

Python Cookbook

Release 3.0.0

çĖŁèĈi

Dec 20, 2018

Contents

1	Copyright	1
2	ĀĹēĀ	1
2.1	éąçŻöäyzeą	1
2.2	ėŀŚēĀĖçŽĎėŀ	1
2.3	äĭĬēĀĖçŽĎėŀ	2
2.4	èĤŽæĬĥăzėéĀĈăĤĤĤĤ	2
2.5	èĤŽæĬĥăzėäyēĀĈăĤĤĤĤ	3
2.6	āĬĬçŻĤçđ' žăĭĤăzčçăĀ	3
2.7	äĭĤçĤĤçđ' žăĭĤăzčçăĀ	3
2.8	èĀĤçşzæĤĤăžň	3
2.9	èĠŀ'ėŕć	4
3	çňăyĀçňăĭjŽæĤŕæőçşæđĎăŤŤçőŬæşŤ	4
3.1	1.1 èğçăŎŇăžŔăĤŬėŀŇăĀĭjçzŽăđ' ŽăyĤăŔŸéĠŔ	5
3.2	1.2 èğçăŎŇăŔŕēĤăžčăŕžėşăęŀŇăĀĭjçzŽăđ' ŽăyĤăŔŸéĠŔ	6
3.3	1.3 äĤĤçŤŽæĬĤăŔŎ Ň äyĤăĖĈçŀ' ä	9
3.4	1.4 æşēæĤĭæĬĤăđ' ġæĤŬæĬĤăŔçŽĎ Ň äyĤăĖĈçŀ' ä	11
3.5	1.5 āōđçŎŕăyĀăyĤăĭjŸăĖĤçžġēŸşăĤŬ	12
3.6	1.6 āŬăĖŸăyēĤŽĎēŤŏæŸăăŕĎăđ' ŽăyĤăĀĭj	15
3.7	1.7 āŬăĖŸăēŎŖăžŔ	16
3.8	1.8 āŬăĖŸçŽĎēĤŔçőŬ	17
3.9	1.9 æşēæĤĭăyđ' āŬăĖŸçŽĎçŸăŕŇçĈz	19
3.10	1.10 āĤăēŽđ' äžŔăĤŬçŸăŕŇăĖĈçŀ' āăžŭăĤĬăŇăĖăžăžŔ	20
3.11	1.11 āŖăŕăĤăĤĤĤĤĤ	22
3.12	1.12 äžŔăĤŬăyăăĠçŎŕăŇăæŤŕæĬĤăđ' ŽçŽĎăĖĈçŀ' ä	23
3.13	1.13 éĀŽēĤĤæşŔăyĤăĖşēŤŏăŬăŎŖăžŔăyĀăyĤăŬăĖŸăĤŬēăĤ	25
3.14	1.14 æŎŖăžŔăyăæŤŕæŇăăŎŖçŤşăŕŤēçĈçŽĎăŕžėşă	27
3.15	1.15 éĀŽēĤĤæşŔăyĤăŬăŏŭăŕĖēŏŕăĤăĤĖçzĎ	28
3.16	1.16 èĤĤæzd' äžŔăĤŬăĖĈçŀ' ä	30
3.17	1.17 äžŎăŬăĖŸăyăæŔŕăŕŬăŕēŽĖ	32
3.18	1.18 æŸăăŕĎăŕăçğŕăĤŕăžŔăĤŬăĖĈçŀ' ä	33
3.19	1.19 èĭŇăçăžŭăŕŇăŬŭēŏăçŏŬăŤŕæő	36
3.20	1.20 āŖĤăžŭăđ' ŽăyĤăŬăĖŸăĤŬăŸăăŕĎ	37

4.15	2.15	ā■Ūcņēäyšäy■æRŠāĒĒāRŸéGR	63
4.16	2.16	äzēæNĠāōŽāLŪāōj;æaijāijRāNŪā■Ūcņēäyš	65
4.17	2.17	āIJā■Ūcņēäyšäy■ād'DçREhtmlāŠNxml	67
4.18	2.18	ā■Ūcņēäyšāzd'çL'NēgčædR	68
4.19	2.19	āōđçŌrāyÄäyļçōĀā■TçŽDēĀŠā;ŠäyNéZ■āLEædRāZĪ	71
4.20	2.20	ā■ŪēŁĆā■ŪcņēäyšäyŁçŽDā■ŪcņēäyšæŠ■ä;IJ	79
5		çññäyL'çñāijŽæTṛā■ŪæŪēæIJšāŠNæŪūéŪt'	81
5.1	3.1	æTṛā■ŪçŽDāZZēL■āzTāĒē	81
5.2	3.2	æL'gèaŊçš;çqōçŽDætōçCzæTṛēfRçōŪ	83
5.3	3.3	æTṛā■ŪçŽDæaijāijRāNŪē;ŠāGž	85
5.4	3.4	āzNāĒnā■AāĒ■ēfZāLūæTt'æTṛ	87
5.5	3.5	ā■ŪēŁĆāLrād'gæTt'æTṛçŽDæL'ŠāNĒäyŌēgčāNĒ	88
5.6	3.6	ād'■æTṛçŽDæTṛā■ēēfRçōŪ	90
5.7	3.7	æŪāçl'ūād'gäyŌNaN	92
5.8	3.8	āLEæTṛēfRçōŪ	94
5.9	3.9	ād'gādNæTṛçzDēfRçōŪ	95
5.10	3.10	çšl'ēYtāyŌçžfæĀgāzçæTṛēfRçōŪ	98
5.11	3.11	éŽRæIJzéĀL'æNl'	100
5.12	3.12	āšzæIJñçŽDæŪēæIJšäyŌæŪūéŪt'èjñæ■c	102
5.13	3.13	ēōaçōŪæIJĀāRŌäyÄäyļāŚlāzTçŽDæŪēæIJš	104
5.14	3.14	ēōaçōŪā;ŠāL■æIJLāz;çŽDæŪēæIJšēNČāZt'	106
5.15	3.15	ā■Ūcņēäyšējñæ■cäyžæŪēæIJš	108
5.16	3.16	çzŠāRLæŪūāNžçŽDæŪēæIJšæŠ■ä;IJ	109
6		çññāZZçñāijŽēf■āzçāZlāyŌçTšæL'RāZĪ	111
6.1	4.1	æL'NāLlÉA■āŌĒēf■āzçāZĪ	111
6.2	4.2	āzççREēf■āzç	112
6.3	4.3	ä;fçTlçTšæL'RāZĪlāLZāzzæŪrçŽDēf■āzçælaqaijR	113
6.4	4.4	āōđçŌrēf■āzçāZlā■Rēōō	115
6.5	4.5	āR■āRŠēf■āzç	117
6.6	4.6	āyææIJL'ād'ŪēČlçLūæĀAçŽDçTšæL'RāZĪlāG;æTṛ	119
6.7	4.7	ēf■āzçāZlāLĠçL'Ġ	120
6.8	4.8	ēūšēfGāRrēf■āzçāržēsāçŽDaijĀāgNéČlāLE	121
6.9	4.9	æŌŠāLŪçzDāRLçŽDēf■āzç	123
6.10	4.10	āzRāLŪāyŁçt'cāijTāĀijēf■āzç	125
6.11	4.11	āRNæŪūēf■āzçād'ŽäyļāzRāLŪ	127
6.12	4.12	äy■āRNéZEāRLāyLāĒČçt'āçŽDēf■āzç	129
6.13	4.13	āLZāzzæTṛæ■ōād'DçREçōāéAŠ	130
6.14	4.14	āsTāijĀātNāēŪçŽDāzRāLŪ	133
6.15	4.15	ēāžāzRēf■āzçāRLāzūāRŌçŽDæŌŠāzRēf■āzçāržēsā	135
6.16	4.16	ēf■āzçāZlāzçæŽfwhileæŪāēŽRā;ļçŌr	136
7		çññāžTçñāijŽæŪGāzūäyŌIO	137
7.1	5.1	ērzaEZæŪGæIJñæTṛæ■ō	137
7.2	5.2	æL'Šā■rē;ŠāGžēGšæŪGāzūäy■	140
7.3	5.3	ä;fçTlāĒūāzŪāLEēŽTçņæāLŪēāNçzLæ■cçņæāL'Šā■r	140
7.4	5.4	ērzaEZā■ŪēŁĆæTṛæ■ō	141
7.5	5.5	æŪGāzūäy■ā■YāIJlāL■ēČ;āEŽāĒē	143

7.6	5.6	āŭņēäyšçŽDI/OæŠ■ä;IJ	144
7.7	5.7	ērzaEŽāŌŅcijl' æŪGāzū	145
7.8	5.8	āŽžāōZād' gārRēōrā;TçŽDæŪGāzūēf■āzč	147
7.9	5.9	ērzaRŪāzŅēfZāLūæTṛæ■ōāLrāRrāRŸcijŠāEšāŅzāy■	147
7.10	5.10	āEĒā■ŸæŸāārDçŽDāzŅēfZāLūæŪGāzū	149
7.11	5.11	æŪGāzūēūrā;DāR■çŽDæŠ■ä;IJ	151
7.12	5.12	ætŅērTṛæŪGāzūæŸrāRēā■ŸāIJl	152
7.13	5.13	ēŌūāRŪæŪGāzūād' zāy■çŽDæŪGāzūāLŪēāl	154
7.14	5.14	āf;çTēæŪGāzūāR■cijŪčāA	155
7.15	5.15	æL'Sā■rāy■āRŁæšTçŽDæŪGāzūāR■	157
7.16	5.16	ācđāŁāæLŪæTzāRŸāūsæL'SāijĀæŪGāzūçŽDcijŪčāA	159
7.17	5.17	ārEā■ŪēŁCāEŽāĒēæŪGæIJŅæŪGāzū	161
7.18	5.18	ārEæŪGāzūæRRēfçņēāŅĒēčĒæLRæŪGāzūāržēšā	162
7.19	5.19	āLZāžzāy' æŪūæŪGāzūāSŅæŪGāzūād' ž	164
7.20	5.20	äyŌäyšēāŅçnrāRççŽDæTṛæ■ōēĀŽāfā	166
7.21	5.21	āžRāLŪāŅŪPythonāržēšā	167
8		çññāĒē■çñāijZæTṛæ■óçijŪčāAāSŅād'DçRE	170
8.1	6.1	ērzaEŽCSVæTṛæ■ō	170
8.2	6.2	ērzaEŽJSONæTṛæ■ō	174
8.3	6.3	ēgčæđRçōĀā■TçŽDXMLæTṛæ■ō	178
8.4	6.4	ācđēGRāijRēgčæđRād' gādNXMLæŪGāzū	181
8.5	6.5	ārEā■ŪāĒyē;Ņæ■cāyžXML	185
8.6	6.6	ēgčæđRāSŅāŁōæTzXML	187
8.7	6.7	āLr'çTlāS;āR■çl'žēŪr'ēgčæđRXMLæŪGæāç	189
8.8	6.8	äyŌāĒšçšzādŅæTṛæ■ōāžSçŽDāzđ' āžŠ	191
8.9	6.9	çijŪčāAāSŅēgčçāAā■AāĒēfZāLūæTṛ	193
8.10	6.10	çijŪčāAēgčçāABase64æTṛæ■ō	194
8.11	6.11	ērzaEŽāžŅēfZāLūæTṛçžDæTṛæ■ō	195
8.12	6.12	ērzaRŪā;ŅāēŪāSŅāRrāRŸēTfāžŅēfZāLūæTṛæ■ō	199
8.13	6.13	æTṛæ■ōçŽDçt'fāŁāäyŌçžšēōæŠ■ä;IJ	208
9		çññäyČçñāijZāG;æTṛ	211
9.1	7.1	ārRæŌēāRŪāzæđRæTṛēGRāRCæTṛçŽDāG;æTṛ	211
9.2	7.2	ārLæŌēāRŪāĒšēTōā■ŪāRCæTṛçŽDāG;æTṛ	212
9.3	7.3	çžZāG;æTṛāRCæTṛācđāŁāāĒČāfāæAr	213
9.4	7.4	ēfTāžđād' ŽāyłāĀijçŽDāG;æTṛ	214
9.5	7.5	āōŽāzL'æIJL'ēžŸēōd' āRCæTṛçŽDāG;æTṛ	215
9.6	7.6	āōŽāzL'āŅfāR■æLŪāĒēĒāTāG;æTṛ	218
9.7	7.7	āŅfāR■āG;æTṛæ■TēŌūāRŸēGRāĀij	219
9.8	7.8	āGRārSārRērČçTlāržēšāçŽDāRCæTṛāyłæTṛ	220
9.9	7.9	ārEā■TṛæŪzæšTçŽDçszē;Ņæ■cāyžāG;æTṛ	223
9.10	7.10	āyēēčlād' ŪçLūæĀAāfāæArçžDāZđērČāG;æTṛ	225
9.11	7.11	āĒēĒāTāZđērČāG;æTṛ	227
9.12	7.12	ēōfēŪōēŪ■āŅĒäy■āōŽāzL'çŽDāRŸēGR	230
10		çññāĒē■çñāijZçszāyŌāržēšā	233
10.1	8.1	æTzāRŸāržēšāçŽDā■ŪņēäyšæŸçđ'ž	233
10.2	8.2	ēGłāōZāzL'ā■ŪņēäyšçŽDæāijāijRāŅŪ	234

10.3	8.3	èol' áržèsæŕŕæŇAäyŁäyŇæŮĠçõaçŔĒāŕĒèõõ	236
10.4	8.4	ālŽāzzād' gēĠŕáržèsæŭŮēŁĆĲIAāĒĒāŕæŮzæsŦ	238
10.5	8.5	āĲĲčšzäyŕAēcĒāsđæĀgāŕ	239
10.6	8.6	ālŽāzzāŕŕçõaçŔĒçŽĐāsđæĀg	240
10.7	8.7	ērČĲŦĲŁūčšzæŮzæsŦ	245
10.8	8.8	āŕŕçšzäyŕæL'ŕāsŦproperty	249
10.9	8.9	ālŽāzzæŮŕçŽĐčšzæŁŮāōđäĲŇāsđæĀg	253
10.108.10		äĲčŦĲāzŭēŕšēõaçõŮāsđæĀg	256
10.118.11		çõĀāŇŮæŦŕæŕçzšæđĐçŽĐāĲiāgŇāŇŮ	259
10.128.12		āōŽāzL' æŌēāŕčæŁŮēĀĒæĲčēsāšžčšz	262
10.138.13		āōđçŌŕæŦŕæŕæĲāđŇçŽĐčšzādŇçžæĲš	265
10.148.14		āōđçŌŕēĠāōŽāzL' āōžāZĲ	270
10.158.15		āsđæĀgçŽĐāzčçŔĒēōēŮō	273
10.168.16		āĲĲčšzäyŕāōŽāzL' āđ' ŽāyĲæđĐĒĀāāZĲ	278
10.178.17		ālŽāzzäyŕērČĲŦĲinitæŮzæsŦçŽĐāōđäĲŇ	279
10.188.18		ālŦçŦĲMixinsæL'ŕāsŦçšzāŁšēČĲ	280
10.198.19		āōđçŌŕçŁūæĀĀŕžèsæŁŮēĀĒçŁūæĀĀæĲž	283
10.208.20		ēĀŽēŕĠāŕŮçņäyšērČĲŦŕŕžèsæŮzæsŦ	286
10.218.21		āōđçŌŕēōēŮōēĀĒāĲāijŕ	288
10.228.22		äyŕŦĲēĀšāŕšāōđçŌŕēōēŮōēĀĒāĲāijŕ	291
10.238.23		āĲčŌŕāijŦçŦĲæŦŕæŕçzšæđĐçŽĐāĒĒāŕçõaçŔĒ	295
10.248.24		èol'čšzæŦŕæŇAæŕŦēĲčæšŕäĲĲ	299
10.258.25		ālŽāzzçĲijšāŕŮāōđäĲŇ	301

11 çññāzĲčñāijŽāĒČĲijŮčĲŇ 305

11.1	9.1	āĲĲāĠçĲæŦŕäyŁæŭzāŁāāŇĒēčĒāZĲ	305
11.2	9.2	ālŽāzzèčĒēēŕāZĲæŮŭāĲçŦŽāĠçĲŦŕāĒČĲæĲæĲŕ	307
11.3	9.3	ēgčēZđ' äyĀäyĲēčĒēēŕāZĲ	308
11.4	9.4	āōŽāzL' äyĀäyĲäyēāŕČæŦŕçŽĐēčĒēēŕāZĲ	310
11.5	9.5	ārŕēĠāōŽāzL' āsđæĀgçŽĐēčĒēēŕāZĲ	312
11.6	9.6	äyēāŕŕēĀL' āŕČæŦŕçŽĐēčĒēēŕāZĲ	315
11.7	9.7	ālŦçŦĲēčĒēēŕāZĲāijzāŁūāĠçĲŦŕäyŁçŽĐčšzādŇæčĀæšē	316
11.8	9.8	ārĒēčĒēēŕāZĲāōŽāzL' äyžçšççŽĐäyĀēČĲāĲĒ	320
11.9	9.9	ārĒēčĒēēŕāZĲāōŽāzL' äyžçšz	322
11.109.10		äyžçšzāšŇēĲæĀĀæŮzæsŦŕēŕäĲžēčĒēēŕāZĲ	325
11.119.11		ēčĒēēŕāZĲäyžēčŇāŇĒēčĒāĠçĲŕāčđāŁāāŕČæŦŕ	327
11.129.12		äĲčŦĲēčĒēēŕāZĲæL'ŕāĒčšççŽĐāŁšēČĲ	329
11.139.13		äĲčŦĲāĒČçšzæŌgāŁūāōđäĲŇçŽĐāŁŽāzz	331
11.149.14		æŦŦēŮçšççŽĐāsđæĀgāōŽāzL' ēāžāžŕ	334
11.159.15		āōŽāzL' æĲĲ' āŕŕēĀL' āŕČæŦŕçŽĐāĒČçšz	337
11.169.16		*argsāšŇ**kwargççŽĐāijzāŁūāŕČæŦŕçŕāŕ	339
11.179.17		āĲĲčšzäyŁāijzāŁūāĲčŦĲçĲijŮčĲŇēgĐçžē	342
11.189.18		āžēçĲijŮčĲŇæŮzāijŕāōŽāzL'čšz	345
11.199.19		āĲĲāōŽāzL'çŽĐæŮŭāĀZāĲiāgŇāŇŮçšççŽĐæĲŕāšŮ	348
11.209.20		ālŦçŦĲāĠçĲæŦŕæšlēgčāōđçŌŕæŮzæsŦŦēĠēĲ	350
11.219.21		ēĀŕāŕēĠāđŕçŽĐāsđæĀgæŮzæsŦ	356
11.229.22		āōŽāzL' äyŁäyŇæŮĠçõaçŔĒāZĲçŽĐçõĀāŕŦæŮzæsŦ	358
11.239.23		āĲĲāšĀēČĲāŕŮēĠŕāššäyŕæL' gēāŇāžčçāĲ	360

11.249.24	èğçæđŘäyŎáĽEæđŘPythonæžŘçāA	363
11.259.25	æŇEèğçPythonā■ŮèĽĆçāA	367
12	çññā■AçñāñijŽælaaiŮäyŎāŇĚ	369
12.1	10.1 æđĐāžžäyÄäyĽælaaiŮçŽĐāsĆçžgāŇĚ	369
12.2	10.2 æŎgāĽŮælaaiŮècñāĚléĆĽārijāĚēçŽĐāĚĚāōž	370
12.3	10.3 ä;ççŦĽçŽyāržeŮrā;ĐāŘ■ārijāĚēāŇĚäy■ā■ŘælaaiŮ	371
12.4	10.4 ārĚælaaiŮāĽEāĽ'sæĽŘād'ŽäyĽæŮĜāžŮ	373
12.5	10.5 āĽĽçŦĽāS;āŘ■çĽ'žéŮŦ'ārijāĚēçŽōā;ŦāĽEæŦççŽĐāžççāA	375
12.6	10.6 éĜ■æŮŦāĽæ;çælaaiŮ	376
12.7	10.7 èĽŘēāŇçŽōā;ŦæĽŮāŎŇçijĽ'æŮĜāžŮ	378
12.8	10.8 èŦžāŦŮā;■āžŎāŇĚäy■çŽĐæŦŦŦæŮæŮĜāžŮ	378
12.9	10.9 ārĚæŮĜāžŮād'žāĽāāĚēāĽŦsys.path	379
12.10	10.10 éĀŽèĽĜā■ŮçñçäyšāŘ■ārijāĚēælaaiŮ	380
12.11	10.11 éĀŽèĽĜéŠĽ'ā■ŘèĽIJçĽĽāĽæ;çælaaiŮ	381
12.12	10.12 ārijāĚēælaaiŮçŽĐāŦŦæŮŮāĽōæŦžælaaiŮ	397
12.13	10.13 āōĽ'èçĚçgAæIJĽçŽĐāŇĚ	399
12.14	10.14 āĽŽāžžæŮŦçŽĐPythonçŎŦāçĈ	400
12.15	10.15 āĽEāŦŦāŇĚ	402
13	çññā■AäyÄçñāñijŽç;ŚçzIJäyŎWebçijŮçĽĽ	403
13.1	11.1 ä;IJäyžāōcæĽŮçñŦäyŎHTTPæIJ■āĽāžd'äžŠ	403
13.2	11.2 āĽŽāžžTCPæIJ■āĽāžĽ	407
13.3	11.3 āĽŽāžžUDPæIJ■āĽāžĽ	411
13.4	11.4 éĀŽèĽĜCIDRāIJŦāĽĀçŦšæĽŦāŦžāžŦçŽĐIPāIJŦāĽĀéŽE	413
13.5	11.5 āĽŽāžžäyÄäyĽçōĀā■ŦçŽĐRESTæŎēāŦç	415
13.6	11.6 éĀŽèĽĜXML-RPCāōđçŎŦçōĀā■ŦçŽĐèĽIJçĽĽŦççŦĽ	419
13.7	11.7 āIJäy■āŦŦçŽĐPythonèğçéĜĽāžĽāžŦŦæŮŮ'äžd'äžŠ	422
13.8	11.8 āōđçŎŦèĽIJçĽĽŦæŮžæçŦŦççŦĽ	423
13.9	11.9 çōĀā■ŦçŽĐāōcæĽŮçñŦèōđ'èŦA	427
13.10	11.10 āIJç;ŚçzIJæIJ■āĽäy■āĽāāĚēSSL	429
13.11	11.11 èĽŽçĽĽŦŦæŮŮ'äijäēĀSSocketæŮĜāžŮāŦŦèĽŦçç	435
13.12	11.12 çŘEèğçāžŦāžŮēĽ'sāĽĽçŽDIO	440
13.13	11.13 āŦŦéĀAäyŎæŎēæŦŦād'gādŦæŦŦçžĐ	446
14	çññā■AāžŦçñāñijŽāžŮāŦŦçijŮçĽĽ	448
14.1	12.1 āŦŦāĽĽäyŎāAĽæ■ççžçĽĽ	448
14.2	12.2 āĽd'æŮ■çžçĽĽŦæŮŦāŦēāŮççžŦāŦŦāĽĽ	451
14.3	12.3 çžçĽĽŦŦæŮŮ'èĀžāŦā	454
14.4	12.4 çžŽāĚšéŦōéĽĽāĽEāĽæŦA	458
14.5	12.5 éŸsæ■cæ■zéŦAçŽĐāĽæŦAæIJžāĽŮ	461
14.6	12.6 āĽĽā■ŸçžçĽĽŦçŽĐçĽŮæĀāĽāæĽŦ	464
14.7	12.7 āĽŽāžžäyÄäyĽçžçĽĽŦæŦā	466
14.8	12.8 çōĀā■ŦçŽĐāžŮēāŦçijŮçĽĽ	469
14.9	12.9 PythonçŽĐāĽĽāŦāĽŦAéŮōéçŸ	473
14.10	12.10 āōžāžĽĽäyÄäyĽŦāŦŦāžžāĽā	476
14.11	12.11 āōđçŎŦæŮŮĽæĽāŦŦŦāŦç/èōéçŸĚæĽāđŦ	480
14.12	12.12 ä;ççŦĽçŦšæĽŦāžĽāžçæžççççĽĽ	483
14.13	12.13 āđ'ŽäyĽçžççĽĽŦæŮŦāĽŮèŦç	491

14.1412.14	āIJÍUnixçşçzçşşäyŁÉİcāRřāŁáōŁæŁd'ēfZçlŃ	494
15	çññā■AäyL'çñāijŽēDŽæIJñcijÚçlŃäyŌçşçzçşçōaçRĒ	498
15.1	13.1 éĀŽēfGéG■āōŽāRŠ/çōaqAŞ/æŪGüzūāŌēāRŪēçŞāĒē	498
15.2	13.2 çZŁæ■ççlŃāzRāzūçZāGžēŤŽēřāŁæAř	499
15.3	13.3 èğçæđRāŚçjāzd'ēāNēĀL'ēāz	499
15.4	13.4 èŁRēāNāēŪūāijzāGžāřEçāAēçŞāĒēāRŘçd'ž	503
15.5	13.5 èŌūāRŪçZŁçñřçŽDād'gāřR	503
15.6	13.6 æL'gēāNād'ŪēČlāŚçjāzd'āzūēŌūāRŪāōČçŽDēçŞāGž	504
15.7	13.7 ād'■āLūæLŪēĀĒçgžāLlāēŪGüzūāšNçZōāçT	506
15.8	13.8 āLŽāzžāšNēgčāŌNāçŞæaçæŪGüzū	508
15.9	13.9 éĀŽēfGæŪGüzūāR■æšæLçæŪGüzū	509
15.10	13.10 èřzāRŪēĒççōæŪGüzū	510
15.11	13.11 çZŽçōĀā■ŤēDŽæIJñāçdāŁāæŪēāŁŪāŁšèČç	514
15.12	13.12 çZāGçæŤřāžŞāçdāŁāæŪēāŁŪāŁšèČç	516
15.13	13.13 āōđçŌřāyĀäyŁēōaqēŪūāZl	517
15.14	13.14 éŽRāLūāEĒā■YāšNçCPUçŽDāçŁçŤlēGR	519
15.15	13.15 āRřāLlāyĀäyŁWEBætŘēgŁāZl	521
16	çññā■AāZŽçñāijŽætNērŤāĀAērČērŤāšNāijČāyŷ	522
16.1	14.1 æŤNērŤstdoutēçŞāGž	522
16.2	14.2 āIJlā■ŤāĒČætNērŤāy■çZāřzēšaqæL'ŞēaqēyĀ	523
16.3	14.3 āIJlā■ŤāĒČætNērŤāy■ætNērŤāijČāyŷāČĒāĒç	527
16.4	14.4 āřEætNērŤēçŞāGžçŤlāēŪēāŁŪēōřāçŤāLřæŪGüzūāy■	528
16.5	14.5 āŁçŤēæLŪēIJšæIJŽætNērŤād'set'ē	530
16.6	14.6 ād'ĐçRĒād'ŽāyŁāijČāyŷ	531
16.7	14.7 æ■ŤēŌūæL'ĀæIJL'āijČāyŷ	533
16.8	14.8 āLŽāzžēGłāōŽāzL'āijČāyŷ	534
16.9	14.9 æ■ŤēŌūāijČāyŷāRŌæLŽāGžāRēād'ŪçŽDāijČāyŷ	536
16.10	14.10 éG■æŪřæLŽāGžèçnā■ŤēŌūçŽDāijČāyŷ	539
16.11	14.11 èçŞāGžē■ēāŚŁāŁæAř	539
16.12	14.12 èřČērŤāšžæIJñçŽDçlŃāžRāt'l'æžČēŤŽēř	541
16.13	14.13 çZāççŽDçlŃāžRāAžæĀgèČçætNērŤ	543
16.14	14.14 āŁāēĀšçlŃāžRēfRēāN	546
17	çññā■AāžŤçñāijŽČēr■ēlĀæL'l'āsŤ	551
17.1	15.1 āçŁçŤlçtypesēōfēŪōCāzççāĀ	553
17.2	15.2 çōĀā■ŤçŽDČæL'l'āsŤælāālŪ	559
17.3	15.3 çijŪāEŽæL'l'āsŤāGçæŤřæŞ■āçIJæŤřçžD	563
17.4	15.4 āIJlČæL'l'āsŤælāālŪāy■æŞ■āçIJēŽRāçæNĠēŚL	565
17.5	15.5 āžŌæL'l'āsŤælāālŪāy■āōŽāzL'āšNārijāGžçŽDAPI	568
17.6	15.6 āžŌČēr■ēlĀāy■ēřČçŤlPythonāžççāĀ	572
17.7	15.7 āžŌČæL'l'āsŤāy■ēGŁæŤçāĒlāsĀēŤĀ	578
17.8	15.8 ČāšNPythonāy■çŽDçžŁçlŃæūūçŤl	578
17.9	15.9 çŤlSWIGāNĒēçĒCāzççāĀ	579
17.10	15.10 çŤlCythonāNĒēçĒCāzççāĀ	584
17.11	15.11 çŤlCythonāEŽēŃYāĀgèČççŽDæŤřçžDæŞ■āçIJ	591
17.12	15.12 āřEāGçæŤřæNĠēŚLēçnā■cāyžāRřērČçŤlāřzēšā	595
17.13	15.13 āijāēĀšNULLçZşāřççŽDā■ŪçñēyşçZCāGçæŤřāžŞ	597

17.14	15.14	äijäéĂŠUnicodeā■ŮčņēäyšçzŽCăĜıæTřăžŞ	601
17.15	15.15	Că■Ůčņēäyšè;ñæ■cäyžPythonā■Ůčņēäyš	605
17.16	15.16	äy■çãõãõŽçijŮčăAæäijâijRçŽĐCă■Ůčņēäyš	606
17.17	15.17	äijäéĂŠæŮĜăžũăR■çzŽCăLı'ăşT	609
17.18	15.18	äijäéĂŠăũşæL'ŞâijĂçŽĐæŮĜăžũçzŽCăLı'ăşT	610
17.19	15.19	ăžŮCér■élĂäy■èrżăRŮçşzæŮĜăžũăržèşă	612
17.20	15.20	ăd'ĐçRĚCér■élĂäy■çŽĐăRřêf■ăžcăržèşă	614
17.21	15.21	èřLæŮ■ăLĚæõťéTŽèřř	615
18		éŽĐă;TA	616
18.1		ăIJčžřètĐæžŘ	616
18.2		Pythonā■çăžăăžçş■	617
18.3		énŸçžgăžçş■	617
19		ăĚşăžŮèrSèĂĚ	618
20		Roadmap	618

Contents:

1 Copyright

ăžçăR■iijŽ āĂLPython CookbookāĂŃ3rd Edition

ă;IJèĂĚiijŽ David Beazley, Brian K. Jones

èrSèĂĚiijŽ çĚLèČ;

çLı'æIJñiijŽ çññ3çLı'

ăĜžçLı'Łçd' ħiijŽ OăĂŽReilly Media, Inc.

ăĜžçLı'ŁæŮčæIJşiiijŽ 2013ăžt' 5æIJŁ08æŮč

Copyright Ăł 2013 David Beazley and Brian Jones. All rights reserved.

æŽt'ăd'ŽăRŠăyCăŁæAřèrũăRČèĂČ

<http://oreilly.com/catalog/errata.csp?isbn=9781449340377>

2 ǎL'■èíĂ

2.1 éążçŻóăÿzéął

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

2.2 èrŚèĂĚçŽĎèrì

äžžçŦşèÑęçş■iijŊæŁŚçŦí PythoniijA

ərSəĀĖäyĀçZt'āiZæNāĭjŁçTĭ Python 3iijNāZāāyžāōČčäzčēalāžE Python
 çŽDæIJĭĭēāĀCēZĭçDūāRŠāRŌāĖiĭjāōžæYřāōČçŽDçañaij'd'iijNāĭEæYřēŁZāyĭlāsĀēĭcéŁšæŮT'aiijZæT'zāRŸçZ
 èĀNāyT Python 3 çŽDæIJĭĭēĭIJĀēēAæřRāyĭlāžžçŽDāyōāL'āŠNæTræNāāĀC
 çŽōāL'■āyČēĭcāyŁçŽDæT'ZĭNāžēçs■iijNçĭSāyŁçŽDæL'NāĖNād'gēČĭāLēĀšžæIJĭēČĭæYř
 2.x çšžāLŮçŽDiiijNāyŠēŮlāšžāžŌ 3.x çšžāLŮçŽDžēçs■ārŠçŽDāRřæĀIJāĀC

[illegible]

erSèĀĒaijŽaiZæŃAāržèGħaušærRäyĂāRēçŽDçfžèrSèt'sèt'čiiJŃāLZæsĆénYèt'léGRāĀCă;EāRŪèČ;āL
æÇædIJērSæŬĞäy■æI JL'ázĀzLéTŽæijRçŽDāIJræŮžèruād'gǎoūègAērĒiijNāžšæñcèŁŌād'gǎoūéŽRæŮūæN

vidao620@gmail.com

2.3 ä¡JèĀĔçŽĎèrí

3
 æĠlāzŌ 2008 āzt'āzælēiijNPython 3 ælġl'zāGžāyŪāzūæĒcæĒcēfZāNŪāĀCPython
 3 çŽDætAæqNāyĀçŽT'ècnēod'āyžēIJĀèeAāġLéTġāyĀæotæUūeŪt'āĀC
 āžNāođāyLēijNāLræĹSāEŽēfZæIJñāzēçŽD 2013 āzt'iijNçzġlād'gēĊlāLēçŽD Python
 ċlNāžRāSŸāz■çĎūaIJġTŝāžgçŌrācČāy■ā;fçTġçŽDæŸrçL'ŁæIJñ 2 çšzāLŪiijN
 æIJĀāyžēeAæŸrāZāāyž Python 3 āy■āRŠāRŌāEijāōzāĀCærñæUāçŪSéŪōiijNāržāžŌāūēajIJāIJléAŪçTŽāzçç
 ā;EæŸræTġçIJijæIJlælēiijNā;āāršāijZāRŠçŌr Python 3 çŽZā;āāyçælēāy■āyĀæāūçŽDæĊŁāŪIJāĀC

æ■čāċ Python 3 äžčēāIæIJlæläyÄæäüijNæŨřçŽĐăĂPython Cook-
bookăĂNçL'LæIJnçŻÿærTè; ČăžNăL'■çŽĐçL'LæIJnæIJL'ăžEäyÄäyİlæElæŨřçŽĐæTzârYăĂČ
éĚUăĒLiiJNăžšæYræIJĂÉG■ēeAçŽĐiiJNēfZăĐRăŠşçİĂæIJnăžæYrăyĂæIJnēIdăyŷal'■æšŁçŽĐăRCêĂČăž
Python 3.3 çL'LæIJnăyNēİcçijŪăEZăŠNætNērTçŽĐiiJN āžūăşæIJL'èĂČèZŚăžNăL'■ēĂAçL'LæIJnçŽĐăĒijă
ăjEæYræĹSăžnăIJĂçZĹÇŽĐçŽōçŽĐæYrăEZăyĂæIJnăoNăĒİlășžăžŌçŎřăžcăuēăEuăŠNēr■ēİĂçŽĐăžēcş■ăĂ
æĹSăžnăyNæIJZæIJnăžēēČjăd' şæNĞărījăžžăžnăj;ŁçTĬ Python 3
cijŪăEZăŨřçŽĐăžćčăAæĹŪēĂĒă■ĞçžgăžNăL'■çŽĐăAŮçTŻăžćčăAăĂČ

ærnæUäçŮŠéŮōriijNčijŮāEŽäyĂæIJñèĹZæăũçŽDžęçzŻcijŮëŁŚăũă;IJăyçæİēäyĂăőŻçŽDæŃSæŁYăĂ
Python çğŸçš■ŽĐērīiijNăijŻăIJerÿâĈ ActiveStateăĂZs Python recipes æŁŮêĂĖ Stack
Overflow çŽDç;ŚçŋZăyLæRİJăĹræTřazěă■ÇèðaçŽDæIJŁçTłçŽDçğŸçš■iijNăjEæYřăĚüäy■çziad'gėĆlálĒéé
èĹZăżZçğŸçš■éŽď'ăžEæYřăşžăžŌ Python 2 cijŮāEŽăznăd' ŮiijNăRřĈç;èĹYăIJŁăŁLăđ'ŽęğcăĔşăŮzæăĹăĹ.
iijŁærTăeĈ 2.3 âŞŇ 2.4 çŁŁæIJñiijŁăĂĈ âŘead' ŮiijNăôČăznèĹYăijŻçzŘăyyă;ŁçTłăyĂăžZèĹGăŮŮçŽDæŁ.

Python 3

ɛʃZæIJnăzɛçŽDæL'ĂæIJL'äyžécYéČ;æYřaşžăžŌăuşçzRă■YăIJčŽDăžččăAăŠNăĽĂæIJřijNěĀNăy■æ
 Python 3 çL'zæIJL'çŽDçğYğś■ăĀĆăIJăŌşæIJL'ăžččăAăşžçăĂăyĹijNăĽSăžnăŏNăĒlă;ççTlæIJĂæŪřçŽD
 Python æĽĂæIJřăŌzæTžéĂăăĀĆăĽĂăžēijNăžză;TæČšă;ççTlæIJĂæŪřăĽĂæIJřcijŪăEžăžččăAçŽDçlNăž

Python éCÉāššæL'ÄæIJL'çŽDäyIJeēfāĀĆ āZāæ■d'iijNæŁŚāznāijYāĒLéĀL'æNl'āžE Python
ēr■ēĀāyāfČēClāLEijjNāzēārLēCčāžZæIJL'čĪāāzfæsŻāžTčŦlécEāššçŽĐUōēcYāĀĆ
āRēād' ŪiijNāEūāy■æIJL'a;Łād'ŽçgYćś■čŦlēāsŦčd'ž Python 3 çŽDæŨřčŁ'záĀgüijN
ēŁZārčāžŌā;Łād'Žāžžælēērt' æYřærŦē; ČēŽNčŦščŽĐiiijNāŠłæĀTæYřā;ŁcŦĪ

Python ěÄÄçŁŁæIŃčŽDčzRéIŃäyřárŃčŽDčIŃäzŘáŠŸāĀĆ
ěfZāžZčd'žāĭŃčIŃäzŘāžšāijŽāAŘāŘŠāžŌāšTčd'žāyÄāžZæIJŁčIÄāžfæšZāžTčTłčŽDčijŮčIŃäŁæIJř
řijŁā■ščijŮčIŃäIqāijRřijŁřijŃěÄŃäy■æŸřāžĚāžĚāšZā;■āIJlāyÄāžZāĚūā;ŠčŽDěŮšćčŸāyŁāĀĆār;čōāāžšæ.
Python ěř■ēĬÄæāyřāfČāšŃæāĠāĠEāžšāĀĆ

2.4 èŁŽæIŋăžęéĂĆăŘĹěřĄ

ɛʃZæIŋnāzɛçŽDçZōæǎGërzèĀĖæYřéĆčāžZæČšæuśăĖěçŘĚèğç Python
 ɛř■ĖĀĖIJžĀLŭăSŇĈŌřāžčcijŮčĹNěćŌăiȳçŽDæIJL'čzŘĖĹNçŽD Python çĹNāžRăŠYăĀĆ
 æIJnāzɛəd'gɛĆĹăĹĖăĖĖăăōzɛZĖäy■ăžŌăIJĹăăĜăĜĖăžSȳijŇăăĖăđŭăSŇnāžTčTĹčĹNāžRăy■ăžĤæšZă;ĤçTĹçŽD
 æIJnāzɛæL'ĀæIJL'çd'žă;ŇăĹĜăĀĜĖō;ɛrżèĀĖăĖŭăIJL'ăyĀăōŽçŽDçijŮčĹNěćNăžřăžŭăyTăRăřăžɛərɤæĜĆçZ
 iijĹæřTăçCăšžæIJŋçŽDɛōaçōŮăIJžçĜSă■ççšɛərĖijŇăTřă■ōçžSăđDçšɛərĖijŇçōŮășTăđ■ăĹĆăžɛijŇçșž
 ɛř■ĖĀĖIçijŮčĹNç■L'iijL'ăĀĆăŘăđ'ŮijŇăřRăyĹçd'žă;Něć;ăRĹăYřăyĀăyĹăĖĖĖŮăNĜăřijŇăçCăđIJɛřzèĀĹ
 æĹSăžnăĀĜăăōZɛrżèĀĖăRăřăžă;ĹĤĖșçžČçŽDă;ĤçTĹăRIJçt'căijTăȘŌăžăRĹĤçšɛĖAșăĀŌăăŭășɛərçăĹJĹçž
 Python æŮĜăęăăĀĆ

æIJL'äyÄäzZæŽt'ǎŁǎénŸçgçŽĐçġŸçs■īijNǎēĆǎđIJèǎǎǎǎĆēŸĖērziijNǎřEǎIJL'ǎŁ'ǎžŌçŘĖègč
Python ǎžTǎsĆçŽĐǎũēǎ;IJǎŌšçŘĖǎǎĆ ǎžŌǎy■ǎ;ǎǎřEǎ■ǎŁǎřǎyÄäzZǎŮřçŽĐǎŁǎǎũǎšNǎŁǎǎIJřīijNǎžǎǎž

2.5 èŁŻæłJňäżęäÿ■éĀĆąŘĹěřA

æƷZæIJnăžęäy■éĂĈăRL Python çŽĐăĹIă■çēĂĚăĂĈăžNăôđăyŁiijNăeIJnăžęăAĞăôŽëržēĂĚăĚŭæIJŁ'
Python æȚȚĹIŃăLŪăĚĚēŪăžęçs■ăy■æL'ĂăȚZăŌĹçŽĐășžçăĂçșēērĚăĂĈ
æIJnăžęăžșăy■æŸřĊççğ■ăfŃéĂșăŔĈēĂĈăL'NăĚŃ iijŁăŁNăçĈăfŃéĂșășēērċășŔăyŁăŁăĹŪăyŃçŽĐășŔă
æIJnăžęăŪăĹIĹăĂȚĈĐăĂĜăăyŁăIJĂĚĜ■ēēĂçŽĐăyžėçŸiijNăeijȚĉđ'žăĜăççğ■ăŔřĊ;çŽĐēğĉăĚșăŪžăăĹiijN
ăŔŔăŁăŽăyĂăyŁēŭșăĹăĹiȚŦărijeržēĂĚēƷZăĚĚăyĂăžžZăŽŦ'énŸçžğçŽĐăĚĂăôžiiȚŁēƷZăžZăŔřăžēăIJĹç;ŠăyŁăĹ

2.6 ĄǃıçžŁčd'żăȚNăzčçăĄ

[http://github.com/dabeaz/
python-cookbook](http://github.com/dabeaz/python-cookbook)

2.9 èGt'èrc

æŁŚäznèaũåŁČæĎšèrcæIJñäzèçŽĎæŁÄæIJfæääåöäžžåŠŸ Jake VanderplasiiĎRobert Kern åŠŇ Andrea Crotti éÍđäyÿæIJL'çŤłçŽĎĕfĎĕðžåŠŇäzžèðóiiĎ èŁŸæIJL' Python çĎ'çåŇžçŽĎäyóåŁ'åŠŇéijŠåŁsāĀČæŁŚäznāŔŇæåũæĎšèrcäyŁäyÄäyłçŁ'ŁæIJñçŽĎçijŮĕŁ Š Alex MartelliĎiiĎAnna Ravenscroft åŠŇ David AscherāĀĆ ār;çðæŁŽäyłçŁ'ŁæIJñæŸfæŮŕåŁŽä;IJçŽĎiiĎNä;EæŸfåŁ■äyÄäyłçŁ'ŁæIJñäyžæIJñäzæŕŔä;ŽäžEäyÄäyŁæŇ æIJāŕŔŌäžšæŸfæIJāéĎ■ĕeAçŽĎiiĎNæŁŚäznĕeAæĎšèrcæŁ'ÄæIJL'æŮ'æIJšéçĎĕgŁçŁ'ŁæIJñçŽĎĕfžĕĀĔiiĎ

3 çññäyĀçñäiiĎŽæŤŕæ■óçžŠæĎĎåŠŇçóŮæşŤ

Python æŕŔä;ŽäžEäĎ'gĕGRçŽĎåĔĔç;őæŤŕæ■óçžŠæĎĎiiĎNāŇĔæŇñåŁŮĕåłiiĎNéZEāŔŁäzĕāŔŁā■ŮāĔ ä;EæŸfiiĎNæŁŚäznāžšäijŽçžŔäyÿçĕŕåŁŕåŁŕĕŕyæČæšĕĕrciiĎNæŌšāžŔāŠŇĔŁĔæzd'ç■Łç■Ł'ĕŁŽäžZæŽóéA■ āZāæ■Ď'iiĎNĕŁŽäyĀçñäçŽĎçŽóçŽĎāŕsæŸfĕóĕðžĕŁŽäžZæŕŤĕ;ČäyÿĕgAçŽĎĕŮĕĕŸāŠŇçóŮæşŤāĀĆ āŔĕāĎ'ŮiiĎNæŁŚäznāžšäijŽçžZāĔžāIJléZEāŔŁæłāāŮ collections ā;Šäy■æŞ■ā;IJĕŁŽäžZæŤŕæ■óçžŠæĎĎçŽĎæŮžæşŤāĀĆ

3.1 1.1 ĕğçåŌŇāžŔāŁŮĕŤŇāĀijçžŽāĎ'ŽäyŁāŔŸĕGR

ĕŮĕĕŸ

çŌŕāIJĀæIJL'äyÄäyŁāŇĔāŔŇ N äyŁāĔĔÇĕŤ'āçŽĎāĔĔÇçžĎæŁŮĕĀĔæŸfāžŔāŁŮĕŤŇāĀŌæåũāŕEāóČĕĔŇĕĕ N äyŁāŔŸĕGRiiĎş

ĕğçåEşæŮžæāŁ

äzžä;ŤçŽĎāžŔāŁŮĕŤŇāĀijŁæŁŮĕĀĔæŸfāŕŕĕŁ■āžčāŕžĕşāiiĎL'āŔfāžĕĕĀŽĕŁĔäyÄäyłçóĀā■ŤçŽĎĕŤŇāĀijĕŕ■ āŤŕāyĀçŽĎāŁ■æŕŔāŕsæŸfāŔŸĕGRçŽĎæŤŕĕGRāŁĔĕāžĕũşāžŔāŁŮĕĔÇĕŤ'āçŽĎæŤŕĕGRæŸfāyĀæåũçŽĎāŁ äžççāAçĎ'žä;ŇiiĎŽ

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

(continues on next page)

(continued from previous page)

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

æĈæđIJăRŸéĜRăyĭæTřăŠŇăžŘăĹŮăĚĈĕt'ăçŽĎăyĭæTřăy■ăNzéĚ■iijŇăijŽăžğĉTšăyĂăyĭaijĆăyyăĂĈ
ăžĉĉăAçđ'žăĬŇiijŽ

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

èõléõž

ăôđéŽĚăyĬiijŇăĚŽçğ■ğĉăŎŇăĬŇăĬjăRřăžĉTĭăIJăžză;TăRřăĚ■ăžĉăřzėşăyĬéĬĉiijŇăĂŇăy■ăžĚăžĚă
ăŇĚăŇăă■ŮĉŇăyşiiijŇăŮĜăžŮăřzėşăiijŇăĚ■ăžĉăŽĭăŠŇĉTšăĹŔăŽĭăĂĈ
ăžĉĉăAçđ'žăĬŇiijŽ

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

æIJĹæŮŮăĂŽiijŇă;ăăRřăĈ;ăRĭæĈşğĉăŎŇăyĂéĈĭăĹĚiijŇăyĉăijĈăĚŮăžŮĉŽĎăĬjăĂĈăřzăžŎĚĚŽçğ■
Python äžŮăşăæIJĹæŘŔăĬŽĉĹ'žăŏĹĉŽĎĕr■ăşTăĂĈă;ĬæŸăŸăřă;ăăRřăžăă;ĬĉTĭăžzăĎĎŔăRŸéĜRăŘ■ăŮăăă;
ăžĉĉăAçđ'žăĬŇiijŽ

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

(continues on next page)

```
91.1
>>>
```

ä;ää£Ééazä£IèrAä;äéÄL'çTlçZDëCçäzZâ■ää;■äRÿéGRäR■äIJläËüäzÜäIJräÜzæsäècñä;£çTlälRäÄC

3.2 1.2 ègçäÕNäRrè£■äzçärzèsäçZDäEÇçt'ääyIæTṛèüEè£GäRÿéGRäyIæTṛæÜüijNäijZæLZäGzäyÄäy

éÜöécY

æÇædIJäyÄäyIäRrè£■äzçärzèsäçZDäEÇçt'ääyIæTṛèüEè£GäRÿéGRäyIæTṛæÜüijNäijZæLZäGzäyÄäy
ValueError äÄC éCçäzLæÄÖæüäL■èC;äzÖè£ZäyIäRrè£■äzçärzèsäç■ègçäÕNäGz N
äyIäEÇçt'ääGzæIèij§

ègçäEşæÜzæäL

Python çZDæYşäRüèäIèç;äijRäRræzèçTlæIèègçäEşæZäyIèÜöécYäÄCærTäçCijNä;ääIJlä■èäzäyÄéÜ
ä;äæÇççz§èöäyNäöüäz■ä;IJäyZçZDäzşäI GäLRçzIijNä;EæYräÖŞÉZd'æÖLçññäyÄäyIäSNæIJäÄRÖäyÄä
ä;EäçæÇædIJäIJL 24 äyIäSçij§è£ZæÜüäZæYşäRüèäIèç;äijRärsæt;äyLçTlälIJzäEijZ

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

äRëäd'ÜäyÄçg■æÇEäEijNäAÇèç;ä;äçÖräIJläIJL'äyÄäZçTlæLüçZDëörä;TäLÜèäIijNærRæIæèörä;T
ä;ääRræzèäÇRäyNéIçè£ZæüäLÈègçè£ZäzZèörä;TijZ

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

äÄijä;ÜæşIæDRçZDæYräyLéIègçäÕNäGzçZD phone_numbers
äRÿéGRærÿè£IJéC;æYräLÜèäIçszädNijNäy■çöæègçäÕNçZDçTṛèIäRüçäAæTṛéGRæYräd'ZärSijLäNÆæN
0 äyIijLäÄC æL'ÄäzëijNäzä;Tä;£çTlälṛ phone_numbers
äRÿéGRçZDäzççäAärşäy■éIJäèçAäAZäd'Zä;ZçZDçszädNæçÄäşèäÖzçäèèd'äöÇæYräRææYräLÜèäIçszäç

æYşäRüèäIèç;äijRäz§èC;çTlälIJläLÜèäIçZDäijÄägNéCälLæÄCærTäçCijNä;ääIJL'äyÄäyIäEñäRyäl
8 äyIäIJLéTäÄTöæTṛæ■öçZDäzRäLÜüijN ä;EæYrä;äæÇççIJNäyNæIJäè£SäyÄäyIäIJLæTṛæ■öäSNäL■éIç
7 äyIäIJLçZDäzşäI GäÄijçZDärzærTäÄCä;ääRræzèè£ZæüäAÇijZ


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

Python 3.6.0

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

Python 3.6.0

Python 3.6.0 is a major release of the Python programming language. It includes several new features and improvements, such as the addition of the `async` and `await` keywords for asynchronous programming, and the introduction of the `dataclass` decorator for simplifying class definitions. The release also includes various performance optimizations and bug fixes. For more information, see the [Python 3.6.0 release notes](#).

```
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)
```

Python 3.6.0 is a major release of the Python programming language. It includes several new features and improvements, such as the addition of the `async` and `await` keywords for asynchronous programming, and the introduction of the `dataclass` decorator for simplifying class definitions. The release also includes various performance optimizations and bug fixes. For more information, see the [Python 3.6.0 release notes](#).

