÿÿäijŽåd'■åĹűä¡ăæŔŘä¿ŻçŽĎå■Ůçňęäÿšæţŕæ■õãĂĆ åęĆædIJæIJL'å£ĚèęĄçŽĎèŕĬïijŇä¡ăéIJĂèęĄåIJĺåŘŐéĺćåŐžéĞŁæŤ¿Cå■Ůçňęäÿšã ĂĆ åŘŇæŰűïijŇäÿžäžĘèől'çĺŃåžŘæŻt'åŁääĄěåčőïijŇä¡ääžŤèřěåŘŇæŰűä¡£çŤĺäÿĂäÿłæŇĞéŠĹåŠŇäÿĂäÿłåd'ğ èĂŇäÿ■æŸŕä¿ÌètŰNULLçżŞář¿æţŕæ■őæİěåĹŻåžžå■Ůçňęäÿšã ĂĆ

17.16 15.16 äÿ■çąőåőŽçijŰçăĄæăijåijŔçŽĎCå■Ůçňęäÿš

éŮőécŸ

ä¡ăèęĄåIJĺCåŠŇPythonçŻt'æŐěæİěåŻđè¡ňæ■ćå■ŮçňęäÿšïijŇä¡ĘæŸŕCäÿ■çŽĎçijŰçăĄæăijåijŔåźűäÿ■çă ä¿ŃåęĆïijŇåŔŕèČ¡Cäÿ■çŽĎæŢŕæ■őæIJ§æIJŻæŸŕUTF-8ïijŇä¡ĘæŸŕåźűæšąæIJL'åijžåĹűåőCå£Ěéążæ Ÿŕã ĂĆ ä¡ăæČşçijŰåĘŹäžčçăĄæİěäžěäÿĂçğ■äijŸéŽĚçŽĎæŰźåijŔåd'ĎçŘĘè£ŹäžŻäÿ■åŘĹæăijæŢŕæ■őïijŇè£Źæăůå

èğčåĘşæŰźæąĹ

äÿŃéÌćæŸŕäÿĂäžŻCçŽĎæŢřæ■őåŠŇäÿĂäÿłåĞ¡æŢřæÌěæijŤçď žè£ŹäÿłéŮőécŸïijŽ

```
/* 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-8åŠŇäÿ■åŘĹæăijæŢřæ■őãĂĆ äÿ■è£ĞïijŇåęĆæđIJçŤÍæĹůåIJÍCäÿ■èřČçŤÍ print_chars(sdata, slen) ïijŇåőČçijžèČ¡æ■čåÿÿåůëä¡IJãĂĆ çŐřåIJÍåĄĞèő¿ä¡ăæČşåřĘ sdata çŽĎåĘĚåőźè¡ňæ■ćäÿžäÿĂäÿłPythonå■Ůçňęäÿšã ĂĆ è£ŻäÿĂæ■ěåĄĞèő¿ä¡ăåIJÍåŘŐéÌćè£ŸæČşéĂŽè£ĞäÿĂäÿłæĽľåśŢåřĘéĆčäÿłå■Ůçňęäÿšäijääÿł print_chars() åĞ¡æŢřãĂĆ äÿŃéÌćæŸŕäÿĂçğ■çŤÍæÌěä£ÌæŁď åŐ§ågŃæŢřæ■őçŽĎæŰźæşŢïijŇåřśçőŮåđ

```
/* 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)

```
  Py_DECREF(bytes);
  Py_RETURN_NONE;
}
```

åęĆædIJä¡ăåIJÍPythonäÿ■åřÌèřŢè£ŹäžŻåĞ¡æŢřïijÑäÿŃéİćæŸŕè£ŘèąŇæŢĹædIJïijŽ

```
>>> 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
>>>
```

äžŤçżĘèğĆår§çżŞædIJä¡ăäijŽåŔŠçŐřïijÑäÿ■åŘĹæăijå■ŮçňęäÿšèćńçijŰçăĄåĹřäÿĂäÿłPythonå■Ůçňęäÿ
åźűäÿŤå¡ŞåőčèćńåžđäijäçżŹCçŽĎæŮűåĂŹïijÑèćńè¡ňæ■ćäÿžåŠÑäźŃåĽ■åŐ§ågŃCå■ŮçňęäÿšäÿĂæăůçŽĎå

èőÍèőž

æIJňèŁĆåśŢçd'žäžĘåIJÍæĽl'åśŢæÍąåİŮäÿ■åd'ĎçŘĘå■ŮçňęäÿšæŮűäijŽéĚ■åĹřçŽĎäÿĂäÿłæčŸæĽŃåŔÍ
äź§åřśæŸŕèŕťïijŇåIJÍæĽl'åśŢäÿ■çŽĎCå■ŮçňęäÿšåŕŕèČ¡äÿ■äijŽäÿěæäijéĄţå¿łPythonæĽĂæIJ§æIJŻçŽĎUni
åŻăæ■d'ïijŇå¿ĹåŔŕèČ¡äÿĂäžŻäÿ■åŘĹæäijCæŢřæ■őäijăéĂŠåĹřPythonäÿ■åŐżãĂĆ
äÿĂäÿłå¿Ĺåě¡çŽĎä¿Ńå■ŘåřśæŸŕæűĽåŘŁåĹřåžŢåśĆçşżçż§èřČçŤÍæŕŤåęĆæŰĞäżűåŘ■è£ŹæăůçŽĎå■Ůçňęäÿ
ä¿ŃåęĆïijÑåęĆædIJäÿĂäÿłçşżçż§èřČçŤÍè£ŤåŻđçżŹèğčéĞŁåŹÍäÿĂäÿłæ■§åİŔçŽĎå■ŮçňęäÿšïijÑäÿ■èČ¡èćńa

äÿĂèĹňæİěèőšïijÑåŕŕäżěéĂŽè£ĞåĹűåőŽäÿĂäžŻéŤŹèŕŕç■ŰçŢěæŕŤåęĆäÿěæäijãĂĄå£¡çŢěãĂĄæŻ£äżč
äÿ■è£ĞïijÑè£ŹäžŻç■ŰçŢěçŽĎäÿĂäÿłçijžçĆźæŸŕåőČäżňæřÿäźĚæĂğçăť'åİŔäžĘåŐ§ågŃå■ŮçňęäÿšçŽĎåĘĚ
ä¿ŃåęĆïijÑåęĆædIJä¿Ńå■ŘäÿçŽĎäÿ■åŘĹæäijæŢřæ■őä¡£çŤÍè£ŹäžŻç■ŰçŢěäźŃäÿĂèğčçăĄïijÑä¡ăäijŽå¿Ů
ä¿ŃåęĆïijŽ

```
>>> raw = b'Spicy Jalape\xc3\xb1o\xae'
>>> raw.decode('utf-8','ignore')
'Spicy JalapeÃśo'
>>> raw.decode('utf-8','replace')
'Spicy JalapeÃśo?'
>>>
```

surrogateescape éŤŹèŕŕåd'ĎçŘEç■ŰçŢěäijŽåřĘæĽĂæIJL'äÿ■åŔŕèğčçăĄå■ŮèŁĆè¡ňåŇŰäÿžäÿĂä
ä¿ŃåęĆïijŽ

```
>>> raw.decode('utf-8','surrogateescape')
'Spicy JalapeÃśo\udcae'
>>>
```

å■ŢçŃňçŽĎä¡Őä¡■äžčçŘĘå■Ůçňęæŕ̈ŤåęĆ \udcae åIJÍUni-
codeäÿ■æŸŕéİđæşŢçŽĎãĂĆ åŻăæ■d'ïijŇè£ŹäÿłåŮçňęäÿšåřśæŸŕäÿĂäÿłéİđæşŢèąĺçd'žãĂĆ
åőđéŹĚäÿŁïijŇåęĆædIJä¡ăåřĘåőČäijăäÿłäÿĂäÿłæĽğèąŇè¿ŞåĞžçŽĎåĞ¡æŢřïijŇä¡ăäijŽå¿ŮåĹřäÿĂäÿłéŤŹèŕŕ

```
>>> s = raw.decode('utf-8', 'surrogateescape')
>>> print(s)
```

(continues on next 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
>>>
```

çĎűèĂŇïijŇåĚĄèőÿäžčçŘĘè¡ňæ■ćçŽĎåĚşéŤőçĆźåIJÍäžŐäżŐCäijăçżŹPythonåŔĹåŻđäijăçżŹCçŽĎäÿ■å¡Şè£Źäÿłå■Ůçňęäÿšåĕ■æňąä¡£çŤÍ surrogateescape çijŰçăĄæŮűïijŇäžčçŘĘå■ŮçňęäijŽè¡ňæ■ćåŻđåŐ