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

(continues on next page)

(continued from previous page)

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

æIJL'æUŭāĀZījNā;āæČšĕğčāŌNāyĀāžZāĒČčt'āāRŌāyčājČāōČāžñījNā;āāy■ēČ;čōĀā■Tāřsā;ččTl
* īijN ājEæYřā;āāRřāžēā;ččTlāyĀāyĽæZōēĀZčZDāžšājČāR■čgřījNāēTāēČ _ æLŮēĀĚ
ign īijLignoreīijLāĀČ

āžččāAčd'žā;NīijŽ

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

āIJlā;Lād'ŽāG;æTřāijRēr■ēlĀāy■īijNāYřāRūēğčāŌNēr■æšTēūšāLŮēālad'DčRĚæIJL'ēōyād'ŽčZyāijjā
ā;āāRřāžēā;LāōžæYščZDāřEāōČāLEāL'sæLŘāL'■āRŌāyd'ēČlāLEīijŽ

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

āēČædIJā;āad'sēAĽæYŌčZDērīijNēfYēČ;čTlēfZčg■āLEāL'sēr■æšTāŌzāūgāēZčZDāōđčŌřeĀšā;ščōŮ

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

čDūāRŌīijNčT'sāžŌēr■ēlĀāsČēlččZDēZŘāLŮīijNēĀšā;šāžūāy■æYř Python
æšĚēTĚčZDāĀČ āZāæ■d'īijNāēIJāāRŌēČčāyĽēĀšā;šāijTčd'žāžĒāžĒæYřāyĽāē;āēGčZDæŌččt'čč;čāžEīijNā

3.3 1.3 āēlčTŽæIJĀāRŌ N āyĽāĒČčt'ā

ēŮōēčY

āIJlēf■āžčæš■ā;IJæLŮēĀĚāĒūāžŮæš■ā;IJčZDæUŭāĀZījNāĀŌæāūāRlāēlčTŽæIJĀāRŌæIJL'ēZŘāGā

èġċàEşæŮzæąŁ

æłİçTŻæIJLéŽŘăŎĚăŘşëŕăĭTæ■çæŸř collections.deque
ăđ'ğæŸĭèžñæL'ŇçŽĐæŮûăĂŽăĂĆæřTăęĆřijŇăyŇéłćçŽĐăžččăĂăIJăđ'ŽëăŇăyŁéłćăĂŽçŏĂă■TçŽĐæŮĠæ
ăžűëŤTăŽđăŇzéĚ■æL'ĂăIJlèăŇçŽĐæIJĂăŔŎŇăqŇřijŽ

```
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)
```

èőłéőž

æŁŚăžñăIJăĚZæşëęřăĚĆçř'ăçŽĐăžččăĂăŮřřijŇéĂŽăyŷăijŽăĭfçTłăŇĚăŔŇ yield
èăłèĭăĭijRçŽĐçTşæŁŔăŽłăĠĭæTřřijŇăžşăřşæŸřæŁŚăžñăyŁéłćçđ'žăĭŇăžččăĂăŷ■çŽĐéĆçæăŷăĂĆ
èŁŽæăŷăŔřăžëăřĚăŔIJçř'ćèŁĠçłŇăžččăĂăŤŇăĭfçTłăŔIJçř'ćçžşæđIJăžččăĂăġçĚăĂăăĂĆăęĆăđIJăĭăèŁŸăy■
4.3 èŁĆăĂĆ

ăĭfçTł deque(maxlen=N) æđĐéĂăăĠĭæTřăijŽæŮřăžžăyĂăyłăŽžăŏŽăđ'ğăřRçŽĐéŸşăŁŮăĂĆăĭşæŮ
æIJăĚĂĂçŽĐăĚĆçř'ăăijŽëĠłăŁłèćŋçġžéŽđ'æŎŁăĂĆ

ăžččăĂçđ'žăĭŇřijŽ

```
>>> 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)
```

āŕĭçōāĭāāzšāŕŕāzēæL'ŊāŁāIJlāyĀāyġāŁŪēāġāyŁāōđçŌŕēŁZāyĀçŽĐæŞ■āĭIJġġLæŕŤæĆācđāŁāāĀAāŁ
 æŽŕ'āyĀēŁŋçŽĐġġŊ deque çşzāŕŕāzēēēŋçŤġāIJlāzzāĭŤāĭāāŕġēIJĀēēAāyĀāyġçōĀā■ŤēŸşāŁŪæŤŕæ■ōç
 āēĆæđIJāĭāāy■ēōĭçĭōæIJĀād'ģēŸşāŁŪād'ģārŖġġŊēĆcāzŁārşāġŹāĭŪāŁŕāyĀāyġæŪāēŽŖād'ģārŖēŸşāŁŪġġŊ
 äžççāAçđ'žāĭŊġġŽ

```

>>> 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
    
```

āIJġēŸşāŁŪāyđ'çŋŕæŖŞāĒēæŁŪāŁāēŽđ'āĒČçŕ'āæŪūēŪŕ'ād'■āĭĆāžēēČĭæŸŕ O(1)
 ġġŊāŊzāŁŋāzŌāŁŪēāġġġŊāIJlāŁŪēāġçŽĐāġĀād't'æŖŞāĒēæŁŪāŁāēŽđ'āĒČçŕ'āçŽĐæŪūēŪŕ'ād'■āĭĆāžēāyž
 O(N) āĀĆ

3.4 1.4 æşēæL'çæIJĀād'ģæŁŪæIJĀārŖçŽĐ N āyġāĒČçŕ'ā

éŪōēčŸ

æĀŌæāūāzŌāyĀāyġēZEāŖġāy■ēŌūāĭŪæIJĀād'ģæŁŪēĀĒæIJĀārŖçŽĐ N
 āyġāĒČçŕ'āāŁŪēāġġġş

ēğçāEşæŪzæāġ

heapq æġāāġŪæIJL'āyđ'āyġāĢĭæŤġġŊnlargest() āŞŊ nsmallest()
 āŖŕāzēāōŊçĭŌēğçāEşæŁZāyġēŪōēčŸāĀĆ

```

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]
    
```

āyđ'āyġāĢĭæŤŕēČĭēČĭæŌēāŖŪāyĀāyġāĒşēŤŌā■ŪāŖĆæŤġġŊŤġāzŌæŽŕ'ād'■āĭĆçŽĐæŤŕæ■ōçzşæđĐā

```

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

(continues on next page)

(continued from previous page)

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

erSèĀĖæšlriijŽäyLéIcäzčçäAāIJlāržærRäyIāĖĈçt' æēfZèaŃāržærTçŽDæUūāĀŽiijŃäijŽäzē
price çŽDāAijèfZèaŃærTè; ČāĀĆ

èõléõž

æĖČæđIJā;äæČšāIJläyĀäyIéZEāRLäy■æšæL;æIJĀārRæLŪæIJĀād' gçŽD N
äyIāĖĈçt' äriijŃäzūäyT N ārRāžŌéZEāRLāĖĈçt' āæTŕeGRiijŃéČčāzLèfZāžZāĖ;æTŕæRRä;ZāžEā;Lāē;çŽDæ
āZāyžāIJlāzTāsČāōđçŌŕeGŃeIciijŃeēŪāĖLāijŽāĖLārEēZEāRLæTŕæ■ōēfZèaŃāāEæŌšāžRāRŌæT;āĖĖäy.

```
>>> 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]
>>>
```

āāEæTŕæ■ōçzŠæđDæIJĀéG■ēæAçŽDçL'žā;AæYŕ heap[0]
ærÿēfIJæYŕæIJĀārRçŽDāĖĈçt' āāĀČāzūäyTāLŕ'ā;ŽçŽDāĖĈçt' āāRŕāžēā;LāōžæYŠçŽDēĀŽēfGērČçTī
heapq.heappop() æŪžæšTā;ŪāLŕiijŃērēæŪžæšTāijŽāĖLārEçñnāyĀäyIāĖĈçt' āäijžāGžæIēriijŃçDūāRŌ
O(log N)iijŃN æYŕāāEāđ' gārRiijL'āĀĆ æŕTāēCiijŃāæČæđIJæČšèæAæšæL;æIJĀārRçŽD 3
äyIāĖĈçt' äriijŃā;āāRŕāžēēfZæāūāĀŽiijŽ

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

ā;ŠèæAæšæL;çŽDāĖĈçt' ääyIæTŕçŽyāržærTè;ČārRçŽDæUūāĀŽiijŃāĖ;æTŕ
nlargest() āŠŃ nsmallest() æYŕā;LāRLéĀĆçŽDāĀĆ
æĖČæđIJā;äāžĖāzĖæČšæšæL;āTŕäyĀçŽDæIJĀārRæLŪæIJĀād' gŕiijLN=1iijLçŽDāĖĈçt' āçŽDērIiijŃéČčāzL
min() āŠŃ max() āĖ;æTŕäijŽæŽt' āfñāžZāĀĆ çšžäijijçŽDriijŃāæČæđIJ N
çŽDād' gārRāŠŃéZEāRLād' gārRæŌēēfSçŽDæUūāĀŽiijŃéĀŽāyāĖLæŌšāžRēfZäyIéZEāRLçDūāRŌāE■ā;
iijL sorted(items)[:N] æLŪēĀĖæYŕ sorted(items)[-N:]
iijL'āĀĆ éIJĀēæAāIJæ■ççāōāIJžāRLā;fçTlāĖ;æTŕ nlargest() āŠŃ
nsmallest() æL■ēČ;āŕSæŃēāōČāžñçŽDäijYāLē iijLāæČæđIJ N
āfñæŌēēfSçZEāRLād' gārRāžEriijŃéČčāzLā;fçTlæŌšāžRæš■ā;IJäijŽæŽt' æē;äžŽiijL'āĀĆ

āŗıçōāāıāæşāæIJL'āŁĒēēAäy'ÄāōZäıŁçŁĲēŁZēĠŃçŽĐæŰzæşTııjŃäıEæYřāāEæTřæ■ōçzŞæđĐçŽĐāōđçŮ
āşzæIJñäyŁāRĲēēAæYřæTřæ■ōçzŞæđĐāŠŃçōŮæşTāzēçş■ēĠŃēĲēČıäıjŽæIJL'æŘŘāRĲāŁřāĀĆ
heapq æĲāĲŮçŽĐāōY'æŰzæŮĠæāçēĠŃēĲēāzşēřēçzEçŽĐäzŃçz■äzEāāEæTřæ■ōçzŞæđĐāzTāśČçŽĐāōđçŮ

3.5 1.5 āōđçŮřäy'ÄäyĲäıjY'äĒŁçžgæŮşāzRçŽĐéYşāĲŮııjş

éŮōēćY

æĀŮæāūāōđçŮřäy'ÄäyĲæŃL'äıjY'äĒŁçžgæŮşāzRçŽĐéYşāĲŮııjş
āzŰäyTāIJĲēŁZäyĲéYşāĲŮäyĲēĲæřRæñā pop æŞ■āıIJæĀzæYřēŁTāZđäıjY'äĒŁçžgæIJĀénYçŽĐéČcäyĲāĒČç

èğcāEşæŰzæāĲ

äyŃēĲçŽĐçşāĲĲçŁĲııheapq æĲāĲŮāōđçŮŮřäy'ÄäyĲçōĀā■TçŽĐäıjY'äĒŁçžgæYşāĲŮııjŽ

```
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]
```

äyŃēĲæYřāōČçŽĐäıŁçŁĲæŰzäıjRııjŽ

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

(continues on next page)

(continued from previous page)

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

azTçzEègCârşâRfrazêâRŞçÖrrijNçnnäyÄäyl pop() æŞ■ä;IJeŦTâZđaijYâĖĹçžgæIJAénYçŽDâĖĈçt'aaĖ
âRêad'ŮæşlæĐRâĹrâeCædIJäyd'äylæIJL'çĹĀçŽyâRŊaijYâĖĹçžgçŽDâĖĈçt'äiijĹ foo âŞN
grok iijL'rijNpop æŞ■ä;IæNL'çĖġâōCäznècñæRŠâĖēâĹrēYşâĹŮçŽDæqžâžRēŦTâZđçŽDâĖĈ

èõlèõž

èŦŽäyÄârRêĹCæĹSäznäyžèeAâĖŞæşĹ heapq æĹaâĹŮçŽDä;ŦçTĹâĖĈ
âĖ;æTŦ heapq.heappush() âŞN heapq.heappop() âĹĖâĹnâĹĹéYşâĹŮ
_queue äyĹæRŠâĖēâŞNâĹæŽd'çnnäyÄäylâĖĈçt'äiijN äžüäyTēYşâĹŮ
_queue äŦĹerAçnnäyÄäylâĖĈçt'âæNēæIJL'æIJAénYäijYâĖĹçžgiiijĹ
1.4 èĹCâušçzRèõlèõžèŦĖèŦŽäylēŮōēçYriijL'âĖĈ heappop()
âĖ;æTŦæĖæYrēŦTâZđâĹæIJAârRçŽDâĖĈçt'äiijNēŦŽârşæYŦäŦĹerAēYşâĹŮpopæŞ■ä;IJeŦTâZđæ
âRêad'ŮriijNçTšâžŮ push âŞN pop æŞ■ä;IæŮŮéŮŦâd'■æĹCâžæyž
O(log N)riijNâĖŮäy■ N æYŦâĖEçŽDâd'ğârRriijNâZâæ■d'ârşçōŮæYŦ N
âĹĹâd'ğçŽDæŮŮâĖZâōCäznèŦRèqNēĖĖâžæžşâĹæŮğâĹĹâŦnâĖĈ

âĹĹäyĹĹéçäzççâÄäy■riijNēYşâĹŮâNĖâRnâžEäyÄäyl (-priority, index,
item) çŽDâĖĈçzDâĖĈ äijYâĖĹçžgäyžet'şæTŦçŽDçŽōçŽDæYŦä;ŦâĹŮâĖĈçt'âæNL'çĖġäijYâĖĹçžgäzŮénY
èŦŽäylēüşæŽōéĖŽçŽDæNL'äijYâĖĹçžgäzŮä;ŮâĹrénYæŮŖâžRçŽDâĖĖæŮŖâžRæAŦâũğçŽyâR■âĖĈ

index âRŸéGRçŽDä;IJçTĹæYŦäŦĹerAârNç■L'äijYâĖĹçžgâĖĈçt'âçŽDæ■ççâōæŮŖâžRâĖĈ
éĖŽèŦGâĹĹâ■YäyÄäyläy■æŮâçdâĹäçŽD index äyNæâGâRŸéGRriijNâRfrazèçâōâŦĹâĖĈçt'âæNL'çĖġâōCä
èĖNäyTriijN index âRŸéGRäzşâĹĹçŽyâRŊaijYâĖĹçžgâĖĈçt'âærTèçççŽDæŮŮâZèŦuâĹrēĖ■èeAä;IJçTĹâ

äyžâžEēYŦæYŮèŦŽâžZriijNâĖĹĹâĖĖōŽ Item âōđä;NæYŦäy■æTŦæNæŮŖâžRçŽDriijŽ

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

âeCædIJä;äâ;ŦçTĹâĖĈçzD (priority, item) iijNâRĹèeÄäyd'äylâĖĈçt'âçŽDäijYâĖĹçžgäy■âRŊâŦ
ä;EæYŦæCædIJäyd'äylâĖĈçt'âäijYâĖĹçžgäyÄæâũçŽDèŦriijNéCçäzĹærTèççæŞ■ä;IJAŦŦaijZeüşâžNâĹ■äyÄ

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

(continues on next page)

(continued from previous page)

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

éĀŽēfĜāijṬāĒēāRēād'ŨčŽĎ index āRŸéGRčzĎāēLŖāyL'āĒČčzĎ
(priority, index, item) iijŊārseČ;ā;Ĺāē;čŽĎéAēāēāyLēlččŽĎēT'ZērfiijŊ
āŽāāyžāy■āRrēČ;æIJL'āyđ'āylāēČčt'āēIJLčŽyāRŊčŽĎ index āĀijāĀCPython
āIJlāAZāēČčzĎārfTē;ČāŨūāĀZiijŊāēČāđIJāL'■ēlččŽĎārfTē;ČāuščzRāRŖāzēčāōāōŽčzSāđIJāžEiijŊ
āRŌēlččŽĎārfTē;ČāS■ā;IJārsāy■āijŽāRŠčTšāžEiijŽ

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

æĆædIJă;ăæĈşăIJăd'ZăyĭĉzĕĭĈNăy■ă;ĕĭTĭăRŇăyĂăyĭĕYşăĭŬĭĭjŇĕĈăzĭLă;ăĕIJăĕĕAăĉďăĽăĕĂĈă;Şĕĭ
ăŖăzĕăşĕĭIJŇ 12.3 ăŖŖĕĽĈĭZĬă;Ňă■ŖăĭjTĉd'zăYŕăŬŬăăăAŹĭZĬăĂĈ

heapq ælǣlŭçŽďǎǎŸæŮžæŮǦæąçæIJL' æŽt' èřęçzEçŽďä; Ná■ŘčlNážRäzëâRŁåržzžŎăăEçŘEèőžâRŁ

3.6 1.6 å■ŮăĚÿăÿ■çŽĐéŤőæŸăăřĎăđ'ŽăÿłăĀij

éŮóécỲ

æAŒæuũaõđŒŒřăĂăyľĚŦŒăřzăŦăđ'ŽăyľăĂijçŽĐă■ŮăĚyřijĹăžšăŦŦ
multidictijĹijš

èġčǎẸșæŮźæǻŁ

äyÄäylā■UāĒyārſæYrāyÄäyleTōārZāzTāyÄäylā■TāĀijcŽDæYāārDāĀĆāēĆædIJä;āæČšēēAäyÄäyleTō.
ærTāēCāLŪēalēLŪēĀĒēZEāRĻēGŅēlĭcāĀĆærTāēCīijNä;āāRrāzēāCRāyŅēlĭcēfZēāuāēdDēĀāēfZēāuēŽDā

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

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


èŁŻäÿÄärRèLCæL'ÀèóíèøžčŽDèŮóécŸèùšæŦræ■óad'ĐčŘĚäÿ■čŽDèóřā;Ŧā;ŠčšzéŮóécŸæIJL'ād'ğčŽD
1.15 ārRèLCčŽDä;Nā■ŘāĂĆ

3.7 1.7 ā■ŮāĚÿæŮŠāžŘ

éŮóécŸ

ā;äæČšāLŽāžžäÿÄäÿlā■ŮāĚÿiijNāžūāÿŦāIJlè■āžčæLŮāžRāLŮāNŮèŁŻäÿlā■ŮāĚÿčŽDæŮūāĂŽèČ;ād

èğčāĚšæŮžæāĹ

äÿžāžĚèČ;æŮğāLŮäÿÄäÿlā■ŮāĚÿäÿ■āĚČčŦ'āčŽDéāžāžŘiijNā;āāRřāžēā;ĚčŦí
collections ælāāĹŮäÿ■čŽD OrderedDict çšžāĂĆ
āIJlè■āžčæš■ā;IJčŽDæŮūāĂŽāōČāijŽāĹlæNāāĚČčŦ'āččnæRŠāĚēæŮūčŽDéāžāžŘiijNčd'žā;NāēČāÿNijŽ

```
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])
```

ā;Šā;äæČšèèAæđĐāžžäÿÄäÿlāŦĚæĹēēIJĚèēAāžRāLŮāNŮæLŮčijŮčāAæLŦāĚūāžŮæāijāijRčŽDæŸāār
OrderedDict æŸŦēĹđäÿÿæIJLčŦíčŽDāĂĆ æŦŦāēČiijNā;äæČšçš;çāōæŮğāLŮāžē
JSON çijŮčāAāŦŮā■ŮæōŦčŽDéāžāžŘiijNā;āāRřāžēāĚĹā;ĚčŦí OrderedDict
æĹēæđĐāžžèŁŽæūčŽDæŦræ■ōiijŽ

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

èóíèøž

OrderedDict āĚĚēČĹčzt'æŁd'čĹÄäÿÄäÿlæāžæ■óēŦōæRŠāĚēēāžāžRæŮŠāžRčŽDāRŦNāRŠēš;ēāĹāĂĆ
āōČāijŽēčnæŦ;āĹŦŦēš;ēāĹčŽDār;ēČĹāĂĆāržāžŮāÿÄäÿlāūščžRā■ŸāIJĹčŽDēŦōčŽDēĠāđ'■èŦNāĀijäÿ■āijŽæŦ

ēIJĚèēAæšĹæĐRčŽDæŸŦiijNäÿÄäÿl OrderedDict čŽDād'ğārRæŸŦäÿÄäÿlæŽōēĂŽā■ŮāĚÿčŽDäÿd'ā
æL'ĂāžēāēČæđIJā;äèēAæđĐāžžäÿÄäÿlēIJĚèēAād'ğēĠŦ OrderedDict
āōđä;NčŽDæŦræ■ōčžšæđĐčŽDæŮūāĂŽiijLærŦāēČēržāRŮ 100,000
ēāN CSV æŦræ■ōāLŦäÿÄäÿl OrderedDict āLŮēāĹäÿ■āŮžiijL'iiijN
ēČčāžĹā;āāršā;ŮāžŦčžĚæĹČēāāÿÄäÿNæŸŦāŦēā;ĚčŦí OrderedDict
āÿææĹēčŽDāē;ād'ĐēēAād'ğēŁĠēčĹād'ŮāĚĚā■ŸæŮĹēĂŮčŽDā;šāš■āĂĆ

3.8 1.8 āUāĖŸçŽĐēRčŮ

ēŮŏēčŸ

æĀŒæăăăIJāŦŕæŋŏăŮāĖŸăŸæLġēāŦăŸĀăžŽēŏaçŏŮæŞăĬJĭijĹăŕŦăĕĆăśĆăIJĀăŕŔăĀĭjăĀĀæIJĀă

èġčāĖşæŮzæąĹ

èĀČēŽŚăŸŦēĬčçŽĐēĆăĉĹăŔăŦŦăžŮăăĭjăŸăăŕĐăŮāĖŸŕĭjŽ

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

ăŸžăžĖăŕžăŮāĖŸăĀĭjăLġēāŦēŏaçŏŮæŞăĬJĭijŦēĀŽăŸŸéIJĀēĕĀăĭĤčŦĭ zip()
ăĠjăŦŕăĖĹăŕĖĕŦŏăŦŦăĀĭjăŔăĕĭŦēĤĠăĹăăĀĆăŕŦăĕĆĭijŦăŸŦēĬčæŸŕăşæLĹăIJĀăŕŔăŦŦăIJĀăđġēĆăĉĹă

```
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')
```

čşžăĭĭjçŽĐĭjŦăŦŕăžăăĭĤčŦĭ zip() āŦŦ sorted()
ăĠjăŦŕăĹăăŒăŦŮăŮāĖŸæŦŕæŋŏĭjŽ

```
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')]
```

æLġēāŦēĤŽăžŽēŏaçŏŮçŽĐæŮăăĀŽĭijŦēIJĀēĕĀăşĹăĐŦçŽĐæŸŕ zip()
ăĠjăŦŕăĹŽăžžçŽĐæŸŕăŸăăŸĹăŦēĤĹēŏĹéŮŏăŸăăŦăçŽĐēĤăăžčăŽĹăĀĆăŕŦăĕĆĭijŦăŸŦēĬčçŽĐăžčçăĀăŕşăĭjŽăžġçŦŦŦéŦŽēŦŕĭjŽ

```
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
```

èŏĹēŏž

ăĕĆăđIJăăăIJăŸăăŸĹăŮāĖŸăŸLæLġēāŦăŽŏéĀŽçŽĐæŦŕăăĕĤŦçŏŮĭijŦăăĭjŽăŦŦçŦŦŦŦăŦŦăžĖăž

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

æŹäyŹçzŞædIJāzūāy■æYřā;āæÇşèeAçZĐrijNāZāyžā;āæÇşèeAāIJlā■UāĖyçZĐāĀijēZEāRĹāyLæL'gèa
æĹŪēōyā;āāijZārĹerTçlĀā;ŹçTlā■UāĖyçZĐ values() æŪzæşTæĹēēğçāEşşēfZāyŹēUōécYrijZ

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

āy■āzŹçZĐæYrijNēĀZāyŹēfZāyŹçzŞædIJāRŹæūāzşāy■æYřā;āæÇşèeAçZĐāĀC
ā;āāRřēC;ēfYæÇşèeAçşēēAşşāřzāzTçZĐēTōçZĐāfāæAřijLærTāeCēCççg■ēCāçēlāzūāāijæYřāIJāā;ŌçZĐ

ā;āāRřāzēāIJl min() āŞN max() āG;æTřāy■æRŘā;Z key
āG;æTřāRČæTřāĹēēŌūāRŪæIJāāRāĀijæĹŪæIJāād'gāĀijārzāzTçZĐēTōçZĐāfāæAřāĀCærTāeCrijZ

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

ā;EæYrijNāēCædIJēfYæÇşèeAā;UāĹræIJāāRāĀijrijNā;āāRĹā;UāL'gèaŹāyĀæñæşēæL'çæş■ā;IJāĀ

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

āL■ēĹççZĐ zip() āG;æTřæŪzæāĹēĀZēfGārEā■UāĖyāĀĹāR■ē;ñāĀĹāyž
(āĀijrijNēTō) āĖČçzĐāzRāĹŪæĹēēğçāEşşāzEāyLēfřēUōécYāĀC
ā;ŞærTē;Čāy'd'āyĹāĖČçzĐçZĐæŪūāĀZrijNāĀijāijZāĖĹēfZēāNærTē;ČrijNçDūāRŌæL■æYřēTōāĀC
ēfZæāūçZĐērĹā;āārşēC;ēĀZēfGāyĀæĹççōĀā■TçZĐēr■āRēārşēC;ā;Ĺē;zæĹçZĐāōđçŌrāIJlā■UāĖyāyĹçZĐ

ēIJāēeAæşĹæĐRçZĐæYřāIJĹēōaçōŪæş■ā;IJāy■ā;ŹçTlāĹrāzE (āĀijrijNēTō)
ārzāĀCā;Şād'ZāyĹāōđā;ŞæNēæIJĹçZyāRŹçZĐāĀijçZĐæŪūāĀZrijNēTōāijZāEşşāōZēfTāZđçzŞædIJāĀC
ærTāeCrijNāIJlæL'gèaŹ min() āŞN max() æş■ā;IJçZĐæŪūāĀZrijNāēCædIJæAřāūgæIJāāRāĹŪæIJāād'

```
>>> 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 æşēæL'çāy'd'ā■UāĖyçZĐçZyāRŹçZ

ēUōécY

æĀŌæūāIJlāy'd'āyĹā■UāĖyāy■ārzāræL'çZyāRŹçCzrijLærTāeCçZyāRŹçZĐēTōāĀAçZyāRŹçZĐāĀi

ēğçāEşşæŪzæāĹ

ēĀČēZŞayNēĹçāy'd'āyĹā■UāĖyrijZ

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

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

āyžāžEāřzæL'äyd'äyġā■ŮāĚyčŽDčZyāRŇčCzījNāRřāžēčŮĀā■ŤčŽDāIJlāyd'ā■ŮāĚyčŽD
 keys() æLŮēĀĚ items() æŮzæšŤēŤāŽdčzšædIJāyŁæL'gëāNēZEāŘLæš■ā;IJāĀCærŤāēCīijŽ

```
# 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) }
```

èŤŽāžZæš■ā;IJāžšāRřāžēčŤlāžŌāŋŌæŤzæLŮēĀĚēŤGæzd'ā■ŮāĚyāĚČt'āāĀC
 æŤāēCīijNāĀGāēČā;āæČšāžēčŌřæIJL'ā■ŮāĚyædĎēĀāyĀāyġæŌŠēŽd'āGāyġæNĠāŋŌŽēŤŏčŽDæŮřā■ŮāĚ
 äyNēlĀl'čŤlā■ŮāĚyæŌlārijælēāŋŋčŌřēŤZæāūčŽDēIJāæšCīijŽ

```
# 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}
```

ēŏlēŏž

āyĀāyġā■ŮāĚyāřsæYřāyĀāyġēŤŏēZEāŘLāyŌāĀijēZEāŘLčŽDæYāārDāĚšçšāĀC
 ā■ŮāĚyčŽD keys() æŮzæšŤēŤāŽdāyĀāyġāšŤčŌřēŤŏēZEāŘLčŽDēŤŏēĠāZ;āržēsāāĀC
 ēŤŏēĠāZ;čŽDāyĀāyġā;LārŠēcnāžEēġččŽDčL'zæĀgāřsæYřāŋŌČāžnāžšæŤřæNāēZEāŘLæš■ā;IJīijNārŤāēC
 æL'ĀāžēīijNāēČædIJā;āæČšāržēZEāŘLčŽDēŤŏæL'gëāNāyĀāžZæŽŏēĀŽčŽDēZEāŘLæš■ā;IJīijNāRřāžēčŽt'
 setāĀC

ā■ŮāĚyčŽD items() æŮzæšŤēŤāŽdāyĀāyġāNēĀRŋ (ēŤŏīijNāĀij)
 āřzčŽDāĚČt'āēĠāZ;āržēsāāĀC èŤZāyġāržēsāRŇæāūāžšæŤřæNāēZEāŘLæš■ā;IJīijNāzūāyŤāRřāžēēčŋčŤ

āř;čŏāā■ŮāĚyčŽD values() æŮzæšŤāžšæYřçšzāijīijNā;EæYřāŋŌČāzūāy■æŤřæNāēŤēŽēĠNāžNçzç
 æšRçġç■ĠNāžēāyŁæYřāZāyžāĀijēĠāZ;āy■ēČ;āŤlērAæL'ĀæIJL'čŽDāĀijāžŠāy■čZyāRŇīijNēŤZæāūāijZār
 āy■ēŤĠīijNāēČædIJā;āçāñēēĀāIJlāĀijāyŁēlĀcæL'gëāNēŤZāžZēZEāŘLæš■ā;IJčŽDērīijNā;āāRřāžēāĚLārEāĀ
 setīijNçĎūāRŌāE■æL'gëāNēZEāŘLēŤRçŏŮāršēāNāžEāĀC

3.10 1.10 aLaeZd'azRaLUcZYaRNaeCct'aaazuäxiæNAeazazR

éUóécY

æAÖæuâaIJläyÄäylâzRâLUäyLéícafiæNæAeCct'æeazazRçZDâRNæUüæüLéZd'égad'■çZDâAijijš

ègčāEşæÚzæqL

æeCædIJâzRâLUäyLçZDâAijéCjæYřhashable çszâdNijNéCczLâRfäzēâjLçōĀa■TçZDâL'çTléZEæ

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

äyNéícaYřäjçTlâyLèfřaGjæTřçZDäjNâ■RijjZ

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

èfZäylæÚzæşTâzEäzEâIJläzRâLUäy■æCct'äayž hashable
çZDæUüâÄZæL'■çōæçTlâĀC æeCædIJäjäæCşæüLéZd'æCct'äay■âRfâŞLâyNijjLærTæC
dict çszâdNijjLçZDâzRâLUäy■éGad'■æCct'æçZDèřijjNä;æeIJĀeçAârEäyLèfřäzççāAçl■âjōæTzâRÿäy/

```
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)
```

èfZéGNçZDkeyâRCæTřæNĠaōZäzEäyÄäylâGjæTřijjNârEäzRâLUäCct'æèjñæ■cæLR
hashable çszâdNâĀCäyNéícaYřaōCçZDçTlæşTçd'žäjNijjZ

```
>>> 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}]
>>>
```

æeCædIJäjäæCşâşžäzŌa■Täylâ■UæōtāĀĀâsđæĀgæLŪëĀĒæşRäylæZt'äd'gçZDæTřæ■ōczŞædDæIææü

èõléõž

âĉĈæđĪä;ăăžĚăžĚăřsæŸræĈşæŭLéŽđ' éĜ■ăđ' ■ăĚĈĉt' āiijNéĂŽăyyăRřăžěĉôĂă■TĉŽĐæđĐéĂăyŸĂăylē

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

ĉĐûēĂŇiijNēĤŽĉğ■æŨzæşTăy■ēĈ;ĉzt' æŁđ' âĚĈĉt' âĉŽĐéąžăžRiijNĉTşæĹŖĉŽĐĉzşæđĪăy■ĉŽĐăĚĈĉt'

ăĪĹæĪJnēĹĈăy■æĹŚăznă;ĤĉTĹăžĖĉTşæĹŖăŽĹăĜ;æTřèol' æĹŚăznĉŽĐăĜ;æTřæŽt' âĹăéĂŽĉTĹiijNăy■ăžĹ
ærTăēĈiijNăēĈæđĪä;âæĈşērŷăRŨăyŸĂăylæŨĜăžŭiijNæŭLéŽđ' éĜ■ăđ' ■ăqNŭiijNă;ăăRřăžěă;ĹăôžæŸ

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

ăyĹēĤrkeyăĜ;æTřăŖĈæTřăĹăžĤăžĖ sorted() , min() âŠŇ max()
ĉ■ĹăĖĖĉ;ôăĜ;æTřĉŽĐĉŽyăiijăĹşēĈ;ăĂĈ âŖřăžěăŖĈēĂĈ 1.8 âŠŇ 1.13
ăŖŖēĹĈăžĖĉæŽt'ăđ'ŽăĂĈ

3.11 1.11 âŚ;ăŖ■ăĹĜĉĹ'Ĝ

éŮôécŸ

âĉĈæđĪä;âĉŽĐĉĹNăžŖăŇĚăŖnăžĖăđ' ĝéĜŖæŮăæşTĉŽt' ēĝĖĉŽĐĉăñĉijŮĉăĂăĹĜĉĹĜiijNăžŭăyTă;ăæĈ

ēĝĉăĖşæŨzæąĹ

ăĂĜăôŽă;ăēĖĂăžŮăyŸĂăylēôŖă;TřijĹærTăēĈæŨĜăžŭăĹŮăĖŭăžŮĉşăiijjæăijăiijŖiijĹăy■ĉŽĐæşŖăžŽăž

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

ăyŮăĖŭéĈĉæăăăĖŽiijNăyžăžĂăžĹăy■æĈşēĤZæăăăŚ;ăŖ■ăĹĜĉĹĜăŚĉiijŽ

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

ăĪĹēĤŽăyĹĉĹĹæĪJnăy■iijNă;ăéĂăăĖ■ăžĖă;ĤĉTĹăđ' ĝéĜŖéŽ;ăžēĉŖĖēĝĉĉŽĐĉăñĉijŮĉăĂăyŸNăăĜăĂĈēĤ

ēōlēōž

āyĀēĻāēēōšīijNāzččāAāy■āēCādIJāGžčŌřad'gēGRčŽDčañčijŪčāAāyNāāGāijŽā;ā;ŪāzččāAčŽDā
ærTāēČīijNāēCādIJā;āāZđēfGāēčIJNčIJNāyĀāzt'āL'■ā;āāEŽčŽDāzččāAīijNā;āāijZāSŷčĀēDŠēcNāČšēČ
ēfZāYřāyĀāyĻā;ĻčōĀā■TčŽDēgčāEšāēŪzāēĻīijNāōČēōl'ā;āēZt'āLāēyĒāēZřčŽDēāēē;āzččāAčŽDčŽōčŽD

āEĒč;ōčŽD slice() āG;āTřāĻZāzžāzEāyĀāyĻāĻGčL'GāržēšāāĀČāēL'ĀēIJL'ā;ēčTĻāĻGčL'GčŽDāIJřā

```
>>> 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]
```

āēCādIJā;āāēIJL'āyĀāyĻāĻGčL'GāržēšāāīijNā;āāRřāzēāĻēāĻnērČčTĻāōČčŽD a.start
, a.stop, a.step āsđāēGāēēēōūāRŪāēZt'ād'ŽčŽDāēāēAřāĀČærTāēČīijŽ

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

āRēād'ŪīijNā;āēfYāRřāzēēĀŽēfGērČčTĻāĻGčL'GčŽD indices(size)
āŪzāēTārēāōČāYāārDāĻřāyĀāyĻāūščšēād'gārRčŽDāžRāĻŪāyĻāĀČ
ēfZāyĻāēŪzāēTēfTāZđāyĀāyĻāyL'āēČčzD (start, stop, step)
īijNāēL'ĀēIJL'čŽDāĀijēČ;āijZēcñčijl'ārRīijNčZt'āĻrēĀČāRĻēfZāyĻāūščšēāžRāĻŪčŽDē;žčTŇāyžā■āĀČ
ēfZāēāūīijNā;ēčTĻčŽDēŪūārsāy■āijZāGžčŌř IndexError āijČāyŷāĀČærTāēČīijŽ

```
>>> 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 āžŖāĹŪäy■āĞžçŎŕæñæŧŕæĴĀāđ'ŽçŽĐāĚČŧ'ā

ēŬōécŸ

æĀŎæăæĹŷāĞžäyĀäyĹāžŖāĹŪäy■āĞžçŎŕæñæŧŕæĴĀāđ'ŽçŽĐāĚČŧ'āāŚcīijŝ

èğĉāĒşæŪzæąĹ

collections.Counter çşzārŝæŸŕäyŞēŬāyžèŁŽçşzéŬōécŸĖĀŃèŏĹèŏaçŽĐīijŃ
āŏČçŤŽèĞşæĴĴäyĀäyĹæĴĴçŤĴçŽĐmost_common() æŪzæşŧçŽŧ'æŎëçžŽāžĒāĵăç■ŤæąĹāĀĆ
äyžāžĒæijŧçđ'žīijŃāĒĹāĀĞèŏĹăĵæĴĴäyĀäyĹā■ŧèŕ■āĹŪèąĹāžŭäyŧæČşæĹŷāĞžāŞĹäyĹā■ŧèŕ■āĞžçŎŕæ

```
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)  
# āĞžçŎŕēćŝçŎĞşĴĴĒēŃŸçŽĐ3äyĹā■ŧèŕ■  
top_three = word_counts.most_common(3)  
print(top_three)  
# Outputs [('eyes', 8), ('the', 5), ('look', 4)]
```

èŏĹèŏž

ăĴĴäyžèŁŞāĒēīijŃ Counter āržèşąāŖŕäžæŎēāŖŬāzzæĐŖçŽĐçŤşāŖŕāŞĹäyŃīijĴhashableīijĴāĚČ
āĴĴāžŧŕāsĈāŏđçŎŕäyĴīijŃäyĀäyĴ Counter āržèşąārŝæŸŕäyĀäyĹā■ŬāĒŸīijŃārĒāĒĚČŧ'āæŸāārĐāĹŕāŏĈāĞžç

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

āēĆæđĴĴāĵæČşæĹŃāĹĹăĉđāĹăèŏaçŧŕīijŃārŕäžèçŏĀ■ŧçŽĐçŤĹāĹăæşŧīijŽ

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

æŁŨèĂĚä;ăăŔŕäzëä;£çŦÍ update () æŨzæşŦiijŽ

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

Counter ǎoḑä; NäyÄäylēsIJäyžāzžçšēcŽĎČL'zæĂgæŸrăoČăznăRărăzēja; ŁăoźæYŠcŽĎeușæTră■eēfRč

```
>>> 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})
>>>
```

æríæUáčŮSéUőiiĴÑ Counter áržèsəǎIJlăGăázŌæL'ĂæIJL'éIJĂeəAăLűeəłŁŬěĂĚèőəæȚræȚra■óçŽI
ǎIJlégčĂŞesēZčszėUőécYćŻDæUŭăĂZă;ăăzTêrëäijYăĒLéĀL'æNl'aôČiiĴNěĂNăy■ăYřæL'NălĲčŽDăĹl'čTlă

3.13 1.13 éĀŽèŁĠæšŘäyİäĚšéŤōā■ŬæŌŠăžŘäyÄäyİā■ŬāĚyāĹŬēāİ

éŮőécŸ

ä:äæIJLäyÄäy!a■ÜaËy!LÜe!aiijNä;äæČšæāzæ■ōæšŘäy!æLŮæšŘ!aGääy!a■ÜaËy!a■Üæō!æ!ææŌš!z!æ!

èġčǎẸșæŮźæǻŁ

éÅžēfGä;ŁçŦí operator ælɑɑiUçŽĎ itemgetter
 āĠ;æŦriijŊāŦrāzēēlāyŷāōzæŸŞçŽĎæŌŖāzŦēfZæāũçŽĎæŦræ■ōçzŞædĎāĀĆ
 āAĠēō;ā;āāzŌæŦræ■ōāŖSäy■æĀĀŦĉāGzæİēç;ŞçŋŽāijŽāŖŸāfæAŦāLŪēālīijŊāzūāyŦāzēāyŊāLŪçŽĎæŦræ

```
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}
]
```

æāžæ■ōāzzæĐŔçŽĐā■ŪāĔÿā■ŪāōŧæĬææŎŠāžŔèçŠāĔĕçžŠæđĬJèāŊæŸŕāçĹāōžæŸŠāōđçŎŕçŽĐĭjŊāžç

```
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)
```

äžçăAçŽĐèçŠāĔžæÇäŸŊĭjŽ

```
[{'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() āĢĭæŦŕāžšæŦŕæŊĀāđ'ŽāŸĭ keysĭjŊæŕŦæÇäŸŊéĬçŽĐäžçăA

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

äĭjŽäžğçŦšæÇäŸŊçŽĐèçŠāĔžĭjŽ

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

ēōlēōž

āĬĬāŸĹéĬcäŊā■ŔāŸ■ĭjŊ rows ècŋāĭjæĀŠçžŽæŎēāŦŪāŸĀāŸĹāĔŖéŦōā■ŪāŦŦæŦŕçŽĐ
sorted() āĔĔç;ōāĢĭæŦŕāĀČ èĔŽāŸĹāŦŦæŦŕæŸŦ callable çšāđŊĭjŊāžŸāŸŦāžŎ rows
āŸ■æŎēāŦŪāŸĀāŸĹā■ŦāŸĀāĔÇçŦ āĭjŊçĐŸāŦŦēĔŦāžđècŋçŦĬæĬæŎŠāžŔçŽĐāĭjāĀČ
itemgetter() āĢĭæŦŕāŦŖæŸŦŕèŦ šèŦ çāĹāžžèĔŽāŸĭ callable āŕžèšaçŽĐāĀČ

operator.itemgetter() āĢĭæŦŕæĬĬ'āŸĀāŸĹècŋ rows
āŸ■çŽĐèŕāŦŦçŦĬæĬæšæŦŦāĭjçŽĐçŦ çāĭŦŦāŦŦæŦŕāĀČāŦŦāžæŸŕāŸĀāŸĹā■ŪāĔŸéŦōāŦŦçğŦĭjŊ
āŸĀāŸĹæŦŦŦāĭjāĀĬŪèĀĔžāŦŦèçĭāđ šāĭjāĀĔēāŸĀāŸĹāŦžèšaçŽĐ __getitem__()
æŪžæšŦçŽĐāĭjāĀČ æÇæđĬāŦāĭjāĀĔēāđ'ŽāŸĭçŦ çāĭŦŦāŦŦæŦŦçžŽ itemgetter()

iijNǎoČčTšæLRčŽD callable áržesqaijŽeŁTāZđäyĀäylāNĒāRñæL'ĀæIJL'āĖČčt'āāĀijčŽDāĖČčzDīijN
 āzūāyT sorted() āGjæTīrijZæāzæāōēfZāylāĖČčzDāyāāĖČčt'āēāzāzRāŌzæŌšāzRāĀC
 ājEājāæČšēēAāRñæŪūāIJlāGāāylāāŪāōjāylĒlēlēfZēāNāŌšāzRīijLærTāēČēĀŽēfGāgšāSñāRāālēæŌšāz
 itemgetter() æIJL'æŪūāĀZāzšāRfāzēčTl lambda
 èāléççāijRāzčæŽēiijNærTāēČīijZ