```
>>> s
'Spicy JalapeÃśo\udcae'
>>> s.encode('utf-8','surrogateescape')
b'Spicy Jalape\xc3\xb1o\xae'
>>>
```

ä¡IJäÿžäÿĂèĹňåĞĘåĹŹïijŇæIJĂåě¡éĄ£åĚ■äžčçŘĘçijŰçăĄâĂŤâĂŤåęĆæđIJä¡ăæ■čçąőçŽĎä¡£çŤÍäžĘçij äÿ■è£ĞïijŇæIJL'æŮűåĂŹçąőåðđäijŽåĞžçŐřä¡ăåźűäÿ■èČ¡æŐğåĹűæŢřæ■őçijŰçăĄåźűäÿŤä¡ăåŔĹäÿ■èČ¡å£¡ç éĆčäźĹåřśåŔŕäžěä¡£çŤÍæIJňèŁĆçŽĎæŁĂæIJŕäžĘãĂĆ

æIJĂåŘŐäÿĂçĆźèęĄæşÍæĎŔçŽĎæŸŕïijŇPythonäÿ■èőÿåd'ŽéÍćåŘŚçşżçż§çŽĎåĞ¡æŢřïijŇçL'źåĹńæŸŕå éČ¡äijŽä¡£çŤÍäžčçŘĘçijŰçăĄãĂĆä¿ŃåęĆïijŇåęĆæđIJä¡ăä¡£çŤÍåČŔ os.listdir() è£ŹæăůçŽĎåĞ¡æŢřïijŇ äijăåĚěäÿĂäÿłåŇĚåŘńäžĘäÿ■åŔŕèğčçăĄæŰĞäžűåŘ■çŽĎçŽőå¡ŢçŽĎèŕÍïijŇåőČäijŽ åŔĆèĂČ5.15çŽĎçŽÿåĚşçńăèŁĆãĂĆ

PEP 383 äÿ■æIJL'æŽť åd'ŽåĚşäžŐæIJňæIJžæŔŘåĹřçŽĎäžěåŔŁåŠŇsurrogateescapeéŤŹèŕŕåd'ĎçŘĘçŽÿåĚşçŽĎä£ąæĄŕãĂĆ

17.17 15.17 äijăéĂŠæŰĞäžűåŘ■çżŹCæĽľåśŢ

éŮőécŸ

ä¡ăéIJĂèęĄåŘŚCåžŞåĞ¡æŢřäijăéĂŠæŰĞäžűåŘ■ïijŇä¡ĘæŸŕéIJĂèęĄçąőä£ÍæŰĞäžűåŘ■æăźæ■őçşżçž§a

èğčåĘşæŰźæąĹ

åĘŹäÿĂäÿłæŐěåŔŮäÿĂäÿłæŰĞäžűåŘ■äÿžåŔĆæŢřçŽĎæĽľåśŢåĞ¡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)

```
  /* Cleanup and return */
  Py_DECREF(bytes)
  Py_RETURN_NONE;
}
```

åęĆædIJä¡ăåůšçżŔæIJL'äžĘäÿĂäÿł PyObject * ïijŇåÿŇæIJŻåřĘåĚűè¡ňæ■ćæĹŘäÿĂäÿłæŰĞäžűåŘ■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

äžěåŔŕçğžæd'■æŰźåijŔæÌěåd'ĎçŘĘæŰĞäžűåŘ■æŸŕäÿĂäÿłå¿ĹæčŸæL'ŃçŽĎéŮőécŸïijŇæIJĂåŘŐäžd'
åęĆædIJä¡ăåIJÍæL'l'åśŢäžčçăĄäÿ■ä¡£çŤÍæIJňèŁĆçŽĎæŁĂæIJŕïijŇæŰĞäžűåŘ■çŽĎåd'ĎçŘĘæŰźåijŘåŠŇåŠ
åŇĚæŃňçijŰçăĄ/çŢŇéÍćå■ŮèŁĆïijŇåd'ĎçŘĘåÍŘå■Ůçňęïij Näžčç ŘĘè¡ňæ■ćåŠŇåĚűäžŰåd'■æÌĆæČĚåĘţãĂĆ

17.18 15.18 äijäéĂŠåůšæL'ŞåijĂçŽĎæŰĞäžűçżŹCæL'l'åśŢ

éŮőécŸ

ä¡ăåIJÍPythonäÿ■æIJL'äÿĂäÿłæL'ŞåijĂçŽĎæŰĞäžűåŕźèšąïijŇä¡ĘæŸŕéIJĂèęĄåřĘåőČäijăçżŹèęĄä¡£çŤÍ

èğčåĘşæŰźæąĹ

èęĄåřĘäÿĂäÿłæŰĞäżűè¡ñæ■ćäÿžäÿĂäÿłæŢť åđŃçŽĎæŰĞäżűæŔŔè£řçňęïijŇä¡£çŤÍ `PyFile_FromFd()` ïijŇåęĆäÿŃïijŽ