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

æƧZçg■æŨzæŁŁäzšäy■éŦZāĀCā;EæYřijŇä;ƧçŦĬ
 æŨzāijRāijZēfRēaŇçZĎçĬ■ā;ōāfŇçCzāĀCāZāæ■d'ijŇāęCāđĬJä;āārzaĀğēC;ēęAæśCærTē;ČénYçŽĎeriārs
 itemgetter() æŨzāijRāĀC

æIJÅŘÖijÑäy■ēēAāfYāzEēfZēŁCäy■āsTçd'zçŽDæŁĀæIJřázšāŘÑæäüēĀĆčTlāžŎ
min() āŠÑ max() c■L'āĜjæTřāĀCærTāēĆijŽ

```
>>> 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 æŌŠăžRäy■æŤræŃAăŌšcŤšærŤè¿ĈĈŽĎăržèsa

éŮőécŸ

ä;äæČsæŎšăžŔčśzādŇčZyăŔŇčŽĎăržèsăijŇNă;EæŸrăzŰăznăy■æŦræŇNăăŎșcŦșcŽĎărŦtè;Čæ\$■ă;Iă

èġčăẸșæŮźæąŁ

âĖĖċ;ôçŽĎ	sorted()	âĖ;æŦræIJL'äyÄäylâĖſeŦôâĖUâĖĆæŦr	key
iijŊâĖŕäzèaiijââĖĖyÄäyl	callable	âržèsaçzŽâôĈiijŊ	èŦŽäyl callable
âržèsaâržæŕRäylaijââĖĖçŽĎâržèsaèŦŦâŽđäyÄäylâĖiijŊèŦŽäylâĖiijäijŽèçŋ			sorted
çŦlælæŦŦâžŕèŦŽäžŽâržèsaâĖĖĖ æŦŦæĈiijŊæĈæđIJâ;ââIJlâžŦçŦlçlŊâžŕèĖŊèlçæIJL'äyÄäyl			
User		âôđä;ŊâžŕâĖUijŊâžžäyŦä;âäyŊæIJŽèÄŽèŦĖâžŦžçŽĎ	
user_id	âŦđæÄĖèŦŽèaŊæŦŦâžŕiijŊ	â;ââŕŕäzèæŕŕä;ŽäyÄäyläzè	User
âôđä;Ŋâ;IJäyžè;ŦâĖèâžžè;ŦâĖžâržâžŦ		user_id	âĖijçŽĎ callable
âržèsaâĖĖæŦŦæĈiijŽ			

```
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))
```

Řeād' Ůäy Äçg ■ Ůzâij Ræ Yřä; fç Tí operator.attrgetter() ælëäzçæŽŁ lambda äĜ;æTřijŽ

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

ěölèőž

éĀL'æŇl'ä; fç Tí lambda äĜ;æTřæL ŮèĀĚæYř attrgetter()
 āRřèČ; āR ŮāEšāž Ůäy lāžžā ŮIJ äē; āĀĆ ä; Eæ YřijŇ attrgetter()
 äĜ;æTřéĀž äy yäij ŽèŁ RēāŇç ŽĎā fñç ČzrijŇāz ůäy TèŁ YèČ; āRŇæ ŮūāĚĀèōy ād' Žäy lā ■ ŮæōtèŁ ŽēāŇæf Tè; ČāĀ
 èŁ Žäy lè ůš operator.itemgetter() äĜ;æTřä; IJç Tí lāž Ůā ■ ŮāĚy çszāđŇā; Łçszäijrij Lā RČèĀĆ 1.13 āRē
 ä; Ňāē ČrijŇāē ČāđIJ User āōđä; ŇèŁ Yæ IJL' äy Ääy l first_name āŠŇ last_name
 āsđæ ÄĝrijŇéČčāž Lā Rřāžēā RŠäyŇé l çèŁ Žæ ůæ ŮŠāž RrijŽ

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

āRŇæ ůūē IJĀèēĀæš læ ĎRç ŽĎæ YřijŇèŁ Žäy ĀāR RēŁ Čç Tí lā Lřç ŽĎæ LĀæ IJřā RŇæ ůūē ĀĆç Tí lāž Ůā ČR
 min() āŠŇ max() äžŇçszç ŽĎā Ĝ;æTřāĀČæf Tāē ČrijŽ

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

3.15 1.15 éĀŽèŁĜæšRäy lā ■ ŮæōtāřEèōřā; Tā LĚçžĎ

éŮőéčY

ä; äæ IJL' äy Ääy lā ■ ŮāĚy æL ŮèĀĚāōđä; Ňç ŽĎāž Rā L ŮrijŇç Ďūā R Ůā; äæ Čšæ āžæ ■ Ůæš Rāy tç L' žāō Žç ŽĎā ■
 date ælēā LĚçžĎèŁ ■ äžçèōŁ éŮőāĀĆ

èĝčāEşæŮzæāL

itertools.groupby() äĜ;æTřār zāž ŮèŁ Žæ ůūç ŽĎæ Třæ ■ Ůā LĚçžĎæš ■ ä; IJé l dāy yāōđç Tí lāĀĆ
 äyžāž Eæij Tçđ' zrijŇāĀĜèō; ä; āā ůšçz Ræ IJL' äž EäyŇā L ŮçžĎā ■ ŮāĚy āL Ůèā lrijŽ

```
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'},
]
```

çŒŕaIJlâAĞèö;ä;äæÇşâIJlæNL date âLEçzĐaŔŒŒŽĐæŦŕæ■ŕaİUäyLèfZèaŦèf■äzçãĂCâyžăŦEèfZæăü.
date)æŒŒăžŔiijŦ çĐŭâŔŒŒŕÇçŦİ itertools.groupby() âĢ;æŦŕiijŽ

```
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)
```

èŦŔèaŦçzŞæđIJiijŽ

```
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'}
```

èŒlèŒž

groupby() âĢ;æŦŕæL'naŔŔæŦŦ'äyLăžŔâLŬăžüäyŦæşşæL;èŦđçz■çŽyâŔŦăŦiijŦLæLŬèĂĖæăžæ■ŕ
key âĢ;æŦŕèŦŦăŽđâĤijçŽyâŔŦiijŦçŽĐâĖÇçŦ'ăăžŔâLŬăĂC
âIJlæŦŕæŦæŦ■ăžççŽĐæŬŭăĂŽiijŦăŕŒÇaijŽèŦŦăŽđäyĤäyLăĤijăŦŦäyĤæŦ■ăžçãŽlŦŕžèşaiijŦ
èŦŽäyLèf■ăžçãŽlŦŕžèşâŦŔăžèçŦşæLŦŦâĖÇçŦ'ăăĤijăĖĖŦç■L'ăžŒäyLèĖŦçĖCăyLăĤijçŽĐçzĐäy■æL'ĂæIJL'âr

äyĤäyLèĖđäyŦŦēŦ■èçAçŽĐâĖĖđ'Ģæ■èĖđ'æŦŦŕèçAæăžæ■ŕăŦăŦŒăŦçŽĐâ■ŬăŕŦârEæŦŕæ■ŕæŒŒăžŔăĂ
ăŽăäyž groupby() äžĖăžĖæçĂæşşèŦđçz■çŽĐâĖÇçŦ'âiijŦăŕçCæđIJăžŦăĖĖLăžüæşæIJL'æŒŒăžŔăŕŦăŦŒŦŦç

æĈædĪĴāāzĒāzĒāRĴæYŕæĈsæāzæŋ date āUæŋŕæEæTŕæŋŋāLĒçzDāLŕäyÄäyĴad' ĝçŽDæTŕæŋŋçzS
 éĈçāzĴā;æĴĴāæĴā;ĴçTĴ defaultdict() æĴædDāzžäyÄäyĴad' ŽāĴijāUāĒyĴijNāĒsāzŌad' ŽāĴijāUāĒy
 1.6 āŕRĒLĈæĴĴ'èĴĠèŕççzĒçŽDāzNçzāĈæŕTāçĈijŽ

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

èĴŽæāüçŽDĒŕĴā;āāŕŕäzēā;Ĵè;žāĴçŽDāŕsèĈ;āŕzæŕŕäyĴæNĠāŋŽæUĒæĴĴsèŋŕĒUŋŕzāžTçŽDĒŕā;TŕijŽ

```
>>> 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'}
>>>
```

āĴĴäyĴĒĴçèĴŽäyĴā;NāŕäyĴijNæĴsāznæšqæĴĴ'āĴĒçAāĒLāŕĒçŕā;TæŌšāzŕāĈāZæd'ijNæĈæd
 èĴŽçĝæŪzāijŕāijŽæŕTāĒLæŌšāzŕçDūāŕŌāĒæĴŽèĴĠ groupby()
 āĠæTŕæfāzççŽDæŪzāijŕèĴŕèāNā;ŪāĴnāyÄāzŽāĈ

3.16 1.16 èĴĠæzd'āžŕāĴUāĒĈçt'ā

éŪŋéçY

āĴæĴĴ'äyÄäyĴæTŕæŋŋāzŕāĴUŕijNæĈsāĴŕçTĴäyÄāzŽèĠDāLŽāzŌäyæŕŕāŕŪāĠzéĴĴæçŽDāĴijæ

èĝçāĒsæŪzæāĴ

æĴĴçŌĀāTçŽDĒĴĠæzd'āžŕāĴUāĒĈçt'āçŽDæŪzæŕTāŕsæYŕä;ĴçTĴāĴUĒāĴæŌĴāŕijāĈæŕTāçĈijŽ

```
>>> 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]
>>>
```

ā;ĴçTĴāĴUĒāĴæŌĴāŕijçŽDäyÄäyĴæ;ĴāĴĴçijžéZūāŕsæYŕæĈædĴĒç;šāĒçĒĴdäyŕad' ĝçŽDæŪūāĴzāijŽāzĝç
 æĈædĪĴā;āāŕzāĒĒāYæŕTĒçTŕæDšijNĒĈçāzĴā;āāŕŕäzēā;ĴçTĴçTšæĴŕāZĴæĴç;āijŕèĴāzççāzĝçTšèĴĠ

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

(continues on next page)

(continued from previous page)

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

æIJLæUúāĀZijNēfGæzd'ēgDāLZæfTēġCād'■æICijNāy■ēČġčōĀā■TġŽDāIJlāLŪēāīæŌlārijæLŪēĀĒč
æfTāēČijNāAĠēōġēfGæzd'čŽDæUúāĀZēIJĀēēAād'ĐġREāyĀāžZāijCāyŷæLŪēĀĒāĒūāzŪād'■æICæČĒāĒē
čDūāRŌāġġTlāĒēĀāžžčŽD filter() āĠæTŕāĀČčd'žāġNāēCāyNijŽ

```
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() āĠæTŕāLZāžžāžĒāyĀāyġēf■āžčāZlīijNāZāæ■d'āēČædIJāġāæČšāġŪāLŕāyĀāyġāLŪēāġčŽDā
list() āŌžēġnæ■čāĀČ

ēōlēōž

āLŪēāīæŌlārijāŠNčTšæLŕāZlēāġēġāijRēĀŽāyŷæČĒāĒāyNāYŕēfGæzd'æTŕæ■ōæIJĀčōĀā■TġŽDæU
āĒūāōđāōČāžnēfYēČġāIJlēfGæzd'čŽDæUúāĀZēġnæ■čæTŕæ■ōāĀČæfTāēČijŽ

```
>>> 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]
>>>
```

ēfGæzd'æS■āġJčŽDāyĀāyġāRŸčg■ārsæYŕāfĒāy■čņēāRLæġāžžčŽDāĀijčTlāŪŕčŽDāĀijāžčæZēijNēĀ
æfTāēČijNāIJlāyĀāLŪæTŕæ■ōāy■āġāāRŕēČġy■āžĒæČšæLġāLŕæ■čæTŕijNēĀNāyTēfYæČšāfĒāy■æYŕæ■
ēĀŽēfĠāfĒēfGæzd'æġāžžāTġāLŕæġāžžēāġēġāijRāy■āŌžijNāRŕāžēāġLāōžæYščŽDēgčāĒēēfZāyġēŪōēčY

```
>>> 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]
>>>
```

```
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]
```

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

ɛƧZéGŇçŽĐăĖŞéTôçCzâIJlăžŎăĖĹăĹZăžzăŸĂăŸt Boolean
 ăžRăĹUîijNăĖŇğçď'žăŞlăžZăĖĈçť'ăçņăRĹăĹăžzŭăĂĈ ċĐúăŘŎ compress()
 ăĢ;æŦřæăžă■œɛŽăŸlăžRăĹUăŎžéĂĹ'æŇĹ'è;ŚăĢžăřzăžTă;■ç;őăŸž True ċŽĐăĖĈçť'ăăĂĈ
 ăŠŇ filter() ăĢ;æŦřçşăiijijijŇ compress() ăžşæŸřɛťTăZđçŽĐăŸĂăŸlɛ■ăžçăŽlăĂĈăZăæ■ď'ij
 éĈçăžĹă;ăéIJăëăĂ;ɛçŦĹ list() æĹăřEçşSăđIJē;ňæ■ăŸŷăĹUăəĹčşăđNăăĂĈ

éŮőécŸ

ä:äæĈsædĎDéĂăäÿĂäÿlā■ŮăĚÿiŋiNăŏĈæŸřăRéăd'ŮäÿĂäÿlā■ŮăĚÿcŽDă■ŘéZEăĀĈ

èġċaEşæŮzæaĹ

æIJĂċŃĂă■TċŽDæŮzâijRæYřăĵċTĹă■ŮăËyæŎĹărijăĂĈăerTăeĈrijŽ

```
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}
```

èŃŃŃŃŃ

ăd'ġăd'ŽăTřăĈĖăĖĵăyNă■ŮăËyæŎĹărijăĈĵăAŽăĹřċŽDrijNéĂŽĖĖĠăLŽăzžăyĂăyĹăĖĈċzDăžRăĹŮċDřt
dict() âĠăTřăžšĖĈĵăŃŃċŮăĂĈăerTăeĈrijŽ

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

ăĵEăYřijNă■ŮăËyæŎĹărijăŮzâijRăăĹăDăŽtăyĖăŽřijNăzŭăyTăŃŃĖŽĖăyĹăžšăijŽăĖRăăNċŽDăŽtă
rijĹăIJĹăĖŽăyĹăNă■Răy■rijNăŃŃĖŽĖăĵNĖrTăĠăăžŎăerTdict()
âĠăTřăŮzâijRăăĹăTăTăyĂăă■rijĹăĂĈ

æIJĹăŮŮăĂŽăŃNăĹRăRăNăyĂăzŭăžNăijŽăIJĹăd'ŽċġæŮzâijRăĂĈăerTăeĈrijNċnăžNăyĹăNă■RċĹN

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

ăĵEăYřijNăĖRăăNăŮŭĖŮtăĵNĖrTċzšădIJăYĵd'žĖŽċġæŮzăĹăd'ġăĖĈăerTċnăyĂċġæŮzăĹă
1.6ăĂăĂĈăĖĈădIJăřċĹNăžRăăĖRăăNăĂġĖĈĵĖĖAăśĈăerTĖĸĈĖnYċŽDĖrĹijNéIJĂĖĖAăĹśĈĈăŮŭĖŮtăăŎžă
ăĖşăžŎăŽtăd'ŽĖŃăŮŮăšNăĂġĖĈĵĖĖNĖrTrijNăRăăžăăRĈĖĂĈ14.13ăřRăĹĈăĂĈ

3.18 1.18 æYăărDăR■ċġăĹăřăžRăĹŮăĖĈċtă

éŮŃŃŃŃŃ

ăĵăæIJĹăyĂăŃŃĖĂŽĖĖĠăyNăăĠĖŃŃŃŃŃŮăĹŮăĹăĹŮăĂĖăĖĈċzDăy■ăĖĈċtăăċŽDăžċċăArijNăĵEăYřĖĖ
ăžŎăYřăĵăăĈşăĂŽĖĖĠăRăċġăĹăĖŃŃŃŃŃŮăĖĈċtăăăĂĈ

èġċaEşæŮzæaĹ

`collections.namedtuple()` aġġæTřéĀŽēĢăĴçTĹäyĀäylæŽōéĀŽçŽDăĔĈçzDărzèşæİēāyōāĵā
ēĴZäylăĢġæTřăōđēŽĔäyĹæYřäyĀäylēĴTăŽđ Python äy■æăĢăĢĔăĔĈçzDçşzăđNă■RçşzçŽDäyĀäylăüēăŌĈă
ăĵăēIJĀēēAăĵăēĀŞäyĀäylçşzăđNăR■ăŞNăĵăēIJĀēēAçŽDă■ŮăŋççzŽăŋĈĵĵNçDŭăRŌăŋĈăřsăĵŽēĴTăŽđäyĀ
ăžççăAçđ'žăĴNĵĴ

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

ărĵçŋ namedtuple çŽDăŋđăĴNçIJNēĵăİēăĈRăyĀäylæŽōéĀŽçŽDçşzăŋđăNĵĵNăĴæYřăŋĈēŋşăĔĈ
ærTăēĈĵĴ

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

ăŞġăR■ăĔĈçzDçŽDäyĀäyläyžēēAçTĹéĀTăYřărĔăĵçŽDăžççăAăžŌăyNăăĢăŞ■ăĴăy■ēġçēDşăĢžæİē
ăŽăă■đ'ĵĵNăēĈăđIJăĵăžŌăTřă■ŋăžŞērĈçTĹäy■ēĴTăŽđăžĔäyĀäylăĴĹăđ'ġçŽDăĔĈçzDăĴŮēăĴĵĵNēĀŽēĴĈ
ăĴŞăĵăăIJĹēăĴăy■ēŭăžăĹăăžĔăŮřçŽDăĴŮçŽDăŮŭăĀŽăĵçŽDăžççăAăRřēĈĵăřsăĵŽăĢžēTŽăžĔăĀĈăĴæYřă
äyžăžĔēřĴæYŌăyĔăēŽĵĵNăyNēĹăēYřăĴçTĹæŽōéĀŽăĔĈçzDçŽDăžççăĀĵĵŽ

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

äyNăăĢăŞ■ăĴĴéĀŽăyĵăĵŽēŋ'ăžççăAēăĹăĎRăy■ăyĔăŽřĵĵNăžŭăyTēĹđăyŷăĴĹēŮēŋăĴçŽDçzŞăđDă
äyNēĹăēYřăĴçTĹăŞġăR■ăĔĈçzDçŽDçĴĹăIJNĵĴ

```
from collections import namedtuple

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

(continues on next page)

(continued from previous page)

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

ěóľěőž

āš;āŘ■āĚĈčzDāRēäyÄäyłçŦléĀŦārsæŸřā;IJāyžā■ŪāĚyçŽDæZĚāzčīijNāZāāyžā■ŪāĚyā■ŸāĆléIJĀēēA
āēĆāđIJā;āēIJĀēēAāđDāzžāyÄäyłēīdāyŷāđ'gçŽDāNĚāRnā■ŪāĚyçŽDæŦræ■ōçzŠāđDīijNēĆčāzLā;ŁçŦlāŠ
ā;EāŸřēIJĀēēAāšlāĐRçŽDæŸřīijNāy■āČRā■ŪāĚyēĆčæāūīijNāyÄäyłāŠ;āŘ■āĚĈčzDæŸřāy■āRfæZŦ'æŦžç

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

āēĆāđIJā;āçIJšçŽDēIJĀēēAāŦžāRŸāśdæĀgçŽDāĀijīijNēĆčāzLāRřāzēā;ŁçŦlāŠ;āŘ■āĚĈčzDāōđā;NçŽ
_replace() æŰžæšŦīijN āōČāijZāLZāzžāyÄäyłāĒlæŰřçŽDāŠ;āŘ■āĚĈčzDāzūārEāržāzŦçŽDā■ŪāōŦçŦlā

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

_replace() æŰžæšŦēŁŸæIJL'äyÄäyłā;ŁæIJL'çŦlçŽDçL'žæĀğārsæŸřā;Šā;āçŽDāŠ;āŘ■āĚĈčzDæNē
āōČāŸřāyÄäyłēīdāyŷāēŰžā;ŁçŽDāānāĒĚæŦræ■ōçŽDæŰžæšŦāĀĆ
ā;āāRřāzēāĒLāLZāzžāyÄäyłāNĚāRnçijžçIJĀāĀijçŽDāŌšāđNāĚĈčzDīijNçDūāRŌā;ŁçŦl
_replace() æŰžæšŦlāLZāzžāēŰřçŽDāĀijēčnæZŦ'æŰřēŁGçŽDāōđā;NāĀČærŦāēČīijŽ

```
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)
```

äyNēlčæŸřāōČçŽDā;ŁçŦlāēŰžæšŦīijŽ

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

æIJĀāRŌēēAērt'çŽDæYřijNāēCædIJā;ăçŽDçŽōæăGæYřăōŽăzL'ăyĂăyléIJĀēēAæŽt'æŪřă;Ĺăd'Žăōdă,
 èfŽæŪŭăĂŽă;ăăžTēřēēĂCēŽSăōŽăzL'ăyĂăylăNĒăRń _____slots_____
 æŪzæşTçŽDçşzřijĹăRĈCēĂĈ8.4ărRēĹĈřijĹăĂĈ

3.19 1.19 è;ñæ■căzúăRŊæŪúèőăçőŪæTřæ■ő

éŪőécŸ

ă;ăéIJĀēēAăIJĹăTřæ■őăžRăĹŪăyĹăL'gēăNèAžÉŽĒăG;æTřijĹăřTăēĆ sum(), min()
 , max() řijĹ'řijŊă;ĒăYřēēŪăĒĹă;ăéIJĀēēAăĒĹē;ñæ■căĹŪēĂĒēēGæzd'æTřæ■ő

èğçăĒşæŪzæăĹ

ăyĂăyléidăyřaijYéŽĒçŽDæŪzăijRăŌžçzŞăRĹăTřæ■őèőăçőŪăyŌē;ñæ■căřsæYřă;ĒçTĹăyĂăylçTşăĹŊ
 æřTăēĈřijNăēCædIJă;ăăCşèőăçőŪăzşæŪzăŊřijŊăRřăzēăĈRăyŊéĹcēēfŽæăŭăĂŽřijŽ

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

ăyŊéĹcæYřæŽt'ăd'ŽçŽDă;Ŋă■ŘřijŽ

```
# 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)
```

èóìèõž

äyŁéİćçŽĐçd'žäŁNāŘŚä;äæijŦčd'žäžEā;ŞçŦŦşæŁŘāŽİēāİē;āijRā;IJäyžäyÄäyİā■ŦçNñāRCæŦräijäēÄŞç
æŦŦæĈijNäyNéİćçŽäžZer■āRēæYŦç■LæŦŁçŽĐijŽ

```
s = sum((x * x for x in nums)) #
→æŸ;āijŦçŽĐäijäēÄŞçÄäyİçŦŦşæŁŘāŽİēāİē;āijRāŦžēšā
s = sum(x * x for x in nums) #
→æŽt'āŁäāijŸēŽĖçŽĐāōđçŦŦæŰžāijŦiijNçIJAçŦēäžEæNñāŦŰ
```

ä;ŁçŦİäyÄäyİçŦŦşæŁŘāŽİēāİē;āijRā;IJäyžāRCæŦräijŽæŦŦāĒLāŁžāžäyÄäyİäyŦ æŰŰāLŰēāİæŽt'āŁäæŦ
æŦŦæĈijNäçCæđIJä;ääy■ä;ŁçŦİçŦŦşæŁŘāŽİēāİē;āijŦçŽĐŦiijNä;āāŦŦēĈ;āijŽēÄĈēŽŚä;ŁçŦİäyNéİćçŽĐā

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

èŁŽçğ■æŰžāijRāŦNæāŰāŦŦāžēē;āŁŦŦæĈşēçAçŽĐæŦŁæđIJijNä;EæŸŦāōĈāijŽād'ŽäyÄäyİā■ēēİd'ijNä
āržāžŦārŦādNāLŰēāİāŦŦēĈ;æşāžÄāžLāĒşçşžiiijNä;EæŸŦæCæđIJāĒĈçŦ'āæŦŦŦēĠŦēİđäyŸād'ğçŽĐæŰŰāÄŽ
āōĈāijŽāLŽāžäyÄäyİāŰāđ'ğçŽĐäžÄäžĒēēñä;ŁçŦİäyÄæñāŦŦēēñäyĈāijĈçŽĐäyŦ æŰŰæŦŦæ■ōçžŞæđĐāĈē.

āIJİā;ŁçŦİäyÄäžŽēAŽēZEāĠ;æŦŦæŦŦæĈ min() āŠŦ max()
çŽĐæŰŰāÄŽā;āāŦŦēĈ;æŽt'āŁāā;āŦŦāžŦā;ŁçŦİçŦŦşæŁŘāŽİçL'ŁæIJñijN
āōĈāžñæŦēāŦŰçŽĐäyÄäyİ key āĒşēŦŦā■ŰāŦŦæŦŦæŦŦŦēōyāržä;āā;ŁæIJL'äyŦāŦŦāĈ
æŦŦæĈijNāIJİäyŁéİćçŽĐŦŦāŁŸä;Nā■Ŧäy■ijNä;āāŦŦēĈ;āijŽēÄĈēŽŚäyNéİćçŽĐāōđçŦŦçL'ŁæIJñijŽ

```
# 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 āŦŁāžŰād'Žäyİā■ŰāĒŸæŁŰæŸāārĐ

éŰōēčŸ

çŦŦāIJİæIJL'ād'Žäyİā■ŰāĒŸæŁŰēĀĒæŸāārĐijNä;æçşārEāōĈāžñāžŦēĀžē;ŚäyŁāŦŁāžŰäyžäyÄäyİā■
æŦŦæĈæşēæL;āāijæŁŰēĀĒæĈĀşēæşŦāžŽēŦŦæŸŦāŦē■ŸāIJİāĈ

èğĈāĒşæŰžæāŁ

āĀĠæĈā;æIJL'æĈĈäyNäyđ'äyİā■ŰāĒŸ:

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

`a` `äy■æL;üijÑæČædIæL;äy■āŁřāE■āIĲ` `b` `äy■æL;üijL'āĀĆ`
`äyÄäyŁēİdäyÿçōĀā■TçŽĐēğčāEşæŪzæāŁārsæYřā;ŁçTĲ` `collections` `æĲāāĲŪäy■çŽĐ`
`ChainMap` `çşzāĀĆæřTāçCüijŽ`

```
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)
```

èõłèõž

`äyÄäyĲ ChainMap æŌēāRŪād'ŽäyĲā■ŪāĒyāzūārEāōČāznāIĲéĀzè;ŚāyŁāRŲāyžāyĀäyĲā■ŪāĒyāĀĆ`
`çĐūāRŌüijÑēŁŽāžŽā■ŪāĒyāzūāy■æYřçIJşçŽĐāRĲāzūāIĲäyĀēĲūāžEüijÑ` `ChainMap`
`çşzāRĲāYřāIĲāEĒēČĲāĲŽāžzāžEäyĀäyĲāōžçşēŁŽāžŽā■ŪāĒyççŽĐāĲŪēāĲ`
`āžūēĠæŪřāōŽāžL'āžEäyĀāžŽāyÿēğAççŽĐā■ŪāĒyæŞ■ā;IJæĲēēA■āŌEēŁŽäyĲāĲŪēāĲāĀĆād'ğēČĲāĲEā■ŪāĒ`

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

`āēČædIĲāĠçžçŌřēĠ■ād'■ēTŌüijÑēČčāžĲçññäyĀæñāāĠçžçŌřçŽĐæYāārĐāĀijāijŽēčñēŁTāžđāĀĆ`
`āŽāæ■d'üijÑā;Nā■RçĲNāžRāy■çŽĐ` `c['z']` `æĀžæYřāijŽēŁTāžđā■ŪāĒy` `a`
`äy■ārřāžTçŽĐāĀijüijNēĀNäy■æYř` `b` `äy■ārřāžTçŽĐāĀijāĀĆ`

`ārřāžŌā■ŪāĒyççŽĐæŽt'æŪřæĲŪāĲāēŽd'æŞ■ā;IJæĀžæYřā;śāŞ■ççŽĐæYřāĲŪēāĲāy■çññäyĀäyĲā■ŪāĒyā`

```
>>> 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` `ārřāžŌçijŪçĲNēr■ēĲĀäy■ççŽĐā;IJçTĲēNČāžt'āRŲēĠRüijŁæřTāçĆ`
`globals` `,` `locals` `ç■ĲüijŁæYřēİdäyÿæIJĲçTĲççŽĐāĀĆ`
`āžNāōđäyĲüijNæIJĲäyĀāžŽæŪzæşTāRřāžēā;ŁāōČāRŲā;ŪçōĀā■TüijŽ`

```

>>> 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})
>>>

```

ChainMap `update()`
 æŰæšTärEäyd'äylä■ŰäËÿäŔĹāzūāĀĆærTäęĆiiJŽ

```

>>> 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
>>>

```

èfZæäüäzšèČ;èaŇä;ŰéĀŽriiJŇä;EæŸrāōČéIJĀèęAä;āāĹZāzzäyĀäyĹāōŇāĒĹäy■āŔŇçŽDā■ŰäËÿärzèsar
 āŔŇæŰüriiJŇäęČædIJāŌšā■ŰäËÿāAŽāžEæŽt'æŰriiJŇèĒŽçg■æTžāŔŸäy■äijŽāŔ■āžTāĹræŰŕçŽDāŔĹāzūā■

```

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

```

ChainMap ä;ęçTĹāŌšæĹęçŽDā■ŰäËÿriiJŇāōČèĠāūsäy■āĹZāzzæŰŕçŽDā■ŰäËÿāĀĆæĹĀäzèāōČāzūāy■

```

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

```

(continues on next page)

(continued from previous page)

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

4 çñăžŇçñăiijŽă■ŮçñęäyśăŠŇăĹĜăĹĴ

ăĜăăžŮăĹĴăĹĴčŤĹčŽĎčĹŇăžŔéČ;ăiijŽăŮĹăĹĹăĹŔăśŔăžŽăĹĜăĹĴăĴăĎĎčŔĒiijŇăy■çŏăĹŔăğ
èĹŽăyĂçñăăŔĒéĜ■čŽăĒşăşĹăĹĜăĹĴŇčŽĎăş■ă;ĹĴăĎĎčŔĒiijŇăŔŤăĉĈăŔŔăŔŮă■ŮçñęäyśiijŇăŔĹĴčŤ
ăĎĎĝéĈĹăĹĒçŽĎéŮŏéčŸéČ;èČ;çŏĂă■ŤçŽĎĕŔčŤĹă■ŮçñęäyśçŽĎăĒăžăĹŮăşŤăŏŇăĹŔăĂĈ
ă;ĒăŸŔiijŇăyĂăžŽăŽŤăyžăĎ■ăĹčŽĎăş■ă;ĹĴăŔŔéČ;éĹĴăĎăĹă■čăĹŽăăĹē;ă;ăiijŔăĹŮăĒăăiijžăĎĝçŽĎĕğčă
ăžŮăyŤăĹĴăş■ă;ĹĴUnicodeăŮăăĂŽçčŔăĹŔçŽĎăyĂăžŽăčŸăĹŇčŽĎéŮŏéčŸăĹĴăĹŽéĜŇăžşăiijŽéčŇăŔŔăŔŮă

Contents:

4.1 2.1 ä;ĹçŤĹăĎŽăyĹçŤŇăŏŽçñęăĹĒăĹśă■Ůçñęäyś

éŮŏéčŸ

ă;ăĹĴăĎăĹŔĒăyĂăyĹă■ŮçñęäyśăĹĒăĹśăyžăĎŽăyĹă■ŮăŏŔiijŇă;ĒăŸŔăĹĒçŽŤçñę(èĹŸăĹĴăŔŤăŽŤčŽĎ

ĕğčăĒşăŮžăăĹ

stringăŔŔžéşçŽĎsplit()ăŮžăşŤăŔĹăĂĈăžŤăžŮéĹăŮyçŏĂă■ŤçŽĎă■ŮçñęäyśăĹĒăĹśăĈĒă;čiiij
ăŏĈăžŮăy■ăĒăĎŏyăĹĴăĎŽăyĹăĹĒçŽŤçñęăĹŮăĒăŸŔăĹĒçŽŤçñęăŔŤăžŤăy■çăŏăŏŽçŽĎčŤăžăăiijăĂĈ
ă;Ťă;ăĹĴăĎăĹŤăĹăĹăŤăŤčŽĎăĹĜăĹśă■ŮçñęäyśçŽĎăŮăăĂŽiijŇăĹĴăăĹă;ĹçŤĹre.
split()ăŮžăşŤiijŽ

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

èŏĹŏž

ăĜăĹŔre.split()ăŸŔéĹăŮyăŏŏčŤĹčŽĎiijŇăŽăăyžăŏĈăĒăĎŏyă;ăăyžăĹĒçŽŤçñęăŇĜăŏŽăĎŽăyĹ
ăŔŤăĉĈiijŇăĹĴăyĹéĹčçŽĎă;Ňă■Ŕăy■iijŇăĹĒçŽŤçñęăŔŔăžăĹŔéĂŮăŔŮiijŇăĹĒăŔŮăĹŮăĒăŸŔăĹŮăăiij
ăŔŔăĹăĹăĹŽăyĹăăiijŔéčŇăĹăĹŔiijŇéČčăžĹăŇăžéĒ■çŽĎăĹĒçŽŤçñęăyĎē;ççŽĎăŏă;ŤéČ;ăiijŽéčŇă;ŤăĹŔă
èĹŤăŽĎčçşăĎĹăyžăyĂăyĹă■ŮăŏŤăĹŮăĹiijŇăĹŽăyĹăŮşstr.split()
èĹŤăŽĎăăiijçşăĎŇăŸŔăyĂăăŮçŽĎăĂĈ

`re.split()` splits the line at the semicolon, and the backslash escapes the semicolon. The result is a list of fields, including the empty string at the end.

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

Now we can use the fields and delimiters to reform the line using the same delimiters.

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

Now we can use the fields and delimiters to reform the line using the same delimiters.

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

4.2 2.2 Using the re module to split a string

Using the re module to split a string

The `re.split()` function splits the line at the semicolon, and the backslash escapes the semicolon. The result is a list of fields, including the empty string at the end.

Using the re module to split a string

The `re.split()` function splits the line at the semicolon, and the backslash escapes the semicolon. The result is a list of fields, including the empty string at the end.

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

(continues on next page)

(continued from previous page)

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

æĈædIJä;äæĈsæĉÄæšëäd'Žçg■āNzéĚ■āRrèĈ;rijNāRlèIJÄèeAārEæL'ÄæIJL'çŽDāNzéĚ■éazæT;āĚēāL
çDúāRÖäijäçzŽ startswith() æLŪēĀĚ endswith() æŪzæsTrijŽ

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

äyNéíæYřāRēäyÄäylä;Nā■RrijŽ

```
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()
```

æĖGæĀlçŽDæYrijNèfZäylæŪzæsTäy■āfĚéazèeAè;ŠāĚēäyÄäyläĚĈçzDä;IJäyžāRCæTřāĀĈ
æĈædIJä;äæAřāügæIJL'äyÄäyĭ list æLŪēĀĚ set çszādNçŽDēĀL'æNĭ'éazrijN
èeAçāōāfIäijäeĀŠāRCæTřāL■āĚĤerĈTĭtuple() āřEāĚŪē;ñæ■cäyžāĚĈçzDçszādNāĀĈæřTāeĈrijŽ

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

èöleöž

startswith() āŠNendswith() æŪzæsTæRRä;ZāžEäyÄäyläIdäyyæŪzä;fçŽDæŪzäijRāŌzāAŽā
çszäijijçŽDæŠ■āIJäžšāRřäzēä;fçTĭāĤGçL'GæĭēāōđçÖrijNä;EæYřāzççāAçIJNèĭuæĭēāsçæIJL'éĈcāzĤäijYé

```
>>> 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;ääRfäzëëÇ;è£YæÇsä;£çTlæ■čāLŽèāè;āijRāŌzāóđçŌřijNæfTæČijŽ

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

è£Žçg■æŰzāijRāzšëāNā;ŰéĀŽřijNā;EæYřāzāzŌçóĀā■TçŽDāNžéĚ■āóđāIJlæYřæIJL'çČzārRæIRād'g
æIJĀāŌŌæRŘäyĀäyNijNā;ŠāŠNāĚüāzŰæS■ā;IJæfTæČæŽóéĀŽæTřæ■óèĀŽāRĹçŽyçzŠāRĹçŽDæŰ
startswith() āšŇ ends with() æŰzæšTæYřā;Läy■éTŽçŽDāĀČ
æfTæČijNāyNéIcè£Žäyler■āRèæčĀæšëæšRäylæŰGäzūād'zäy■æYřāŘēā■YāIJlæNĜāóŽçŽDæŰGäzūçšzād

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

4.3 2.3 çTÍShelléĀŽéĚ■çñęāNžéĚ■ā■Űçñęäyš

éŰóécY

ä;ääÇsä;£çTÍ **Unix Shell** äy■äyycTlçŽDēĀŽéĚ■çñę(æfTæČ *.py,Dat[0-9]*.csv
ç■L')āŌzāNžéĚ■æŰGæIJnā■Űçñęäyš

èğčāEşæŰzæāL

fnmatch æĹāāIŰæRŘä;ŽāžEäyđ'äyĹāĜ;æTřāĀTāĀT fnmatch() āšŇ
fnmatchcase() řijNāRfäzëçTlæIēāóđçŌřefZæāūçŽDāNžéĚ■āĀČçTlæšTæČäyNijŽ

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

(continues on next page)

(continued from previous page)

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

`fnmatch()` `ãĜ;æTřä;£çTlăzTăsCæŞ■ä;IJşşçzşçşçZDăd'gărRăEŻæTRæDşèĝDălZ(ăy■ăRŇŇZDçşçzşçşç`

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

`æĈCædIJă;ăărze£ZăylăNžăLăŋăLăIJlăDRijNăRřazěă;£çTl` `fnmatchcase()`
`æIěazcæZ£ăĀCăoČăoŇăĒlă;£çTlă;ăçZDălăqăijRăd'gărRăEŻăNžéĚ■ăĀCăerTăĈCijZ`

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

`è£Zăyd'ăylăĜ;æTřéĀZăyyăijZēcŋă£;çTēcZDăyĂăylçL'zăĀĝæYřăIJlăd'DçRĚéIdăŮĜăzŭăR■çZDă■Ůç`
`ærTăĈCijNăĀĜěoĵă;ăæIJL'ăyĂăylëqŮéAşşIJřălĀçZDălŮëălăTřæ■ōijZ`

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

`ăjăăRřazěăĈRè£ZăăŭăEŻăLŮëălăŌlăřijijZ`

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

ëőlëőž

`fnmatch()` `ãĜ;æTřăNžéĚ■Ĉ;ăLZăzNăžŌçôĂă■TçZDă■ŮçŋăyşæŮzæşTăŠNăijžăd'ĝçZDă■čăLZăæ`
`æĈCædIJăIJlăTřæ■ōăd'DçRĚæŞ■ă;IJăy■ăRlėIJĂēēAçőĂă■TçZDěĂŽéĚ■çŋăřsèĈ;ăoŇăĒRçZDăŮŭăĂZijl`

`æĈCædIJă;ăçZDăzčçăAéIJĂēēAăAŻæŮĜăzŭăR■çZDăNžéĚ■ijNăIJĂăë;ă;£çTl` `glob`
`æIăălŮăĀCăRČèĀĈ5.13ărRèĹCăĀĈ`

4.4 2.4 āŮčņęäÿšāŅzéĚāŠŅæŘĪĵťć

éŮóécŸ

äĵäæČšāŅzéĚāĽŮĚĀĚæŘĪĵťćčĽ'záǒŽæĹajĪŔčŽĎæŮĠæĪŅ

èġčāĒşæŮzæąĹ

āęĆæđĪäĵäæČšāŅzéĚčŽĎæŸřāŮéĹčāŮŮčņęäÿšĭĭŅéĆčázĹäĵäéĀŽăÿÿăŔĹéĪĀĕĕĀĕřČĵŦĹăşžæĪŅăŮ
æřŦăĕĆ str.find() , str.endswith() , str.startswith()
æĽŮĚĀĚčşzäĭĭĭĵčŽĎæŮzæşŦĭĭŹ

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

årzäžŌād'æĹĆčŽĎāŅzéĚæĪĀĕĕĀăĲčŦĹæčāĹŽæĹĕĹăĭĭŔăŠŅ re æĹăăĪŮăĀĆ
äÿžăžĒęġčĕĠĹæčāĹŽæĹĕĹăĭĭŔčŽĎăşžæĪŅăŌşĵŔĒĭĭŅăĀĠĕǒĹăĵäæČšāŅzéĚæŦřāŮăăĭĭăĭĭŔčŽĎæŮĕæ
11/27/2012 ĭĭĭŅăĵăăŔřăžĕĕĲŽæăŭăĀŽĭĭŹ

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

æĈæđIJă;æĈşă;ĤĉTlăŔŇăyĂăytlăıăijŔăŌzăAžăđŽăñăăŇzéĚĚĚijŇă;ăăžTêrêăĚĹăŔEăıăijŔăĚŮĉņēă

```
>>> 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() æĀžæŸŕăzŌăŮĉņēăyşăijĂăğŇăŌzăŇzéĚĚĚijŇăæĈæđIJă;æĈşăşşæL'ăŮĉņēăyşăzæĎŔéăĤĉTlă findall() æŰzæşŤăŌžăžĉæŽăĂĈæŕTăĉĆĚijŽ

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

ăIJlăŌžăzLăĈăĹZăijŔĉŽĐăŮăăĂžĚijŇăĂžăyăijŽăĹl'ĉTlăŇăŔăŮăŌzăŤêŌăĹĚĉzĐăĂĈæŕTăĉĆĚij

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

æŤêŌăĹĚĉzĐăŔŕăzēăĤăŮăŔŌéĭĉĉŽĐăđĐĉŔĚăŽtăĹăĉŏĂăŤĚijŇăZăăyăzăŔŕăzēăĹĚăĹŇăŔĚăŕŔăy

```
>>> 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)

(continued from previous page)

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

`findall()` returns a list of all non-overlapping matches of the regular expression in the string. The matches are returned as a list of tuples, where each tuple contains the matched groups. The `finditer()` method returns an iterator object that yields match objects for each non-overlapping match of the regular expression in the string.

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

Regular Expressions

Regular expressions are a powerful tool for matching patterns in text. They are used in many applications, including text processing, data validation, and search engines. The `re` module in Python provides a set of functions for working with regular expressions. The `re.compile()` function is used to compile a regular expression into a regular expression object. The `match()` function is used to check if a regular expression matches a string at the beginning. The `findall()` function is used to find all non-overlapping matches of a regular expression in a string. The `finditer()` function is used to find all non-overlapping matches of a regular expression in a string, returning an iterator object.

The `match()` function returns a match object if the regular expression matches the string at the beginning, and `None` otherwise. The `findall()` function returns a list of all non-overlapping matches of the regular expression in the string. The `finditer()` function returns an iterator object that yields match objects for each non-overlapping match of the regular expression in the string. The `match()` function is used to check if a regular expression matches a string at the beginning. The `findall()` function is used to find all non-overlapping matches of a regular expression in a string. The `finditer()` function is used to find all non-overlapping matches of a regular expression in a string, returning an iterator object.

The `match()` function returns a match object if the regular expression matches the string at the beginning, and `None` otherwise. The `findall()` function returns a list of all non-overlapping matches of the regular expression in the string. The `finditer()` function returns an iterator object that yields match objects for each non-overlapping match of the regular expression in the string.

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

The `match()` function returns a match object if the regular expression matches the string at the beginning, and `None` otherwise. The `findall()` function returns a list of all non-overlapping matches of the regular expression in the string. The `finditer()` function returns an iterator object that yields match objects for each non-overlapping match of the regular expression in the string.