```
PyObject *fobj;     /* File object (already obtained somehow) */
int fd = PyObject_AsFileDescriptor(fobj);
if (fd < 0) {
   return NULL;
}
```

çżŞæđIJæŰĞäżűæŔŔè£řçňęæŸŕéĂŽè£ĞèřČçŤÍ `fobj` äÿ■çŽĎ `fileno()` æŰźæşŢèŐůå¿ŮçŽĎãĂĆ åŻăæ■ď ïijŇäżżä¡Ţäžèè£Źçğ■æŰźåijŔæŽť éIJšçżŹäÿĂäÿłæŔŔè£řåŹÍçŽĎåŕźèšąéČ äÿĂæŮęä¡ăæIJĽäžĘè£ŹäÿłæŔŔè£řåŹÍïijŇåőČåřśèČ¡èćńäijăéĂŠçżŹåď Žäÿłä¡ŐçžğçŽĎåŔŕåď ĎçŘĘæŰĞäżű

åęĆæđIJä¡ăéIJĂèęĄè¡ñæ■ćäÿĂäÿłæŢť åđŃæŰĞäżűæŔŔè£řçňęäÿžäÿĂäÿłPythonåŕźèšąïijŇéĂĆçŤÍäÿŃé `PyFile_FromFd()` :

```
int fd;     /* Existing file descriptor (already open) */
PyObject *fobj = PyFile_FromFd(fd, "filename","r",-1,NULL,NULL,NULL,
↪1);
```

`PyFile_FromFd()` çŽĎåŔĆæŢřåŕźäžŤåĘĚç¡őçŽĎ `open()` åĞ¡æŢřãĂĆ NUL-LèąÍçď žçijŰçăĄãĂĄéŤŹèŕŕåŠŇæ■ćèąŇåŔĆæŢřä¡£çŤÍéżŸèőď åĂijãĂĆ

èőÍèőž

åęĆæđIJåřĘPythonäÿ■çŽĎæŰĞäżűåŕźèšąäijăçżŹCïijŇæIJĽäÿĂäžŽæşÍæĎŔäžŃéąźãĂĆ éęŰåĚĹïijŇPythonéĂŽè£Ğ `io` æÍąåİŮæĽğèąŇèĞłåůšçŽĎI/OçijŞåĘšãĂĆ åIJÍäijăéĂŠäżżä¡Ţçśżåđ ŃçŽĎæŰĞäżűæŔŔè£řçňęçżŹCäźŃåĽ■ïijŇä¡ăéČ¡èęĄéęŰåĚĹåIJÍçŽÿåžŤæŰĞäżűåŕź äÿ■çĎűçŽĎèŕİïijŇä¡ääijŽæĽŞäźśæŰĞäżűçşżçż§äÿŁéÍćçŽĎæŢřæ■őãĂĆ

åĚűæňąïijŇä¡ăéIJĂèęĄçĽźåĹńæşÍæĎŔæŰĞäżűçŽĎå¡Šåśđè ĂĚäžěåŔŁåĚşéŮ■æŰĞäżűçŽĎèĄŇèť čãĂĆ åęĆæđIJäÿĂäÿłæŰĞäżűæŔŔè£řçňęèćńäijăçżŹCïijŇä¡ĘæŸŕåIJÍPythonäÿ■è£ŸåIJÍèćńä¡£çŤÍçİĂïijŇä¡ăéIJĂèę çśżäijijçŽĎïijŇåęĆæđIJäÿĂäÿłæŰĞäżűæŔŔè£řçňęèćńè¡ñæ■ćäÿžäÿĂäÿłPythonæŰĞäżűåŕźèšąïijŇä¡ăéIJĂèęĄ `PyFile_FromFd()` çŽĎæIJĂåŘŐäÿĂäÿłåŔĆæŢřèćńèő¿ç¡őæĹŘ1ïijŇçŤÍłæİěæŇĞåĞžPythonåžŤèŕěåĘşéŮ

åęĆæđIJä¡ăéIJĂèęĄäżŐCæăĞåĞĘI/OåžŞäÿ■ä¡£çŤÍåęĆãĂĂ`fdopen()` åĞ¡æŢřæİěåĹŻåžžäÿ■åŘŇçśżåđŃçŽĎæŰĞäżűåŕźèšąæŕŤåęĆ `FILE *` åŕźèšąïijŇ ä¡ăéIJĂèęĄçĽźåĹńåřŔå£ČäžĘãĂĆè£ŹæăůåĄŽäijŽåIJÍI/OåăĘæăĹäÿ■äžğçŤ§äÿđ äÿłåőŇåĚÍäÿ■åŘŇçŽĎI/Oç ïijĹäÿĂäÿłæŸŕæİěèĞłPythonçŽĎ `io` æÍąåİŮïijŇåŘęäÿĂäÿłæİěèĞłCçŽĎ `stdio` ïijĽãĂĆ åČŔCäÿ■çŽĎ `fclose()` äijŽåĚşéŮ■PythonèęĄ¡£çŤÍçŽĎæŰĞäżűãĂĆ åęĆæđIJèőľä¡ăéĂĽçŽĎèŕİïijŇä¡ääžŤèŕěäijŽéĂĽæŇľåŐžæđĎåžžäÿĂäÿłæĽľåśŢäžčçĄæİěåđ ĎçŘĘåžŢåśĆ èĂŇäÿ■æŸŕä¡£çŤÍæİěèĞł<stdio.h>çŽĎéŃŸåśĆæŁ¡ésąåŁ§èČ¡ãĂĆ

17.19 15.19 äżŐCèŕ■èÍÄäÿ■èŕżåŔŰçśżæŰĞäżűåŕźèśą

éŮőécŸ

ä¡ăèęĄåĘŹCæĽ'ľåśŢæİěèŕżåŔŰæİěèĞłäżżä¡ŢPythonçśżæŰĞäżűåŕźèśąäÿ■çŽĎæŢřæ■őïijĹæŕŤåęĆæŹőé

èğčåĘşæŰźæąĹ

èęĄèŕżåŔŰäÿĂäÿłçśżæŰĞäżűåŕźèśąçŽĎæŢřæ■őïijŇä¡ăéIJĂèęĄéĞ■åd'■èřČçŤÍ
`read()` æŰźæşŢïijŇçĎűåŘŐæ■čçąőçŽĎèğčçăĄèŐůå¿ŮçŽĎæŢřæ■őãĂĆ

äÿŃéİćæŸŕäÿĂäÿłCæĽ'ľåśŢåĞ¡æŢřä¿Ńå■ŘïijŇäżĚäżĚåŘłæŸŕèŕżåŔŰäÿĂäÿłçśżæŰĞäżűåŕźèśąäÿ■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)