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

æIJĀāŔŌīījNāēĆæđIJā;āāzĒāzĒæYŕāAŽāyĀæñāçōĀā■TçŽĐæŪĠæIJñāNžéĒ■/æŔIJçt'ćæŞ■ā;IJçŽĐēŕI
re ælāāIŪçžgāLŋçŽĐāG;æTŕāĀĆæŕTāēĆīījŽ

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

ā;EæYŕéIJĀēēAæşlæĐŔçŽĐæYŕīījNāēĆæđIJā;āæL'ŞçōŪāAŽād'gēĠŔçŽĐāNžéĒ■āŠNæŔIJçt'ćæŞ■ā;IJ
ælāāIŪçžgāLŋçŽĐāG;æTŕāījŽāŕEæIJĀēēŞçījŪēŕSēēĠGçŽĐælāāījŔçījŞā■YēŭæIēīījNāZāæ■d'āzūāy■āījŽæŪL
ā;EæYŕāēĆæđIJā;ĤçTīēćĐçījŪēŕSælāāījŔçŽĐēŕIīījNā;āāŕEāījŽāĠŔāŕSæşæL;āŠNāyĀāzŽéćIād'ŪçŽĐād'Đ

4.5 2.5 ā■ŪçñēäyşæŔIJçt'ćāŠNæŽĚæ■ć

éŪŌéćY

ā;āæČşāIJlā■Ūçñēäyşäy■æŔIJçt'ćāŠNāNžéĒ■æNĠāōŽçŽĐæŪĠæIJñælāāījŔ

èğçāEşæŪzæāL

ārzāžŌçōĀā■TçŽĐā■ŪéIćælāāījŔīījNçŽt'æŌēā;ĤçTī str.replace()
æŪzæşTā■şāŔīījNæŕTāēĆīījŽ

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

ārzāžŌād'■æIĆçŽĐælāāījŔīījNēŕūā;ĤçTī re ælāāIŪāy■çŽĐ sub()
āG;æTŕāĀĆ āyžāžEēŕt'æYŌēĤZāyIīījNāAĠēō;ā;āæČşāŕEā;ćāījŔāyž 11/27/2012
çŽĐæŪēæIJşā■ŪçñēäyşæTzæLŔ 2012-11-27 āĀĆçd'žā;NāēĆāyNīījŽ

```
>>> 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() āG;æTŕāy■çŽĐçñāyĀāylāŔĆæTŕæYŕēcñāNžéĒ■çŽĐælāāījŔīījNçññāžNāylāŔĆæTŕæYŕæŽĚæ
\\3 æNĠāŔSāL■éIćælāāījŔçŽĐæ■TēŌūçžĐāŔūāĀĆ

āēĆæđIJā;āæL'ŞçōŪçTīçŽyāŔNçŽĐælāāījŔāAŽād'ZæñæZĚæ■çīījNēĀČēZŞāĒLçījŪēŕSāōČæIēæŔŔā

```
>>> 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.'
>>>
```

ārzāžŌæŽt'āLāād'■æIĆçŽĐæŽĚæ■çīījNāŔŕāzēāījāēĀŞāyĀāylæZĚæ■ćāZdēŕČāG;æTŕāIēāzçæŽēīījNæŕ

äyÄäylæŽŁæ■cāZðerCāĜ;æTṛçŽDāRĆæTṛæYřäyÄäyl match áržèsajijŇāžšārśæYř
 match() æĽŮĚĀĚ find() èĤTāZđçŽDāržèsāāĀĆ äjĤçTĪ group()
 æŮžæşTæIēæRŖāRŮçĽžáoŽçŽDāŇzéĚ■éĈlāLEāĀĆāZðerCāĜ;æTṛæIJĀāRŌèĤTāZđæŽŁæ■cāŮçņäysāÄ
 æĈædIJēŽd'āžEæŽŁæ■cāRŌçŽDçzŞædIJād'ŮijŇā;æèĤYæĈşçşēéAşæIJĽād'ŽārŚæŽŁæ■cāRŚçTşāžE
 re.subn() æIēäžcæŽŁāĀĆærTāeĈiijŽ

èóíèőž

4.6 2.6 ă■ŮčņăyřšăĲ;cTěăd'ğăřRăĲčŽĎăŘIĲct'ćăŽĲă■ć

éŮőécÿ

èğčǎEşæŮźæąŁ

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

æIJĀāRŌčŽDěČčäyläĴNā■RæR■cd'žāžEäyÄäylārRçijžēŽūijNæŽŁæ■cā■Ūçņäyšāzūäy■aijŽēĠāLléūš
äyžāžEāŁōād'■ēfŽāyĥijNā;āāRrēČĴēIJĀēēAäyÄäylēĴĒāL'āĠ;æTřijNārśāČRāyNéÍčçŽDēŁŽæūijŽ

```
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
```

äyNéÍcæYřäĴčTřāyLēŁřāĠ;æTřçŽDæŪzæšTřijŽ

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

ērSēĀĒæšĥijŽ matchcase('snake') ēŁTāŽdāžEäyÄäylāŽdērČāĠ;æTř(āRCæTřāŁĒēāzæYř
match āřzēsā)ijNāL■ēÍcäyÄēŁCæRŘāLřēŁĠijN sub()
āĠ;æTřēŽd'āžEāŌēāRŪæŽŁæ■cā■Ūçņäyšād'ŪijNēŁYēČĴæŌēāRŪäyÄäylāŽdērČāĠ;æTřāĀČ

ēōlēōž

āržāžŌäyÄēLŋčŽDāŁĴčTēād'ġārRāĒŽçŽDāNzéĒ■æŠ■äĴIijNčōĀā■TçŽDāijāēĀŠäyÄäyl
re.IGNORECASE æāĠāŁŪāRCæTřārśāūšçžRēūšād'šāžEāĀČ
āĴEäYřēIJĀēēAäšĴæĎRçŽDæYřijNēŁŽāylāržāžŌæšRāžŽēIJĀēēAād'ġārRāĒŽēĴ;ñæ■čçŽDUnicodeāNzéĒ■ā
āRCēĀČ2.10ārRēŁČāžEēġčæŽt'ād'ŽçžEēŁČāĀČ

4.7 2.7 æIJĀçš■āNzéĒ■æĴāaijR

éŪōécY

äĴāæ■cāIJĴērTçĴĀčTřāē■čāLŽēāĴēĴāijRāNzéĒ■æšRāylæŪĠæIJñāĴāaijRijNā;EäYřāōČæLĴāŁřçŽDæY
ēĀNāĴāæČšāŁōāTřāōČāRŸæŁRæšēæLĴæIJĀçš■çŽDāRrēČĴāNzéĒ■āĀČ

ēġčāEšæŪzæāĴ

ēŁŽāylēŪōécYäyÄēLŋāĠžçŌřāIJĴēIJĀēēAāNzéĒ■äyÄāržāŁēēŽTçņēāžNéŪt'çŽDæŪĠæIJñçŽDæŪūāĀ
äyžāžEēřt'æYŌäyYēæēŽijNēĀČēŽŚāçCäyNçŽDäĴNā■RřijŽ


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

āIġēŁZāyġā;Nā■Rāy■īījNāġāījR r'\\"(.*)\\\"' ċŽDæĐRāŽġæYřāNzéĚ■ècāRŇāījTāRūāNĚāRŋċŽġ
 ājEæYřāIġā■čāŁZēāġēġāījRāy■*æŠ■ā;IġċġæYřēťġāġġŽĐīījNāZāæ■d'āNzéĚ■æŠ■ā;IġāījZæšēāLġæIġġēť
 āžŌæYřāIġġġāžNāyġā;Nā■Rāy■æRIġťć text2 ċŽDæŪāĀŽēťTāZđċzŠæđIġāžūāy■æYřāēŁSāžnāēČšēēAġ

āyžāžEāŁōæ■čēŁZāyġēŪōēćYīījNāRřāžēāIġāġāījRāy■ċŽĐ*æŠ■ā;IġċġāRŌēġāŁāāyŁ?āŁōēēřċġēīījNā

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

ēŁZæāūāřsā;ġāġ;ŪāNzéĚ■āRŸæŁRēġđēťġāġġāġāījRīījNāžŌēĀNāġ;ŪāŁræIġġš■ċŽĐāNzéĚ■īījNāžšāřsā

èŌēŌž

ēŁZāyġēŁČāsťċđ'žāžEāIġāēZāNĚāRŋċČz(.)ā■ŪċġġēċŽDæ■čāŁZēāġēġāījRċŽDæŪāĀŽēAġāŁřċŽDāy
 āIġāyġāyġāēāījRā■Ūċġēāyšāy■īījNċČz(.)āNzéĚ■ēZđ'āžEæ■čēāNāđ'ŪċŽĐāžzā;Tā■ŪċġēāĀĆ
 ċĐūēĀNīījNāēČāđIġā;āārEċČz(.)āRūāēTġāIġāījġāġNāyŌċzŠæġšċġē(ærťāēČāījTāRū)āžNēŪťċŽDæŪāĀŽē
 ēŁZæāūēĀŽāyġāījZārījēġťāġ;Łāđ'Žāy■ēŪťċŽĐēcāījġāġNāyŌċzŠæġšċġēāNĚāRŋċŽDæŪġæIġġēcāāġġťTēā
 ēĀŽēŁġāIġ * æŁŪēĀĚ + ēŁZæāūċŽDæŠ■ā;IġċġāRŌēġāēūzāŁāāyġāyġ ?
 āRřāžēāījzāŁūāNzéĚ■ċŌŪāšTæťzæŁRāřzæLġæIġġš■ċŽĐāRřēČ;āNzéĚ■āĀĆ

4.8 2.8 āđ'ŽēāNāNzéĚ■æġāījR

ēŪōēćY

ājāæ■čāIġēťĤġġġāġ;ġċťġāē■čāŁZēāġēġāījRāŌzāNzéĚ■āyġāāđ'ġāġŪċŽDæŪġæIġīījNēĀNā;āēIġāēēAēūġ

ēġčāEšæŪzæāġ

ēŁZāyġēŪōēćYāġ;ŁāĚyāđNċŽĐāġžċŌřāIġā;Šā;āċťġċČz(.)āŌzāNzéĚ■āžzæĐRā■ŪċġġēċŽDæŪāĀŽēīījNā
 æřťāēČīījNāAġēŌġā;āæČšēřĤġġġāŌzāNzéĚ■Čēř■ēġāġġēāŁšċŽDæšġēġġīījŽ

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

(continues on next page)

(continued from previous page)

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

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

re.compile(r'\/*((?:.|\\n)*?)\/', re.DOTALL)

4.9

re.compile(r'\/*((?:.|\\n)*?)\/', re.DOTALL)

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

re.compile(r'\/*((?:.|\\n)*?)\/', re.DOTALL)

4.9 2.9

4.9

re.compile(r'\/*((?:.|\\n)*?)\/', re.DOTALL)

4.9

re.compile(r'\/*((?:.|\\n)*?)\/', re.DOTALL)

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

(continues on next page)

(continued from previous page)

```
'Spicy JalapeÃ±o'
>>> s2
'Spicy JalapeÃ±o'
>>> s1 == s2
False
>>> len(s1)
14
>>> len(s2)
15
>>>
```

normalize() çññäÿÄäÿlâRĆæTṛæŃĠăōŽăŰŰçñęäÿşæăĠăĠĚăŃŰçŽĎæŰzâijRăĂĆ
NFCealçd'žăŰŰçñęăžTërëæŸræTṛ'ä;ŞçzĎæĹŔ(æŕTăeĈăRrèĈ;çŽĎërĭârşă;£çTlăŰTăÿAçijŰçăA)iiijŃëĂŃNFI
PythonăŔŃăăuăŦṛæŃAæL'ŦăşTçŽĎæăĠăĠĚăŃŰŰă;ćâijŔNFKCăŞŃNFKDiiijŃăőĈăžŃăĬJlăd'ĎçŔĚæşŔ

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

normalize() çññäÿÄäÿlâRĆæTṛæŃĠăōŽăŰŰçñęäÿşæăĠăĠĚăŃŰçŽĎæŰzâijRăĂĆ
NFCealçd'žăŰŰçñęăžTërëæŸræTṛ'ä;ŞçzĎæĹŔ(æŕTăeĈăRrèĈ;çŽĎërĭârşă;£çTlăŰTăÿAçijŰçăA)iiijŃëĂŃNFI

PythonăŔŃăăuăŦṛæŃAæL'ŦăşTçŽĎæăĠăĠĚăŃŰŰă;ćâijŔNFKCăŞŃNFKDiiijŃăőĈăžŃăĬJlăd'ĎçŔĚæşŔ

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

[illegible]

æIJAãRÕäyÄäyļa;Nã■RāsTçd'žāžE unicodedata ælɑɑlUçŽDāRēäyÄäyļēG■ēAæŪzélciijNāzšārsæ
combining() āĢ;æTŗāRfāzēætNērTāyÄäyļa■UçņæYŗāRēäyžāŠNēššā■UçņēāĀC
āIJlēfZāylælaɑlUäy■ēfYæIJLāĒūāzŪāĢ;æTŗçTlāzŌæšēæL;ā■UçņęçszāLñijNæ;NērTæYŗāRēäyžæTŗā■Ūā
UnicodeæY;çDūæYŗäyÄäyļa;Ļād'gçŽDäyžécYāĀCāēCādIJæCšæZt'æuśāĒēçŽDāžEēgčāĒšāžŌæāGāC
ērūçIJNēĀC UnicodeāōYç;Šāy■āĒšāžŌēfZēČlāĻēçŽDērt'æYŌ
Ned BatchelderāIJl āzŪçŽDç;ŠçñŽ äyĻārZPythonçŽDUni-
codeād'DçRĒēŪōécYāzšæIJLäyÄäyļa;Ļāē;çŽDāzNçz■āĀC

éŮőécŸ

ä; äæ■čāIĴä;ŁçTīæ■čāLZēālē; ,; āiĴRād'DčŘEæŮĞæIĴniiĴNä;EæŸrāĚšæšĲčZĎæŸrUnicodeā■Ůčņēād'Dč

```

ézYèòd' æČĚăĖtäyN
codeā■ŮčņęšzæIJL'ăžĖāšžæIJņčŽDæŦræŊĀăĂĆ
re
āūščžŦĀNzéĖ■ăžzæĎŦŕčŽDunicodæŦŦā■Ůā■ŮčņęăžĖīījŽ
æŦāēĆīījN
\ \ d

```

æĈæđIjä;äČšāIJlælaijRäy■āNĖāŔnēNǦăōŽčŽDUnicodeā■ŮčņēijNā;āāŔřāzēā;ęçȚİUnicodeā■Ůčņ
\uffff æŁŬēÄĲ \UFFFFFFFF)ăĂĆ æŕŤăeÇiiJNäyNeİcāYřāvĀävłāNžēĒ■ăĞăävļv■ārNēYľæNL'aijrçijŮc

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

ā;ŞæL'gëaŊāŊzéĚāšŊæŔIJçt'ćæŞā;IJçŽDæŮūāĀŽiijŊæIJĀāē;æŸŕāĒLæăĠăĠĚāŊŮāzūāyŤæyĚçŔĒ
ä;EæŸŕāŔŊāūāzşāžŤērēæşĬæĐŔäyĀāžŽçL'zæŏLæĈĚāĒiijŊæŕŤæĈāIJĬāŧ;çŤēād'ġārŔāĒZāŊzéĚāšŊād'

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

èõlèõž

æūūāŔLä;ŧçŤĬUnicodeāšŊæāĬLŽèāĬē;āijŔéĀŽāyāijŽèŏŦā;āæLŞçŊĈāĀĈ
āēĈæđIJä;ăçIJşçŽDæL'ŞçŏŮèĚæāūāĀŽçŽDērĬiijŊæIJĀāē;èĀĈèŽŚāyŊāŏL'èĉĚçññāyL'æŮzæāĬLŽāijŔāžŔ
āŏĈāžñāijŽāyžUnicodeçŽDād'ġārŔāĒZè;ŋæāšŊāĒūāzŮād'ġéĠŔæIJL'èūĉçL'zæĀġæŔŔä;ZāĒĬēĬççŽDæŦŕ

4.11 2.11 āĬæŽd'āŮçņęäyşäyāyēĬJĀēęAçŽDāŮçņę

éŮŏécŸ

ä;āæĈşāŐzæŐL'æŮĠGæIJñāŮçņęäyşāijĀād't'iijŊçzŞār;æĬŮèĀĚäyēŮt'äyæĈşęęAçŽDāŮçņęiijŊæŕ

èġĈāĒşæŮzæāĬ

strip() æŮzæşŤèĈ;çŤĬāžŐāĬæŽd'āijĀāġŊæĬŮçzŞār;çŽDāŮçņęāĀĈ
rstrip() āšŊ rstrip() āĬĒāĬŊāzŐāūęāšŊāzŐāŔşæL'gëaŊāĬæŽd'æŞā;IJāĀĈ
ézŸèŏd'æĈĚāĒġāyŊiijŊæŧZāžZæŮzæşŤāijZāŐzéŽd'çĬ'žçŽ;āŮçņęiijŊä;EæŸŕä;āāzşāŔŕāžæŊĠāŏZāĒūāzŮ

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

(continues on next page)

(continued from previous page)

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

ěóľěőž

ěĚžžž strip() æŮžæšŤāIJlérzāRŮāšNæŷĚčŘĚæŤřæ■őäžěād'ĜāŘŎçž■ād'ĎčŘĚçŽĎæŮūāĀŽæŸřç
æŕŤāęĈiijNā;āāŔŕāžęčŤíāőĈāžñæĹēāŎžæŎŮč!žæāijīijNāijŤāŔūāšNāőNæĹŔāĚūāžŮāžžāŁāāĀĈ
ā;ĚæŸřēIJĀęęAæšĹæĎŔçŽĎæŸŕāŎžéŽd'æš■ā;IJäŷ■āijŽāŕžā■ŮçņęäŷšçŽĎäŷ■éŮŕçŽĎæŮĜæIJñžğçŤ

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

āęĈæđIJā;āæĈšād'ĎčŘĚäŷ■éŮŕçŽĎç!žæāijīijNéĈčāžĹā;āēIJĀęęAæšĈāĹ!āĚūāžŮæĹĀæIJŕāĀĈæŕŤāę
replace() æŮžæšŤæĹŮēĀĚæŸřçŤíæ■čāĹŽēāĹēĹāijŔæŽĚæ■čāĀĈçd'žäĹNāęĈāŷNīijŽ

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

ēĀŽāŷŷæĈĚāĚŷāŷNā;āæĈšāŕĚā■Ůçņęäŷš strip æš■ā;IJāšNāĚūāžŮēĚ■āžçæš■ā;IJçŽŷçžšāŔĹiijNæŕ
āęĈæđIJæŸřēĚāūçŽĎŕīiijNéĈčāžĹçŤšæĹŔāŽĹēāĹēĹāijŔāŕšāŔŕāžěād'ğæŸĹēžñæĹNāžĚāĀĈæŕŤāęĈiijŽ

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

āIJĹēĚŽēĜNīijNēāĹēĹāijŔ lines = (line.strip() for line in f)
æĹğēāNæŤřæ■őē;ñæ■čæš■ā;IJāĀĈ ěĚžçğ■æŮžāijŔēĹđāŷŷēŷŷæŤĹiijNāŽāŷžāőĈāŷ■ēIJĀęęAęĈĎāĚĹŕžāŕ
āőĈāžĚāžĚāŔĹæŸŕāĹāžžāŷĀäŷĹçŤšæĹŔāŽĹiijNāžūāŷŤæŕŔæñæĚŤāŽĎēāNāžNāĹ■āijŽāĚĹæĹğēāN
strip æš■ā;IJāĀĈ

āŕžāžŎæŽŕénŸēŸŮçŽĎstripīijNā;āāŔŕēĈ;ēIJĀęęAā;ĚçŤĹ translate()
æŮžæšŤāĀĈčŕūāŔĈéŸĚäŷNāŷĀēĹĈāžĚęğçæŽŕād'ŽāĚšāžŎā■ŮçņęäŷšæŷŷĚčŘĚçŽĎāĚĚāőžāĀĈ

(continued from previous page)

```
'pÃ;tÄëÃúÃs is awesome\n'
>>> b.translate(cmb_chrs)
'python is awesome\n'
>>>
```

```
äyLéIcä;NāRāyRijNéÅŽèfGā;fçTl dict.fromkeys()
æŰzæſTædDēÄäyÄäylāUāËÿijNæfRāyUnicodeaŠNéſſçñæ;IJäyžéTōijNārzážTçŽDāĀijāÉléČlāyž
None āĆ
```

```
çDúāRŌā;fçTl unicodedata.normalize() āřEāŌſāgNè;ŠāĒēāGāGĒāNŰäyžāLĒēgčā;čāijRā
çDúāRŌāEēççTl translate āĠ;æTřāLāéŽd'æL'ĀæIJL'éĠéſſçñæāĆ
āRŊæūçŽDæL'ĀæIJřāžšāRřāžèèçñçTlāĒēāLāéŽd'āĒūāžŰçſžādnçŽDāŰçñç(æřTāçĀēŌgāLūāŰçñçL)ā
ā;IJäyžāRēāyÄäylā;NāRijNéfŽéGŊædDēÄäyÄäylāEāL'ĀæIJL'UnicodeæTřāŰāŰçñçæYāārDāL
```

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

```
āRēāyĀçġæyĒçRĒæŰĠæIJñçŽDæL'ĀæIJřāūL'āRĒāLřI/OègčçāĀäyŌçijŰçāĀāĠ;æTřāĀÇèfŽéGŊçŽD
çDúāRŌāEēççšāRĒ encode() æLŰēÄē decode() æſā;IJāĒēæyĒēŽd'æLŰāfōæTžāōČāĀÇæřTāçĀijŽ
```

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

```
èfŽéGŊçŽDæāGāGĒāNŰæſā;IJārEāŌſæĒççŽDæŰĠæIJñāLĒēgčāyžāTçNñçŽDāŠNéſſçñæāĀÇæŌē
ā;ſçDūijNéfŽçġæŰzæſTæžĒäžĒāRĒāIJāĒāRŌçŽDçŽōæāĠāřsæYřēŌūāRŰāLřæŰĠæIJñārzážTĀCSIIēā
```

èõléõž

```
æŰĠæIJñāŰçñçæyĒçRĒæyÄäylāIJÄäyžèèAçŽDēŰōécYāžTēřæYřēfRēāNçŽDæĀgēČ;āĀÇāyĀēLñā
āržāžŌçōĀāTçŽDæŽfæçæſā;IJijN str.replace() æŰzæſTēĀŽāyÿæYřæIJĀāfñçŽDijNçTžèĠſāIJ
æřTāçĀijNäyžāžEæyĒçRĒçl'žçŽ;āŰçñçijNä;āāRřāžèèfŽæūāAžijŽ
```

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

(continues on next page)


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

translate() æŁŨëĀĒæ■čāŁŻeāĹë;ĹāijRēeAāŁāā;Łād'ŽāĀĆ

translate() æŨzæşŦāijŽēīdāyŷçŽDāŁāĀĆ

āzŌād'gçŽDæŨzēīcæĹēēōšīijNārzāzŌā;āçŽDāzŦçŦīçĹNāzRæĹēērt'æĀgēČ;æŸrā;āāy■ā;Ũāy■āŌzēĠāūsāy■āzŷçŽDæŸrīijNæŁŚāzñāy■āRfēČ;çzŽā;āāzžēōōāyĀāyŦçŁ'zāōŽçŽDæŁĀæIJfīijNā;ŁāōČēČ;ād'šéĀĆāzŦāāZāæ■d'āōdēŽĒæČĒāĒŦāy■ēIJĀēeAā;āēĠāūsāŌzārĹērŦāy■āRŦçŽDæŨzæşŦāzŭērDāijrāōČāĀĆ

ār;çōāēŁZāyĀēŁCéZEāy■ēōĹēōžçŽDæŸræŨGæIJñīijNā;ĒæŸrçşzāijijçŽDæŁĀæIJfāzşāRfāzēēĀĆçŦīāz

4.13 2.13 ā■Ũçņęäyşārzé;Ř

éŨōécŸ

ā;āæČşēĀŽēŁGæşŘçg■ārzé;ŘæŨzāijRæĹēæāijāijRāŦŨā■Ũçņęäyş

èğcāĒşæŨzæāŁ

ārzāzŌāşzæIJñçŽDā■Ũçņęäyşārzé;Řæş■ā;IJīijNāRfāzēā;ŁçŦīā■ŨçņęäyşçŽD ljust(), rjust() āŦŦ center() æŨzæşŦāĀĆærŦāeČīijŽ

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

æŁ'ĀæIJŁ'èŁZāzŽæŨzæşŦéČ;èČ;æŌēāRŨāyĀāyĹāRfēĀŁçŽDāāāāĒĒā■ŨçņęāĀĆærŦāeČīijŽ

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

āĠ;æŦŦ format() āRŦæāūāRfāzēçŦīāēā;ŁāōzæŸşçŽDārzé;Řā■ŨçņęäyşāĀĆ
ā;āēeAāĀZçŽDārşæŸrā;ŁçŦī<, > æŁŨëĀĒ ^ ā■ŨçņęāRŌēĹçŦ'gēūşāyĀāyĹāŦGāōŽçŽDāō;āžēāĀĆærŦāeČīijŽ

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

æÇædIJä;äæÇsæŃGăŃZăyÄäyŁéİdçl'zæäijçŽĐăăŃăĚĚă■ŮçņēijŃăŕĒăŃČăĚŽăĽŕăŕzé;Ŕă■ŮçņēçŽĐăĽ■

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

ă;ŞæäijäijŔăŃŮăđ'ŽăyĽăÄijçŽĐæŮŮăĂŽiijŃĕŁZăžŽæäijäijŔăžččăĂăžşăŔŕăžĕĕčŋçŦĽăIJĽ
format() æŮzæşŦăy■ăĂÇæŕŦăĕÇiijŽ

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

format() äĢ;æŦŕçŽĐăyÄäyŁăĕ;ăđ'DæŸŕăŃČăy■ăžĚĕĂČçŦĽăžŎă■ŮçņăyşăĂČăŃČăŔŕăžĕçŦĽăĬæäij
æŕŦăĕÇiijŃă;ăăŔŕăžĕçŦĽăŃČăĬæäijäijŔăŃŮæŦŕă■ŮiijŽ

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

ēŏĽēŃž

ăIJĽĕĂĂçŽĐăžččăĂăy■iijŃă;ăçžŔăyŷäijŽçIJŃăĽŕĕčŋçŦĽăĬæäijäijŔăŃŮæŮĢæIJŋçŽĐ
% æŞ■ă;IJçņĕăĂÇæŕŦăĕÇiijŽ

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

ă;ĒæŸŕiijŃăIJĽæŮŕçĽĽăIJŋăžččăĂăy■iijŃă;ăăžŦĕŕĕäijŸăĔĽăĽĽăŃĽ
format() äĢ;æŦŕæĽŮĕĂĔæŮzæşŦăĂÇ format() ĕĕĂæŕŦ %
æŞ■ă;IJçņēçŽĐăĽşĕČ;æŽŕăyžăijžăđ'ğăĂÇ äžŮăyŦ format() äžşæŕŦă;ĕçŦĽ
ljust(), rjust() æĽŮ center() æŮzæşŦăŽŕĕĂŽçŦĽiijŃ
ăŽăäyžăŃČăŔŕăžĕçŦĽăĬæäijäijŔăŃŮăžžæĐŔăŕžĕşăŕiijŃĕĂŃăy■ăžĔăžĔæŸŕă■ŮçņăyşăĂČ

æĈædIJæĈşèeAăŃăĒlăžEğç
èrûăŔCèĂĈ âIJlçžPythonæŪĞæqç

format()

ăĜjæŦŕçŽĐæIJLçŦlçL'záĂĝiijŃ

4.14 2.14 âŔĹăžúæNijæŌěăŪçņęäŷš

éŬŌécŸ

ăjăæĈşârEăĜăăŷlăŕŔçŽĐăŪçņęäŷšăŔĹăžúăŷăŷăĂăŷlăd'ğçŽĐăŪçņęäŷš

èğĉăEşşæŪzæqĹ

æĈædIJăjăæĈşèeAăŔĹăžúçŽĐăŪçņęäŷšæŸŕăIJlăŷĂăŷlăžŔăĹŬăĹŪèĂĒ iterable
ăŷŋiijŃĒéĈăžĹæIJĂăŋŋçŽĐæŪzăijŔăŕşæŸŕăjçŦl join() æŪzæşŦăĂĈæŕŦăeĈiijŽ

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

ăĹlçIJŃèŧăŭăĲiijŃĒéŹçğèŕæşŦçIJŃăŷĹăŌzăijŽæŕŦèĈăĂŋiijŃăjEăŸŕ
join() èĉŋăŃĜăŏŽăŷžăŪçņęäŷşçŽĐăŷĂăŷlăŪzæşŦăĂĈ
èŹŽăăŭăĂžçŽĐéĈlăĹăŌşăZăæŸŕăjăæĈşăŌžèŹđæŌççŽĐăŕžèşăŕŕèĈjæĲèĈĜăŔĐçğăŷăŕŃŋçŽĐæŦŕæ
æĈædIJăIJăĹĂăæIJL'èŹŽăžŽăŕžèşăăŷĹéĈjăŏŽăžĹăŷĂăŷl join()
æŪzæşŦăŸŌăŸjæŸŕăEŭăjŽçŽĐăĂĈăZăăđ'ăjăăŔĲIJĂèeAăŃĜăŏŽăjăæĈşèeAççŽĐăĹăĹšăŪçņęäŷşă
join() æŪzæşŦăŌžăŕEăŪĞăIJŋçĹĜăŏŧçžĐăŔĹèŧăŭăĲăĂĈ

æĈædIJăjăăžĒăžĒăŔĲæŸŕăŔĹăžúăŕşæŦŕăĜăăŷlăŪçņęäŷşiijŃăjçŦlăĹăăŕŭ(+)
éĂžăŷŷăŭşçžŔèŭşăd'ş

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

ăĹăăŕŭ(+)
æŞăjIJçņęăIJăjIJăŷăŷăĂăžŽăđ'ăĲăŪçņęäŷşăăijăijŔăŃŪççŽĐæŽăžçæŪzæqĹçŽĐæŪŭă

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

æĈædIJăjăæĈşăIJăžŔçăĂăŷăŕEăŷd'ăŷlăŪéĲăŪçņęäŷşăŔĹăžúèŧăŭăĲiijŃăjăăŔĲIJĂèeAçŏĂăŦçŽ

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

ëõlëõž

■ÜçñäyşâRLâzûâRrêÇ;çIJNâyLâÖzâzûäy■éIJĀēēAçTīäyĀæTt'èLCæIëèõlëõžāĀĆ
 äjEæYřäy■āžTēřēārRçIJNēēZāyIēŮōēēYñijNçlNāzRāSŸéĀŽāyāIJĀ■ÜçñäyşæāijāijRāNŮçŽDæŮūāĀŽāZ
 æIJĀēēG■ēēAçŽDēIJĀēēAāijTētuāşlæDRçŽDæYñijNā;ŞæLŠāznā;ççTīāLāāRū(+)æŞ■ā;IJçñæāŌzēēfðæē
 āŽāyāzāLāāRūēēfðæŌēāijŽāijTētuāEĒā■Yād'■āLūāzēāRLādČāIJ;āŽDæTūæŞ■ā;IJāĀĆ
 çL'zāLñçŽDñijNā;ææyēēIJēČ;äy■āžTāČRāyNéIēēēZæāūāEŽā■ÜçñäyşēēfðæŌēāzççāAñijŽ

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

ēēŽçg■āEŽæşTāijŽæřTā;ççTī join() æŮzæşTēfRēāNçŽDēēAæĒcāyĀāžŽñijNāZāyāzæřRāyĀæñææL
 äjāæIJĀēē;æYřāĒLæTūēēZEæL'ĀæIJL'çŽDā■ÜçñäyşçL'ĠæōtçDūāRŌāE■ārEāōČāznēēfðæŌēētūæIēāĀĆ
 äyĀäyIçŽyāřzæřTē;ÇēAļæYŌçŽDæLĀāūgæYřāL'çTīçTşæL'RāZlēāIē;ççāijR(āRÇēĀĆ1.19ārRēĒC)ējñā

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

āRŊæāūēēYā;ŮæşlæDRāy■āēēēēAçŽDā■ÜçñäyşēēfðæŌēæŞ■ā;IJāĀĆæIJL'æŮūāĀŽçlNāzRāSŸéIJĀēē

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

ā;ŞæūūāRLā;ççTīI/OæŞ■ā;IJāSŊā■ÜçñäyşēēfðæŌēæŞ■ā;IJçŽDæŮūāĀŽñijNæIJL'æŮūāĀŽēIJĀēēAāij
 æřTāēČñijNēĀČēŽSāyNéIēēēŽDāyē'çñrāzççāAçL'ĠæōtñijŽ

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

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

āēČædIJāyē'äyIā■Üçñäyşā;LārRñijNēČcāzLçññāyĀäyIçL'LæIJñæĀgēČ;āijŽæŽt'āē;āžŽñijNāZāyāzI/Oç
 āRēād'ŮāyĀæŮzēēēñijNāēČædIJāyē'äyIā■Üçñäyşā;Lād'gñijNēČcāzLçññāzNāyIçL'LæIJñāRrêÇ;āijŽæŽt'āē
 āŽāyāzāōČēAāēāēāzEāLZāzāyĀäyIā;Lād'gçŽDāyē'æŮūçZşædIJāzūāyTēēAād'■āLūād'gēGRçŽDāEĒā■Yā
 ēēYæYřēČcāRēēřñijNæIJL'æŮūāĀŽæYřēIJĀēēAæāzæ■ōā;āçŽDāžTçTīçlNāzRçL'zçČzæIēāEşāōŽāžTēřēā;ç

æIJĀāRŌērLäyÄäyNīijNāeCædIJā;āāGĒāđ'ĠçijŪāEZæđDāzZāđ'ġeĠRārRā■ŪçņęäyşçŽĐē;ŞāĠzāzççāA
 ä;āæIJĀāē;ēĀCēZŚāyNā;ŁçTłçTşæLRāZlāĠ;æTŗijNāL'çTlyieldēr■āRēāzġçTşē;ŞāĠzçL'ĠæōtāĀĆæfTāeĆ

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

ēŁŻçġ■æŪzæşTāyÄäyIæIJL'ēūççŽĐæŪzéIcæYrāōCāzūæşæIJL'āržē;ŞāĠzçL'ĠæōtāLrāzTēeAæĀŌæāŭ
 ä;NāeCīijNā;āāRrāzēçōĀā■TçŽĐā;ŁçTł join() æŪzæşTārEēŁZāzŽçL'ĠæōtāRLāzūētūæIēijŽ

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

æLŪēĀĒä;āāzşāRrāzēārEā■ŪçņęäyşçL'ĠæōtēĠ■āōZāRŚāLr/OīijŽ

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

āE■æLŪēĀĒä;āēŁYāRrāzēāEŽāĠzāyÄāzŽçzŞāRLI/OæŞ■ä;IJçŽĐæūūāRLæŪzæāLīijŽ

```
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)

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

ēŁZēĠNçŽĐāEşēTōçCzāIJlāzŌāŌşāġNçŽĐçTşæLRāZlāĠ;æTŗāzūāy■ēIJĀēeAçşēeAŞā;ŁçTłçzEēŁĆīij

4.15 2.15 ā■Ūçņęäyşäy■æRŚāĒēāRŸéĠR

éŪŏécŸ

ä;āæCşāLZāzāyÄäyIæEĒātNāRŸéĠRçŽĐā■ŪçņęäyşīijNāRŸéĠRēcāōCçŽĐāĀijæL'ĀeāŁçd'żçŽĐā■Ū

èġċăEşşæŮzæąŁ

PythonăzŮæşşæIJL'ărzăIJlă■Ůçņęäyşäy■çõĂă■TæZŁæ■căRŸéĠRăĂijæŘŘăĭZçZt' æŌëçŽDæŤræŃAăĂăĭEæŸréĂŽèŁĠăĭŁçŤlă■ŮçņęäyşçŽD format() æŮzæşTæİëèġċăEşşèŁZäyİéŮŌéçŸăĂĆærŤăĊĭijŽ

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

æŁŮèĂĖĭijŃăĊăđIJèĊAèćnăZŁæ■ççŽDăRŸéĠRèĊĭăIJlăRŸéĠRăşşäy■æL'ăĬŤĭijŃéĆçăzĬăĭăăŘrăzèçzŞăŤlăĭŁçŤlă format_map() ăŞŃ vars()ăĂĆărşăĊŤăyŃéİçèŁZæăĭijŽ

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

vars() èŁŸæIJL'ăyĂăyĬæIJL'æĐŤæĂİçŽĐçL'zæĂġărşæŸŤăŌĊăzŃşéĂĆçŤlăzŌăŤzèşăăŌđăĭŃăĂĆærŤă

```
>>> 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 ăŞŃ format_map() çŽDăyĂăyĬçijžéZăăŤşæŸŤăŌĊăzŃăzŮăy■èĊĭăĬLăèçŽDăđ'ĐçŤEăŤŸéĊ

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

ăyĂçġ■éAŁăĂ■èŁZçġ■éŤŽèŤŤçŽDæŮzæşTæŸŤăŤăđ' ŮăŌŽăzL'ăyĂăyĬăŤŤæIJL'__missing__() æŮzæşTçŽDă■ŮăĖyărzèşăĭijŃăŤşăĊŤăyŃéİçèŁZæăĭijŽ

```
class safesub(dict):
    """éŸşă■keyăL'ĭăy■ăĬŤ"""
    def __missing__(self, key):
        return '{' + key + '}'
```

çŌŤăIJlăĭăăŘrăzèăĬŤlăŁçZăyĬçşăŤŃĖèçĖèŁŞăĖĖăŤŖŌăĭjæĂŞçzŽ format_map() ĭijŽ

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

æCædIJä;ääRŖäzëäČRäyNéIcéfZæuüäEŽäzEijŽ

```
import sys

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

çŖäIJlä;ääRŖäzëäČRäyNéIcéfZæuüäEŽäzEijŽ

```
>>> 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}
>>>
```

ëöleöž

äd'Žäzt'äzëæIëçTšäzŖPythonçijžäzRärzäRŸéGRæZfæ■çŽDäEĚç;õæTræNÄæÄNärijeGt'äzEäRĎçg■äy
ä;IJäyžæIJñëLČäy■äsTçd'žçŽDäyÄäyIäRrëČ;çŽDëgčäEşæŮzæaLijNä;ääRŖäzëæIJL'æŮüäÄŽäijŽçIJNäLräČ

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

ä;ääRŖëČ;èfYäijŽçIJNäLrä■ŮçñäyşæIæIëçŽDä;ççTliijŽ

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

çDüëÄNrijN format() äŠN format_map() çŽyæfTè;ČäyLéIcéfZäzZæŮzæaLèÄNäüšæZt'äLäaÉL
ä;ççTl format() æŮzæşTèfYæIJL'äyÄäyIäe;äd'DärsæYrä;ääRŖäzëèŮüä;Ůärfzä■ŮçñäyşæäijäijRäNŮçŽD
èÄNëfZäzZçL'zæÄgæYrä;ççTlāČRæIæIä■ŮçñäyşäzNçşçŽDæŮzæaLäy■äRŖëČ;èŮüä;ŮçŽDäÄC

æIJñæIJzèfYéČlāLEäzNçz■äzEäyÄäzŽénYçžççL'zæÄgäÄCæYäärDæLŮëÄĚä■ŮäEÿçşzäy■ésIJäyžäzž
__missing__() æŮzæşTäRŖäzëèõl'ä;ääŖŽäzL'æçCä;Täd'DçRĚçijžäd'şçŽDäÄijäÄC äIJl

SafeSub çşzäy■iijNëfZäylæŮzæşTëcñáoŽžzL'äyžárzçijžad'şçŽDāAijèfTāZđäyÄäylā■ää;■çñēāĀĆ
 äjāāRřazēāRŠçŎřçijžad'şçŽDāAijäijŽāGžçŎřāIJlçzŞæđIJā■Ůçñēäyşäy■(āIJlërÇerTçŽDæŮūāĀŽāRřēČ;ā;Lā
 KeyError äijCāyŷāĀĆ

```
sub() āĜ;æTřä;fçTÍ sys.__getframe(1) èfTāZđërČçTíèĀĚçŽDæāLāyğāĀĆāRřazēāzŎäy■èøŁéŮ
f_locals ælëèŎūā;ŮāšĀéCíāRŸéGRāĀĆ ærñæŮāçŮŠéŮōçzÍāđ'ğéCíāLĒæČĒāĒtäyNāIJlāzççāĀäy■āŎžç
ä;ĒæŸřijNārzažŎāČRā■ŮçñēäyşæZŁæ■cāuēāĒūāĜ;æTřēĀNēlĀāōČæŸřēlđäyŷæIJL'çTÍçŽDāĀĆ
āRēāđ'ŮřijNāĀijā;ŮāşlæĐRçŽDæŸř f_locals æŸřäyĀäylāđ'■āLūērČçTÍāĜ;æTřçŽDæIJñāIJřāRŸéGRçŽ
ār;çōāā;āāRřazēāTzāRŸ f_locals çŽDāĒĒāōžijNā;ĒæŸřēfZäylāŁōæTzārzažŎāRŎēlççŽDāRŸéGRēøŁé
æL'ĀāžēřijNēŽ;èrt'èøŁéŮōäyĀäylæāLāyğçIJNāyLāŎžā;LéCíæĀřijNā;ĒæŸřārzažŎČçŽDāzžā;TæŞ■ā;IJäy■ā
```

4.16 2.16 äžēæNĜāōŽžĀLŮāō;æäijāijRāNŮā■Ůçñēäyş

éŮōécŸ

äjāæIJL'äyĀāžZéTŁā■ŮçñēäyşřijNæČşäzēæNĜāōŽçŽDāLŮāō;ārĒāōČčāznēĜ■æŮřæäijāijRāNŮāĀĆ

èğčāĒşæŮzæāŁ

ä;fçTÍtextwrap ælāāĀŮælēæäijāijRāNŮā■ŮçñēäyşçŽDē;ŞāĜžāĀĆærTæČřijNāĀĜæČā;āæIJL'äyNā

```
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."
```

äyNēlćæijTçđ'žā;fçTÍtextwrap æäijāijRāNŮā■ŮçñēäyşçŽDād'Žçğ■æŮžāijRřijŽ

```
>>> 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.
```

ěóíèőž

```
textwrap ħġaġiŮaržāžŌā■ŮčņäyšæL'Sā■ræŸréġāyŷæIJL'çŦlçŽDġijŊçL'žāĻnæŸrā;Šā;āāyŊæIJŽē;S
ā;āāRfāžēā;ŧçŦġ os.get_terminal_size() æŰžæŧæŦæĲēŌūāRŮçzĻçŋrçŽDād'ğārRāržāŷāĀĆærŦæĆ
```

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

```
fill() æŰžæŧæŦæŌēāRŮäyÄāžZāĒūāzŮāRréĀĻāRĆæŦræĲæŌğāĻŮtabġijŊēr■āRēçzŠār;ç■Ļ'āĀĆ
ārĆēŸĒ textwrap.TextWrapperæŰĢæç ēŌūāRŮæŽŦād'ŽāĒĒāōžāĀĆ
```

4.17 2.17 āĲġā■Ůčņäyšäy■ād'DçŘĒhtmlāŠŊxml

éŮőécŸ

```
ā;āæČšārĒHTMLæĻŮēĀĒXMLāōđā;ŠæĆ &entity; æĻŮ &#code;
æŽĒæ■cāyžāržāžŦçŽDæŰĢæIJñāĀĆ āĒēĀĒġijŊā;āēIJĀēæĲēñæ■cæŰĢæIJñäy■çĻ'žāōŽçŽDā■Ůčņæ(ærŦæĆ
>, æĻŮ &)āĀĆ
```

èğčāĒşæŰžæąĻ

```
āēĆæđIJā;āæČşæŽĒæ■cæŰĢæIJñā■Ůčņäyšäy■çŽD āŸŸ<āŸŹ æĻŮēĀĒ āŸŸ>āŸŹ
ġijŊā;ŧçŦġ html.escape() āĢ;æŦŦāRfāžēā;ĻāōžæŸŞçŽDāōŊæĻŦāĀĆærŦæĆġijŽ
```

```
>>> 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 '<tag>text</tag>'.

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

```
āēĆæđIJā;āæ■cāĲġād'DçŘĒçŽDæŸŦASCIIæŰĢæIJñġijŊāžūāyŦæČšārĒēĲāŮASCIIæŰĢæIJñāržāžŦçŽDç
ārŦāžēçzŹæŞŦāžŹĻŮāĢ;æŦŦāġijāēĀŦāRĆæŦŦ errors='xmlcharrefreplace'
æĲēē;āĻŦēŧŹāyĻçŽōāĀĆærŦæĆġijŽ
```

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

äyžāẸæẒæ■céŨĜæIJñäy■çŽĎçijŮčāAāōđä;ŠiijŇä;æéIJĎēAä;ŁçŤlāŘēāđ'ŮäyĀçĝ■æŮzæšŤāĀĆ
 āēĆāđIJä;ăæ■cāIJlād'ĐçŘĚHTMLæĹŮèĀĚXMLæŮĜæIJñijŇēřŤçlĀāĚĹä;ŁçŤlāyĀäyĹāŘĹéĀĆçŽĎHTML
 éĀŽāyŷæĈĚāĒŷyŇijŇēŁŽāžZāūēāĚūāijŽēĠlāĹlæŽŁæ■céŁŽāžŽçijŮčāAāĀijñijŇä;ăæŮæéIJæŇĚāŁĈāĀĆ

æIJL'æŮūāĀŽiijŇāēĆāđIJä;ăæŌēæŤūāĹrāžĒäyĀāžZāŘŇæIJL'çijŮčāAāĀijçŽĎāŌšāĝŇæŮĜæIJñijŇēř
 éĀŽāyŷä;ăāŘĹéIJĎēAä;ŁçŤlĪHTMLæĹŮèĀĚXMLēĝçæđŘāŽlçŽĎäyĀāžŽçZyāĚšāūēāĚūāĠ;æŤř/æŮzæšŤā

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

ēōlēōž

āIJlçŤšæĹŖHTMLæĹŮèĀĚXMLæŮĜæIJñçŽĎæŮūāĀŽiijŇāēĆāđIJæ■ççāōçŽĎē;Ňæ■céŁ'zæōŁæāĜēō
 çĹ'žāĹŇæŸřā;Šā;ăä;ŁçŤlĪprint() āĠ;æŤřæĹŮèĀĚāĚūāžŮā■ŮçŇēäyšæāijāijŘāŇŮæĹēāžĝçŤšēçŠāĠžçŽĎā
 ä;ŁçŤlāĈŘhtml.escape() çŽĎāūēāĚūāĠ;æŤřāŘrāžēāĠ;ĹāōžæŸšçŽĎēĝçāĒšēŁŽçšzéŮōécŸāĀĆ

āēĆāđIJä;ăæĈšāžēāĚūāžŮæŮzāijŘād'ĐçŘĒæŮĜæIJñijŇēřŸæIJL'äyĀāžZāĚūāžŮçŽĎāūēāĚūāĠ;æŤřā
 xml.sax.saxutils.unescape() āŘrāžēāyōāĹl'ä;ăāĀĆ
 çĎūēĀŇijŇä;ăāžŤēřāĚĹēřĈçāŤäyĒæēŽæĀŌæāūā;ŁçŤlāyĀäyĹāŘĹéĀĆçŽĎēĝçæđŘāŽlāĀĆ
 æřŤāēĈiijŇāēĆāđIJä;ăāIJlād'ĐçŘĚHTMLæĹŮXMLæŮĜæIJñijŇ
 ä;ŁçŤlāēšŘāyĹēĝçæđŘāĹāĹŮæřŤāēĈhtml.parse æĹŮ xml.etree.ElementTree
 āūšçžŘāyōā;ăēĠlāĹlād'ĐçŘĒæžĒçŽyāĚšçŽĎæŽŁæ■ççzĒēŁĈāĀĆ

4.18 2.18 ā■ŮçŇēäyšāzd'çĹŇēĝçæđŘ

ēŮōécŸ

ä;ăæIJL'äyĀäyĹā■ŮçŇēäyšijŇæĈšāžŮāūēēĠšāŘšārĒāĚūēĝçæđŘäyžäyĀäyĹāzd'çĹŇæřĀāĀĆ

ēĝçāĒšæŮzæāĹ

āĀĠæĈā;ăæIJL'äyŇéĹcéŁžæāūäyĀäyĹæŮĜæIJñā■ŮçŇēäyšijŽ

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

äyžāẒEāzđ'çL'ÑāÑŨā■ŮčņēāyšīijNā;āāy■āzĒēIJĀēēAāÑzéĒ■āēlāāijRīijNēfYā;ŮāēNĠāōŽāēlāāijRçŽĐç
æŕTāēČīijNā;āāRŕēČ;æČšāŕEā■ŮčņēāyšāČRāyNēlčēfZāāūē;ñā■čāyžāžRāLŮārziijŽ

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

äyžāẒEāL'gēāNēfZāāūçŽĐāLĠāLēīijNçññāyĀā■ēārśæYŕāČRāyNēlčēfZāāūāLl'çTlāŚ;āR■ā■TēŌūçz

```
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]))
```

āIJlāyLēlčçŽĐāēlāāijRāy■īijN ?P<TOKENNAME> çTlāžŌçzZāyĀāyLāēlāāijRāŚ;āR■īijNā;ZāRŌēlčā;ççT

äyNāyĀā■ēīijNāyžāẒEāzđ'çL'ÑāÑŨīijNā;ççTlāēlāāijRāržēsāā;LārŚēcñāžççšēēAšçŽĐ
scanner() æŮzæšTāĀČ èfZāyLæŮzæšTāijŽāLZāžzāyĀāyL
scanner āržēsāīijN āIJlèfZāyLāržēsāāyLāy■æŮ■çŽĐèŕČçTl match()
æŮzæšTāijŽāyĀā■ēā■ēçŽĐāL'ñāRŔçŽōāāĠæŮĠæIJñīijNæŕRāē■āyĀāyLāÑzéĒ■āĀČ
äyNēlčæYŕāijTçđ'žāyĀāyL scanner āržēsāāçČā;Tāūēā;IJçŽĐāzđ'āžŠāijRā;Nā■RīijŽ

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

(continues on next page)

(continued from previous page)

```
>>>
```

āōđéŽĚä;ŁçŦlèŁŻçġæŁĀæIJŕçŽĎæŮŭāĀŽiijŇāŔŕāzēā;ŁāōžæŸŞçŽĎāČŔäyŇéÍcèŁŽæăŭăŕĒäyŁèŁŕāz

```
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')
```

āēČæđIJä;ăæČşèŁĠæzd'ăzd'çŁŇăŦĀiijŇă;ăăŔŕāzēāōŽăzŁ'æŽt'ăđ'ŽçŽĎçŦşæŁŔăŽÍăĠ;æŦŕæŁŮèĀĚă;æŕŦăēČiijŇăyŇéÍcæijŦçđ'žæĀŌæăŭèŁĠæzd'æŁĀæIJLçŽĎçŁ'žçŽ;ăzd'çŁŇiijŽ

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

èóíèőž

éĀŽăyŷæÍèèőšăzd'çŁŇăŇŮæŸŕă;Łăđ'ŽénŸçžġæŮĠæIJñèġçăđŔăyŌăđ'ĎçŔĒççŽĎçñăyĀæ■ēăĀČăyžăžĒä;ŁçŦlăyŁéÍcçŽĎæŁ'ŇăŕŔæŮžæşŦiijŇă;ăéIJăēçĀēōŕă;ŔèŁŽéĠŇăyĀăžŽéĠ■ēçĀçŽĎăĠăçČžăĀČçñăyĀçČžăŕşæŸŕă;ăăŁĒéăžçăōēōđ'ă;ăă;ŁçŦlă■čăŁŽèălè;ăiijŔæŇĠăōŽăžĒæŁĀæIJLè;ŞăĒēăy■ăŔŕēČ;ăĠăēČăđIJæIJLăžză;Ŧăy■ăŔŕăŇzéĒ■çŽĎæŮĠæIJŇăĠççŌŕăžĒiijŇăŁ'ŇăŕŔăŕşăiijŽçŽt'æŌēăĀIJæ■čăĀČèŁŽă

ăzd'çŁŇçŽĎéăžăžŔăžşæŸŕæIJLă;şăŞ■çŽĎăĀČ re æłăăłŮăiijŽæŇLçĒġæŇĠăōŽăē;çŽĎéăžăžŔăŌžăĀăžăæ■đ'iijŇăēČăđIJăyĀăyłæłăiijŔæĀŕăē;æŸŕăŔēăyĀăyłæŽt'éŦŁæłăiijŔçŽĎă■Ŕă■ŮçŇēăyşŦiijŇéČčăžŁă;ăē

```
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
```

çñăžăŇăyłæłăiijŔæŸŕéŦŽçŽĎiijŇăŽăăyžăōČăiijŽăŕĒæŮĠæIJŇ<=ăŇzéĒ■ăyžăzd'çŁŇLTçŦ'ġèŭşçĀĒEQŕæIJăăŔŌiijŇă;ăéIJăēçĀçŦŽæĎŔăyŇă■Ŕă■ŮçŇēăyşă;çăiijŔçŽĎæłăiijŔăĀČæŕŦăēČiijŇăĀĠèō;ă;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')

```

PyParsing module provides a simple way to create a parser for a language. It is a part of the `PLY` (Python Lex-Yacc) package. The `PLY` package is a collection of modules that provide a simple way to create a parser for a language. The `PLY` package is a collection of modules that provide a simple way to create a parser for a language.

4.19 2.19 `PLY` module

4.19.1 `PLY` module

The `PLY` module provides a simple way to create a parser for a language. It is a part of the `PLY` (Python Lex-Yacc) package. The `PLY` package is a collection of modules that provide a simple way to create a parser for a language.

4.19.2 `PLY` module

The `PLY` module provides a simple way to create a parser for a language. It is a part of the `PLY` (Python Lex-Yacc) package. The `PLY` package is a collection of modules that provide a simple way to create a parser for a language.

```

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

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

factor ::= ( expr )
        | NUM

```

The `PLY` module provides a simple way to create a parser for a language. It is a part of the `PLY` (Python Lex-Yacc) package. The `PLY` package is a collection of modules that provide a simple way to create a parser for a language.

```

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

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

factor ::= ( expr )
         | NUM

```

çÖrãIJĩĩijÑãçCædIJä;ããrzBNFçŽDãũçä;IJæIJžãLũeŁYäy■æYřãŁæYŎçŽ;çŽDẽrĩĩijÑãrsæŁŁãŏCã;ŠãA
 eŁÑãlẽeõsĩijNẽgçædRçŽDãŎççRĖãřsæYřã;ããL'çTĩIBNFãŏNæLŘad'ŽãylæŽŁæ■cãŠNæL'ãŠTãžẽãNžẽ
 ãŽæiijTçd'žiiijÑãAĝeõŁä;ãæ■cãIJlẽgçædŘã;çãeC 3 + 4 * 5 çŽDẽalẽŁä;ãijRãĀC
 äylẽalẽŁä;ãijRãĒLẽeAẽĀŽẽŁGã;ŁçTĩ2.18eŁCäy■ãžNçž■çŽDæŁĀæIJřãLĖgçäyžãYĀçžDãžd'çL'ÑætĀãĀ
 edIJãRřẽC;æYřãČRäyNãLũeŁŽæũçŽDãžd'çL'ÑãžRãLŮĩijŽ

ǎIjæ■' aʂzçAäyŁiiŃ ěğcæđŘāŁlā; IjāijZērTçiAăŌzéĂŽefĜăŻfæ■céS■ä; IjāNzéĚēr■æsTālRè; ŠăĚ

äyÑéíçæL'ÄæIJL'çŽĐëğçædRæ■ēēld'āRrēČ;éIJĀēēAēŁšçCzæUūēŮr'āijDæYŌçŽ;īijNā;EæYřāōČāznāČ
 çññāyÄāyŕē;ŠĀēēāzd'çL'NæYřNUMīijNāZāæ■d'æŽŁæ■céēŮāĒLaijŽāNzéĒ■ēČçāyŕēČlāŁEāĀČ
 äyÄāUēāNzéĒ■æLŘāŁšīijNāršāijŽēfZāĒēäyNāyÄāylāzd'çL'N+īijNāzēæ■d'çšzæŌlāĀČ
 ā;ŠāušçzRçāōāōŽāy■ēČ;āNzéĒ■āyNāyÄāylāzd'çL'NçŽDæUūāĀŽīijNāršē;žçŽĐēČlāŁE(æřTāēČ
 { (* /) factor } *)āršāijŽēcnāyĒēçRĒæŌLāĀČ āIJāyÄāylæLŘāŁšçŽĐëğçædRāy■īijNæTt'āylāRšē;žç

æIJL'äZĖĀL■ēIcčŽDčšēēfĖēČŅæŽriiŃäyŃēIcæĻSāznäy;ävÄäyĭcōĀā■Tčd'žä;ŃæIēāsTčd'žæČä;Tædl

(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>

```


(continued from previous page)

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

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

èóìèőž

æŮGæIJñèġcædŘæŸräyÄäyłäŁad'ğçŽDäyzécŸtījŇäyÄeŁnäijŽā■āçŦlā■ēçŦšā■ēāzāçijŮerŠerçlNæŮ
āēČædIJä;āāIJæLçārzāĚšāžŌer■æşŦtījNèġcædŘçóŮæşŦç■LçŽyāĚşçŽDèČNæŽrçşēerEçŽDèrIrijŇä;āāžTèr
āçŁæŸçDūiijŇāĚşāžŌēfZæŮzéÍççŽDāĚĚāōžād'lad'ŽtījŇäy■ārèC;āIJlèfZéGŇāĚlÉČlāsŦāijĀāĀC

ārçóāāēČæ■d'tijŇçijŮāĚŽäyÄäyłéĀšā;ŠäyŇéŽ■èġcædŘāZlçŽDæŦt'ä;ŞæĀlèurfæŸræŦTèçÇçóĀā■ŦçŽ
āijĀāğNçŽDæŮūāĀŽtījŇä;āāĚLèŮūāçŮæL'ĀæIJLçŽDèr■æşŦtēğDāLŽtījŇçDūāRŌārEāĚūē;ñæ■cäyžäyÄäy
āZāæ■d'āēČædIJä;āçŽDèr■æşŦçşzāijijèfZæūiijŽ

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

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

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

ä;āāžTèrēēēŮāĚLārEāōČzñè;ñæ■cæLRäyĀçzDāČRäyŇéÍcèfZæūççŽDæŮzæşŦtījŽ

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

æŦRäyłæŮzæşŦtēçAāōŇæLRçŽDäzzāŁāāçŁçóĀā■Ŧ - āōČāfĚēāzāzŌāūçèGşāRşéA■āŌĚer■æşŦtēğDāLŽ

āzŌæšRċg■æĎRāzL'äyLèðšīijNæŪzæšTçŽDçŽōçŽĎārsæYřèçAāzL'ād'DçRĒāōNēr■æšTēgĎāLŽīijNēçAāzL'äyžāžEēfZæāuāAŽīijNéIJĀéGĠçTīāyNéIcçŽĎēfZāžZāōđçŌræŪzæšTīijŽ

- æĈædIJēgĎāLŽāy■çŽĎāyNāyIçņēāRūæYřāRēād'ŪāyĀāyIēr■æšTēgĎāLŽçŽĎāR■āŪ(ærTāçĆtermæēfZārsæYřēfēçŌŪæšTāy■āĀIāyNéZ■āĀIçŽDçT'sæIē -
æŌgāLŪāyNéZ■āLrāRēāyĀāyIēr■æšTēgĎāLŽāy■āŌzāĀĆ
æIJL'æŪāāŽēgĎāLŽāijZērČçTīāušçzRæL'gēāNçŽĎæŪzæšT(ærTāçĆīijNāIJl
factor ::= '('expr ')'
ēfZārsæYřçŌŪæšTāy■āĀIēĀSā;ŠāĀIçŽDçT'sæIēāĀĆ
āy■ārēzexprçŽĎērČçTī)āĀĆ
- æĈædIJēgĎāLŽāy■āyNāyĀāyIçņēāRūæYřāyIçL'zæŌLçņēāRū(ærTāçĆ())īijNā;āā;ŪæšēæL'çāyNāyĀāy
æĈædIJāy■āNžēĒ■īijNārsāžgçT'sāyĀāyIēr■æšTēTŽērrāĀĆēfZāyĀēLÇāy■çŽĎ
_expect() æŪzæšTārsæYřçTīāIēāAŽēfZāyĀæ■ēçŽĎāĀĆ
- æĈædIJēgĎāLŽāy■āyNāyĀāyIçņēāRūāyžāyĀāzZāRrēČ;çŽĎēĀL'æNl'ēāz(ærTāçĆ +
æLŪ-)īijNā;āāfĒēāzārsæfRāyĀçg■āRrēČ;æĈēāĒtāçĀæšēāyNāyĀāyIāz'd'çL'NīijNāRlæIJL'ā;ŠāŌČāN
ēfZāzšæYřæIJnēLÇçd'žā;Nāy■ _accept() æŪzæšTçŽDçŽōçŽĎāĀĆ
āŌČçŽyā;ŠāžŌ_expect()æŪzæšTçŽĎāijsāNŪçL'LæIJīijNāZāāyžæĈædIJāyĀāyIāNžēĒ■æL'çāLrāžEā
ā;EæYřāçĈædIJæšqæL'çāLīijNāŌČāy■āijZāžgçT'sēTŽērrēĀNæYřāZdæzŽ(āĒAēōyāRŌçz■çŽĎæçĀæ;
ā;EæYřāçĈædIJæšqæL'çāLīijNāŌČāy■āijZāžgçT'sēTŽērrēĀNæYřāZdæzŽ(āĒAēōyāRŌçz■çŽĎæçĀæ;
- ārzāžŌæIJL'ēG■ād'■ēČlāLēçŽĎēgĎāLŽ(ærTāçĆāIJlēgĎāLŽēālēçāijR ::= term {
('+' | '-') term } * āy■īijNēG■ād'■āLlā;IJēĀŽēfGāyĀāyIwhileā;IçŌrāIēāŌđçŌrāĀĆ
ā;IçŌrāyžā;ŠāijZæTūēZēæLŪād'DçRĒæL'ĀæIJL'çŽĎēG■ād'■āĒČçT'āçZt'āLræšqæIJL'āĒūāzŪāĒČçT'ā
ā;IçŌrāyžā;ŠāijZæTūēZēæLŪād'DçRĒæL'ĀæIJL'çŽĎēG■ād'■āĒČçT'āçZt'āLræšqæIJL'āĒūāzŪāĒČçT'ā
- āyĀæŪçæTt'āyIēr■æšTēgĎāLŽād'DçRĒāŌNæL'RīijNærRāyIæŪzæšTāijZēfTāZdæšRċg■çzšædIJçzŽē
ēfZārsæYřāIJlēgçædRēfGçlNāy■āĀijæYřæĀŌæāuçt'fāLāçŽĎāŌšçRĒāĀĆ
ærTāçĆīijNāIJlēālēçāijRæšČāĀijçlNāzRāy■īijNēfTāZdāĀijāzçēālēālēçāijRēgçædRāRŌçŽĎēČlāLēç
æIJĀāRŌæL'ĀæIJL'āĀijāijZāIJlæIJĀēāuāsČçŽĎēr■æšTēgĎāLŽæŪzæšTāy■āRlāzūētūāIēāĀĆ

ār;çŌāāRŠā;āæijTçd'žçŽĎæYřāyĀāyIçŌā■TçŽĎā;Nā■RīijNēĀSā;ŠāyNéZ■ēgçædRāZlāRrāzēçTīāIēā
ærTāçĆīijNPythonēr■ēlĀæIJnēžnārsæYřēĀZēfGāyĀāyIēĀSā;ŠāyNéZ■ēgçædRāZlāŌzēgçēGēçŽĎāĀĆ
æĈædIJā;āārzæ■d'æĎšāĒt'ēūčīijNā;āāRrāzēēĀZēfGæšēçIJNPythonæžRçāAæŪGāzūGrammar/Grammarēl
çIJNāŌNā;āāijZāRŠçŌīijNēĀZēfGæL'NāLlæŪzāijRāŌzāŌđçŌrāyĀāyIlēgçædRāZlāĒūāŌđāijZæIJL'ā;Lād'Žç

āĒūāyāyĀāyIāSĀēZĎārsæYřāŌČāznāy■ēČ;ēcncTlāžŌāNēāRnāzžā;TāūēēĀSā;ŠçŽĎēr■æšTēgĎāLŽāy

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

āyžāžEēfZæāuāAŽīijNā;āāRrēČ;āijZāČRāyNéIcēfZæāuā;fçTl items() æŪzæšTīijŽ

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

āTřāyĀçŽĎēŪŌēçYæYřēfZāyIæŪzæšTæāžæIJnāy■ēČ;āūēā;IJīijNāzNāŌđāyLīijNāŌČāijZāžgçT'sāyĀāy
āĒšāžŌēr■æšTēgĎāLŽæIJnēžnā;āāRrēČ;āžšāijZççrāLrāyĀāzZæçYæL'NçŽĎēŪŌēçYāĀĆ
ærTāçĆīijNā;āāRrēČ;æČçšēēĀšāyNéIcēfZāyIçŌā■TæL'ijēr■æšTæYřāRēēālēfřā;Ūā;ŠīijŽ

```

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

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

```

èfZäylèr■æşTçIJNäyLăŌzæşqāTēēŪōécŸiijNä;EæYřăŌČă■t'äy■èČ;ărşèğL'ăĹræăĜăĜĖăZăĹZăēfŘçŌ
 æřTăēČiijNēaĹēĹĹăijR "3 + 4 * 5" äijŽăĹŪăĹr35ēĂNäy■æYřæIJşæIJZçŽĎ23.
 ăĹEăijĂă;ĤçTĹăĂlexprăĂİăŞNăĂİtermăĂİēğĐăĹZăRřăzēēŌ'ăŌČă■čçăŌçŽĎăŭčă;IJăĂĆ
 ărzăžŌăđ'■ăİČçŽĎēr■æşTiiijNä;ăæIJĂăē;æYřăĹL'æNĹ'æşRăyĹēğčăđRăŭēăĖŭærTăēČPyParsingăĹŪēĂ
 äyNēİēæYřă;ĤçTĹPLYăİēēĜ■ăĖZăēaĹēĹĹăijRăşČăĂijçĹNăžRçŽĎăžčçăĂiijŽ

```

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()

```

èfZäyłçlNäZRäy■rijNæL'ÄæIJL'äzççäAéČj;ä;■äžÖäyÄäyłærTè;ČénYčŽDásĆæñqāĀĆä;ääRłéIJĀèçAäy;
 èĀNāōđéŽĚčŽDèfRēqNèğčæđŘāŽlījNæŌēāRŪāzd'çL'Ňç■L'ç■L'āžTāsĆāLía;IJāũščzRēcñāžŠāĜ;æTŗāōđçČ
 äyNéIcæYřayÄäyłæĀŌæăüă;£çTlā;UāLřçŽDèğčæđŘārżèśaçŽDä;Nā■ŘiijŽ

æĆæđIĲă;ăăĈșăIĲă;ăĉŽĐċijŮĉlŇěĜĉlŇăy■æIēĆĈzăŇŚăĹŸăŚŇăĹzăĤĂġijŇċijŮăEŽēĝĉăđŘăŽlăŚŇă
ăE■ăăġijŇăyĂăIĲŇċijŮērŚăŽlĉŽĐăžēĉș■ăijŽăŇĚăŔăŇăĹăđ'ŽăŽŤăśĈĉŽĐĉRĚēōžĉșēērĚăĂĈăy■ēĜăĹăđ'
PythonēĜăŭșĉŽĐăstăĹăăIŮăžșăĂġă;ŮăŮžĉIĲăyĂăyŇăĂĈ

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

èõìèõž

ād'gād'ŽæTṛæČĚāĖtāyNījNāIJlæŮĜæIJnā■ŮčņäyšāyŁçŽDæŠ■ā;IJāIĜāRfçTlāžŌā■ŮèŁCā■Ůčņäyšā
çDūèĀNījNēfZēGNāžšæIJLāyĀāžZēIJĀēēAæšlæDRçŽDāy■āRŃçCzāĀCēēŮāĒLījNā■ŮèŁCā■Ůčņäyšç

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

èfŽçğ■èr■āzL'āyŁçŽDāNžāLñāijŽāržāžŌād'DçRĖēIcāRŠā■ŮèŁCçŽDā■ŮčņæTṛæ■ōæIJL'ā;šāŠ■āĀC
çñnāžNçCžījNā■ŮèŁCā■Ůčņäyšāy■āijŽæRĀz;ZāyĀāyŁç;ŌèğCçŽDā■ŮčņäyšēāŁçd'žījNāžšāy■ēČ;ā

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

çszāijijçŽDījNāžšāy■ā■ŮāIJlāzā;TēĀCçTlāžŌā■ŮèŁCā■ŮčņäyšçŽDæāijāijRāNŮæŠ■ā;IJījŽ

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

(continues on next page)

(continued from previous page)

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

æĈædIJä;äæĈsæäijäijRāNŨā■UēLCā■UçņäyšrijNä;äâ;UāĒLä;ĒçTlæĀGāĠEçŽDæŨGæIJñā■Uçņäyš

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

æIJĀāRŌéIJĀèēAæslæDRçŽDæYrijNä;ĒçTlā■UēLCā■UçņäyšāRrēČ;äijZæTzāRŸäyĀāžZæS■ä;IJçŽL
æŕTæČrijNäæĈædIJä;äâ;ĒçTlāyĀäyĻçijŨçāAäyžā■UēLCçŽDæŨGāzūāR■rijNēĀNäy■æYŕäyĀäyĻæŽōéĀŽçŽ

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

æslæDRä;Nā■Räy■çŽDæIJĀāRŌéČlāĒEçzŽçŽōā;TāR■äijæĀŠäyĀäyĻā■UēLCā■UçņäyšæYŕæĀŌæāŨ
āIJlçŽōā;Täy■çŽDæŨGāzūāR■āNēāRñāŌšāĠNçŽDUTF-8çijŨçāAāĀČ
āRČēĀČ5.15ārRēLCēŌūāRŨæŽŕād'ŽæŨGāzūāR■çŽyāĒšçŽDāĒēāōzāĀČ

æIJĀāRŌæRŕäyĀçCzrijNäyĀāžZçlNāzRāSŸäyžāzEæRŕā■ĠçlNāzRæL'ġēāNçŽDēĀšāžēäijZāĀ;āRŠā
ār;çōāæS■ä;IJā■UēLCā■UçņäyšçāōāōđäijZæŕTæŨGæIJñæŽŕāĒénŸæTl(āŽāäyžād'DçRĒæŨGæIJñāŽžæI
ēfZæāūāAŽéĀŽāyāijZārijēGrēlđäyāēlCāzšçŽDāzççāAāĀČä;ääijZçzRāyāRŠçŌŕā■UēLCā■Uçņäyšāzūāy
āzūāyTā;āēfŸā;ŨæL'NāĒlāđ'DçRĒæL'ĀæIJLçŽDçijŨçāA/ēġççāAæS■ä;IJāĀČ
āĒçç;ēōšrijNäæĈædIJä;āāIJlāđ'DçRĒæŨGæIJñçŽDēlrijNāršçŽŕæŌēāIJlçlNāzRäy■ä;ĒçTlæŽōéĀŽçŽDæŨG

5 çņnäyL'çnäijZæTŕā■UæUěæIJšāŠNæUúéŮŕ

āIJlPythonäy■æL'ġēāNæTŕæTŕāŠNæŭçCzæTŕçŽDæTŕā■ēēRçōŮæUūā;ĒçōĀā■TçŽDāĀČ
ār;çōāæČæ■d'rijNäæĈædIJä;āéIJĀèēAæL'ġēāNāĒEæTŕāĀAæTŕçzDæLŮēĀæYŕæŮēæIJšāŠNæUúéŮŕçŽD
æIJñçnäēZĒäy■ēōlēōžçŽDāršæYŕēfZāžZäyžēcŸāĀČ

Contents:

5.1 3.1 æṬṙǎ■ŬçŽĐǎŽŽèĹ■ǎžṬǎĔĕ

éŬóéćŸ

äĳǎæĈşǎřzæṭõçĆzæṬṙǎĹ'ğèǎŊæŊŊǎōŽçşĳǎžèçŽĐèĹ■ǎĔĕèĔŔçõŬǎǺĆ

èğĉǎĔşǎŬzæǎĹ

ǎřzǎžŎçõǺǎ■ṬçŽĐèĹ■ǎĔĕèĔŔçõŬĳĳŊǎĳçṬĹǎĔĔçĳõçŽĐ round(value, ndigits) ǎĜĳæṬṙǎ■şǎŔřǎǺĆæŕṬǎèĆĳĳŽ

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

ǎĴşǎŷǺǎŷĹǎǺĳǎĹŽǎèĳǎĴĹǎŷd'ǎŷĹèĳçṬŊçŽĐǎŷ■éŬŕçŽĐæŬŭǎǺĳĳŊŊ round
ǎĜĳæṬṙèĔṬǎžĐçèzǎōĆæĴǺèĔŔçŽĐǎŬǎæṬṙǎǺĆ ǎžşǎřsæŸŕèŕ'ĳĳŊǎřz1.5æĹŬèǺĔ2.5çŽĐèĹ■ǎĔĕèĔŔçõŬéĆ

ǎĳĳǎçŽŽ round() ǎĜĳæṬṙçŽĐ ndigits ǎŔĆæṬṙǎŔřǎžèæŸŕèŕ şæṬĳĳŊŊèĔŽçğ■ǎĈĔǎĔǎŷŊŊĳŊŊ
èĹ■ǎĔĕèĔŔçõŬǎĳžǎĳĴçṬĹǎĴǎ■Ǻǎ■ǎǺǺçŽçĳ■ǎǺǺǎ■Ĉǎ■ç■ĹǎŷĹéĳǎǺĆæŕṬǎèĆĳĳŽ

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

èóĹèőž

ǎŷ■èçǺǎŕĔèĹ■ǎĔĕǎŊŊǎĳĳŔǎŊŬèĳşǎĜžæŔđæŭŭæŭĔǎžĔǎǺĆ
ǎèĆǎđĴǎĳçŽĐçŽõçŽĐǎŔĹæŸŕçõǺǎ■ṬçŽĐèĳşǎĜžǎŷǺǎōžǎōĳǎžèçŽĐæṬĳĳŊŊǎĳǎŷ■éĴǺèçǺǎĳçṬĹ
round() ǎĜĳæṬṙǎǺĆ èǺŊǎžĔǎžĔǎŔĹéĴǺèçǺǎĴĴǎĳĳĳŔǎŊŬçŽĐæŬŭǎǺžæŊŊǎōžçşĳǎžèǎ■şǎŔřǎǺĆæŕṬ

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

(continues on next page)

(continued from previous page)

```
>>> 'value is {:.3f}'.format(x)
'value is 1.235'
>>>
```

ǎRñæuüijNäy■ēēAērTçIǎǎŌzēL■ǎĚēætōçCzǎĀijǎlēǎĀIǎǎŌǎ■čǎĀIēǎléIcǎyŁçIJNētūǎlēǎ■čçǎoçŽǎDēŁ

```
>>> a = 2.1
>>> b = 4.2
>>> c = a + b
>>> c
6.3000000000000001
>>> c = round(c, 2) # "Fix" result (???)
>>> c
6.3
>>>
```

áržāžŌād' gād' ŽæȚřaj; ǣçȚlǎLřæȚōçĆççŽDćlNāžRiijNǣšqæIJL' ǎĚĚeçAǎžšǎy■æŌlè■ŘeřŽæuǎAŽǎǎĆ
 ǎř;çōǎǎIJlĚōǎçōŮçŽDæUǎǎĀŽǎijŽæIJL' ǎyǎçĆççĆžǎRçŽDěřřǎuōiijNǎj;EǎŸřeřŽǎžŽǎRçŽDěřřǎuōǎŸřeç;ǐ
 ǎçCǎđIJǎy■ēç;ǎĒAēōyēřŽæǎuçŽDǎRřeřřǎuō(ǎřȚǎēCǎul'ǎRĚǎLřeĜSēđ■éçEǎšš)iiijNéCčǎžLǎřsǎ;ŮēǎCēŽ
 decimal ælǎǎlŮǎžEiijNǎyNǎyǎēLçCǎLšǎžnǎijŽeřeççEēōlēōžǎǎĆ

5.2 3.2 æL'ǵeaŃçšŁçaoćŻDætōćĆzæTřèƐŘćóŮ

éŮőécÿ

ä:äeIJÄèeAårzætoćĆzæTræL'gèaÑçş;çaoćŽĐèoaçoŮæS■ä:IJijIÑázúäyŤäy■äyÑæIJZæIJL'äzzä;TårRèrr

èǧčǎẸșæŮźæǻŁ

æ̥t̥o̥ç̥Ćzæ̥T̥r̥ç̥ŽD̥äy̥Ḁ̈äy̥tæ̥Ž̥o̥é̥Ḁ■ē̥U̥o̥é̥ć̥Y̥æ̥Y̥r̥ḁo̥Č̥äz̥n̥ḁž̥u̥äy̥■ē̥Č̥;ç̥ş̥;ç̥ḁo̥ç̥Ž̥D̥è̥ḁl̥ç̥d̥'z̥ḁ■Ḁē̥Ł̥Z̥ḁl̥U̥ḁ̄T̥r̥ḁ̄Ā̇Ć̇
ā̇ž̇u̇äẏṪu̇j̇Ṅā̇■ṡä̇;Ł̇æ̇Ẏṙæ̇İJ̇ā̇ċō̇Ā̇■ṪċŽ̇Ḋæ̇Ṫṙā̇■ē̇ē̇Ł̇Ṙċō̇U̇ā̇ž̇ā̇j̇J̇ä̇ž̇ġċṪṧā̇ṘċŽ̇Ḋē̇ṙṙȧu̇ō̇i̇j̇Ṅæ̇ṙṪā̇ē̇Ć̇i̇j̇Ž̇

```
>>> a = 4.2
>>> b = 2.1
>>> a + b
6.3000000000000001
>>> (a + b) == 6.3
False
>>>
```

æŹăžŽēŤZėrræYřcŦśāžŦśĆCPUăŠÑIEEE754ăǺăĜăĖéĂŽēŹĜēĞlăũścŽDætőćĆză■Ţăj■ăŌzăLğèaŃ
 cŦśāžŎPythoncŽDætőćĆzăŦrăë■őcszādŇä;ŁcŦłāžŦśĆeălcd'ză■ŸăĆlăŦrăëōīīNăŻăăd'ă;ăæşaaŁdæsŦăŐ

æĈæđIä;äĈſæŽt'ăŁăſſ;çąǫ(ăžűëĈ;ăǫžăſ■ăſĂăǫŽçŽĐæĂğëĈ;æ■ſèĂŮ)üjNă;ăăRăžžă;ŁçŦĬ
decimal æłăăIŮüjŽ

```
>>> 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
```

decimal module provides a way to represent floating point numbers with arbitrary precision. It is useful when you need to perform calculations that require high precision, such as financial calculations or scientific computations. The Decimal class is used to create decimal numbers, and it can be used in arithmetic operations like addition, subtraction, multiplication, and division.

decimal module also provides a way to control the precision of calculations. You can use the `getcontext()` method to retrieve the current context, and then modify the `prec` attribute to set the precision. This is useful when you need to perform calculations with a specific number of decimal places.

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

Conclusion

decimal module provides a way to represent floating point numbers with arbitrary precision. It is useful when you need to perform calculations that require high precision, such as financial calculations or scientific computations. The Decimal class is used to create decimal numbers, and it can be used in arithmetic operations like addition, subtraction, multiplication, and division.

Python's decimal module is a powerful tool for handling high-precision arithmetic. It allows you to perform calculations with a specific number of decimal places, which is essential in many applications. The `Decimal` class is the core of the module, and it provides methods for creating, manipulating, and comparing decimal numbers. The `localcontext()` function is used to create a new context object, which can be used to temporarily change the precision of calculations. This is useful when you need to perform a series of calculations with a specific precision, without affecting the global context. The decimal module is a standard part of Python, and it is widely used in many applications.

decimal module provides a way to represent floating point numbers with arbitrary precision. It is useful when you need to perform calculations that require high precision, such as financial calculations or scientific computations. The Decimal class is used to create decimal numbers, and it can be used in arithmetic operations like addition, subtraction, multiplication, and division.

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

äyŁéÍćŻĐéŤŽèřřáŘřäžěáŁ'ćŤÍ math.fsum() æL'ĂæŘŘä;ŽćŽĐæŽt'ćš;ćąõèőąćóŬèĈ;ăŁŽæİèèğĉăE

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

ćĐŬèĂŇrijŇáržäžŎăĚŭäžŮćŽĐćóŬæşŤrijŇă;ăăžŤèřěäžŤćžEĉăŤćŤ'ŭăőĈăžŭćŘEèğĉăőĈćŽĐèřřăŭőăžğĉŤ
 æĂžćŽĐæİèèřŤ'rijŇ decimal æÍăăİŬăÿžèèAĉŤÍăIJăŭL'ăŘŁăĹréĜŚèđ■ćŽĐécEăşşăĂĈ
 âIJİèŁŽćşşćİŇăžRăy■rijŇăŞĭæĂŤæŸřăÿĂĉĆžăřRăřRĉŽĐèřřăŭőăIJİèőąćóŬèŁĜĉİŇăÿ■èŤŞăžŭéĈ;æŸřăÿ■ăĚĂ
 âŽăæ■d'rijŇ decimal æÍăăİŬăÿžèğĉăEşşēŁŽćşşēŬőécŸæŘŘă;ŽăžEæŮžæşŤăĂĈ
 â;ŞPythonăŞŇæŤřæ■őăžŞæL'Şăžd'éAŞćŽĐæŮŭăĂŽăžşéĂŽăÿÿăijŽéAĜăĹř Decimal
 áržèşăijŇăžŭăÿŤrijŇéĂŽăÿÿăžşæŸřăIJăđ'ĐĉŘEéĜŚèđ■æŤřæ■őćŽĐæŮŭăĂŽăĂĈ

5.3 3.3 æŤřă■ŮćŽĐæăijăijRăŇŮè;ŞăĜž

éŬőécŸ

ă;ăéIJăèĉAăřEæŤřă■ŮæăijăijRăŇŮăŘŎè;ŞăĜžrijŇăžŭăŎĝăĹŭæŤřă■ŮćŽĐă;■æŤřăĂĂăřzé;ŘăĂĂă■Ĉ

èğĉăEşşæŮžæăĹ

æăijăijRăŇŮè;ŞăĜžă■ŤăÿĭæŤřă■ŮćŽĐæŮŭăĂŽrijŇăŘřäžă;ŁĉŤÍăEĚĉ;őćŽĐ
 format() âĜ;æŤrijŇăřŤăĉĈijŽ

```
>>> x = 1234.56789

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

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

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

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

(continues on next page)

(continued from previous page)

```
' 1234.6 '
```

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

æĈæđĬä;ăæĈşă;ŁçŦĭæŊĜæŦřèőřæşŦiijŊăřĖfæŦżæĹŔæĹŨèĂĖĖ(ăŔŨăĖşăžŎæŊĜæŦřèĹşăĜżçŹĐă

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

ăŔŊæŨăæŊĜăőŹăő;ăžăăŊŋçş;ăžęçŹĐăyĂēĹŋă;ăiijŔæŦř

'[<>^]?width[,]?(.digits)?' iijŊă ņăŨăy width

ăŊŊ digits äyžæŦř æŦřiijŊiijşăžçăăĹăŔŕéĂĹéĈĹăĹăĂĈ

ăŔŊæăŭçŹĐăăijăijŔăžşèçŋçŦĭăĬăăŨçŋęăyşçŹĐ format() æŨżæşŦăyăăĂĈæŦřăęĈiijŹ

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

èőĹèőž

æŦřăăŨăăijăijŔăŊŨèĹşăĜžéĂŹăyŷæŦřæŦřèĹççőĂăăŦçŹĐăĂĈăyĹēĹăijŦçđ'žçŹĐăĹăăĬŔăŊæŨă

decimal æĹăăĬŨăyŹĐ Decimal æŦřăăŨăŕžèşăăĂĈ

ă;şæŊĜăőŹæŦřăăŨçŹĐă;æŦřăŔŎiijŊçžşæđĬăăăijăijŹæăžæăő round()

ăĜ;æŦřăŔŊæăŭçŹĐèĝĐăĹŹèĤŹèăŊăŹŹèĹăăžŦăĖčăŔŎèĤŦăŹđăĂĈæŦřăęĈiijŹ

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

ăŊĖăŔŋăăĈă;ăŋęçŹĐăăijăijŔăŊŨèŭşæĬŋăĬŔăŊŨăşăæĬĹăăĖşçşăăĂĈ

ăĈæđĬä;ăéĬĂăęĂæăžæăăăĬŔăŊŊæĹăæŦçđ'žăăĈă;ăŋęçiijŊă;ăéĬĂăęĂēĜĹăŭşăŎžèŦĈæşăyŊ

locale æĹăăĬŨăyŹĐăăĜ;æŦřăžĖăăĂĈăăăŔŊæăŭăžşăŔŕăžèă;ŁçŦĭăăŨçŋęăyşçŹĐ

translate() æŨżæşŦăĹăăžđ'æăăăăĈă;ăŋęăăĂĈæŦřăęĈiijŹ

(continued from previous page)

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

æTt'æTṛæYṛæIJL'çñæRûçZḌiijNæL'ÄäzææCædIJä;äaIJlâd'DçREèr'şæTṛçZḌèrIijNèçŞâGžçzŞædIJäij

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

æCædIJä;äæCşäzğçTşäyÄäylæUäçñæRûâÄijriijNä;æIJÄæAäçdâLääyÄäylæNĜçd'zæIJÄâd'gä;æTḞäz

```
>>> x = -1234  
>>> format(2**32 + x, 'b')  
'1111111111111111111111111111111101100101110'  
>>> format(2**32 + x, 'x')  
'fffffb2e'  
>>>
```

äyžäZÆäzæy■āRŃçZḌèfZāLūē;ñæ■cæTt'æTṛā■ŮçñæyşiiijNçóĀā■TçZḌä;ççTlāyææIJL'èfZāLūçZḌ
int() āĜ;æTṛā■şāRriijZ

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

ëöleöž

âd'gâd'ZæTṛæCĚâEṭäyNâd'DçREäzNèfZāLūāĀāĒnèfZāLūāŠNā■AāĒ■èfZāLūæTt'æTṛæYṛä;LçóĀā
ârIèæAèöřä;RèfZäzZè;ñæ■câsðäzŌæTt'æTṛâŠNāĒūârZâzTçZḌæŮĜæIJnèaIçd'žäzNèŮt'çZḌè;ñæ■cā■şāRfā

æIJÄāRŌriijNä;ççTlāĒnèfZāLūçZḌçlNāžRāSŸæIJL'äyÄçCzéIJÄæAæşlæDRäyNāĀC
PythonæNĜäóZāĒnèfZāLūæTṛçZḌèr■æşTēuşāĒūäzŮèr■èlĀçl■æIJL'äy■āRŃāĀCærTæCiiijNæCædIJä;äâC

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

éIJĀçāōāflāĒnèfZāLūæTṛçZḌâL■çijĀæYṛ 0o iijNārsâCRäyNéIçèfZæäüriijZ

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

5.5 3.5 à■ÙèĽĆáĹřád'ǵæŦt'æŦřçŽǾL'ŠaŇĚäyŎëğčāŇĚ

éŮőécÿ

ä:äæIJL'äyÄäyIa■ÜeLĆa■ÜçņäyśazūaČśârEaōČègċaŌŃæLŔäyÄäyIæTt'æTŕăĂĆæLŪĖĂĖiijNä:äeIJĀ

èġčăẸșæŮźæąŁ

āAǧēō; ä; ăcŽDčÍNăžRéIJĀēēAăd' DčŘEäyÄäyĭæNěæIJL' 128ä; ■ēTfçŽD16äyĭăĚČčt' ăcŽDă■ŪēLČă■Ūç

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

äyžžĚårĚbytesèğçæđŘäyžæȚʼæȚriijŃä;ŁçȚĬ int.from_bytes()
æŰžæȚriijŃäžüäČŘävŇÉíçèŁŽæäüæŇĞăŰŽă■ŰèŁCéazžŘriijŽ

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

äyžāžĒårĒäyÄäyłād'gæŦt'æŦrējñæ■cāyžāyÄäyłā■ŰēŁĆā■ŰçņäyšiiijŃä;ŁćŦł int.
to_bytes() æŰžæsŦiiijŃāžūāĆŔäyŃēlćēŁžæāūāŃĠāōžā■ŰēŁĆæŦŕāšŃā■ŰēŁĆēāžāžŦiiijž

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

èóíèőž

ād'gæTt'æTṛāšNā■UēŁCā■ŮčņäyšāzNéŮt'čŽDè;ñæ■cæš■ā;Ijāzūäy■āyṽēgAāĀĆ
 čDūēĀNriiNāIĴlāyĀāzŽāzTčTlécEāššæIĴL'æUūāĀŽāzšāijŽāGžČŌriiNærTāēČārEčāAā■ēāŁŮēĀĒč;ŚczIĴā.
 ä;NāēČriiNIPv6č;ŚczIĴāIṛāĀā;ŁčTlāyĀäy1128ä;■čŽDæTt'æTṛēālčd'žāĀĆ
 āeČædIĴā;āēēAāzŌäyĀäyŁæTṛæ■ōēōrā;Täy■æRṚāRŮēŁZæāūčŽDāAijčŽDæUūāĀŽiiNā;āāršāi;ŽēlčāržēŁZ.

ä;IJäyžäyÄçg■æZfäzçæŮzæĹLijNä;ġāŖŕĕČ;æČšä;ĤçŤl6.11ārŖĕŁĆäy■æL'ÄäzNçz■çŽD
struct æĹaĹiŮæĪēēçġāŌŃā■ŮĕŁCăĀĆ ēŁZæăūăzšēaŃă;ŮĕĀŽiijNăv■ēŁĠĹl'cŤl

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

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

```
>>> 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'
```


ĕġċăĖşăŮzăăĹ

ăđ'■ăŤrăŔrăzĕċŤlă;ĤċŤlăĜ;ăŤŕ complex(real, imag)
ăĹŮĕĂĖăŸŕăŷăĖIJL'ăŔŎċijĂjċŽĐăŤŏċĈzăŤŕăĹăăŊĜăŔăĂĈăŕŤăċŤiijŽ

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

ărzăžŤċŽĐăŏđĕĈlăĂăĖŽŽĕĈlăŤŊăĖĖ;■ăđ'■ăŤrăŔrăzăăĹăŔăŔăŷăŸŤċŽĐăŔăŮăĂĈăŕŝăĈŔăŷŊĕĹĕĖŤ

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

ăŔĕăđ'ŮiijŊăĹ'ĂăIJL'ăŷŷĕġĂċŽĐăŤŕă■ĕĖŔċŔŮĕĈ;ăŔŕăzăăŭĕă;IJiijŽ

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

ăĖĈăđIJĕĖĂăĹ'ġĕăŊăĖŮăžŮċŽĐăđ'■ăŤŕăĜ;ăŤŕăŕŤăĖĈăăċăijăăĂăă;ŽăijăăĹŮăžăăŮăăžiiijŊă;ĤċŤlă
cmathăĹăăĹŮiijŽ

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

èõìèõž

Pythonäy■ād' ġéČlāĹĒäyŎæTřā■ęçŽyāĚşçŽĎæĹaĹĹŮéČĵèČĵād'ĐçŘĒād'■æTřāĀĆ
æřTāęČæČæđĪĴĵääĵčTĪ numpy ĩijŇāŘřäzēāĴĹāōžæŸşçŽĎæđĎéĀäyĀäyĹād'■æTřæTřçzĎāžŮāĪĴĹèĴŽäyĹæ

```
>>> 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])
>>>
```

PythonçŽĎæāĠāĠĒæTřā■ęāĠĵæTřçāōāōđæČĚāĒtāyŇāžŮäy■èČĵāžġçTřād'■æTřāĀĵĳĳŇāŽāæ■d'äĵāçŽ

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

æęČæđĪĴĵääČşçTřşæĹŘäyĀäyĹād'■æTřèĴTāŽđçzşæđĪĳĳŇāĵāĴĚéāzæŸĴçd'žçŽĎäĵčçTĪ
cmath æĹaĹĹŮĳĳŇāĹŮèĀĚāĪĴæşŘäyĹæTřæŇĀād'■æTřçŽĎāžşäy■āçřæŸŎād'■æTřçşzādŇçŽĎäĵčçTĪāĀĆæ

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

5.7 3.7 æŮäçĴŮād'ġäyŎNaN

éŮóécŸ

äĵääČşāĹŽāžžæĹŮætŇërTřæ■čæŮäçĴŮāĀĀèt'şæŮäçĴŮāĹŮNaN(éĪđæTřā■Ů)çŽĎætōçČzæTřāĀĆ

èġčāĒşæŮzæāĴĹ

PythonāžŮæşæĪĴĴĴĴ'čĴ'žæōĴçŽĎër■æşTřæĹèēāĴçd'žèĴŽāžŽçĴĴ'žæōĴçŽĎætōçČzāĀĳĳĳŇāĴæŸřāŘřäzēāĴ
float() æĹèāĹŽāžžāōČžžñāĀĆæřTāęČĳĳŽ

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

(continues on next page)

(continued from previous page)

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

äyžāžEætŊerŤetŽāžZāĀijçŽDā■ŸāIJlījNā;£çŦĪ math.isinf() āŠŊ math.
isnan() āĠ;æŦřāĀĆærŦæĆījŽ

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

ěóľěőž

æČšāžEëğčæZŦ'ād'Žè£ŽāžZçL'žæōŁæŦōçČzāĀijçŽDā£æAŦījNāŦřāžēāŦĆèĀČIEEE
754èğDèNČāĀĆ çDūēĀŊījNāžšæIJL'äyĀāžZāIJræŮzéIJĀèçAä;ăçL'zāLŋæšlæĎŦījNçL'zāLŋæŸrèùšærŦè;
æŮăçl'ŭăđ'ğæŦřāIJlæL'ğēăNæŦřā■èōăçŦŮçŽDæŮŭāĀŽāijŽāijæŠ■ījNærŦæĆījŽ

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

ä;EæŸŦæIJL'āžZæŠ■ä;IJæŮŭæIJāōŽāžL'çŽĎāžŭāijŽè£ŦāŽđäyĀäyŦNaNçzšæđIJāĀĆærŦæĆījŽ

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

NaNāĀijāijŽāIJlæL'ĀæIJL'æŠ■ä;IJäy■āijæŠ■ījNèĀŦäy■āijŽāžğçŦšāijČāyŷāĀĆærŦæĆījŽ

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

(continues on next page)

(continued from previous page)

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

NaN is a floating point value that represents "Not a Number". It is created when an operation results in a value that is not a number, such as dividing by zero or taking the square root of a negative number.

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

NaN is not equal to itself. This is because NaN represents an undefined or unrepresentable value, and there is no specific value to compare it to. Therefore, `NaN == NaN` returns `False`.

NaN is also not equal to any other value, including itself. This is because NaN represents an undefined or unrepresentable value, and there is no specific value to compare it to. Therefore, `NaN == 0` returns `False`.

5.8 3.8 `float('nan')`

`float('nan')`

NaN is a floating point value that represents "Not a Number". It is created when an operation results in a value that is not a number, such as dividing by zero or taking the square root of a negative number.

`fractions.Fraction`

The `Fraction` class in the `fractions` module represents rational numbers. It is created by passing a numerator and a denominator to the `Fraction` constructor.