```
    }

    /* 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£Źäÿłäžčçă Ąïij ŇåĚĹædĎéĂääÿĂäÿłçśżæŰĞäžűåŕźèśąæŕŤåęĆäÿĂäÿłStringIOåőđä¿ŃïijŇç

```
>>> import io
>>> f = io.StringIO('Hello\nWorld\n')
>>> import sample
>>> sample.consume_file(f)
Hello
World
>>>
```

èőÍèőž

åŠŇæŹőéĂŽçşżçż§æŰĞäžűäÿ■åŘŇçŽĎæŸŕïijŇäÿĂäÿłçśżæŰĞäžűåŕźèśąåźűäÿ■éIJĂèęĄä¡£çŤÍä¡ŐçžğĄ
åŻăæ■ď'ïijŇä¡ääÿ■èČ¡ä¡£çŤÍæŹőéĂŽçŽĎCåžŞåĞ¡æŢřæİèèő£éŮőåőČãĂĆ
ä¡ăéIJĂèęĄä¡£çŤÍPythonçŽĎC APIæİèåČŔæŹőéĂŽæŰĞäžűçśżäijijçŽĎéĆčæăůæŞ■ä¡IJçśżæŰĞäžűåŕźèśąãĂ

åIJÍæĹŚäžňçŽĎèğčåĘşæŰźæąĹäÿ■ïijŇ`read()` æŰźæşŢäžŐèćńäijăéĂŠçŽĎåŕźèśąäÿ■æŔŘåŔŰåĞžæİě
äÿĂäÿłåŔĆæŢřåĹŮèąĺèćńæđĎåžžçĎűåŘŐäÿ■æŰ■çŽĎèćńäijăçżŹ `PyObject_Call()`
æİèěřČçŤÍè£ŹäÿłæŰźæşŢãĂĆ èęĄæčĂæ§ěæŰĞäžűæIJńář¿ïijĹEOFïijL'ïijŇä¡£çŤÍäžĘ
`PySequence_Length()` æİěæ§ěçIJŃæŸŕåŘęè£ŤåŻđåŕźèśąéŢ£åžęäÿž0.