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

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

(continues on next page)

(continued from previous page)

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

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

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

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

èóíèõž

āIJlād' gād' ŽæTřčlNāžRāy■āyĀēLnāy■āijŽāGžčŎřāLEæTřčŽDèõaçõŮéŮõécŸijNā;EæŸræIJL'æŮūāĀ
ærTāēĆijNāIJlāyĀāyĪāĒĀēõyæŎēāRŮāLEæTřā;ćāijRčŽDætNērTā■Tā;■āžūāzēāLEæTřā;ćāijRæL'gēāNēĒR
čŽt'æŎēā;ĲčTlāLEæTřāRřāzēāGRārSæL'NāLē;ñæ■cāyžārRæTřæLŮætõçCzæTřčŽDāũēā;IJāĀĆ

5.9 3.9 ād'gādNæTřčzDèĒŘčõŮ

éŮõécŸ

ä;āēIJĀēçAāIJlād' gæTřæ■õéŽĒ(ærTāēĆæTřčzDæLŮč;Sæāij)äyLéÍcæL'gēāNēõaçõŮāĀĆ

èğčāEşæŮzæāĻ

æŮL'āRĻāĻræTřčzDčŽDēĠēGRčžgēĒŘčõŮæŞ■ā;IJijNāRřāzēā;ĲčTl NumPy āžSāĀĆ
NumPy çŽDāyĀāyĪāyžēçAçL'žā;AæŸrāõĆāijŽčžPythonæRŘā;ŽāyĀāyĪæTřčzDāržēsāijNçŽyærTāēĀGāGE
āyNēÍcæŸrāyĀāyĪçõĀ■TčŽDārRā;Nā■RijNāRŠā;āāsTčd'žæāGāGEāLŮēāĻrřzēsāāŠŇ
NumPy æTřčzDāržēsāāžNēŮt'çŽDāũōāĻnīijŽ

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

(continues on next page)

(continued from previous page)

```
>>> x + 10
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: can only concatenate list (not "int") to list
>>> x + y
[1, 2, 3, 4, 5, 6, 7, 8]
>>> # Numpy arrays
>>> import numpy as np
>>> ax = np.array([1, 2, 3, 4])
>>> ay = np.array([5, 6, 7, 8])
>>> ax * 2
array([2, 4, 6, 8])
>>> ax + 10
array([11, 12, 13, 14])
>>> ax + ay
array([ 6,  8, 10, 12])
>>> ax * ay
array([ 5, 12, 21, 32])
>>>
```

æ■čāēĆæL'ĀèġAīijNāyd'çġ■æŪzæāLāy■æTṛçzDçŽDāšzæIJnæTṛā■ēèfRçõŪçzŞædIJāzūäy■çŽyāRŅāĀ
çL'zāLŋçŽDīijN NumPy äy■çŽDæāGēGRèfRçõŪ(ærTāēĆ ax
* 2 æLŪ ax + 10)äijŽā;IJçTlāIJlærRāyĀäylāĒÇçt'āāyLāĀĆ
āRēād'ŪīijNā;Şāyd'äylæŞ■ā;IJæTṛēČ;æYṛæTṛçzDçŽDæŪūāĀZæL'gēāNāĒÇçt'āārżç■L'ä;■ç;õèøaçõŪīijNāzū
āṛzæTt'äylæTṛçzDäy■æL'ĀæIJL'āĒÇçt'āāRŅæŪūæL'gēāNæTṛā■ēèfRçõŪāRṛäzēä;£ā;Ūā;IJçTlāIJlærTṛā
ærTāēĆīijNāēCædIJā;āæCşèøaçõŪād'ŽēazāijRçŽDāĀīijNāRṛäzēèfZæāūāĀZīijŽ

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

NumPy $\text{e}\text{f}\text{Y}\text{ä}\text{y}\text{z}\text{æ}\text{T}\text{ř}\text{c}\text{z}\text{D}\text{æ}\text{S}\text{■}\text{ä}\text{I}\text{j}\text{æ}\text{R}\text{Ř}\text{ä}\text{;}\text{Z}\text{ä}\text{Z}\text{E}\text{ä}\text{d}'\text{g}\text{é}\text{G}\text{R}\text{ç}\text{Ž}\text{D}\text{é}\text{Ä}\text{Ž}\text{ç}\text{T}\text{l}\text{ä}\text{G}\text{;}\text{æ}\text{T}\text{ř}\text{i}\text{i}\text{j}\text{N}\text{e}\text{f}\text{Z}\text{ä}\text{ž}\text{Z}\text{ä}\text{G}\text{;}\text{æ}\text{T}\text{ř}\text{ä}\text{R}\text{rä}\text{z}\text{e}\text{ä}\text{;}\text{I}\text{j}\text{ä}\text{}$
 $\text{math}\text{ä}\text{l}\text{ä}\text{ä}\text{I}\text{Ü}\text{ä}\text{y}\text{■}\text{c}\text{s}\text{z}\text{ä}\text{i}\text{i}\text{j}\text{j}\text{ä}\text{G}\text{;}\text{æ}\text{T}\text{ř}\text{c}\text{Ž}\text{D}\text{æ}\text{Z}\text{f}\text{ä}\text{z}\text{c}\text{ä}\text{Ä}\text{C}\text{ä}\text{r}\text{T}\text{ä}\text{e}\text{C}\text{i}\text{i}\text{j}\text{Ž}$

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

$$\begin{aligned} & \mathfrak{a};\mathfrak{f}\mathfrak{c}\mathfrak{T}\mathfrak{l}\mathfrak{e}\mathfrak{f}\mathfrak{Z}\mathfrak{a}\mathfrak{z}\mathfrak{Z}\mathfrak{e}\mathfrak{A}\mathfrak{Z}\mathfrak{c}\mathfrak{T}\mathfrak{l}\mathfrak{a}\mathfrak{G};\mathfrak{x}\mathfrak{T}\mathfrak{r}\mathfrak{e}\mathfrak{e}\mathfrak{A}\mathfrak{x}\mathfrak{f}\mathfrak{T}\mathfrak{a};\mathfrak{l}\mathfrak{c}\mathfrak{O}\mathfrak{r}\mathfrak{a}\mathfrak{T}\mathfrak{r}\mathfrak{c}\mathfrak{z}\mathfrak{D}\mathfrak{a}\mathfrak{z}\mathfrak{u}\mathfrak{a};\mathfrak{f}\mathfrak{c}\mathfrak{T}\mathfrak{l} \\ \text{math} & \mathfrak{a}\mathfrak{l}\mathfrak{a}\mathfrak{i}\mathfrak{U}\mathfrak{a}\mathfrak{y}\mathfrak{Z}\mathfrak{D}\mathfrak{a}\mathfrak{G};\mathfrak{x}\mathfrak{T}\mathfrak{r}\mathfrak{a}\mathfrak{L}\mathfrak{g}\mathfrak{e}\mathfrak{a}\mathfrak{N}\mathfrak{e}\mathfrak{o}\mathfrak{a}\mathfrak{c}\mathfrak{O}\mathfrak{U}\mathfrak{e}\mathfrak{e}\mathfrak{A}\mathfrak{a}\mathfrak{f}\mathfrak{h}\mathfrak{c}\mathfrak{Z}\mathfrak{D}\mathfrak{a}\mathfrak{d}'\mathfrak{Z}\mathfrak{a}\mathfrak{A}\mathfrak{C} \\ \mathfrak{a}\mathfrak{Z}\mathfrak{a}\mathfrak{a}\mathfrak{d}'\mathfrak{i}\mathfrak{i}\mathfrak{j}\mathfrak{N}\mathfrak{a}\mathfrak{R}\mathfrak{l}\mathfrak{e}\mathfrak{e}\mathfrak{A}\mathfrak{x}\mathfrak{I}\mathfrak{J}\mathfrak{l}'\mathfrak{a}\mathfrak{R}\mathfrak{r}\mathfrak{e}\mathfrak{C};\mathfrak{c}\mathfrak{Z}\mathfrak{D}\mathfrak{e}\mathfrak{r}\mathfrak{l}\mathfrak{a}\mathfrak{r};\mathfrak{e}\mathfrak{G}\mathfrak{R}\mathfrak{e}\mathfrak{A}\mathfrak{L}'\mathfrak{x}\mathfrak{N}\mathfrak{l}' & \text{NumPy} \mathfrak{c}\mathfrak{Z}\mathfrak{D}\mathfrak{a}\mathfrak{T}\mathfrak{r}\mathfrak{c}\mathfrak{z}\mathfrak{D}\mathfrak{a}\mathfrak{U}\mathfrak{z}\mathfrak{a}\mathfrak{q}\mathfrak{L}\mathfrak{a}\mathfrak{A}\mathfrak{C} \\ \mathfrak{a}\mathfrak{z}\mathfrak{T}\mathfrak{a}\mathfrak{s}\mathfrak{C}\mathfrak{a}\mathfrak{o}\mathfrak{d}\mathfrak{c}\mathfrak{O}\mathfrak{r}\mathfrak{a}\mathfrak{y}\mathfrak{i}\mathfrak{i}\mathfrak{j}\mathfrak{N} & \text{NumPy} \mathfrak{x}\mathfrak{T}\mathfrak{r}\mathfrak{c}\mathfrak{z}\mathfrak{D}\mathfrak{a};\mathfrak{f}\mathfrak{c}\mathfrak{T}\mathfrak{l}\mathfrak{a}\mathfrak{z}\mathfrak{E}\mathfrak{C}\mathfrak{a}\mathfrak{L}\mathfrak{U}\mathfrak{e}\mathfrak{A}\mathfrak{E}\mathfrak{F}\mathfrak{o}\mathfrak{r}\mathfrak{t}\mathfrak{r}\mathfrak{a}\mathfrak{n}\mathfrak{e}\mathfrak{r}\mathfrak{e}\mathfrak{l}\mathfrak{A}\mathfrak{c}\mathfrak{Z}\mathfrak{D}\mathfrak{a}\mathfrak{I}\mathfrak{J}\mathfrak{z}\mathfrak{a}\mathfrak{L}\mathfrak{u}\mathfrak{a}\mathfrak{L}\mathfrak{e}\mathfrak{E}\mathfrak{B}\mathfrak{a}\mathfrak{E}\mathfrak{E}\mathfrak{a}\mathfrak{N}\mathfrak{Y}\mathfrak{a} \end{aligned}$$

āzšārsæYřert'ijNāōČāznæYřayĀäylēldāyād'gčŽDēŁdčz■čŽDāžūčTšāRŇčsžādNæTřæ■ōczDæLŔčŽDāEĚā
æL'ĀāžērijNā;āāRřāzēædDēĀāyĀäylærTæŽōéĀŽPythonāLŮèāĹād'gčŽDād'ŽčŽDæTřčzDāĀČ
ærTāēČijNāēČædIJā;āæČšædDēĀāyĀäy10,000*10,000čŽDætōčČzæTřāžNčzt'č;ŠæāijijNā;Lè;zæĹijŽ

```
>>> 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.]])
>>>
```

æL'ĀæIJL'čŽDæŽōéĀŽæŠ■ā;IJēŁYæYřaijŽāRŇæŮūā;IJčTĹāIJL'ĀæIJL'āĒČčt'āyĹijŽ

```
>>> 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]])
>>>
```

āĚšāžŮ NumPy æIJL'āyĀčČzéIJĀèēAçL'žāLŇčŽDāyžæĐRijŇéČčārsæYřāōČæL'āšTPythonāLŮèāĹčŽL
-çL'žāLŇæYřāržāžŮād'Žčzt'æTřčzDāĀČ āyžāžEērt'æYŌæyĒæēŽijNāĒLædDēĀāyĀäylčōĀā■TčŽDāžNčzt'

```
>>> 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]])
>>>

```

èóìèõž

NumPy æŸřPythonécEâššäy■āçĹād'ŽçğŚā■ēäyŌāũēçĹNāžŚçŽDāšžçāĀiijNāRŃæUūāžšæŸřèçñāžŁæšŽ
ā■šāçŁæçCæ■d'rijNāIJlāLŽāijĀāğNçŽDæUūāĀŽéĀŽēŁGāyĀāžŽçóĀā■TçŽDāçNā■RāŠNçŌĹāĒūçĹNāžRāžš

éĀŽāyŷæĹŚāžñārijāĒē NumPy æĹāāĹUçŽDæUūāĀŽāijŽāçŁçTĹér■āRē import numpy
as np āĀĆ èŁZæāũçŽDèrlā;āāřsāy■çTĹāE■āçŽDçĹNāžRéGŃēĹcāyĀéA■éA■çŽDæTšāĒē
numpy iijNāRĹēIJĀēçAēçŚāĒē np āřsēāNāžEiijNēŁCçIJĀāžEāy■āřSæUūēŮt'āĀĆ

āçCædIJæČšèŌūāRŮæŽt'ād'ŽçŽDāŁæAřrijNā;āā;ŚçDūāçŮāŌž NumPy
āōŸç;ŚéĀŽéĀŽāžEiijNç;ŚāĹĀæŸřijŽ <http://www.numpy.org>

(continued from previous page)

```
-229.99999999999983

>>> # Eigenvalues
>>> numpy.linalg.eigvals(m)
array([-13.11474312,  2.75956154,  6.35518158])

>>> # Solve for x in mx = v
>>> x = numpy.linalg.solve(m, v)
>>> x
matrix([[ 0.96521739],
        [ 0.17391304],
        [ 0.46086957]])

>>> m * x
matrix([[ 2.],
        [ 3.],
        [ 4.]])

>>> v
matrix([[2],
        [3],
        [4]])

>>>
```

èóìèõž

åĴŁæŸçĎŮçžŁæĀğāzčæŦræŸrāyłéłđāyŷād'ğçŽĎāyžécŸiijŇāũščžRèuĒāĜžāžEæIJñāžęèČ;èóìèõžçŽĎæ
äĵEæŸriijŇāçCæđIJä;äéIJĀèçAæŞ■äĵIJæŦrçžĎāŠŇāŔŠéĜŔçŽĎèŦriijŇ NumPy
æŸrāyĀäyłāy■éŦŽçŽĎāĒēāŔççĆzāĀĆ āŦrāžèèóŁéŮ NumPy āóŸç;Ś <http://www.numpy.org>
èŮāŦŔŮæŽŦ'ād'ŽāŁæAŦāĀĆ

5.11 3.11 éŽŔæIJžéĀŁ'æŇŦ'

éŮóécŸ

äĵāæČšāžŮāyĀäyłāžŔāŁŮäy■éŽŔæIJžæŁ;āŦŮŮèŇēāžšāĒČçŦ'āriijŇæŁŮèĀĒæČšçŦšæŁŦāĜāyłéŽŔæIJ

èğčāEşæŮžæāŁ

random æłāāĴŮæIJŁ'ād'ğéĜŔçŽĎāĜĵæŦrçŦłæłēāžğçŦšéŽŔæIJžæŦŦāŠŇēŽŔæIJžéĀŁ'æŇŦ'āĒČçŦ'āāĀĆ
æŦŦāçĆiijŇēçAæČšāžŮāyĀäyłāžŔāŁŮäy■éŽŔæIJžçŽĎæŁ;āŦŮŮäyĀäyłāĒČçŦ'āriijŇāŦŦāžēā;ŁçŦŦ
random.choice() iijŽ

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

(continues on next page)

(continued from previous page)

```
2
>>> random.choice(values)
3
>>> random.choice(values)
1
>>> random.choice(values)
4
>>> random.choice(values)
6
>>>
```

random.sample() **iiž**

```
>>> 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]
>>>
```

random.shuffle() **iiž**

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

random.randint() **iiž**

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

random.random() returns a random float between 0 and 1.

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

random.getrandbits(k) returns a random integer with k bits.

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

Seeding

random.seed() or random.seed(x) initializes the random number generator. If x is omitted, it uses the system time or os.urandom(). If x is an integer, it uses that integer. If x is a bytes object, it uses that data.

```
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
```

random.uniform(a, b) returns a random float between a and b. random.gauss(mu, sigma) returns a random float from a Gaussian distribution with mean mu and standard deviation sigma.

random.randrange(a, b, step) returns a random integer from the range [a, b) with a step of step. random.random() returns a random float between 0 and 1. random.getrandbits(k) returns a random integer with k bits. random.seed() or random.seed(x) initializes the random number generator.

3.12 Random Number Generation

Seeding

random.seed() or random.seed(x) initializes the random number generator. If x is omitted, it uses the system time or os.urandom(). If x is an integer, it uses that integer. If x is a bytes object, it uses that data.

Random Number Generation

random.random() returns a random float between 0 and 1. random.getrandbits(k) returns a random integer with k bits. random.seed() or random.seed(x) initializes the random number generator. random.randrange(a, b, step) returns a random integer from the range [a, b) with a step of step. random.random() returns a random float between 0 and 1. random.getrandbits(k) returns a random integer with k bits. random.seed() or random.seed(x) initializes the random number generator.

```

>>> 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
>>>

```

æĈædĪĵāæĈşēāċd'zæŅĜāōŽçŽĎæŮæĪJšāŠŅæŮŭéŮrĭĵŅāĒĹāĹZāzzāyĀäyĭ
 datetime āōđāĵŅçĎŭāŔŌäĵġçŦĹæāĜāĠĖçŽĎæŦŕā■ēēŦŔçōŮæĪæş■äĵĪāōĈāznāĀĈærŦāēĈĭĵŽ

```

>>> 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
>>>

```

āĪĪēōāçōŮçŽĎæŮŭāĀŽĭĵŅēĪJĀēēAæşĹæĎŔçŽĎæŸŕ datetime
 äĭjŽēĠāĹāđ'ĎçŔĖēŮŕāzt'āĀĈærŦāēĈĭĵŽ

```

>>> 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
>>>

```

èõléõž

årzād' gād' ŽæŦrāšžæIJñçŽDæŦæIJšāšNæŦŦæŦ' ad' DçRĖĖŦŦécŦŦijN datetime
æŦāāŦŦŦŦçzRèŦŦād' šāžĖāĀC āĖCædIJā;æĖIJĀèĖAæL' gèāNæŽt' āLāād' ■æĪCçŽDæŦæIJšæ\$■ā;IJŦijNæŦŦāĖC
årRrāžèèĀCèŽSā;ŦçŦĪ dateutilæŦāāŦŦ

èõŦād' ŽçšžāijjçŽDæŦŦæŦ' èõāçŦŦŦRrāžèä;ŦçŦĪ dateutil.relative_delta()
āĖ;æŦŦrāžçæŽĖāĀC ā;ĖæŦŦijNæIJL' äŦĀçCžĖIJĀèĖAæšĖāDŦçŽDāŦsæŦŦijNāŦŦĀijŽāIJĪad' DçRĖæIJĪäz;(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 èõāçŦŦŦæIJĀāRŦŦäŦŦäŦŦāSĪäžŦçŽDæŦæIJš

éŦŦécŦŦ

ā;æĖIJĀèĖAæšæL'æŦŦŦæIJšäŦŦæšRäŦŦĀad'ŦæIJĀāRŦŦāGžçŦŦçŽDæŦæIJšŦijNæŦŦāĖCæŦŦŦæIJšäžŦŦā

èğçāĖşæŦŦæŦĪ

PythonçŽD datetime æŦāāŦŦŦŦæIJL' āŦŦāĖŦŦāĖ;æŦŦŦŦçšžāRrāžèäŦŦāL' ā;ææL' gèāNæŦŦæŦŦçŽDèõ
äŦŦŦĖĪæŦŦŦçšžāijjçĖŦæŦŦçŽDèŦŦécŦŦçŽDäŦŦäŦŦŦŦçŦŦŦŦççāĖşæŦŦæŦĪŦijŽ

```
#!/usr/bin/env python
# -*- encoding: utf-8 -*-
"""
Topic: æIJĂăŘŮčŽĎăŚĺăžŤ
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
```

ăIJăžď'ăžŚăijŘëğćëĠăŽĺăy■ă;čŤĺăçCăyŇijŽ

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

ăŖŕéĂĹčŽĎ start_date ăŖCăŤŕăŖŕăzëçŤśăŖëăď'ŮăyĂăyĭ datetime
ăőďă;ŇăĹăăŖŖă;ŽăĂCăŕŤăçCĭijŽ

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

ëőĹëőŽ

ăyĹĹĹčŽĎčŏŮăşŤăŎşçŖĒăŸŕëŹăăŭčŽĎijŽăĹĹăŖĒăijĂăğŇăŮăăIJşăŚŇçŽŏăăĠăŮăăIJşăŸăăŖĎ.
çĎŮăŖŎéĂžëŹĠăŹăŖŖçŮŮëŏăçŮŮăĠžçŽŏăăĠăŮăăIJşëçAçžŖëŹĠăď'ŽăŖŚăď'ŤăĹ■ëCĭăĹŖëĹăijĂăğŇăŮă

ăçCăďIJă;ăëçAăŖŖëŹăăŭăĹğëăŇăď'ğëĠŖçŽĎăŮăăIJşëŏăçŮŮăçŽĎŕĭijŇă;ăăIJĂăë;ăŏĹëçĬčňăyĹ
python-dateutil æĹăăžçăŽăĂC æŖŤăçCĭijŇăyŇéĹăŸŕăŸŕă;čŤĺ dateutil

relativedelta() aġġiebi l-ħin ta' relattiva għall-ħin ta' iss.

```
>>> 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 eżempju ta' kodiċi ta' Python għall-ħin ta' relattiva

eżempju

għall-ħin ta' relattiva għall-ħin ta' iss.

eżempju ta' kodiċi ta' Python

għall-ħin ta' relattiva għall-ħin ta' iss.

għall-ħin ta' relattiva għall-ħin ta' iss.

```
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)
```

għall-ħin ta' relattiva għall-ħin ta' iss.


```

>>> 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...

```

èóìeóž

äyŁéİççŽDäzççäAäĖĖŁeðaçóŮäGžäyÄäyġärzāžTæIJLāz;çññäyÄād'ŦçŽDæŮeæIJšāĀĆ
äyÄäyġäŦnéĀšçŽDæŮzæşTārşæYřä;ŁçTĪ date æLŮ datetime āržèšaçŽD
replace() æŮzæşTçóĀā■TçŽDārĖ days āsdæĀğèð;ç;óæĹR1ā■şāRřāĀĆ replace()
æŮzæşTäyÄäyġäe;ād'DārşæYřäóĀijZāLZāzzāšNā;āāijĀāgNāijāāĖĖāržèšaçşzādNçZyāRŦçŽDāržèšāāĀĆ
æL'ĀāžērijNāeĀcædIJe;ŠāĖĖāRĀĆæTřæYřäyÄäyġ date āóðä;NrijNēĀcāzLçzŞædIJāzşæYřäyÄäyġ
date āóðä;NāĀĆ āRŦæāũçŽDrijNāeĀcædIJe;ŠāĖĖæYřäyÄäyġ datetime
āóðä;NrijNēĀcāzLā;āā;ŮāĹřçŽDārşæYřäyÄäyġ datetime āóðä;NāĀĆ

çDūāRŮrijNā;ŁçTĪ calendar.monthrange() āĜ;æTřæĖæL;āĜžèřæIJLçŽDæĀzād'ŦæTřāĀĆ
āzzā;TæŮūāĀZārĖeAā;āæĀşèŮūā;ŮæŮāŮĖāĖæAřrijNēĀcāzL
calendar æġāāĪŮārşēĪdāyŷæIJLçTĪāzĖāĀĆ monthrange()
āĜ;æTřāijZèĤTāZdāNĖāRŦæYşæIJšāšNĖēřæIJLād'ŦæTřçŽDāĖĀçzDāĀĆ

äyĀæŮeřæIJLçŽDād'ŦæTřāũşçşēāžErijNēĀcāzLçzŞæĪşæŮeæIJšārşāRřāžēēĀŽēĤĜāIJġāijĀāgNæŮeæ
æIJLāyġēIJĀēeAæşġæĀRçŽDæYřçzŞæĪşæŮeæIJšāzūāy■āNĖāRŦāIJġēZāyġæŮeæIJšēNĀZt'āĖĖ(āžNāóðāy
ēĤZāyġāšNPythonçŽD slice äyŮ range æŞ■ā;IJēāNāyžāĤġæNĀāyĀēĜt'rijNāRŦæāũāžşāy■āNĖāRŦçzŞār

äyžāžĖāIJġæŮeæIJšēNĀZt'äyġā;ŁçŮrijNēeAā;ŁçTĪġġæāĜāĖĖçŽDæTřā■ēāšNæřTē;ĀæŞ■ā;IJāĀĆ
æřTāeĀrijNāRřāžēāĤ'çTĪ timedelta āóðä;NāĖēĀšācðæŮeæIJšrijNārRāžŮāRū<çTĪæĖæĀæşēāyÄäyġ

çŘæĀçşæĀĖāĖäyNrijNāeĀcædIJeç;äyžæŮeæIJšēĤ■āzçāLZāzzāyÄäyġāRŦāĖĖç;óçŽD
range() āĜ;æTřāyĀæāũçŽDāĜ;æTřārşæ;āžĖāĀĆ āžyēĤŘçŽDæYřrijNārRāžēā;ŁçTĪäyÄäyġŁçTşæĹRāZġāĖ

```

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

```

äyNēĪæYřä;ŁçTĪēĤZāyġŁçTşæĹRāZġçŽDä;Nā■RrijZ

```
>>> 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
...
>>>
```

èŁŻçġ■āōđçŎřāzŊæŁ'ĂăžèŁŻāzŁçōĂā■ŤiijŊèŁŸā;Ůā;ŠāŁšāžŎPythonāy■çŽĐæŮæIJšāŠŊæŮéŮŤ

5.15 3.15 ā■Ůçņęäyšè;ŋæ■căyžæŮëæIJš

éŮóécŸ

ä;ăçŽĐāžŤçŤlćlŊāzŤæŎëāŤŮā■ŮçņęäyšæāijāijŤçŽĐè;ŠāĚëijŊā;ĒæŸŤā;ăæČšāŤĒāōČāzŋè;ŋæ■căyž
datetime āŤžèšāžæä;ŁāIJläyŁéłćæŁġëāŊéłđā■Ůçņęäyšæš■ā;IJāĂĆ

èġčāĒşæŮzæąŁ

ä;ŁçŤlŤPythonçŽĐæāĠāĠĒāŁāŮ datetime āŤŤāžæā;ŁāōžæŸşçŽĐèġčāĒşèŁŻāyŁéŮóécŸāĂĆæŤāçŮ

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

èőlèőž

datetime.strptime() æŮzæşŤæŤŤæŊAā;Łād'ŽçŽĐæāijāijŤāŊŮāžčçāĀiijŊ
æŤāçŮ %Y āžčēāł4ā;■æŤŤāžŤ'āž;ijŊ %m āžčēāłāyđ'ā;■æŤŤæIJŁāž;āĂĆ
èŁŸæIJL'āyĂçČzāĀijā;ŮæşŁæĐŤçŽĐæŸŤèŁŻāžŽæāijāijŤāŊŮā■ā;■çņęäžşāŤŤāžæāŤŤèŁĠæĒā;ŁçŤlŤijŊāŤŤ
æŤāçŮĀiijŊāĠĠèō;ä;ăçŽĐāžčçāĀāy■ŤşæŁŤāžĒāyĀāyŁ datetime āŤžèšāijŊ
ä;ăæČšāŤĒāōČæāijāijŤāŊŮāyžæijČāžōæŸşèŤzā;ćāijŤāŤŮæŤ;āIJĒĠāŁłćŤşæŁŤŤçŽĐäŁāžūæŁŮèĂĚæŁēā

```
>>> z
datetime.datetime(2012, 9, 23, 21, 37, 4, 177393)
>>> nice_z = datetime.strptime(z, '%A %B %d, %Y')
>>> nice_z
'Sunday September 23, 2012'
>>>
```

ðŁŸæIJL'äyÄçÇzéIJÄèeAæslæĐRçŽĐæYřijŃ
çŽĐæĀğèČĭèeAærfTä;ăæČšèsaÿ■çŽĐăũőăĹăđ'ŽřijŃ âŽăÿžăőCæYřă;ŁçŤĹçžřPythonăőđçŎřřijŃăžŭăyŤăŁ
ăeČăđIJă;ăèeAăIJăžčçăAăÿ■eIJÄèeAeğçăđRăđ'gēGRçŽĐæŮeæIJšăžŭăyŤăũšçžRçšèeAšăžEæŮeæIJšă■Ů
æřŤăeČřijŃăeČăđIJă;ăăũšçžRçšèeAšăL'ĂăžèæŮeæIJšăăijăijRæYř
řijŃă;ăăRăřăžăăČRăyŃéĹcèŁŽæăăăđçŎřăyĂăylègçăđRăĠæŤřijŽ

strptime()
YYYY-MM-DD

```
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))
```

ăőđéŽĚæřŃèřŤăy■řijŃèŁŽăylăĠ;æŤræřŤ datetime.strptime() âŁŃ7ăĂăđ'ŽăĂČ
ăeČăđIJă;ăèeAăđ'ĐçRĚăđ'gēGRçŽĐæŮL'ăRĹăĹræŮeæIJšçŽĐæŤræ■őçŽĐèřřijŃéČcăžĹăIJăăe;èĂČèŽŠă

5.16 3.16 çžŠăŘĹæŮăăŃžçŽĐæŮeæIJšăš■ăIJ

éŮeéčŸ

ă;ăæIJL'äyĂăylăăŁ'ăŎšăIJĹ2012ăžŤ'12æIJĹ21æŮeæŮŤ'äyĹ9:30çŽĐçŤřerăijŽèőřijŃăIJřçČăIJĹèĹăŁă
èĂŃă;ăçŽĐæIJŃăRŃăăIJă■ăžççŽĐçRă■ăŁăç;ŮăřŤřijŃéČcăžĹăžŮăžŤèřăăIJă;šăIJræŮeæŮŤ'ăĠăçČăăRČăăŁă

èğçăEşæŮžæăĹ

ăržăĠăăžŎăL'ĂæIJL'ăũĹăRĹăĹræŮăăŃžçŽĐéŮeéčŸřijŃă;ăeČăăžŤèřăă;ŁçŤĹ
pytz æĹăăĹŮăĂČeŁŽăylăŃĚæRŘă;ŽăžEŮolsonæŮăăŃžæŤræ■ăăžšřijŃ
ăőCæYřæŮăăŃžăæAæAřçŽĐăžŃăăđăyĹçŽĐæăĠăĠEřijŃăIJăĹăĹăđ'Žèř■eĹĂăšŃăš■ăIJçšçžçšéĠŃéĹcèČăăR
pytz æĹăăĹŮăyĂăylăyžèeAçŤĹéĂŤæYřăřE datetime
ăžšăĹŽăžççŽĐçăăŤæŮeæIJšăřžèšæăIJăăIJrăŃŮăĂČ æřŤăeČřijŃăyŃéĹcăeČă;ŤeăĹçđ'žăyĂăylèĹăĹăăšă

```
>>> from datetime import datetime
>>> from pytz import timezone
>>> d = datetime(2012, 12, 21, 9, 30, 0)
>>> print(d)
2012-12-21 09:30:00
>>>

>>> # Localize the date for Chicago
>>> central = timezone('US/Central')
>>> loc_d = central.localize(d)
```

(continues on next page)

(continued from previous page)

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

äyÄæÛæÛæIJšècnæIJñâIJřâŇŮäzEijŇ äöČâršâRřázèè;ñæ■cäyžâĚúázŮæŮúâŇžçŽDæŮúéŮt' äzEāÄæ
äyžāzEā; ŮāLřçR■āLāç; ŮārTāržāžTçŽDæŮúéŮt' iijŇä; āāRřázèè£ZæāūāAŽiijŽ

```
>>> # Convert to Bangalore time
>>> bang_d = loc_d.astimezone(timezone('Asia/Kolkata'))
>>> print(bang_d)
2012-12-21 21:00:00+05:30
>>>
```

āēČædIJā; āæL' ŠçōŮāIJæIJñâIJřâŇŮæŮæIJšäyLæL' gëāŇèōaçōŮiijŇä; āēIJĀèēAçL' žāLŇæšlæDŘād' Rā
ærTāēČiijŇâIJĪ2013āžt' iijŇç; ŌāŽ; æāGāGĚād' Rāzd' æŮúæŮúéŮt' āijĀāgŇāžŌæIJñâIJřæŮúéŮt' 3æIJL13æŮ
āēČædIJā; āæ■cāIJæL' gëāŇæIJñâIJřèōaçōŮiijŇä; āāijŽā; ŮāLřāyĀäyLéTŽèrrāĀČærTāēČiijŽ

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

çzŠædIJēTŽèrræYřāŽāäyžāöČāzūæšææIJL' èĀČèŽŠāIJæIJñâIJřæŮúéŮt' äy■æIJL' äyĀārRæŮúçŽDèûšèü
äyžāzEāfōæ■çè£ZāyLéTŽèrrījŇāRřázèä; £çTlæŮúâŇžāržèšā normalize()
æŮzæšTāĀČærTāēČiijŽ

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

èõlèõž

äyžāzEäy■èõl' ä; äècnè£ZāžZäyIJäyIJāijDçŽDæŽTād' t' è; ŇāRŠiijŇād' DçRĚæIJñâIJřâŇŮæŮæIJšçŽDèÄ
āžūçTlāöČāēæL' gëāŇæL' ĀæIJL' çŽDäy■éŮt' ā■YāCīāŠŇæš■ä; IJāĀČærTāēČiijŽ

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

äyÄæÛë;ñæ■cäyžUTCiijNä;äärsäy■çTlãÖzæNĚäfÇèùšād' Rāz d' æÛçŽyāĖşçŽDēŮóécYāzĖāĀĆ
 āZāæ■d'iijNä;āāRfāzēēšāzNāL'■äyÄæūāTlāfČçŽDæL'gēāNāyēēgAçŽDæŮēæIJşēōaçōŮāĀĆ
 ā;Şā;āæČşārĖē;ŞāGžāRŸäyžæIJñāIJræŮūēŮt' çŽDæŮūāĀZiijNä;çTlāRĹéĀĆçŽDæŮūāNžāŌzē;ñæ■cäyNā

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

ā;ŞæŮL'āRĹāĹræŮūāNžæŞ■ā;IJçŽDæŮūāĀZiijNæIJL'äyĹēŮóécYārsæYræĹSāznāēČā;Tā;ŮāĹræŮūāN
 ærTāēČiijNāIJĹēfZāyĹā;Nā■Rāy■iijNæĹSāznāēČā;TçşēēAşāĀIJAsia/KolkataāĀĹārsæYrā■rāzēārāzāTçŽDæ
 äyžāzĖæşēæL'iijNāRfāzēā;çTlISO 3166āZ;āōūāzççāAā;IJäyžāĖşēTōā■ŮāŌzæşēēYĖā■ŮāĖy
 pytz.country_timezones āĀĆærTāēČiijŽ

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

æşliijZā;Şā;æYĖērzaĹrēfZēGŃçŽDæŮūāĀZiijNæIJL'āRrēČ; pytz
 æĹāāĹŮāūşçzRāy■āĖ■āzžēōōā;çTlāzĖiijNāZāäyžPEP431æRĹāGžāzĖæZt' āĖĹēfZçŽDæŮūāNžæTræNāāĀĆ
 ā;ĖæYrēfZēGŃērĹāĹrçŽDā;Ĺād'ZēŮóécYēfYæYræIJL'āRČēĀCāzūāĀijçŽD(ærTāēČā;çTlUTCæŮēæIJşç

6 çññāZZçñāiijZēē■āzčāZlāyŌçTşæĹRāZl

ēē■āzčæYrPythonæIJĀiijzād'gçŽDāLşēČ;āzNāyĀāĀCāĹIçIJNēŮāĹēiijNā;āāRrēČ;āijZçōĀā■TçŽDēō
 çDūēĀNriijNçzĹēĹdāzĖāzĖārsæYræēČæ■d'iijNēfYæIJL'ā;Ĺād'Zā;āāRrēČ;äy■çşēēAşçŽDriijN
 ærTāēČāĹZāzžā;æēĠāūşçŽDēē■āzčāZlāržēsāiijNāIJĹitertoolsæĹāāĹŮāy■ā;çTlæIJL'çTlçŽDēē■āzčæĹāāijRiij
 ēēfZāyĀçñāçZōçŽDārsæYrāRŞā;āāsTçd'žēūşēē■āzčæIJL'āĖşçŽDāRĹDçg■äyēēgAēŮóécYāĀĆ

Contents:

6.1 4.1 æĹ'NāĹĹéA■āŌĖēē■āzčāZl

ēŮóécY

ā;āæČşēA■āŌĖäyÄäyĹāRrēf■āzčāržēsāy■çŽDæL'ĀæIJL'āĖČçt'āiijNä;ĖæYrā■t'äy■æČşā;çTlforā;ĹçČ

ēğçāĖşæŮzæāĹ

äyžāzĖæĹ'NāĹĹçŽDēA■āŌĖāRrēf■āzčāržēsāiijNä;çTl next()
 āG;æTrāzūāIJlāzççāAäy■æ■TēŌū StopIteration āijČāyāĀĆ
 ærTāēČiijNāyNēĹççŽDā;Nā■RæĹ'NāĹĹērzaŮŮäyÄäyĹæŮGāzūāy■çŽDæL'ĀæIJL'ēāNriijŽ

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

(continues on next page)

(continued from previous page)

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

éĀŽāyāæİëèőšīījŃ StopIteration çTlæİëæŃĞçd'žèf■āzççŽDçzŞārĭāĀĆ
çDūeĀŃīījŃāeĆædİJā;ăæL'ŃāLlā;fçTlāyLēİcāijTçd'žçŽD next()
āĠ;æTřçŽDērīījŃā;æfYāRřāzēēĀŽēfĞēfTāZđāyĀāyĭæŃĞāōZāĀijæİëæāĞēōřçzŞārĭīījŃærTāçĆ
None āĀĆ āyŃēİcæYřçd'žāĭŃīījŽ

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

èőİèőž

ād'ğād'ŽæTřæČĒāEřāyŃīījŃæĹSāznāijŽā;fçTl for āĭlçŌřēr■āRēçTlæİëēA■āŌĒāyĀāyĭāRřēf■āzčāržē
ā;EæYřīījŃāAūārTāzŞēİJĀēçAāržēf■āzčāAŽæŽt'āLāçş;çāōçŽDæŌğāLūīījŃēfZæUūāĀZāžEğçāžTāsĆēf■
āyŃēİcçŽDāžd'āzŞçd'žāĭŃāRŠæĹSāznāijTçd'žāžEēf■āzçæIJşēŪt'æL'ĀāRŠçTşçŽDāşžæIJñçzEēLĆīīj

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

æIJñçāæŌēāyŃæİcāĞāārRēLĆāijŽæŽt'æūsāĒēçŽDēōşēğçēf■āzççŽyāĒşæLĀæIJīījŃāL'■æRŘæYřā;ă
æL'Āāzēçāōāflā;ăāūsçzRæĹLēfZçñāçŽDāĒĒāōzçL'ççL'cèōrāIJlāfČāy■āĀĆ

6.2 4.2 äžččŘĚèŁ■äžč

éŮóécŸ

ä;äæđĎāžžāŽĚäyÄäyĥëĠāōŽāzL'āōžāŽĪāřžèsāijNéĠNéĪcāNĚāŘnæIJL'āLŮeāĪāĀAāĚČčzĎæLŮāĚŮāzŮ
ä;äæČšçŽť æŮēāIJā;äçŽĎēfŽāyĥæŮřāōžāŽĪāřžèsāyĥæL'gēāNēf■äžčæ\$■ä;IJāĀĆ

èğčāĒşæŮzæāĹ

āōđéŽĚäyĤä;äāŘĥéIJĀēēAāōŽāzL'äyÄäyĥ _____iter____()
æŮzæşŤijNāřĒēf■äžčæ\$■ä;IJäžččŘĚāĹrāōžāŽĪāĒēČĪçŽĎāřžèsāyĥLāŮžāĀĆæřŤæĆijŽ

```
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)
```

āIJāyĤĒĪcāžžčāAäy■ijN _____iter____() æŮzæşŤāŘĥæŸřčōĀā■ŤçŽĎāřĒēf■äžčæřŮæśCāijæĀŠçzŽāĒē
_children āśđæĀğāĀĆ

èóĪéőž

PythonçŽĎēf■äžčāŽĪā■ŘēōōéIJĀēēA _____iter____() æŮzæşŤēfŤāŽđäyÄäyĥāōđçŮřāžĒ
____next____() æŮzæşŤçŽĎēf■äžčāŽĪāřžèsāāĀĆ æĆæđIJā;äāŘĥæŸřēf■äžčæA■āŮēāĒŮāzŮāōžāŽĪçŽĎāĒēĀ

èfŽéĠNçŽĎ _____iter____() āĠ;æŤřçŽĎä;ŤçŤĪçōĀāNŮāžĒæžččāAijN
iter(s) āŘĥæŸřčōĀā■ŤçŽĎéĀŽēfĠēřČçŤĪ s.____iter____()

6.3 4.3 ä;ȚćŤşæĹŔăZíaĹZăzzæŮřčŽĎè■äzčælaajR

ä|äČşăđđŎřăŸĂăylēĠlaōŽăzLēř■ăzčăláăijRiijŊëüşăŽőéĂžČŽĎăĚĚç|ôăĠăŤrăřŤăăČ
range() , reversed() äŷ■ăŷĂăăăăĂČ

æCædIJä;äæČšăođčŎräyĂçg■æŮřčŽDěŁ■äzčælaajRiiJŇä;ŁçŤlăyĂäyŁçŤšæŁŔăZlăG;æŤræleăŎZăZL'ă
äyŇÉlĆæYřăyĂäyŁçŤšăzğæšŔăylēŇČăZt'ăEĚætōcĆzæŤrcŽDčŤšæŁŔăZl'ijZ

`äyžāẒĖǻ;ŁęȚl̥ēŹăyl̩GɿæTřijN̊ ä;aaRrāzēcȚlforaꞤıçÖřef■āzčáoČælÚěĂĕǻ;ŁęȚl̥ǎËüázÜæŐêârUäýÄ`

`sum()`,`list()` c■L'ãĂĆcđ'żă;ÑaeCăyÑiiJŽ

äyÄäylāĜ;æTṛäy■éIJĀēæAæIJL'äyÄäyl y i e l d ē r■āRēā■šāRfāŕEāĒūē;ñæ■cäyžäyÄäylçTšæLŖāŽlāĀĆ
 èùšæŽōéĀŽāĜ;æTṛäy■āRŇçŽDæYřiiJŇçTšæLŖāŽlāRlēČ;çTlāžŌēf■āzčæS■ā;IJāĀĆ
 äyNéIcæYřäyÄäylāōđéIŇiiJŇāRŠä;āāsTcd'žēfZæāuōŽDāĜ;æTṛāžTāsCāuēā;IJæIJžāLūiiJž


```

>>> 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
>>>

```

äyÄäyİçTşæLŘăZÍăĜ;æTřäyžèĕAçL'žă;AæYřăôČăRłaijŽăZďăžTăIJİēf■ăžčäy■ă;ĕçTłăLřçŽĎ
 next æŞ■ă;IJăĂĆ äyÄæŮĕçTşæLŘăZÍăĜ;æTřēfTăŽďēĂăĜžiiJNēf■ăžčçZŁæ■čăĂĆæĹSăžňăIJİēf■ăžčäy■é

6.4 4.4 ăôđċŎřēf■ăžčăZÍă■Řēőő

éŮőéćŸ

ă;ăæČşæđĎăžžăyĂăyİēČ;æTřæŃAēf■ăžčæŞ■ă;IJçŽĎēĜłăôŽăZŁăržēsăiiJŃăžŮăyŃæIJŽæL'ăĹřăyĂăyİē

èġčăEşæŮzæąĹ

çŽôăĹ■ăyžæ■ċiiJŃăIJăyĂăyİăržēsăyŮăôđċŎřēf■ăžčæIJăçôĂă■TçŽĎæŮžăiiJŮæYřă;ĕçTłăyĂăyİçTşæ
 ăIJİ4.2ăřRēĹĆăy■iiJŃă;ĕçTİNodeçşzæİēēăĹċd'žæăŞă;ćæTřæ■őçzşæđĎăĂĆă;ăăRřēČ;æČşăôđċŎřăyĂăyİăžēă

äyÑéíæYřázččäAçd'žä;ÑijŽ

```
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)
```

âIJléfŽæōřázččäAäyÑijŽdepth_first() æŰžæşŦçōĂăŦçŽt'èğĆăĂĆ
ăōČéęŰăĚĹèfŦăŽdèĠăûsæIJñèznázűef■ăžčæřRäyĂäyĹăŦRèĹĆçĆžăžű
éĂŽèfĠërČçŦĹăŦRèĹĆçĆžčŽĐ depth_first() æŰžæşŦ(ăĴçŦĹ yield from
ërăRè)èfŦăŽďăržăžŦăĚČçŦăăĂĆ

èőléőž

PythonçŽĐèf■ăžčăŦRèőőëęAæśĆäyĂäyĹ __iter__() æŰžæşŦèfŦăŽďäyĂäyĴL'žæŴĹçŽĐèf■ăžčăŽĹăržesăijŦ èfŽăyĹèf■ăžčăŽĹăržesăăōđčŎřăžE
__next__() æŰžæşŦăžűéĂŽèfĠ StopIteration âijĆăyŷæăĠërĚèf■ăžččŽĐăŵŦăĹŦăĂĆ
ăĴĚæYřijŦăăōđčŎřèfŽăžŽéĂŽăyŷăijŽæřŦèĴČçžAçŦŦăĂĆ äyÑéíæĹŦsăžñæijŦçd'žăyŦèfŽčğæŰžăijŦijŦă
depth_first() æŰžæşŦijŽ

```

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 çśzāŠŇäyŁéİcā;ŁçŦİçŦŦşæŁŖāZİçŽĐçL'ŁæIJñāüēä;IJāŦŦşçŖĖçśzāijijijŦ
ä;ĖæŸŖāŦŦĈāĖŽēŦūæİēā;ŁçzAçŖŖijŦāZāyžēŁ■āzčāZİāŁĖĖēāzāIJĖŁ■āzčād'ĐçŖĖĖŁĖçİŦāy■çzt'æŁd'ād'ģēŦ
āİēçZ;ĖİēēŦŦijŦāşāžžæĐŁæĐŖāĖŽēŁZāzŁæŽēæŦŦ'çŽĐāzčçāAāĈĈāŖĖä;āçŽĐēŁ■āzčāZİāŦŦZāzL'äyžäyĀäy

6.5 4.5 āŖ■āŖŠēŁ■āzč

éŦŦēčŸ

ä;āæČşāŖ■æŦŦzāŖŠēŁ■āzčāyĀäyŁāzŖāŁŦ

èğçāĖşæŦŦzæāŁ

ä;ŁçŦŦāĖĖç;ŦçŽĐ reversed() āĖ;æŦŦijŦāŖŦāĖČijŽ

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

āŖ■āŖŠēŁ■āzčāzĖāzĖā;ŞāržēşāçŽĐād'ğārŖāŖŖēçĐāĖŁçāŦŦŦžæŁŦŦēĀĖāržēşāŦŦŦçŦŖāžĖ
__reversed__() çŽĐçL'zæŦŁæŦŦzæşŦæŦŦæL'ēČ;çŦŦşæŦŦāĈ
āĖČādIJāyđ'ēĀĖēČ;äy■çņēāŖŦijŦēČčā;āāŁĖēāzāĖŁāŖĖāržēşāē;ñæ■cāyžäyĀäyŁāŁŦŦēāŁ'ēāŦŦijŦāŖŦāĖČijŽ

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

ēēAæşŁæĐŖçŽĐæŸŖāĖČæđIJāŖŖēŁ■āzčāržēşāĖĖČçt'āā;Łād'ŽçŽĐŖŦijŦāŖĖāĖŦēçĐāĖŁē;ñæ■cāyžäyĀäy

èŦĖēŦž

ā;Łād'ŽçİŦāžŖāŠŸāzŦüä■çşēēAşşāŖŖāžēēĀŽēŁĖĖāIJĖĖĖāŦŦZāzL'çşzäyŁāŦŦçŦŖ
__reversed__() æŦŦzæşŦæİēāŦŦçŦŖāŖ■āŖŠēŁ■āzčāĈĈāŖŦāĖČijŽ

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

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

(continues on next page)

äyžāẒĒā;ŁçŦlĕŦŽāyŁçšzījŦā;āāŦŕāzēāŦĒāōČā;ŠāAŽāĲŕāyĀāyŁāŽōēĀŽçŽĐçŦšāĹŦāŽlāĠ;āŦŕāĀČ
 çĐūēĀŦījŦçŦšāžŌāŦŕāzēāĹŽāžzāyĀāyŁāōđā;ŦāŦzēšāījŦāžŌāĲŕā;āāŦŕāzēēōĕēŬōāĒĒēČlāšđāĀġāĀījŦījŦ
 āŦŦāēČ history āšđāĀġāĹŦēĀĒāĲŦ clear() āŦžāšŦŦāĀČāžčçāĀçđ'žā;ŦāēČāyŦījŽ

```
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='')
```