åŕźäžŐæL’ĂæIJL’çŽĎI/OæŞ■ä¡IJïijŇä¡ăéIJĂèęĄåĚşæşĺåžŢåśĆçŽĎçijŰçăĄæăijåijŔïijŇè£ŸæIJL’å■ŮèŁ
æIJňèŁĆæijŤçd’žäžĘåęĆä¡Ţäžěæ ŰĞæIJňæĺąåijŔèŕžåŔŰäÿĂäÿłæŰĞäžűåźűåřĘçżŞæđIJæŰĞæIJňèğčçăĄäÿž
åęĆæđIJä¡ăæČşäžěäžŇè£ŽåĹűæĺąåijŔèŕžåŔŰæŰĞäžűïijŇåŔłéIJĂèęĄä£őæŤźäÿĂçĆźçĆźå■şåŔŕïijŇä¿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¡Şåď’ĎçŘĘ `PyObject *` åŔŸéĞŔçŽĎæŮűåĂŹïijŇéIJĂèęĄæşĺæĎŔçőąçŘĘåijŢçŤĺèőąæŢřäžěåŘŁåIJĺäÿ■
åŕź `Py_DECREF()` çŽĎèřČçŤĺåřśæŸŕæİěåĄŽè£ŹäÿłçŽĎãĂĆ

æIJňèŁĆäžčçăĄäžěäÿĂçğ■éĂŽçŤĺæŰźåijŔçijŰåĘŹïijŇåŽäæ■ď’äžŰäź§èČ¡éĂĆçŤĺäžŐåĚűäžŰçŽĎæŰĞ
ä¿ŃåęĆïijŇèęĄåĘŹæŢřæ■őïijŇåŘłéIJĂèęĄèŐůåŔŰçśżæŰĞäžűåŕźèśąçŽĎ `write()`
æŰźæşŢïijŇåřĘæŢřæ■őè¡ňæ■ćäÿžåŘĹéĂĆçŽĎPythonåŕźèśą ïijĹå■ŮèŁĆæĹŰŰni-
codeïijL’ïijŇçĎűåŘŐèřČçŤĺèŕěæŰźæşŢåřĘè¿ŞåĚěåĘŹåĚěåĹřæŰĞäžűãĂĆ

æIJĂåŘŐïijŇår¡çőąçśżæŰĞäžűåŕźèśąéĂŽåÿÿè£ŸæŔŘä¿ŻåĚűäžŰæŰźæşŢïijĹærŤåęĆreadline(),
read_info()ïijL’ïijŇ æĹŚäžňæIJĂåě¡åŔłä¡£çŤĺå§žæIJňçŽĎ `read()` åŠŇ `write()`
æŰźæşŢãĂĆ åIJĺåĘŹCæL’ľåśŢçŽĎæŮűåĂŹïijŇèČ¡çőĂå■Ţåřśår¡éĞŔçőĂå■ŢãĂĆ

17.20 15.20 åď’ĎçŘĘCèŕ■èÍĂäÿ■çŽĎåŔŕè£■äžčåŕźèśą

éŮőécŸ

ä¡ăæČşåĘŹCæL’ľåśŢäžčçăĄåď’ĎçŘĘæİěèĞłäżżä¡ŢåŔŕè£■äžčåŕźèśąåęĆåĹŮèąĺãĂĄåĚČçżĎãĂĄæŰĞäž

èğčåĘşæŰźæąĹ

äÿŃéİćæŸŕäÿĂäÿłCæL’ľåśŢåĞ¡æŢřä¿Ńå■ŘïijŇæijŤçď’žäžĘæĂŐæăůåď’ĎçŘĘåŔŕè£■äžčåŕźèśąäÿ■çŽĎå

```
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("");
}
```