ēōlēōž

āĒšāžŌçŦšāĹŦāŽlāījŦā;ĹāōžāĲšāŌĹēŦŽāĠ;āŦŕāŦāāĹĀāy■ēČ;çŽĐēŽūēŲšāĀČ
 āēČādĪçŦšāĹŦāŽlāĠ;āŦŕēĪĀēēĀēūšā;āçŽĐçlŦāžŦāĒūāžŦēČlāĹēāĹŦšāžđ'ēĀšçŽĐēŦl(āŦŦāēČāŽŦ'ēĪšā
 āŦŕēČ;āījŽāŦījēĠŦ'ā;āçŽĐāžčçāĀāījČāyŲçŽĐād'■āīČāĀČ āēČādĪāĲŦŕēŦŽçġ■āēČēĀĒçŽĐēŦlījŦāŦŕāzēēĀČ
 āĪĹ __iter__() āŦžāšŦŦāy■āōŽāžĹā;āçŽĐçŦšāĹŦāŽlāy■āījŽāŦžāŦŲā;āāžzā;ŦçŽĐçōŬāšŦŦēĀžē;ŠāĀČ
 çŦšāžŌāōČāĲŦçšžçŽĐāyĀēČlāĹēījŦāēĹĀāžēāĒēōyā;āāōŽāžĹāŦŦçġ■āšđāĀġāŠŦāŦžāšŦŦēĀ;ŽçŦlāēĹ

āyĀāyĹēĪĀēēĀēšāēĎŦŦçŽĐāŦŦāĪŦāŦžāĲŦījŦāēČādĪā;āāĪĹēŦ■āžčāēš■ā;ĪāŦūāy■ā;ŁçŦlŦŦā;ŁçŦŦē
 iter() āĠ;āŦŦāĀČāŦŦāēČījŽ

```
>>> 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 ēē■āžčāŽlāĹĠçĹĠĠ

ēŬōēčŲ

ā;āāČšā;ŦāĹŦāyĀāyŁçŦšēē■āžčāŽlçŦšāĹŦŦçŽĐāĹĠçĹĠĠŦāŦzēšāījŦā;ĒāĲŦŦāēāĠāĠēĀĹĠçĹĠĠēš■ā;Īāž

ēġčāĒšāŦžāēāĹ

āĠ;āŦŦŦertools.islice() ā■čāē;ēĀČçŦlāžŌāĪĹēŦ■āžčāŽlāšŦçŦšāĹŦāŽlāyĹāĀžāĹĠçĹĠĠēš■ā;Īāž

```

>>> 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
>>>

```

èõléõž

è£■āzčāŽlāšNčŤšæLRāŽlāy■ēČ;ä;£çŤlæāGāGĖçŽDāLGçL'Gæš■ā;IJiijNāZāāyžāóČāzñçŽDēŤ£āžēāžN
 āG;æŤŤŝlice() è£ŤāZđāyĀāyġāRřāzēçŤšæLRæNĠāóZāĖČçŤ'āçŽDè£■āzčāŽlāy■ēČ;ä;£çŤlæāGāGĖçŽDāLGçL'Gæš■ā;IJiijNāZāāyžāóČāzñçŽDēŤ£āžēāžN
 çDūāRŌāL'■āijĀāğNāyĀāyġāyġçŽDè£ŤāZđāĖČçŤ'āijNāžūçZŤ'āLŝāL'GçL'GçzšæĲšçŤ'čāijŤā;■ç;ōāĀĆ

è£ŽéGŇēçAçĲĀéG■āijžērČçŽDāyĀçČzæŸŝ islice()
 āijŽæŭLèĀŮāŌL'āijāāĖēçŽDè£■āzčāŽlāy■ēČŽDæŤŝŝāĀĆ ā£ĖēāzèĀČèZšāLŝŝē£■āzčāŽlāy■ēČŝāRŝŝĀĖçŽ
 æL'ĀāžēāçČæđIJā;āēIJĀēçAāžNāRŌāĖ■ēñāçō£ēŮōē£Žāyġē£■āzčāŽlāy■ēČŽDēŤiijNēČčā;āāŝā;ŮāĖLāŝĖāōČēČ

6.8 4.8 èũşè£ĠāRŝŝē£■āzčāržèšāçŽDāijĀāğNéČlāĖ£

éŮōéćŸ

ā;āæČšēA■āŌĖāyĀāyġāRŝŝē£■āzčāržèšāijNā;ĖæŸŝāōČāijĀāğNçŽDæšRāžZāĖČçŤ'āā;āāžūāy■ēDšāĖŤ'è

èġċăEşæŮzæąŁ

itertools æÍąąİŮäy■æIJL'äyÄäzZăĜ;æTŗăRřäzëăŃæLŘëfZäyłäzzăŁąăĂĆ
éęŮăĚŁäzŃçz■çŽĐæŸř itertools.dropwhile() âĜ;æTŗăĂĆă;ŁçŤłæŮŭřijŃă;ăçzZăŃăŃăăĂšăyĂäy
ăŃăijZëfŤăZđäyĂäyłëf■ăzčăZłăřzëšăijŃăyčăijČăŮšæIJL'ăzRăĹŮäy■çŽt'ăĹrăĜ;æTŗëfŤăZđFlaseăzŃăĹ
äyžăžEæijŤčđ'zřijŃăAĜăŃăZă;ăăIJłérzăRŮäyĂäyłăijĂăĝŃéĆłăŁEæŸřăĜăăNăşłéĜŁçŽĐæžŘæŮĜăžŭă

```
>>> 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
...
>>>
```

ăęĆăđIJă;ăăČşëŭşëfĜăijĂăĝŃéĆłăŁEçŽĐăşłéĜŁăăŃçŽĐërĹijŃăRřäzëëfZăăŭăĂŽřijŽ

```
>>> 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
...
>>>
```

ëfZäyłä;Ńă■RăŸřăşzăžŮăăzăæ■ŃăşŘăyłætŃërŤăĜ;æTŗëŭşëfĜăijĂăĝŃçŽĐăĚČčť'ăăĂĆ
ăęĆăđIJă;ăăŭşçzŤăŸŮăŃăŃşëéAşăžEëęAëŭşëfĜçŽĐăĚČčť'ăçŽĐäyłæTŗçŽĐërĹijŃéĆčăzŁăRřäzëă;ŁçŤł
itertools.islice() æłëăzčăZăăĂĆăřŤăęĆřijŽ

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

```
    aIJleſZäyläçNā■Räy■iijN islice() aĜjæTřæIJĀāRŌéĆčäyl None
aŔĆæTřæNĜāōZāžEājāēēAēŌūāRŪāzŌčññ3äylāLřæIJĀāRŌčŽDæL'ĀæIJL'āĚČčt'āiijN
āēČædIJ None āŠN3çŽDäj■çjōāržērČiijNæĎRæĀIārſæYřāzĚāzĚēŌūāRŪāL■äyL'äylāĚČčt'āæAřæAřçŽyāŔ
(ēſZäylēūšāLĜçL'ĜçŽDçŽyāR■æŠ■äjIJ [3:] āŠN [:3] āŌšçRĚæYřäyĀæäüçŽD)āĀĆ
```

èõlèõž

```
aĜjæTřdropwhile() āŠN islice() āĚūāōdāŕſæYřäyd'äylāyōāL'aĜjæTřiijNäyžçŽDārſæYřéAſāL
```

```
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)
```

```
ēūšēſĜäyĀäylāRřēf■āzčāržēsāçŽDāijĀāĝNéČlāLĚēūšéĀŽāyŷçŽDēſĜæzd'æYřäy■āRŅçŽDāĀĆ
ærTāēČiijNäyLēſrāzčçāAçŽDçññäyĀäylēČlāLĚāRřēČjāijŽēſZæäüéĜ■āĚŽiijŽ
```

```
with open('/etc/passwd') as f:
    lines = (line for line in f if not line.startswith('#'))
    for line in lines:
        print(line, end='')
```

```
ēſZæūūāĚZçāōdāRřāzēēūšēſĜāijĀāĝNéČlāLĚçŽDæſléĜLēāNriijNājĚæYřāRŅæāūāzšāijŽēūšēſĜæŪ
æ■cāRēērĪēōšiiijNæLŠāzñçŽDēĝcāĚſæŪzæāLæYřāzĚāzĚēūšēſĜāijĀāĝNéČlāLĚæzæēūšætNērTæĪāzūçŽDē
```

```
æIJĀāRŌéIJĀēēAçĪĀéĜ■āijžērČçŽDäyĀçČzæYřiijNæIJñēŁČçŽDæŪzæāLēĀČçTlāžŌæL'ĀæIJL'āRřēſ
ærTāēČçTšæLŔāŽĪiijNæŪĜāzūāRĻāĚūçšzāiijçŽDāržēsāāĀĆ
```

6.9 4.9 æŌŠāLŪçzDāRĻçŽDēſ■āzč

éUōécŸ

```
ājāæČšēſ■āzčēA■āŌĚäyĀäylēZEāRĻäy■āĚČčt'āçŽDæL'ĀæIJL'āRřēČjçŽDæŌŠāLŪæLŪçzDāRĻ
```

èġċaEşæŮzæaĹ

itertoolsaĹaĹŮæRŘă;ZăŹEăyL'ăyĹăĜĵæTŗæĹèèġċaEşæfZşşzéŮőécŸăĂĆ
ăĔŮăy■ăyĂăyĹăŸř itertools.permutations() iĵŃ
ăŎŢăŎăRŮăyĂăyĹăŹEăRĹăzŮăzġġTşşyĂăyĹăĔĈġzĎăŹRăĹŮiĵŃăŕRăyĹăĔĈġzĎġTşşEăRĹăy■ăL'ĂăIJL'
ăzşăŕşăŸřèřťéĂŹéŹĜăL'ŞăzşéŹEăRĹăy■ăĔĈġťăăŎŖăĹŮéăŹăŹRġTşşăL'RăyĂăyĹăĔĈġzĎiĵŃăŕTăeĈiĵŹ

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

ăeĈădIJăjăăĈşăĹŮăĹŕăŃĜăŏŹéTřăŹġġŹĎăL'ĂăIJL'ăŎŖăĹŮiĵŃăjăăŕăŹéăiĵăeĂŞăyĂăyĹăŕŕéĂĹġŹ

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

ăĵġĤĹitertools.combinations() âŖŕăĹŮăĹŕèĹşăĔééŹEăRĹăy■ăĔĈġťăġŹĎăL'ĂăIJL'ġŹĎġzĹ

```
>>> from itertools import combinations
>>> for c in combinations(items, 3):
...     print(c)
...
('a', 'b', 'c')

>>> for c in combinations(items, 2):
...     print(c)
...
('a', 'b')
('a', 'c')
('b', 'c')

>>> for c in combinations(items, 1):
...     print(c)
```

(continues on next page)

(continued from previous page)

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

```
arżaző combinations() æİëëőšijŇăĚĈt'ăçŽĐéążăžRăũşçzRăy■éĜ■èęAăžEăĂĈ
ăžşărşæYřèrt'ijŇçzĐăŘĹ ('a', 'b') èűş ('b', 'a')
ăĚűăőđæYřăyĂăűçŽĐ(æIJĂçzĹăŘĹăijŽè;ŞăĜăăĚűăy■ăyĂăyĹ)ăĂĈ
```

```
ăIJĹëőăçőŮçzĐăŘĹçŽĐăŮűăĂŽijŇăyĂăŮęăĚĈt'ăèćnéĂĹ'ăŔŮăřşăijŽăžŎăĂŽéĂĹ'ăy■ăĹ'ŤéŽđ'æŎĹ'
èĂŇăĜ;æŦř itertools.combinations_with_replacement()
ăĚĂëőyăŔŇăyĂăyĹăĚĈt'ăèćnéĂĹ'ăŇĹ'ăđ'ŽăŋăijŇăŕŦăęĈijŽ
```

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

èőİëőž

```
èĚŽăyĂăŕŔèĹĈăĹŚăžŇăŔŚă;ăăşŦçđ'žçŽĐăžĚăžĚăYř itertools
ăĹăăĹŮçŽĐăyĂéĈĹăĹĚăĹşèĈ;ăĂĈ âŕ;çőăă;ăăžşăŔŕăžèèĜăũşăĹ'ŇăĹăăđçŎŕăŎşăĹŮçzĐăŘĹçőŮăşŦijŇă
ă;ŞăĹŚăžŇçŕăĹŕçIJŇăyĹăŎžăIJĹ'ăžŽăđ'■ăĹĈçŽĐèĚ■ăžçéŮőécYăŮűijŇăIJĂăę;ăŔŕăžèăĚĹăŎžçIJŇçIJŇă
ăęĈăđIJĹëŦŽăyĹéŮőécYăĹăŦăžőéĂ■ijŇéĈçăžĹăĹăIJĹ'ăŔŕèĈ;ăijŽăIJĹéĜŇéĹăĹ;ăĹŕëğçăĚşăŮžăăĹijĂ
```

6.10 4.10 ăžŔăĹŮăyĹçŦ'căijŦăĂijèĚ■ăžč

éŮőécY

```
ă;ăăĈşăIJĹëĚ■ăžçăyĂăyĹăžŔăĹŮçŽĐăŔŇăŮűëűşèyĹă■căIJĹéćŇăđ'ĐçŔĚçŽĐăĚĈt'ăçŦ'căijŦăĂĈ
```

èğçăĚşăŮžăăĹ

```
ăĚĚç;őçŽĐ enumerate() âĜ;æŦŕăŔŕăžèăĹăăç;çŽĐèğçăĚşăèĚŽăyĹéŮőécYijŽ
```

```
>>> my_list = ['a', 'b', 'c']
>>> for idx, val in enumerate(my_list):
...     print(idx, val)
...
0 a
1 b
2 c
```

äyžāẒEæŃL'äijăçzşèaŃăRûè;ŞăĜž(èaŃăRûăzŌ1ăijĂăğŃ)îijŃă;ăăRřăzèăijăéĂŞăyĂăyĹăijĂăğŃăRĆæŢŕ

```
>>> my_list = ['a', 'b', 'c']
>>> for idx, val in enumerate(my_list, 1):
...     print(idx, val)
...
1 a
2 b
3 c
```

è£Žçğ■æĈĖaĖĹăĹă;ăéA■ăŌEæŪĜăzŭăŪăăĈşăĹĹéŢŢZèŕăŭĹăAŕăy■ă;£çŢĹèaŃăRûăăžă;■æŪăăĂŽéĹ

```
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() âřzāẒŌèŭşèyĹæŞŔăžZăĂijăĹăĹăŪèaĹăy■ăĜžçŌřçŽĎă;■ç;óæŸŕăĹăĹăĹçŢĹçŽĎăĂĈ
æĹ'ĂăžēĹijŃăæĈăđĹă;ăæĈşăŕEăyĂăyĹăŪĜăzŭăy■ăĜžçŌřçŽĎă■Ţŕ■æŸăăŕĎăĹăăĈăĜžçŌřçŽĎăaŃăRûăy
enumerate() æĹèăŏŃăĹŕĹijŽ

```
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)
```

ăæĈăđĹă;ăăđ'ĎçŔĖăŏŃăŪĜăzŭăŔŌæĹ\$ă■ŕ
îijŃăijŽăŔŚçŌŕăŏĈæŸŕăyĂăyĹă■ŪăEŷ(ăĖEçăŏăĹèèŏşæŸŕăyĂăyĹ
)îijŃă âřzāẒŌæŕŔăyĹă■Ţŕ■æĹĹăăyĂăyĹ key îijŃăŕŔăyĹ key
âřzāẒŢçŽĎăĂijăŸŕăyĂăyĹçŢşè£ŽăyĹă■Ţŕ■ăĜžçŌřçŽĎăaŃăRûçzĎăĹŔçŽĎăĹŪèaĹăĂĈ
ăæĈăđĹăŞŔăyĹă■Ţŕ■ăĹăyĂèaŃăy■ăĜžçŌřè£Ĝăyđ'æŋăijŃéĈăzĹè£ŽăyĹèaŃăRûăzşăijŽăĜžçŌŕăyđ'æŋă

āŕŅæŮüäzšāŕfäzēä;IJäyžæŮĜæIJŋçŽDäyÄäylçōÄā■TçzšèōāāĀĈ

èõlèõž

ā;Šä;āæĈšécĪāđ' ŮāōŽāzL'äyÄäylèōāæTŕāŔŸéĜŔçŽDæŮüāÄŽiijŅä;ŕçTĪ
enumerate() āĜ;æTŕäijŽæŽt' āLāçōÄā■TāĀĈä;āāŔŕèĈ;äijŽāĈŔäyŅéĪçèŁŽæāüāEžZāzčçāAiiijŽ

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

ä;EæŸŕæĈæđIJä;ŕçTĪ enumerate() āĜ;æTŕæĪèäzçæŽŁāŕšæŸ;ā;ŮæŽt' āLāäijŸéŽĒäžEiiijŽ

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

enumerate() āĜ;æTŕèŁTāZđçŽDæŸŕäyÄäyl enumerate āŕžèšāāōđä;ŅiiijŅ
āōĈæŸŕäyÄäylèŁ■äzçāŽiijŅèŁTāZđèŁđçz■çŽDāŅĒāŔŅäyÄäylèōāæTŕāŠŅäyÄäylāÄijçŽDāĒĈçzDiiijŅ
āĒĈçzDäy■çŽDāÄijéÄŽèŁĜāIJāijāāĒèäžŔāĻŮäyĻèŕĈçTĪ next() èŁTāZđāĀĈ

èŁŸæIJL'äyÄçĈzāŔŕèĈ;āzūäy■ā;ĻéĜ■èçAiiijŅä;EæŸŕäzšāÄijā;ŮæšĻæĎŔiiijŅ
æIJLæŮüāÄŽä;Šä;āāIJläyÄäylāüšçzŔèĝçāŌŅāŔŌçŽDāĒĈçzDāžŔāĻŮäyĻä;ŕçTĪ
enumerate() āĜ;æTŕæŮüā;ĻāōžæŸŠèŕĈāĒèéŽüéŸsāĀĈ
ä;āā;ŮāĈŔäyŅéĪçæ■ççāōçŽDæŮžäijŔèŁŽæāüāEžZiijŽ

```
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 āŕŅæŮüèŁ■äzçāđ'ŽäyĻāžŔāĻŮ

éŮōécŸ

ä;āæĈšāŕŅæŮüèŁ■äzçāđ'ŽäyĻāžŔāĻŮiijŅæŕŔæŋāāĻEāĻŅāzŌäyÄäylāžŔāĻŮäy■āŔŮäyÄäylāĒĈçt' āāÄ

èĝçāEšæŮžæāĻ

äyžāžEāŕŅæŮüèŁ■äzçāđ'ŽäyĻāžŔāĻŮiijŅä;ŕçTĪ zip() āĜ;æTŕāĀĈæŕTæĈŕiijŽ

```
>>> 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) (x, y)

çŽĐēēāzčāŽlījŇāĚūāy■xāĪēēĠlāiijŇyāĪēēĠbāĀĆ āyĀāŮēāĚūāy■āšŘāyġāzŘāĹŮāĹřāžŤčzŠāřġiijŇēē■āzāŽāē■d'ēē■āzčēŤēāžēēūšāŖĆāŤřāy■āĪĀçš■āžŘāĹŮēŤēāžēāyĀēĠ'āĀĆ

```
>>> a = [1, 2, 3]
>>> b = ['w', 'x', 'y', 'z']
>>> for i in zip(a,b):
...     print(i)
...
(1, 'w')
(2, 'x')
(3, 'y')
>>>
```

āēĆāēdĪēēŽāyġāy■āŸřāġāēČšēēĀçŽĐāŤĹāēdĪīijŇēĆčāzĹēēŸāŖřāzēāġēçŤĪ

itertools.zip_longest() āĠġāŤřāĪēāzčāŽēāĀĆāēŤāēĆīijŽ

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

èóìèõž

ā;Šā;āæČšæĹŔāřžāđ'ĐčŘĚæŤřæ■óçŽĐæŮūāŽž zip()
āĠ;æŤřæŸřā;ĹæIJĹ'čŤĭçŽĐāĀČ æřŤāæČřijNāAĠĚō;ā;āāđ't'āĹŮēāĹāŠNāyĀāyĹāĀijāĹŮēāĹijNāřsāČŘāyNéĹ

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

ā;ĤçŤĭzip()āŔřāžēēōŤ'ā;āāĚāōČāžnæĹ'ŠāNĚāžūçŤšæĹŔāyĀāyĹā■ŮāĚyijŽ

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

æĹŮēĀĚā;āāžšāŔřāžēāČŘāyNéĹcēĤæūūāžgçŤšē;ŠāĠžijŽ

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

ēŽ;çĐūāy■āyÿēĠAijNā;ĚæŸř zip() āŔřāžēæŌēāŮāđ'ŽāžŌāyđ'āyĭçŽĐāžŔāĹŮçŽĐāŔČæŤřāĀČ
ēĤŽæŮūāŽžæĹ'ĀçŤšæĹŔçŽĐçzŠæđIJāĚČçzDāy■āĚČçt'āāyĹæŤřēūšē;ŠāĚēāžŔāĹŮāyĹæŤřāyĀæūūāĀČæřŤ

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

æIJāāŔŌāijžērČāyĀçČžāřsæŸřijN zip() āijŽāĹŽāžžāyĀāyĹē■āžčāŽĹæĹēā;IJāyžçzŠæđIJēĤāŽđāĀČ
āēČæđIJā;āēIJāēĠAāřĚçzŠāřçžŽĐāĀijā■ŸāČĹāIJāĹŮēāĹāy■ijNēĠAā;ĤçŤĭ list()
āĠ;æŤřāĀČæřŤāēČřijŽ

```
>>> zip(a, b)  
<zip object at 0x1007001b8>  
>>> list(zip(a, b))  
[(1, 10), (2, 11), (3, 12)]  
>>>
```

6.12 4.12 āy■āŔNéŽĚāŔĹäyĹāĚČçt'āçŽĐēĤ■āžč

éŮōécŸ

ā;āæČšāIJāđ'ŽāyĹāřžēsæĹ'gēāNçŽyāŔNçŽĐæŠ■ā;IJijNā;ĚæŸřēĤŽāžŽāřžēsāāIJāy■āŔNçŽĐāōžāŽĹā

èġċaEşæŮzæaĹ

itertools.chain() æŮzæşTāRřäzëċTĹæİëċōĀāNŮëfZāyĹäzzāŁaāĀĆ
ăŌĈæŌëāRŮäyĀäyĹāRřëf■äzċāržèśāāĹŮëāĹāĹJäyžèĹŞāĒëĹijNāzŮëfTāZđäyĀäyĹëf■äzċāZĹijNæIJL'æTĹċZĹ
äyžäZĒæijTċd'žæyĒæëZĹijNëĀĈèZŚäyNéİċëfZāyĹāĹNā■RĹijZ

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

äĹċTĹ chain() ċZĐäyĀäyĹäyžèġAāIJæZřæYřāĹŞāĹæĈşāržäy■āRŹċZĐéZĒāRĹäy■æL'ĀæIJL'ăĒĈċ

```
# 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
```

ëfZċġ■èġċaEşæŮzæaĹLëeAæřTāĈRäyNéİċëfZæāüāĹċTĹäyđ'äyĹā■TċNñċZĐāĹċŌřæZř'āŁāāijYéZĒijN

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

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

ëŋĹëŋž

itertools.chain() æŌëāRŮäyĀäyĹæĹŮād'ZāyĹāRřëf■äzċāržèśāāĹJäyžèĹŞāĒëāRĈæTřāĀĆ
ċĐüāRŌāĹZāzzäyĀäyĹëf■äzċāZĹijNāĹIænāeĹđċž■ċZĐëfTāZđæřRäyĹāRřëf■äzċāržèśāāy■ċZĐāĒĈċř'āāĀĆ
ëfZċġ■æŮzāijRëeAæřTāĒĹāřĒāzRāĹŮāRĹāzŮāĒëf■äzċëeAēñYæTĹċZĐād'ZāĀĆæřTāeĈijZ

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

(continues on next page)


```

...

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

```

čňňäyĂçğ■æŰzæŁäy■iijŇ a + b æŞ■äiIJäijŽăĹŽăzzăyĂäyĹăĒĹæŰřçŽĎăžŔăĹŰăzŭèèAæśĆăăŇbçŽ
 chian() äy■äijŽæIJĹèĤŽăyĂæ■ēiijŇæĹĂăžēăèĆæđIJèĭŞăĒēăžŔăĹŰēĹđăyŷăđ'ğçŽĎæŰŭăĂŽăijŽăĭĹçIJA
 âzŭăyŤăĭŞăŔŕèĤ■ăzčăržèśăçşăđŇăy■ăyĂæăŭçŽĎæŰŭăĂŽ chain()
 âŔŇæăŭăŔŕăžēăĭĹăēĭçŽĎăŭēăiIJăĂĆ

6.13 4.13 âĹŽăzzæŤŕæ■óăđ'ĎçŔĖçőăéAŞ

éŰóécŸ

ăĭăăÇşăžæŤŕæ■óçőăéAŞ(çşzăijijUnixçőăéAŞ)çŽĎæŰzăijŔèĤ■ăzčăđ'ĎçŔĖæŤŕæ■óăĂĆ
 æŕŤăèĆiijŇăĭăæIJĹăyĹăđ'ğéŔŔçŽĎæŤŕæ■óéIJăèèAăđ'ĎçŔĖiijŇăĭĖæŸŕăy■èĆĭăŕĖăőČăžňăyĂæňăæĂğăŤĭ

èğčăĖşæŰzæăĹ

çŦşăĹŔăŽĹăĠĭæŤŕæŸŕăyĂäyĹăđçŦŔçőăéAŞæIJăĹăĹçŽĎăēĭăĹđăşŤăĂĆ
 äyžăžĖæijŤçđ'žiiijŇăAĠăđŽăĭăèèAăđ'ĎçŔĖäyĂäyĹăĹđăyŷăđ'ğçŽĎæŰēăĹŰăŰĠăžŭçŽăăĭŤiijŽ

```

foo/
  access-log-012007.gz
  access-log-022007.gz
  access-log-032007.gz
  ...
  access-log-012008
bar/
  access-log-092007.bz2
  ...
  access-log-022008

```

ăAĠèőĭæŕŔăyĹæŰēăĹŰăŰĠăžŭăŇĒăŔŕèĤŽæăŭçŽĎæŤŕæ■óiiijŽ

```

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 -
...

```

äyžăžĖăđ'ĎçŔĖèĤŽăžŽæŰĠăžŭiijŇăĭăăŔŕăžēăăđŽăžĹăyĂäyĹçŦśăđ'ŽăyĹæĹĭğăăŇçĹĭăăđŽăžăăĹăçŇňçŇŇ

```
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(filenamees):
    """
    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 filenamees:
        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
```

```

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)

```

æĈæđIārEæIēçŽDæŮŭāĀŽā;ăæĈşæL'ĭāsŤçōăeAŞŭijNă;ăçŤŽeĜşăRfăzeăIJĭçŤşæĹŖăZĭeăĭēĭăĭjRăy
ærŤăĈĭijNăyNéIcēfZăyĭçL'ĹæIJnēōaçōŮăĜzăijăēĭŞçŽDăŮēĹCæŤŖăzŭēōaçōŮăĒŭăĀzăŞNăĀĈ

```

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))

```

èóIèőž

ăzēçōăeAŞæŮzăijRăd'ĐçŘEæŤŖæŭăRfăzēçŤĹæIēēĝcăEşăRĐçşzăĒŭăzŮēŮōécŸĭijNăNĒæNnēĝcæđŘĭ
ăyžăEçŘEēĝcăyĹēfŖăzçcăAĭijNéĜŭĈzæŸŕeæAæŸŌçŽ; yield
ērăRēă;IJăyžæŤŖæŭōçŽĐçŤşăzĝeAĒĒĀN for âĭçŖēŕăRēă;IJăyžæŤŖæŭōçŽDæŮĹet'zēĀĒăĀĈ
ăĭŞēfZăzŽçŤşæĹŖăZĭcñēfđăIJăyĀēŭăRŌĭijNăŕRăyĭ yield
ăĭjZărEăyĀăyĭăŤçNnçŽDæŤŖæŭăĒĈŕ'ăăĭjăeĀŞçzŽēfăăzçăđ'ĐçŘEçōăeAŞçŽDăyNăyĀēŸŭăōŭăĀĈ
ăIJĭăĭNăŖăIJăăRŌēĈĹăĹEĭijN sum() âĜĭæŤŖæŸŕæIJăçzĹçŽĐçĹNăzŖēŕsăĹĹēĀĒĭijNăŕRăŋăzŌçŤşæĹŖă
ēfZçĝŭæŮzăijRăyĀăyĭēĹdăyŷăē;çŽĐçL'zçCzæŸŕæŕRăyĭçŤşæĹŖăZĭăĜĭæŤŖăĹăŕRăzŭăyŤēĈ;æŸŕçNnç
ăĭĹăđ'ZæŮŭăĀŽĭijNēfZăzŽăĜĭæŤŖæĈæđIærŤēĭĈēĀŽçŤĭçŽĐēŕĹăRfăzeăIJăĒŭăzŮăIJzæŖŕēĜăđ'ăĭ;ĕçŤ
ăzŭăyŤæIJăçzĹăŕEēfZăzŽçzĐăzŭçzĐăŖĹēŭăIēçŽDăzçcăAçIJNăyĹăŌzēĹdăyŷçōĀăŤĭijNăzşăĭĹăōzæŸŞç
ăĭ;ĕçŤĭēfZçĝŭæŮzăijRçŽDăĒĒăŸæŤĹçŌĜăzşăyăĭŮăyăŕRăĀĈăyĹēfŖăzçcăAăşăĭæŸŕăIJăyĀăy
ăžNăōđăyĹĭijNçŤşăzŌăĭ;ĕçŤĹăzEēfăăzçæŮzăijRăd'ĐçŘEĭijNăzçcăAēfŖēăNēfĜçĹNăyăŕĹēIJăēçAăĭĹăŕRă
ăIJĭērĈçŤĭ gen_concatenate() âĜĭæŤŖçŽDæŮŭăĀŽă;ăăŖŕēĈ;ăĭjZæIJĹăzZăyăđ'ĹæŸŌçŽ;ăĀĈ
ēfZăyĭăĜĭæŤŖçŽĐçŽçŽDæŸŕăŕEē;ŞăĒēăzŖăĹŮăNĭjæŌēæĹŖăyĀăyĭăĭ;ĹēŤĕçŽDēăNăzŖăĹŮăĀĈ
itertools.chain() âĜĭæŤŖăŖNăăŭăIJĹçşzăĭijçŽDăĹşēĈ;ĭijNă;EæŸŕăŏĈēIJăēçAăŕEæL'ĀæIJĹăŖŕă
ăIJăyĹēĹcēfZăyĭă;NăŖăyăĭijNă;ăăŖŕēĈ;ăĭjZăĒZçşzăĭijjēfZăăŭçŽĐērăŖē
lines = itertools.chain(*files) ĭijN ēfZărEăŕĭjēĜŕ
gen_opener() çŤşæĹŖăZĭcñæŖŖăĹăĒĹēĈĹæŮĹet'zæŌĹăĀĈ âĭEçŤşăzŌ
gen_opener() çŤşæĹŖăZĭăŕŖæŋaçŤşæĹŖăyĀăyĭăĭ;ŖăĭjăēfĜçŽDæŮĜăzŭĭijN
çŭĹăĹŖăyNăyĀăyĭēfăăzçăŕēĹd'æŮŭăŮĜăzŭăŕsăĒēŖŮăăzEĭijNăZăăđ' chain()
ăIJĭēfZēĜNăyăēĈ;ēfZăăŭăĭ;ĕçŤĭăĀĈ äyĹēĹççŽDæŮzæăĹăŖăzēēAăĒăēfZçĝŭæĈĒăĒăĀĈ
gen_concatenate() âĜĭæŤŖăyăăĜzçŖŕēfĜ yield from ēŕăŖēĭijNăŏĈăŕE
yield æŞăĭIJăzççŘEăĹŕçĹŭçŤşæĹŖăZĭăyĹăŌzăĀĈ ēŕăŖē yield from
it çōĀăŤçŽDēfŤăZđçŤşæĹŖăZĭ it æĹăăzĝçŤşçŽDæĹăæIJĹăĀĭjăĀĈ
ăĒşăzŌēfZăyĭăĹŖăzŋăIJĹ4.14ăŕŖēĹCăĭjZæIJĹæZŕ'ēfZăyĀæŕēçŽDæŖŖēŕăĀĈ

æIJĀāŔŌēƒŸæIJL'äyĀçCzéIJĀèçAæşlæDŔçŽDæŸriijŊçōæéAşæŪzāijRāzūāy■æŸrāyĠēČ;çŽDāĀĆ
 æIJL'æŪūāĀŽā;āæČşçñŊā■şād'ĐçŘEæL'ĀæIJL'æŦræ■ōāĀĆ çDūèĀŊriijŊā■şā;£æŸrēƒŽçğ■æČĒāEŦriijŊā;£
David Beazley āIJlāzŪçŽD Generator Tricks for Systems Programmers
 æŦŽçlŊāy■ārřāžŌēƒŽçğ■æL'ĀæIJræIJL'ēlđāyŸæūsāĒēçŽDèōşēğçāĀĆāŔřāzēāŔĆēĀČēƒŽāyġæŦŽçlŊēŌūāŔ

6.14 4.14 āŦāijĀāŦŊāēŪçŽDāžŔāLŪ

éŪŌéćŸ

ā;āæČşārEäyĀäyġlād'ŽāsČātŊāēŪçŽDāžŔāLŪāsŦāijĀæLŔāyĀäyġlā■ŦāsČāLŪēāġ

èğçāEşæŪzæāġ

āŔřāzēāEŦāyĀäyġlāŊĒāŔñ yield from ēŕ■āŔēçŽDēĀŖā;ŞçŦşæLŔāZlāēē;zāġ;èğçāEşēƒŽāyġéŪŌéćŸ

```
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)
```

āIJlāyġēlčāzççāAäy■riijŊ isinstance(x, Iterable)
 æČĀşēæşŔāyġlāĒČçŦ'āæŸrāŔēƒ■āzççŽDāĀĆ æČæđIJæŸŕçŽDēŦriijŊ yield
 from āŕşāijŽēƒŦāŽđæL'ĀæIJL'ā■Ŕā;ŊçlŊçŽDāĀijāĀĆæIJĀçžLēƒŦāŽđçzşæđIJārşæŸrāyĀäyġlāşæIJL'āŦŊā

ēlđād'ŪçŽDāŔĆæŦŦ ignore_types āŖŊæčĀæŦŊēr■āŔē isinstance(x,
 ignore_types) çŦlāēārEā■ŪçņēāyşāŖŊā■ŪēLČæŌŖēŽđ' āIJlāŔŕēƒ■āzçārřzēşāād' ŪriijŊēŸşæ■čārEāŌČā
 èƒŽæāūçŽDēŦlā■ŪçņēāyşæŦŦçžDārşēČ;æIJĀçžLēƒŦāŽđæL'ŖāzēāŔĆēĀČēƒŽāyġæŦŦæç

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

èõìèõž

èr■āRē yield from āIJā;āæČšāIJčTšæLŘāZlāy■ērČčTlāĚūāzŮčTšæLŘāZlā;IJāyžā■Řā;NčlNčŽDa
āęČædIJā;āāy■ā;ŁçTlāōČčŽDērIijNéČčāzLāřšāŁĚēāzāĚŽéclād' ŮčŽD for
ā;ŁçŮřāžĚāĀČærTāęĆiijŽ

```
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
```

ār;čōāāRlæTžāžĚāyĀčCžčCžiiNā;ĚæYř yield from
èr■āRēčIJNāyLāŮzāēDšēgŁæŽt' āę;iiijNāžūāyTžāžšā;Łā;ŮāžččāAæŽt' čōĀæt' AæyĚčL;āĀČ

āžNāL'■æRŘāLřčŽDāržāžŮā■ŮčņēāyšāŠNā■ŮēLČčŽDēclād' ŮæčĀæšēæYřāyžāžĚēYšæ■čārĚāōČāžnā
āęČædIJēŁYæIJL'āĚūāzŮā;āāy■æČšāsTāijĀčŽDčšzādNiiijNāŁōæTžāRČæTř
ignore_types ā■šāRřāĀČ

æIJāāRŮēēAæšlæDŘčŽDāyĀčCžæYřiiijN yield from
āIJlæūL'ārĚāLřāšžāžŮā■RčlNāŠNčTšæLŘāZlčŽDāžūāRŠcijŮčlNāy■æL'ōæijTčlĀæŽt' āŁāčG■ēēAčŽDēgŠ
ārřāžēāRČēĀČ12.12ārRēLČæšēčIJNāRēād' ŮāyĀāyĹā;Nā■RāĀČ

6.15 4.15 ēāžāžRēŁ■āžčāRĹāžūāRŮčŽDæŮŠāžRēŁ■āžčāržèšā

éŮóécY

ā;āæIJL'āyĀčšzāLŮæŮŠāžRāžRāLŮiiijNæČšārĚāōČāžnāRĹāžūāRŮā;ŮāLřāyĀāyĹæŮŠāžRāžRāLŮāžūā

èğčāĚşæŮzæāŁ

heapq.merge() āĜ;æTřāRřāžēāyōā;āęğčāĚşēŁZāyĹēŮóécYāĀČærTāęĆiijŽ

```
>>> 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

èõìèõž

heapq.merge āRřèĚāzčçL'zæĀğæĎRāŚşçİĀāōČäy■āijŽčnNél'ñèrżāRŪæL'ĀæIJL'āžRāĹŪāĀĆ
èĚŽārsæĎRāŚşçİĀā;āāRřāzēāIJlédāyēTĚçŽĎāžRāĹŪāy■ā;ĚçTĹāōČrijNèĀNāy■āijŽæIJL'ād'Ĺād'ğçŽĎāijĀē
ærTāēČrijNāyNéİcæYřāyĀāyĹā;Nā■RāĹēāijTčd'žāēČā;TāRĹāžūāyđ'āyĹæŌŠāžRæŪGāžūrijŽ

```
with open('sorted_file_1', 'rt') as file1, \
    open('sorted_file_2', 'rt') as file2, \
    open('merged_file', 'wt') as outf:

    for line in heapq.merge(file1, file2):
        outf.write(line)
```

æIJL'āyĀçČžèēAāijžèrČçŽĎæYřheapq.merge() éIJĀèēAæL'ĀæIJL'è;ŠāĒēāžRāĹŪāĚĒēāzæYřæŌŠ
çL'žāĹnçŽĎrijNāōČāžūāy■āijŽēčĎāĒĹérzāRŪæL'ĀæIJL'æTřæ■ōāĹrāāĒæāĹāy■æL'ŪēĀĒēčĎāĒĹæŌŠāžRrij
āōČāžĒēāzæYřæčĀæšēæL'ĀæIJL'āžRāĹŪçŽĎāijĀāğNéČĹāĹēāžūēĚTāžđæIJĀārRçŽĎēČčāyrijNèĚŽāyĹēč

6.16 4.16 èĚāzčāŽĹāzčæŽwhileæŪāéŽRā;ĹçŌř

éŪōéčY

ā;āāIJlāzčçāAāy■ā;ĚçTĹwhile ā;ĹçŌřæĹēēĚāzčād'ĎçRĚæTřæ■ōrijNāZāyžāōČēIJĀēēAērČçTĹæšRāy
èČ;āy■èČ;çTĹēĚāzčāŽĹāēēčĚāĒēĚāyĹā;ĹçŌřāŚčrijš

èğčāĒşæŪzæāĹ

āyĀāyĹāyēğĀçŽĎIOæŞ■ā;IJçĹNāžRāRřèČ;āijŽæČşāyNéİcèĚæāūrijŽ

```
CHUNKSIZE = 8192

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

èĚŽçğ■āzčçāAēĀŽāyžāRřāzēā;ĚçTĹiter() æĹēāzčæŽrijNāēČāyNæL'Āçđ'žrijŽ

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

(continues on next page)

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

æĈædIJä;æÄÄŮŠăŮĈăĹrăžTēĈ;äy■ēĈ;æ■čäyÿăüēä;IJijNăRăžēērTēlNăyNăyÄäyŁčŏĂ■TĉŽĎä;Nă

```
>>> 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
...
>>>
```

ěŏlěŏž

iter āĜ;æTŕäyÄäyĹésIJäyžăžžĉšĉŽĎĈL'žæĀĝæYŕăŏĈæŌēăRŮäyÄäyĹăRŕéĀL'ĉŽĎ
callable āržēsăŮNăyÄäyĹăăĜēŏŕ(ĉžŠăŕĹ)ăĀijä;IJäyžēĹŠăĒēăRĈæTŕăĀĈ
ă;ŠăžēēfŽĉĝ■æŮžăijRă;ĤĉTĹĉŽĎæŮŭăĀŽijNăŏĈăijŽăĹZăžžäyÄäyĹēf■ăžčăŽĹijN
ēfŽăyĹēf■ăžčăŽĹăijŽăy■æŮ■ērĈĉTĹ callable āržēsăĉŽŕ'ăĹŕēfTăŽďăĀijăŠNăăĜēŏŕăĀijĉŽyĉ■L'äyžæ■čăĂ
ēfŽĉĝ■ĈL'žæŏĹĉŽĎæŮžæšTŕăžăžŌăyÄăžŽĈL'žăŏŽĉŽĎăijŽēĉnéĜ■ăď'■ērĈĉTĹĉŽĎăĜ;æTŕăĹLăIJL'æTŕă
äyĹăĹNăĹēēŏšijNăēĈædIJä;æĈšăžŌăēŮăŌēă■ŮăĹŮăŮĜăžŭäy■ăžēæTŕă■ŏăĹŮĉŽĎæŮžăijRēŕžăRŮăTŕă
read() æĹŮ recv() ijNăžŭăIJăRŌēĹĉĉŕ'ĝēŭšăyÄäyĹæŮĜăžŭĉžŠăŕĹăŕNērTŕăĹăEšăŏŽæYŕăRĉĉžĹæ■čă
iter() ēŕĈĉTĹăŕšăRŕăžēăŕEäyď'ēĀĒĉžŠăRĹĹēŭăĹēăžEăĀĈ äĒŭäy■ lambda
ăĜ;æTŕăRĈæTŕăYŕăyžăžEăĹZăžžäyÄäyĹæŮăăRĈĉŽĎ callable āržēsăijNăžŭäyž recv
æĹŮ read() æŮžæšTŕăRŕăĹZăžE size âRĈæTŕăĀĈ

7 ĉňňăžTĉňăijŽæŮĜăžŭäyŮŌŌ

æĹ'ÄæIJL'ĈĹNăžRĉĈ;ēĉĂăď'ĎĉRĒēĹŠăĒēăŠNēĹŠăĜžăĀĈ
ēfŽăyÄĉňăăŕEăŭĉĉŽŮăď'ĎĉRĒäy■ăŕNĉšăďNĉŽĎæŮĜăžŭijNăNĒæNňæŮĜæIJňăŠNăžNēfŽăĹŭæŮĜăžŭ
ăŕžæŮĜăžŭăŕ■ăŠNĉŽŏă;TĉŽĎæŠ■ă;IJăžšăijŽæŭL'ăŕĹăĹŕăĀĈ

Contents:

7.1 5.1 èrZàEŽæÚGæIJñæTřæ■ó

éÚóéćŸ

ä;äëIJÄëAèrZàEŽàRĎçg■äy■âRŇçijÚčāAçŽĎæÚGæIJñæTřæ■óijNæřTāeCASIiijNUTF-8æLÚUTF-16çijÚčāAç■LāĀĆ

èğčāEşæÚzæaĹ

ä;ŁçTĹäyæIJL' rt æĹaĹijRçŽĎ open() äĜ;æTřèrZàRÚæÚGæIJñæÚGäzūāĀĆæCäyNæL' Āçd' žiijŽ

```
# 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
    ...
```

çšzäijijçŽĎijNäyžāžEāEŽāĒëäyĀäyĹæÚGæIJñæÚGäzūiijNä;ŁçTĹäyæIJL' wt
æĹaĹijRçŽĎ open() äĜ;æTřiijN' æçCædIJāzNāL'■æÚGäzūāEĒāóžā■ŸāIJāĹZæyĒéŽd' āzūëEçŽÚæŌL'āĀĆ

```
# 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)
    ...
```

æçCædIJæŸřāIJāūsā■ŸāIJæÚGäzūāy■æūzāLāāEĒāóžijNä;ŁçTĹæĹaĹijRäyž at çŽĎ
open() äĜ;æTřāĀĆ

æÚGäzūçŽĎèrZàEŽæŞ■ä;IJézŸëóđ'ä;ŁçTĹçşçzşçijÚčāAĹijNāRřāzëéĀŽëŁĜërČçTĹ
sys.getdefaultencoding() æĹēā;ŪāĹřāĀĆ āIJĹād' ġād'ŽæTřæIJžāŽĹäyĹēĹcéČ;æŸřutf-
8çijÚčāAāĀĆæçCædIJā;āāūsçzRçşëéAŞā;äëAèrZàEŽçŽĎæÚGæIJñæŸřāĒūāzŪçijÚčāAæŪzāijRiijN
éĆçāzĹLāRřāzëéĀŽëŁĜāijäéĀŞäyĀäyĹāRřéĀL'çŽĎ encoding
āRĀCæTřçzŽopen()äĜ;æTřāĀĆæçCäyNæL' Āçd' žiijŽ

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



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

æĈædIJăĜžĉŎřēŁŻăyléTŻèrrīijNēĀŽăyÿēalċd'zä;ăerzâRŪæŪĜæIJñæŪŭæŃĜăőŻĉŽĎċijŪĉăAăy■æ■ĉ
 ä;ăæIJĂăē;ăzTĉzEĉYĒēřzēřt'æYŎăzŭĉăőēōd'ă;ăĉŽĎæŪĜăzŭĉijŪĉăAæYřæ■ĉăăĉŽĎ(æřTăēĈă;ċĉTĪUTF-
 8ēĀŃăy■æYřLatin-1ċijŪĉăAæLŪăĒŭăzŪ)ăĀĈ æĈædIJċijŪĉăAēTŻèřrēŁYæYřă■YăIJċŽĎèřrīijŃăjăăRřăzē
 open() äĜ;æTřăijăēĀŠăyĀăylăRřéĀLĉŽĎ errors āRĈæTřălēăd'ĎĉŘĒēŁŻăzŽēTŻèřrăĀĈ
 äyŃēĪĉæYřăyĀăzŽăd'ĎĉŘĒăyÿēĜAēTŻèřrĉŽĎæŪzæşTīijŽ

```
>>> # 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!'
>>>
```

æĈædIJăăĉzŘăyÿă;ċĉTĪ errors āRĈæTřălēăd'ĎĉŘĒċijŪĉăAēTŻèřrīijŃăRřēĈ;ăijŽēōĪ'ă;ăĉŽĎĉTşæt
 āřzăžŎăŪĜæIJñăd'ĎĉŘĒĉŽĎĎēŪēĉAăŎşăĀLŽæYřĉăăăĪă;ăæĀzæYřă;ċĉTĪĉŽĎæYřæ■ĉăăĉċijŪĉăAăĀĈă;Şă
 8)ăĀĈ

7.2 5.2 æL'Şă■rēĹŞăĜžèĜşæŪĜăzŭăy■

éŬőécŸ

ăjăæĈşărE print() äĜ;æTřĉŽĎēĹŞăĜžēĜ■ăăőZăRŞăĀLřăyĀăylăæŪĜăzŭăy■ăŎzăĀĈ

èĝĉăEşşæŪzæqĹ

ăIJĪprint() äĜ;æTřăy■æŃĜăăőŽ file äĒşēTŏă■ŪăRĈæTřīijŃăĈRăyŃēĪĉēŁZăăŭīijŽ

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

èõíèõž

āĖšāžŒēĭŠāĠžēĠāōŽāŘŠāĹŕæŮĠāžūāyāŕŕſēŁŻāžZāžĖāĀĆāĭĖæŸŕæĪĴĹāyĀċĆžēēĀæſĹæĎŔċŽĎŕſæāĖĆæĎĪæŮĠāžūāŸŕāžŅēŁŽāĹŭāĹāĭĭŔċŽĎĕŕĪĭjŅæĹŠāāŕāŕſāĭjŽāĠžēŤŽāĀĆ

7.3 5.3 āĭĖċŤĹāĖŮāžŮāĹĖĖŽŤċņæĹŮĖāŅċžĹæāċċņæĹŠāāŕ

éŮŏéćŸ

āĭāæĈšāĭĖċŤĹĭprint() āĠĭæŤŕēĭŠāĠžæŤŕæāŕĭĭjŅāĭĖæŸŕæĈšæŤžāŔŸēžŸēŏĎċŽĎāĹĖĖŽŤċņæĹŮĖ.