èőĺèőž

æĲňèŁĆäÿ■çŽĎäżčçăĄåŠŇPythonäÿ■åŕźåžŤäżčçăĄçśżäijijãĂĆ `PyObject_GetIter()` çŽĎèřČçŤĺåŠŇèřČçŤĺ `iter()` äÿĂæăůåŘŕèŐůå¿ŮäÿĂäÿłè£■äżčåŹĺãĂĆ `PyIter_Next()` åĞ¡æŢřèřČçŤĺ `next` æŰźæşŢè£ŤåŻđäÿŃäÿĂäÿłåĚČçť äæĹŰNULL(åęĆæđĲæšąæĲĽåĚČçť äážĘ)ãĂĆ èęĄæşĺæĎŔæ■čçąőçŽĎåĘĚå■Ÿçőąçŕ ĘâĂŤâĂŤ `Py_DECREF()` éĲĂèęĄåŔŇæŮűåĲĺäžğçŤ§çŽĎåĚČçť äåŠŇè£■äżčåŹĺåŕźèśąæĲňèžńäÿŁåŘŇæŮűèćńèřČçŤĺïijŇ äžěéĄ£åĚ■åĞžçŐřåĘĚå■ŸæşĎéĲšãĂĆ

17.21 15.21 èŕŁæŰ■åĹĘæőţéŤŹèŕŕ

éŮőéćŸ

èğčéĞŁåŹĺåŻăäÿžæ§ŘäÿłåĹĘæőţéŤŹèŕŕãĂĄæĂżçž£éŤŹèŕŕãĂĄèő£éŮőèűŁçŢŇæĹŰåĚűäżŰèĞť åŠ¡éŤ
ä¡ăæČşèŐůå¿ŮPythonåăĘæăĹä£ąæĄŕïijŇäżŐèĂŇæĽ¿åĞžåĲĺåŔŠçŤ§éŤŹèŕŕçŽĎæŮűåĂŹä¡ăçŽĎçĺŃåžŔè£

èğčåĘşæŰźæąĹ

`faulthandler` æĺąåİŮèČ¡èćńçŤĺæİĕåÿőä¡ăèğčåĘşèĿŹäÿłéŮőéćŸãĂĆ åĲĺä¡ăçŽĎçĺŃåžŔäÿ■åijŢåĚěäÿŃåĹŮäżčçăĄïijŽ

```
import faulthandler
faulthandler.enable()
```

åŔęåd'Ůè£ŸåŔŕäžěåČŔäÿŃéİćè£ŹæăůäĮ£çŢĺ -Xfaulthandler æİěè£ŘèąŇPythonïijŽ

```
bash % python3 -Xfaulthandler program.py
```

æIJĂåŘŐïijŇäĮäåŔŕäžěèő¿ç¡ő PYTHONFAULTHANDLER çŐŕåćČåŔŸéĞŔãĂĆ åijĂåŘŕ-faulthandleråŘŐïijŇåIJÍCæL'l'åśŢäÿ■çŽĎèĞt'åŠ¡éŤŹèŕŕäijŽåŕijèĞt'äÿĂäÿłPythonéŤŹèŕŕåäĘæăĹèćńæL'Şå■ř

```
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
```

ař¡çőąè£Źäÿłåźűäÿ■èČ¡åŠŁèŕL'ä¡ăCäžčçăĄäÿ■åŞłéĞŇåĞžéŤŹäžĘïijŇä¡ĘæŸŕèĞşåřŠèČ¡åŠŁèŕL'ä¡ăPyth

èőÍèőž

faulthandleräijŽåIJÍPythonäžčçăĄæL'ğèąŇåĞžéŤŹçŽĎæŮűåĂŹåŘŠä¡ăåśŢçd'žèů§èÿłä£ąæĄŕãĂĆ èĞşåřŠïijŇåőČäijŽåŠŁèŕL'ä¡ăåĞžéŤŹæŮűèćńèřČçŤÍçŽĎæIJĂéąűçžğæL'l'åśŢåĞ¡æŢřæŸŕåŞłäÿłãĂĆ åIJÍpdbåŠŇåĚűäžŰPythonèřČèŕŢåŹÍçŽĎåÿőåŁl'äÿŃïijŇä¡ăåřšèČ¡è£¡æăźæžŕæžŘæL'¿åĹréŤŹèŕŕæL'ĂåIJÍçŽĎ

faulthandleräÿ■äijŽåŠŁèŕL'ä¡ăäżżä¡ŢCèŕ■èÍĂäÿ■çŽĎéŤŹèŕŕä£ąæĄŕãĂĆ åŻăæ■d'ïijŇä¡ăéIJĂèęĄä¡£çŤÍäijăçż§çŽĎCèřČèŕŢåŹÍïijŇæŕŤåęĆgdbãĂĆ äÿ■è£ĞïijŇåIJÍfaulthandlerè£¡èÿłä£ąæĄŕåŘŕäžěèől'ä¡ăåŐżåĹd'æŰ■äžŐåŞłéĞŇçİĂæL'ŃãĂĆ è£ŸèęĄæşÍæĎŔçŽĎæŸŕåIJÍCäÿ■æ§ŘäžŽçśžåđŇçŽĎéŤŹèŕŕåŘŕèČ¡äÿ■åd'łåőźæŸŞæĄćåd'■ãĂĆ ä¿ŃåęĆïijŇåęĆæđIJäÿĂäÿłCæL'l'åśŢäÿćåijČäžĘçÍŃåžŘåäĘæăĹä£ąæĄŕïijŇåőČäijŽèől'faulthandleräÿ■åŘŕçŤ éĆčäźĹä¡ääź§å¿Ůäÿ■åĹŕäżżä¡Ţè¿ŞåĞžïijĹéŹd'äžĘçÍŃåžŘåěŤæžČåd'ŰïijL'ãĂĆ

18 éŹĎå¡ŢA

18.1 åIJÍçž£èţĎæžŘ

http://docs.python.org

åęĆæđIJä¡ăéIJĂèęĄæůśåĚěäžĘèğčæŐćçl'űèŕ■èÍĂåŠŇæÍąåİŮçŽĎçżĘèŁĆïijŇéĆčäźĹäÿ■å£Èèŕt'ïijŇPyth 3 çŽĎæŰĞæąčèĂŇäÿ■æŸŕäžěåL'■çŽĎèĂĄçL'ĹæIJň

http://www.python.org/dev/peps

åęĆæđIJä¡ääŘŠçŘĘèğčäÿžpythonèŕ■èÍĂæůżåŁăæŰřçL'źæĂğçŽĎåŁÍæIJžäžěåŔŁåőđçŐřçŽĎçżĘèŁĆïijŇ Enhancement ProposalsâĂŤ-PythonåijĂåŘŠçijŰçăĄèğĎèŇČïijL'çżÍåŕźæŸŕéİđåÿÿåőÍèt'tçŽĎèţĎæžŘãĂĆåŕd'

http://pyvideo.org

è£ŹéĞŇæIJL’æİěèĞłæIJĂè£ŚçŽĎPyConåď ğäijŽãĂĄçŤĺæĹůçżĎèğĄéİćäijŽç■L’çŽĎåď ğéĞŔèğĘéćŚæi
3äÿ■æůżåŁăçŽĎçŽĎæŰřçL’źæĂğãĂĆ

http://code.activestate.com/recipes/langs/python

éŢ£æIJ§äžěæİěïijŇActiveStateçŽĎPythonçL’ĹåİŮåůšçżŔæĹŘäÿžäÿĂäÿłæL’¿åĹřæŢřäžěå■Čè őąçŽĎéŠĹ

http://stackoverflow.com/questions/tagged/python

Stack Overflow çŽőåL’■æIJL’èűĚè£Ğ175,000äÿłéŮőéćŸèćńæăĞèőřäÿžPythonçŻÿåĚşïijĹèĂŇåĚűäÿ■åď
3çŽĎïijL’ãĂĆår¡çőąéŮőéćŸåŠŇåŻđç■ŤçŽĎèť´íéĞŔäÿ■åŘŇïijŇä¡ĘæŸřäż■çĎűèČ¡åŔŚçŐřå¿Ĺåď Žåě¡äijŸçğ

18.2 Pythonå■ęäźăäźęçś■

äÿŃéİćè£ŹäžŻäźęçś■æŔŘä¿ŻäžĘåŕźPythonçijŰçÍŃçŽĎåĚěéŮÍäżŃçż■ïijŇäÿŤéĞ■çĆźæŤ¿åIJÍäžĘPytho
3äÿŁãĂĆ

- *Learning Python*ïijŇçňňåŻŻçL’Ĺ ïijŇä¡IJèĂĚ Mark LutzïijŇ OâĂŹReilly & Associates åĞžçL’Ĺ (2009)ãĂĆ
- *The Quick Python Book*ïijŇä¡IJèĂĚ Vernon CederïijŇ 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■æŔŘä¿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:

```
|    åŕźåĚĺéČĺç£żèŕŚè£ŻèąŇæăąåŕźäÿĂæňą
```

2016/01/01 - 2016/01/10:

```
|    åŕźåd’ŰåĚňåijĂåŔŚåÿČåőŇæŢt’çL’Ĺ1.
↪0ïijŇåŇĚæŃňè¡ňæ■ćåŘŐçŽĎPDFæŰĞäżű
```