èġċāĖšæŮžæāĹ

āŔŕāžēāĭĖċŤĹāĪĴĭprint() āĠĭæŤŕāyāāĭĖċŤĹĭsepāšŅend
āĖſēŤŏāŮāŔĈæŤŕĭĭjŅāžēāĭāæĈšēēĀċŽĎæŮžāĭjŔēĭŠāĠžāĀĆæŕŤāĖĈĭĭjŽ

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

āĭĖċŤĹĭend āŔĈæŤŕāžšāŔŕāžēāĪĴēĭŠāĠžāyāċēĀæāāċēāŅāĀĆæŕŤāĖĈĭĭjŽ

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

èõíèõž

āĭŠāĭāæĈšāĭĖċŤĹēĭĎċĹ'žæāĭjāĹĖĖŽŤċņæĹēēĭŠāĠžæŤŕæāŕĭĭjŽĎæŮŭāĀŽĭĭjŅċžŽ
print() āĠĭæŤŕāĭjāēĀšāyĀāyĹĭsepāŔĈæŤŕæŸŕæĪĴŏĀāĭĤċŽĎæŮžæāĹāĀĆ
æĪĴĹæŮŭāĀŽāĭāĭjŽċĪĴŅāĹŕāyĀāžŽċĹŅāžŔāŖŸāĭjŽāĭĖċŤĹĭstr.join()
æĹēāŏŅæĹŔāŕŅæāŭċžĎāžŅæĈĖāĀĆæŕŤāĖĈĭĭjŽ

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

`str.join()` çŽĎéŮőécŸâIJläžŎăőČăžĚăžĚéĂĆčŤlăžŎă■ŮčņęäÿšăĂĆèŁŻăĎŔăŚşçİĂă;ăéĂŽăÿŷéI

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

ă;ăă;ŞçĎŮăŔŕăžěäÿ■čŤléĆčăžĹéžžçČęiijŇăŔlėIJĂèęAăČŔăÿŇéİćèŁŻăăŮăĚŽiijŽ

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

7.4 5.4 èŕžăĚŽă■ŮèŁĆăŤŕăëŎ

éŮőécŸ

ă;ăăČşèŕžăĚŽăžŇèŁŻăĹŮăŮŮĜăžŮiijŇăŕŤăęČăŽçŁĜiijŇăčŕéşşăŮŮĜăžŮç■Ł■ăĂĆ

èĝčĂĚşăŮžăăĹ

ă;ŁçŤlăİăiijŔăÿž rb æŁŮ wb çŽĎ open() âĜ;ăŤŕăİèèŕžăŔŮăŁŮăĚŽăĚăžŇèŁŻăĹŮăŤŕă■ăĂĆăŕŤ

```
# 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')
```

ăIJlėŕžăŔŮăžŇèŁŻăĹŮăŤŕă■ăŮŮiijŇéIJĂèęAăŇĜăŸŎçŽĎăŸŕăĹĂăIJĹ'èŁŤăŽđçŽĎăŤŕă■ăŎéČ;ăŮ
çşžăiijçŽĎiijŇăIJăĚŽăĚéçŽĎăŮăăĂŽiijŇăŁĚéăžăŁİėŕAăŔĆăŤŕăŸŕăžěă■ŮèŁĆă;ăăiijŔăŕžăđ'ŮăŽŤ'éIJşăŤ

èőİèőž

ăIJlėŕžăŔŮăžŇèŁŻăĹŮăŤŕă■ăçŽĎăŮăăĂŽiijŇă■ŮèŁĆă■ŮčņęäÿšăŇăŮŮĜăIJŇă■ŮčņęäÿşçŽĎėŕ■ăžŁ
çŁ'žăĹnéIJĂèęAăşăĹăĎŔçŽĎăŸŕiijŇçŤăăiijŤăŇŇèŁ■ăžčăĹă;IJèŁŤăŽđçŽĎăŸŕă■ŮèŁĆçŽĎăĂiijèĂŇăÿ■ăŸ

```

>>> # 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
...
>>>

```

æCædIJä;äæČšžÖäžÑëfZålŮæłajRçŽĐæŮĞäzŭäy■ëržâRŮæLŮâEžâĚëæŮĞæIJñæŤræ■ōijŃâfĚéa

```

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'))

```

äžÑëfZålŮI/OëfŸæIJL'äyÄäylésIJäyžäžžçšëçŽĐçL'žæĂğârşæŸræŤrçzĐâŠŃCçzŞæđĐä;ŞçşzâđÑëČ;ç

```

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

```

èfZäyléĂĆçŤlăžŌăzză;ŤăôđçŎřăžEëcñçğrăžNăyžâĂlçijŞâEşæŎëâRcâĂlçŽĐâržèsajijÑëfŽçg■âržèsajij
äžÑëfZålŮæŤræ■ôçŽĐâEžâĚëârşæŸrëfŽçşzæŞ■ă;IJăžNăyĂăĂĆ

ăĴLăđ'ŽâržèsăèfŸăĚĂèçŏyéĂžëfĞă;ŤçŤlăŮĞäzŭâržèsăçŽĐ readinto()
æŮzæşŤçŽŤ'æŎëëržâRŮäžÑëfZålŮæŤræ■ôâlŤrăĚŭăžŤăsĆçŽĐâEžâ■Ÿäy■ăŎžăĂĆærŤăçĆijŽ

```
>>> 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;æÿrä;ççTíèçZçg■æLÄæIJçZDæUúäÄZéIJÄèçAæäijäd'ÚärRáfÇrijNäZäyÿäöÇéÄZäyÿäËüæIJL'áz
ärRäzææšççIJN5.9ärRèLCäy■äRèad'ÚäyÄäylerzaRÜäzNèçZäLüæTæ■öäLräRäçöæTzçijSäEšäNzçZDä;Nä

7.5 5.5 æÚGäzúäy■ä■YäIJæL■èÇjâEzäËë

éUóécY

ä;äæÇsäÇRäyÄäyLæÚGäzúäy■äEzäËëæTæ■öijNä;EæYfäL■æRäRäçEéazæYrèçZäyLæÚGäzúäIJæÚG
ázšärsæYräy■äEÄèöyèçEçZÜäüšä■YäIJçZDæÚGäzúäEËäözäÄÇ

èççäEçsæÚzæäL

ärRäzèäIJí open() äGjæTäy■ä;ççTí x ælääijRæIëäzçæZç w
ælääijRçZDæÚzæçTæIëèççäEçsèçZäyLéUóécYäÄÇærTæÇrijZ

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

äçÇädIJæÚGäzúæYräzNèçZäLüçZDrijNä;ççTí xb æIëäzçæZç xt

èöIèöž

èçZäyÄärRèLCæijTçd'žäZÈäIJäEzæÚGäzúæUüéÄZäyÿäijZéAĞäLrçZDäyÄäyLéUóécYçZDäöNç;Öèg
äyÄäyLæZçäzçæÚzæäLæYräELætNërTèçZäyLæÚGäzúæYräRèä■YäIJliijNäČRäyNéIcéçZæäüijZ

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

(continues on next page)

(continued from previous page)

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

`open()` `io.StringIO()` `io.BytesIO()` `io.StringIO('Hello\nWorld\n')` `io.BytesIO(b'binary data')`

7.6 5.6 `io.StringIO()` `io.BytesIO()`

`io.StringIO()`

`io.StringIO('Hello\nWorld\n')` `io.BytesIO(b'binary data')`

`io.BytesIO()`

`io.StringIO('Hello\nWorld\n')` `io.BytesIO(b'binary data')`

```
>>> 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('Hello\nWorld\n')` `io.BytesIO(b'binary data')`

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

èõléõž

ā;Šā;āæČšæĭæNšāyĀāyĭæŽōéĀŽčŽDæŮĠāzūčŽDæŮūāĀŽ StringIO āŠN
BytesIO çšzæYřā;ĹæIJL'çTĭçŽDāĀĆ ærTæÇĭijNāIJĭāTāĒČætNērTāyĭijNā;āāRřāzēā;ĤçTĭ
StringIO æĹēāĹŽāžžāyĀāyĭāNĒāRnætNērTæTřæĭōçŽDçšzæŮĠāzūāřžēšāijN
èĤŽāyĭāřžēšāāRřāzēèèñāijāçžZæšRāyĭāRĆæTřāyžæŽōéĀŽæŮĠāzūāřžēšāçŽDāĠ;æTřāĀĆ

éIJĀèèAæšĭæDRçŽDæYřĭijN StringIO āŠN BytesIO
āōdā;NāžūæšāæIJL'æĭççāōçŽDæTř æTřçšzādNçŽDæŮĠāzūæRŘèĤřçñēāĀĆ
āŽāæĭ'ĭijNāōČāznāyĭēČ;āIJĭéČčāžŽéIJĀèèAā;ĤçTĭIJšāōdčŽDçšzçžšçžgæŮĠāzūāēČæŮĠāzūĭijNçōāéAš

7.7 5.7 èrzāĒZāŌNçijl'æŮĠāzū

éŮōécY

ā;āæČšèřzāĒZāyĀāyĭgziæĹŮbz2æāijāijRçŽDāŌNçijl'æŮĠāzūāĀĆ

èğčāĒšæŮzæāĹ

gzip āŠN bz2 æĭāāĹŮāRřāzēā;ĹāōžæYšçŽDād'ĎçRĒèĤZāžZæŮĠāzūāĀĆ
āyđ'āyĭæĭāāĹŮēČ;āyž open() āĠ;æTřæRŘā;ZāžĒāRēād'ŮçŽDāōdčŌřæĹēèğčāĒšèĤZāyĭéŮōécYāĀĆ
æŕTæÇĭijNāyžāžĒāzēæŮĠæIJnā;čāijRēržāRŮāŌNçijl'æŮĠāzūĭijNāRřāzēèĤZæāūāĀŽĭijŽ

```
# 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()
```

çšzāijijçŽDĭijNāyžāžĒāēZāĒēāŌNçijl'æTřæĭōĭijNāRřāzēèĤZæāūāĀŽĭijŽ

```
# 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)
```

āēČāyĹĭijNæĹ'ĀæIJL'çŽDI/Oæšĭā;IJéČ;ā;ĤçTĭæŮĠæIJnāĭāijRāžūæĹgēāNUnicodeçŽDçijŮčāA/èğčç
çšzāijijçŽDĭijNāēČādIJā;āæČšæšĭā;IJāžNēĤZāĹūāTřæĭōĭijNā;ĤçTĭ rb æĹŮēĀĒ wb
æŮĠāzūāĭāijRā;šāRřāĀĆ

èõléõž

ad'gëČlálEæČĚāEṭāyNërzaEŽāŎŇcijl' æTṛæ■óéČ;æYřā;ŁçóĀā■TčŽDāĀCā;EæYřèeAæslæDŘčŽDæY
æeČædIJā;āy■æŇĠāōŽælaāijRiijŇéČčāzLézYëod'čŽDārśæYřāžŇēŁāLūælaāijRiijŇāeČædIJèŁZæŮūāĀZč
gzip.open() āšŇ bz2.open() æŎěāRŮèūšāEĚč;ōčŽD open()
āĠ;æTṛāyĀæāūčŽDāRČæTṛiijŇ āŇĚæŇŇ encodingiijŇerrorsiijŇnewline
č■L'č■L'āĀČ

ā;ŠāEŽāĚĚāŎŇcijl' æTṛæ■óæŮūiijŇāRřāžēā;ŁčTl compresslevel
èŁZāyĹāRřéĀLčŽDāĚšéTōā■ŮāRČæTṛælēæŇĠāōŽāyĀāyĹāŎŇcijl'čžġāLŇāĀČærTāeČriijŽ

```
with gzip.open('somefile.gz', 'wt', compresslevel=5) as f:  
    f.write(text)
```

ézYëod'čŽDč■L'čžġæYř9iijŇāzšæYřæIJĀénYčŽDāŎŇcijl'č■L'čžġāĀČč■L'čžġēūLā;ŎæĀġēČ;ēūLāē;iiij
æIJĀāRŎāyĀčČzriijŇ gzip.open() āšŇ bz2.open()
èŁYæIJL'āyĀāyĹā;LārŠècŋčšēēAščŽDčL'záĠiijŇ āōČāznāRřāžēā;IJčTlāIJlāyĀāyĹāūšā■YāIJlāzūāzēāžŇēŁ

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

èŁZæāūārśāĒAèōy gzip āšŇ bz2 ælaāiŮāRřāžēāūēā;IJāIJlèōyād'ŽčšzæŮĠāzūāržèšāyLriijŇærTāeČāē

7.8 5.8 āŽžāōŽad'ġārRèōrā;TčŽDæŮĠāzūèŁ■āžč

éŮóécY

ā;āæČšāIJlāyĀāyĹāŽžāōŽéTŁāžèèōrā;TæLŮèĀĚæTṛæ■óāiŮčŽDèZEāRĹāyLèŁ■āžčriijŇēĀŇāy■æYřāIJ

èġčāEšæŮzæāL

éĀŽèŁĠāyŇéiŁèŁZāyĹārRæLĀāūġā;ŁčTl iter āšŇ functools.partial()
āĠ;æTṛiijŽ

```
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:  
        ...
```

èŁZāyĹā;Nā■Rāy■čŽD records āřžèšæYřāyĀāyĹāRřèŁ■āžčāržèšāiijŇāōČāijZāy■æŮ■čŽDāžġčTšāŽž
èeAæslæDŘčŽDæYřæČædIJæĀžèōrā;Tad'ġārRāy■æYřāiŮad'ġārRčŽDæTt'æTṛāĀ■čŽDèriijŇæIJĀāRŎāy

èõléõž

`iter()` āĠ;æŦŕæIJL'äyÄäyléšIJäyžāžžçšëçŽDçL'žæĀġārsæŸŕiijŇæÇæđIJā;āçžŽāōČaijāēĀŠäyÄäylāēēfŽäyŸēf■āzčāŽlāijŽäyĀçŽt'ērČçŦlāijāāĒëçŽDāŦŕērČçŦlāržžèšaçŽt'ālŦāōČēfŦāŽđæāĠēōŕāĀijäyžæ■čiiŇēf

āIJlā;Ňā■Ŧäy■iiŇ `functools.partial` çŦlāēāLŽāžžäyÄäylæŦŕæñæçñērČçŦlāŮüāžŌæŮĠäzūāæāĠēōŕāĀij b' ' āŕsæŸŦā;ŠāŦŕè;çæŮĠäzūçžŠāŦçæŮüçŽDēfŦāŽđāĀijāĀĆ

æIJĀāŦŦŌāE■æŦŦäyĀçČziiŇNäyLēlççŽDä;Ňā■Ŧäy■çŽDæŮĠäzūāŮüāžēāžŇēfŽāLūāēlāijŦæL'ŠāijĀçāēÇæđIJæŸŦēržāŦŮāŽžāōŽād'ġārŦçŽDēōŕā;ŦiiŇŇēfŽéĀŽäyŸæŸŦæIJæŽōēA■çŽDæČĒāEŦāĀĆēĀŦāŦžāžŌæŮĠæIJñæŮĠäzūiiŇNäyĀēāNäyĀēāŇçŽDēržāŦŮ(ēžŸēōd'çŽDēf■āzčēāNäyž)æŽt'æŽōēA■çŽā

7.9 5.9 èŕžāŦŮāžŇēfŽāLūæŦŕæ■ōāŦŕāŦŕāŦŦçijŠāEšāŇžäy■

éŮōécŸ

ä;āæČšçŽt'æŌēēržāŦŮāžŇēfŽāLūæŦŕæ■ōāŦŕäyÄäylāŦŕāŦŦçijŠāEšāŇžäy■iiŇēĀŇäy■ēIJĀēēAāAžžāæŦŮēĀĒā;āæČšāŌšāIJŦāŦōæŦžæŦŕæ■ōāžūārEāōČāEŽāŽđāŦŦäyÄäylæŮĠäzūāy■āŌžāĀĆ

èġčāEšæŮžæāŦ

äyžāžEēržāŦŮæŦŕæ■ōāŦŕäyÄäylāŦŕāŦŦæŦŦçžDäy■iiŇNä;ççŦlāŮĠäzūāržžèšaçŽD
`readinto()` æŮžæšŦāĀĆæŦāēČiiŇŽ

```
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
```

äyŇēlçæŸŦäyÄäylæijŦçd'žēfŽäyŸāĠ;æŦŦä;ççŦlāŮžæšŦçŽDä;Ňā■ŦiiŇŽ

```
>>> # 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
>>>
```

èóìèőž

æŮĜäzŭärzèšaçŽĎ readinto() æŮzæsŦëČ;ècñçŦlæìëäyžécĎāĒĹāĹēĒ■āĒĒā■ŸçŽĎæŦřçzĎāāñāĒ
 array æĹāāĹŮæĹŮ numpy āžšāĹZāžžçŽĎæŦřçzĎāĀĆ āšŦæŽóéĀŽ read()
 æŮzæsŦäy■āŦŦçŽĎæŸřijŦ readinto() āāñāĒĒāŭšā■ŸāĹĹçŽĎçijšāĒšāŦžèĀŦäy■æŸřäyžæŮřärzèšaçĒ
 āŽāæ■d'rijŦä;āāŦřäzèä;ĲçŦĹāōČæĹēĒĀĲāĒ■ād'gēĠŦçŽĎāĒĒā■ŸāĹēĒēĒ■æš■ā;ĹāĀĆ
 æŦŦāēČrijŦāēČæĎĹā;æēřzāŦŮäyĀäyĲçŦšçŽyāŦŦāĎ'gārŦçŽĎēōřā;ŦçzĎāĹŦçŽĎžŦēĲZāĹŮæŮĠäzŭæŮŭii

```
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
    ...
```

āŦēād'ŮæĹĹ'äyĀäyĲæĹĹ'ēūčĲL'zæĀgāršæŸř memoryview iiŦŦ
 āōČāŦřäzèēĀžēĲĠēŽŭād'■āĹŮçŽĎæŮžāijŦārřzāŭšā■ŸāĹĹçŽĎçijšāĒšāŦžæĹ'gēāŦāĹĠçĲĠĠæš■ā;ĹāijŦçŦžē

```
>>> buf
bytearray(b'Hello World')
>>> m1 = memoryview(buf)
>>> m2 = m1[-5:]
>>> m2
<memory at 0x100681390>
>>> m2[:] = b'WORLD'
>>> buf
bytearray(b'Hello WORLD')
>>>
```

ä;ĲçŦĹf.readinto() æŮŮēĹĀēēĀæšĹāēĎŦçŽĎæŸřijŦä;āāĲĒēāzæčĀæšēāōČçŽĎēĲŦāžĎāĀijrijŦā
 āēČæĎĹā■ŮēĲČæŦřärŦäžŌçijšāĒšāŦžād'gārŦrijŦēāĹæŸŌæŦřæ■ōēcñæĹĲæŮ■æĹŮēĀĒēcñçāt'āĹŦäžēĲ
 æĹĀāŦŦŦrijŦçŦžāĲČēgČāršāĒēŭāzŮāĠ;æŦřäžšāšŦæĹāāĹŮäy■āšŦ into
 çŽyāĒšçŽĎāĠ;æŦř(æŦŦāēČ recv_into() iiŦŦ pack_into() ç■Ĳ)āĀĆ
 PythonçŽĎā;Ĺād'ŽāĒŭāzŮēČĹāĹēāŭšçzŦēČ;æŦřæŦĀçŽŦ'æŌēçŽĎĹŌæĹŮæŦřæ■ōēōĲēŮōæš■ā;ĹāijŦēĲZā
 āĒšāžŌēgčæĎŦäžŦēĲZāĹŮçzšæĎĎāšŦ memoryviews
 ä;ĲçŦĹæŮzæsŦçŽĎæŽŦ'énŸçžgä;Ŧā■ŦrijŦēŦŕāŦČēĀĆ6.12ārŦēĲČāĀĆ

7.10 5.10 ħĖĖā■ŸæŸāārĎçŽĎžŇèŁŽāĹūæŰĠžū

éŰóécŸ

ä;ăæČšāĖĖā■ŸæŸāārĎäŸÄäŸlāžŇèŁŽāĹūæŰĠžžūāĹŕäŸÄäŸlāŔŕāŔŸā■ŰēĹĆæŦŕçžĎäŸ■iijŇçŽóçŽĎžā

èğčāĖşæŰžæąĹ

ä;ĤçŦĪ mmap æĹāāĪŰæĹēāĖĖā■ŸæŸāārĎæŰĠžžūāĀĆ
äŸŇēĹēæŸŕäŸÄäŸlāūēāĖŰāĠæŦŕiijŇāŔŖŖā;ăæijŦçĎ'žžæĖāçĆā;ŦæĹŖāijÄäŸÄäŸlāæŰĠžžūāžžūāžæäŸÄçğ■ä;Ĥæ

```
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)
```

äŸžžæĖä;ĤçŦĪēĤŽäŸlāĠæŦŕiijŇä;ăēĪĀēçĀæĪĹäŸÄäŸlāūŖāĹŽāžžāžžūāŸŦāĖĖāōžäŸ■äŸžçĹ'žçŽĎæŰĠžžūā
äŸŇēĹēæŸŕäŸÄäŸlā;Ňā■ŔiijŇæŦŦžä;ăæĀŌæāūāĹĪāğŇāĹŽāžžäŸÄäŸlāæŰĠžžūāžžūārĖāĖŰāĖĖāōžæĹŦ'āĖĖāĹŦ

```
>>> size = 1000000
>>> with open('data', 'wb') as f:
...     f.seek(size-1)
...     f.write(b'\x00')
...
>>>
```

äŸŇēĹēæŸŕäŸÄäŸlāĹŦçŦĪ memory_map() āĠæŦŦçšžāĖĖā■ŸæŸāārĎæŰĠžžūāĖĖāōžçŽĎžā;Ňā■Ŕiijž

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

```
>>> with memory_map('data') as m:
...     print(len(m))
...     print(m[0:10])
...
1000000
b'Hello World'
>>> m.closed
True
>>>
```

```
m = memory_map(filename, mmap.ACCESS_READ)
```

```
m = memory_map(filename, mmap.ACCESS_COPY)
```

äyžäžEéŽRæIJžèðféUöæŨĠäzúçŽDāEĖĀðžījŅā;£çTĪ mmap
 ārEæŨĠäzūæYāārDāLrāEĖĀ■Yāy■æYřāyĀäylēnYæTŁāŠŅāijYéZĚçŽDæŨzæşTāĀĆ
 äĴNāçCřījŅā;āæUāēIJāæLŠāijĀäyĀäylæŨĠäzūāzūāLgēaŅād'gēGRçŽD seek() iijŅ
 read() iijŅ write() èrCçTĪiijŅārLéIJāēæAçõĀā■TçŽDæYāārDæŨĠäzūāzūā;£çTĪāŁĠçŁĴGæŞ■ä;IJèðfé
 äyĀēLŅāēlèèðšījŅ mmap() æLĀæŽt'ēIJçŽDāEĖĀ■YçIJŅāyŁāŌzārśæYřāyĀäylāzŅē£ZāLūæTřçzDār
 ä;EæYřījŅā;āāRřāzēä;£çTĪāyĀäylāEĖĀ■YēgEāŽ;æIēēgçædRāĖūäy■çŽDæTřæ■ōāĀĆæřTæCřījŽ

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

157

æĊædIJăd' ŽăyI PythonèġċéĠŁăŽlăEĖă■YæYăârDăRŃăyĂăyŁæŰĠăzŭiijNăĭŰăĹŕċŽĎ
 mmap ârzésăċĭăd' šċċŋċŦlăIċăIJlċġċċéĠŁăŽlċŽt' æŰċăzd' æ■ċæŦŕæ■ăăĂĊ
 äžšâršæYŕĕŦ' iijNăL' ĂæIJL'èġċéĠŁăŽlċĊĭċĊ;ăRŃăŰŰĕŕzăEŽæŦŕæ■ŏiijNăžŭăyŦăĖŭăy■ăyĂăyŁèġċéĠŁăŽlă
 âĭŁăYŰăYĭiijNĕĤŽéĠŃĖIJĂĖĖAĖĂĊĊēZŠăRŃă■ĖċŽĎĖŰŏċYăĂĊăĭEăYŕĕĤŽċġ■æŰžăŝŦæIJL'æŰŭăĂŽăR
 èĤŽăyĂăŕRĖĤĂy■ăĠ;æŦŕăŕĭĖĠŕăEŽăĭŰăĭŁăĂŽċŦlĭijNăRŃăŰŰĖĂĊċŦlăžŰUnixăŠŃWindowsăžšăR
 ĖĖAăŝlăĎŦŕċŽĎăYŕăĭĤċŦlĭ mmap () âĠ;æŦŕæŰŰăijŽăIJlăžŦŝĊæIJL'ăyĂăžŽăžšăRŕċŽĎăŭŏăijĊăĂġăĂĊ
 âŦĖăd' ŰiijNĖĤYăIJL'ăyĂăžŽĖĂL'ĖăžăŦŕăžĖċŦlăIĖăĹŽăžžăŢăŦŕ■ċŽĎăEĖă■YæYăârDăNžăššăĂĊ
 æĊædIJăĭăâržĖĤŽăyŁăĎšăĖŦ'ĖŭċiijNċăŏăĤlăĭăăžŦċzEċăŦĕŕzăžE PythonæŰĠăĖăċăy■
 èĤŽæŰžĖĤċċŽĎăEĖăŏž āĂĊ

7.11 5.11 æŰĠăžŭĖŭŕăĭĎăŦ■ċŽĎăŝ■ăĭIJ

ĖŰŏċY

ăĭăĖIJĂĖĖAăĭĤċŦlĖŕăĭĎăŦ■æIĖĖŰăŦŰæŰĠăžŭăŦŕ■iijNċŽŏăĭŦăŦŕ■iijNċžĭâržĖŭŕăĭĎċ■Ĥċ■ĤăĂĊ

èġċăEŝăŰžăĖăĹ

ăĭĤċŦl os.path æĭăăĤŰăy■ċŽĎăĠ;æŦŕæIĖăŝ■ăĭIĖŭŕăĭĎăŦ■ăĂĊ
 äyŢĖĤăYŕăyĂăyŁăzd' äžŠăijRăĭNă■ŦăIĖăijŦċd' žăyĂăžŽăĖŝĖŦŏċŽĎċĤăĂġiijŽ

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

èõléõž

```
árzäžÖäzzä;TçŽDæŮĠäzúäR■çŽDæŠ■ä;IJiijNä;äéČ;äžTèréä;ŁçTl os.path
ælaaiŮiijNèĀNäy■æYrä;ŁçTlæăĠăGEă■ŮçņäyšæŠ■ä;IJæIēædDēĀăĠăŮšçŽDäzčçăĀăĀĆ
çL'žāLŋæYräyžāžĒāRrcgžæd'■æĀġèĀĆèŽŠçŽDæŮŮăĀŽæŽt'ăžTăĕCæ■d'iijN äZäyž os.
path ælaaiŮçšĕĒAšUnixăŠNWindowsçšçžçžšăžNéŮt çŽDăŮŮăijCăžŮăyTēČ;ăd' šăRfēlăăIJrăd'DçRĒçšzäiij
Data/data.csv äŠN Data\data.csv èŁZæăŮçŽDæŮĠäzúäR■ăĀĆ
ăĒŮăŋăiijNă;ăçIJšçŽDäy■ăžTèréæŁt'zæŮŮéŮt'ăŮžéĠăd'■éĀăē;Ůă■RăĀĆéĀŽăyŷæIJĀăē;æYrcŽt'æŮēă;ŁçŽyăĒççŽ
èĒAæšlæDRçŽDæYf os.path èŁYæIJL'æŽt'ăd'ŽçŽDăLšēČ;ăIJlèŁZéĠNăžŮăšăæIJL'ăLŮăy;ăĠZæIēă
ăRfäzææšĕĒYĒăŮYæŮzæŮĠæăçæIēēŮăRŮŮæŽt'ăd'ŽăyŮăŮĠäzúăŮtNērTrijNçņăRŮēŠ;æŮēç■L'çŽyăĒççŽ
```

7.12 5.12 æŁNērTæŮĠäzúæYräRĕă■YăIJl

éŮŮéćY

ă;ăæČšætNērTäyĀăylæŮĠäzúæLŮçŽŮă;TæYräRĕă■YăIJlăĀĆ

èġcăĒşæŮzæăŁ

ă;ŁçTl os.path ælaaiŮæIēætNērTäyĀăylæŮĠäzúæLŮçŽŮă;TæYräRĕă■YăIJlăĀĆæŁTăĕCiiijŽ

```
>>> import os
>>> os.path.exists('/etc/passwd')
True
>>> os.path.exists('/tmp/spam')
False
>>>
```

ă;ăēŁYēČ;ēŁZăyĀă■æŁNērTēŁZăylæŮĠäzúæŮŮăžĀăžŁçšžădNçŽDăĀĆ
ăIJlăyNēlĕēŁZăžZætNērTäy■iijNăēĆădIJætNērTçŽDæŮĠäzúăy■ă■YăIJlçŽDæŮŮăĀŽiijNçžšădIJéČ;ăiijŽè

```
>>> # 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'
>>>
```

æċædIJä;æċŸæĈšèŌüâRŪâĔĈæTŗæ■ō(æŕTæĈæŪĠäzûâd'ġârRæĹŪèĀĔæŸŕäĤōæTŹæŪèæIJ§)iijNäz
os.path æġäġŪæĬèġċâĔšijŽ

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

èőĬèőž

ä;ĤċTĬ os.path æĬèċŹæġNæŪĠäzûæġNèŕTæŸŕä;ĹċōĀâ■TċŽĎăĀĈ
âIJġâĔŹæĤZäZžĎŽæIJnæŪüiijNâŔŕèĈ;âTŗäŸĀĔIJæċæĤŕċŽĎăŕsæŸŕä;æĬIJæċæĤĀĈèŽŖæŪĠäzûæĬĈ

```
>>> 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 èŌüâRŪæŪĠäzûâd'žäŸ■ĈŽĎæŪĠäzûâĹŪèġġ

éŪōéċŸ

ä;ăæĈšèŌüâRŪæŪĠäzûċšžċžšäŸ■æšŔäŸĹċŽōâ;TäŸNċŽĎæĹĀæIJĹæŪĠäzûâĹŪèġġăĀĈ

èġċâĔšæŪzæġĹ

ä;ĤċTĬ os.listdir() âĠ;æTŗæĬèċŌüâRŪæšŔäŸĹċŽōâ;TäŸ■ĈŽĎæŪĠäzûâĹŪèġġiijŽ

```
import os
names = os.listdir('somedir')
```

ċžŖædIJäijŽèĤTâZċŽōâ;TäŸ■æĹĀæIJĹæŪĠäzûâĹŪèġġiijNâNĔæNnæĹĀæIJĹæŪĠäzûiijNâ■ŔċŽōâ;
æċædIJä;æĬIJæċæĤĀĈèŽŖæšŔċġ■æŪzâijŔèĤĠæzd'æTŗæ■ōiijNâŔŕäzèèĀĈèŽŖŖæŖĹ
os.path äžŖäŸ■ĈŽĎäŸĀžZâĠ;æTŗæĬèä;ĤċTĬâĹŪèġġæŌġŕijăĀĈæŕTæĈiijŽ


```
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))]
```

startswith() ends with()

```
pyfiles = [name for name in os.listdir('somedir')
            if name.endswith('.py')]
```

glob fnmatch

```
import glob
pyfiles = glob.glob('somedir/*.py')

from fnmatch import fnmatch
pyfiles = [name for name in os.listdir('somedir')
            if fnmatch(name, '*.py')]
```

èóíèőž

os.path

```
# Example of getting a directory listing

import os
import os.path
import glob

pyfiles = glob.glob('*.py')

# Get file sizes and modification dates
name_sz_date = [(name, os.path.getsize(name), os.path.
    getmtime(name))
                 for name in pyfiles]
for name, size, mtime in name_sz_date:
    print(name, size, mtime)
```

(continues on next page)

(continued from previous page)

```
# Alternative: Get file metadata
file_metadata = [(name, os.stat(name)) for name in pyfiles]
for name, meta in file_metadata:
    print(name, meta.st_size, meta.st_mtime)
```

æIJĀāŔŌèĚŸæIJL'äyĀčĆzèĚAæşlæĐŔčŽĐāŕsæŸŕijŊæIJL'æŮŭāĀŽāIJlād'ĐčŔĚæŮŮGāzŭāŔ■çijŮčāAé
éĀŽāŷyæĪèèōšijŊāŮĜĭæŦŕ os.listdir() èĚŦāŽđčŽĐāōđäĭŞāLŮèāĪijŽæāžæ■ōçşzçzşēzŸèōd'čŽĐæŮŮGā
äĭĚæŸŕæIJL'æŮŭāĀŽāzşäijŽčŕāĹŕāyĀāžZāy■èČĭæ■čāyŷèğčçāAçŽĐæŮŮGāzŭāŔ■āĀĆ
āĚşāžŌæŮŮGāzŭāŔ■çŽĐād'ĐčŔĚéŮōécŸŕijŊāIJĪ5.14āŖŊ5.15ārŔèĹĆæIJL'æŽŦ'èŕççzĚçŽĐèōšèğčāĀĆ

7.14 5.14 āŖĭçŦĚæŮŮGāzŭāŔ■çijŮčāA

éŮōécŸ

äĭāæČşäĭĚçŦĪāŌşāğŊæŮŮGāzŭāŔ■æL'ğèāŊæŮŮGāzŭçŽĐĪ/OæŞ■äĭIJŕijŊāzşāŕsæŸŕèŦ'æŮŮGāzŭāŔ■āzŭāŔ■

èğčāĚşæŮŮzæāĹ

ézŸèōd'æČĒāĚŷäyŊŕijŊæL'ĀæIJL'čŽĐæŮŮGāzŭāŔ■éČĭäijŽæāžæ■ō sys.
getfilesystemencoding() èĚŦāŽđčŽĐæŮŮGāĪŊçijŮčāAæĪèçijŮčāAæLŮèğčçāAāĀĆæŦĪæČŕijŽ

```
>>> sys.getfilesystemencoding()
'utf-8'
>>>
```

āēČæđĪJāZāyŷæşŔçğ■āŌşāZāāĭāæČşāĹĭçŦĚèĚŽçğ■çijŮčāAŕijŊāŖŕāzèäĭĚçŦĪāyĀāyŦāŌşāğŊā■ŮèĹĆā

```
>>> # Write a file using a unicode filename
>>> with open('jalape\xflo.txt', 'w') as f:
...     f.write('Spicy!')
...
6
>>> # Directory listing (decoded)
>>> import os
>>> os.listdir('.')
['jalapeĀŕso.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■çäĈä;äæL'ÄëĠAīījNāIJāIJāāRŌäyð'äyġæŞ■ä;IJäy■īījNā;Şä;ăçzZæŪGäzūçZyāĖŞāĠ;æTŗæĈ
open() āŠN os.listdir() āijäēĀŠā■ŪēŁĈā■ŪçņēäyşæŪūīījNæŪGäzūāR■çZDād'DçŖĒæŪzāijRāijZçġ
```

èõlèõž

```
éĀŽāyŷæĪēðõīījNā;äāy■éIJāēçAæNĖāŁĈæŪGäzūāR■çZDçijŪçāAāŠNēġççāAīījNæŽðéĀŽçZDæŪGäz
ā;ĒæŸīījNæIJL'āzZæŞ■ā;IJçşçzçşāĖAēðyçTġæŁūēĀŽēĠGāŪçDūāŁŪæAūāĎRæŪzāijRāŌzāŁZāzžāR■ā■
ēġZāzZæŪGäzūāR■āRfēĈ;āijZçēðçġŸāIJrāy■æŪ■ēĈcāzZēIJāēçAāđ'DçŖĒāđ'ġēĠRæŪGäzūçZDPythonçġN
```

```
ērzāRŪçZōā;TāzūēĀŽēĠGāŌşāġNæIJġēġççāAæŪzāijRāđ'DçŖĒæŪGäzūāR■āRfāzēæIJL'æTġçZDēAġā
ār;çōāēġZæāūāijZāyçæĪēäyĀāõŽçZDçijŪçġNēŽ;āzçāĀĈ
```

```
āĖşāzŌæL'Şā■rāy■āRfēġççāAçZDæŪGäzūāR■īījNērūāRĈēĀĈ5.15ārRēŁĈāĀĈ
```

7.15 5.15 æL'Şā■rāy■āRġæşTçZDæŪGäzūāR■

éŬóécŸ

```
ā;ăçZDçġNāzRēŌūāRŪāzĒäyĀäyġçZōā;Tāy■çZDæŪGäzūāR■āŁŪēāġīījNā;ĒæŸīījNæŞāōĈērTçġĪāāŌzæL'
āĠççŌrāzĒ UnicodeEncodeError āijĈāyŷāŠNāyĀæĪāāēĠGāġçZDæŪLæAġāĀTāĀT
surrogates not allowed āĀĈ
```

èġçāĒşæŪzæāŁ

```
ā;ŞæL'Şā■ræIJçşççZDæŪGäzūāR■æŪūīījNā;ġçTġāyNēġççZDæŪzæşTāRfāzēēAġāĒ■ēġZæāūçZDēTŽē
```

```
def bad_filename(filename):
    return repr(filename)[1:-1]

try:
    print(filename)
except UnicodeEncodeError:
    print(bad_filename(filename))
```

èõlèõž

```
ēġZāyĀārRēŁĈēõlèõžçZDæŸīījġijŪāĒZāġĒēāzāđ'DçŖĒæŪGäzūçşççççZDçġNāzRæŪūāyĀäyġāy■āđ
ézŸēðđ'æĈĒāĒjāyNīījNPythonāAĠāōZæL'ĀæIJL'æŪGäzūāR■ēĈ;āūşçzRæāzæ■ō
sys.getfilesystemencoding() çZDāġijçijŪçāAēġGāzĒāĀĈ
ā;ĒæŸīījNæIJL'āyĀāzZæŪGäzūçşçççşāzūāşæIJL'āijzāŁūēçAæşĈēġZæāūāĀŽīījNāZāæ■đ'āĒAēðyāŁZāzžā
ēġZçġ■æĈĒāĒjāy■āđ'ġāyŷēġAīījNā;ĒæŸīījNæĀzāijZæIJL'āzZçTġæŁūāĒĒēZġ'ēġZæāūāĀZæŁŪēĀĒæŸīījNæŪāēĪ
```

āŖŕēČ;æŸŕāIJāyĀāyĭæIJL'çijžéŽŭçŽDāzčçāAāy■çžŽ open ()
 āĠ;æŦŕāijāēĀSāžEāyĀāyĭāy■āŖLēğDēNČçŽDæŨĠāzūāŖ■)āĀĆ

ā;ŠæL'gēāNčsžāijij os.listdir() èŁŽæāŭçŽDāĠ;æŦŕæŨiijNèŁŽāžŽāy■āŖLēğDēNČçŽDæŨĠāzūāyĀæŨzéIčijNāōČāy■ēČ;āžĒāžĒāŖĭæŸŕāyčāijČèŁŽāžŽāy■āŖLæāijçŽDāŖ■ā■ŨāĀĆēĀNāŖēāyĀæŨzéIčijPythonāržēŁŽāyĭēŮōēčŸçŽDēğčāEšæŨzæāLæŸŕāžŌæŨĠāzūāŖ■āy■ēŌūāŖŨæIJhēğčçāAçŽDā■ŨēŁČāĀijæ\xhhāžūāŖEāōČæŸāŕDæŖUnicodeā■Ũçņē \udchh èāĭçd'žçŽDæL'ĀērŠçŽDāĀiāžčçŖEçijŨçāAāĀĭāĀĆāyNēIčāyĀāyĭā;Nā■ŖāijŦçd'žāžEā;ŠāyĀāyĭāy■āŖLæāijçŽōā;ŦāLŨēāĭāy■āŖNæIJL'āyĀāyĭæŨĠāzūāŖ■āyžt1ēĀNāy■æŸŦUTF-8çijŨçāA)æŨççŽDæāūā■ŖiijŽ

```
>>> import os
>>> files = os.listdir('.')
>>> files
['spam.py', 'b\udce4d.txt', 'foo.txt']
>>>
```

āēČāēdIJā;āæIJL'āžčçāAēIJĀēçAæŠ■ā;IJæŨĠāzūāŖ■æLŨēĀĒāŖEæŨĠāzūāŖ■āijāēĀŠçžŽ
 open () èŁŽæāŭçŽDāĠ;æŦŕiijNāyĀāLĠēČ;ēČ;æ■çāyāūēā;IJāĀĆ
 āŖĭæIJL'ā;Šā;āæČšēēAē;ŠāĠžæŨĠāzūāŖ■æŨūæL■āijŽççŕāŖāžŽēžçČē(æŖŦāēČæL'Šā■Ŗē;ŠāĠžāŖāšŖāžçŁ'žāŖNçŽDīijNā;Šā;āæČšæL'Šā■ŖāyĭēIčçŽDæŨĠāzūāŖ■āLŨēāĭæŨiijNā;āçŽDçĭNāžŖāŖsāijŽāt'ĭæžČiijŽ

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

çĭNāžŖāt'ĭæžČçŽDāŌšāŽāāŖsæŸŕā■Ũçņē \udce4 æŸŕāyĀāyĭēIēdæšŦçŽDUni-
 codeā■ŨçņēāĀĆ āōČāĒūāōdæŸŕāyĀāyĭēčnçğŕāyžāžčçŖEā■ŨçņēāržçŽDāŖNā■ŨçņēçžDāŖLçŽDāŖŌā■LēČ
 çŦšāžŌçijžārŠāžEāL■ā■LēČĭāŖēiijNāŽāæ■d'āōČæŸŕāyĭēIēdæšŦçŽDUnicodeāĀĆ
 æL'ĀāžēŕiijNāŦŕāyĀēČ;æŖŖāLšē;ŠāĠžçŽDæŨzæšŦŕāŖsæŸŕā;ŠēAĠāŖāy■āŖLæšŦæŨĠāzūāŖ■æŨūēĠGāŖ
 æŖŦāēČāŖŕāžēārEāyĭēŖāžççāAāŁōæŦžāēČāyNīijŽ

```
>>> for name in files:
...     try:
...         print (name)
...     except UnicodeEncodeError:
...         print (bad_filename(name))
...
spam.py
b\udce4d.txt
foo.txt
>>>
```

āIJĭ bad_filename() āĠ;æŦŕāy■æĀŌæāūād'Ďç;ōāŖŨāEšāžŌā;āēĠāūšāĀĆ
 āŖēād'ŨāyĀāyĭēĀL'æNŖ'āršæŸŖēĀŽēŁGæšŖçğ■æŨzāijŖēĠ■æŨŖçijŨçāĀiijNçd'žā;NāēČāyNīijŽ

```
def bad_filename(filename):
    temp = filename.encode(sys.getfilesystemencoding(), errors=
↳ 'surrogateescape')
    return temp.decode('latin-1')
```

ěřŠèĀĚæšÍ:

```
surrogateescape:
ěřžčġ■æŸrPythonāIJlčziāđ' ġéčlālĒēíćāřšOSčžĐAPIäŷ■æL' Ää; ěčťlčžĐéťžěřřāđ' ĎčŘēāžlīi.
āōčěč; äžěäŷĀčġ■äijŸéžĚčžĐæŮžāijřāđ' ĎčŘēčťšæš■ä; IJčšžčžšæřřāžžčžĐæťřæ■ōčžĐčijŮčāAē
āIJlēġččāĀāĠžéťžæŮüāijžārēāĠžéťžā■ŮēŁč■ŸāčlālřāŷĀäŷlā; Łāršēcñā; ěčťlālřčžĐUnicode
āIJlčijŮčāAæŮüārēēččāžžéžžŘēŮřāāijāřlēŷŸāōšāžđāōšāēlēġččāĀāđ' sèt' ěčžĐā■ŮēŁčāžřālŮ
āōčāŷ■āžĚāržāžŮOS_
↳ APIēíđāŷŷæIJL' čťlīijňāžšēč; ā; ŁāōžæŷščžĐāđ' ĎčŘēāēŮāžŮāčĚāēŷäŷŷŷčžĐčijŮčāAēťžěřřā
```

ä; ěčťlēčžāŷlčLŁæIJňāžġčťščžĐē; šāĠžæčāŷŷŷŷ

```
>>> for name in files:
...     try:
...         print(name)
...     except UnicodeEncodeError:
...         print(bad_filename(name))
...
spam.py
bĀđ'd.txt
foo.txt
>>>
```

ěřžāŷĀārēŁčāŷžēcŸārřēč; äijžēcñāđ' ġéčlālĒēřžēĀĚæL' ÄāŷčťēāĀčā; ēæŸřāēčāđIJā; āāIJlčijŮāē
āršāŷĚēāžā; ŮēĀčēžšāłřēčžāŷlāĀčāřēāłžā; āārřēč; äijžāIJlāšřāŷlāšlāIJñēcñāřñāłřāđāēñāōđ' āōžěřč

7.16 5.16 āčđāŁæŁŮæťžāŘŸāũšæL'šāijĀæŮĠāžŷčžĐčijŮčāA

éŮōēcŸ

ä; āæčšāIJlāŷ■āĚšēŮāŷĀäŷlāũšæL'šāijĀčžĐæŮĠāžŷāŁ■æřřāŷŷñāčđāŁæŁŮæťžāŘŸāōččžĐUnicode

ēġčāĒšæŮžæāŁ

āēčāđIJā; āæčščžžāŷĀäŷlāžēāžŷŷēčžāłŮāēlāāijřāēL'šāijĀčžĐæŮĠāžŷāũšāŁUnicodečijŮčāA/ēġččāA
ārřāžēā; ěčťl io.TextIOWrapper() āřžēšāāŷēčĚāōčāĀčærťāēčřijž

```
import urllib.request
import io

u = urllib.request.urlopen('http://www.python.org')
```

(continues on next page)

(continued from previous page)

```
f = io.TextIOWrapper(u, encoding='utf-8')
text = f.read()
```

```

    æĈæđĬä;äæĈşăĕŏæŤzäyÄäyĭăũşçzRæL'SăijĂçŽDæŮGæĬñăĭăijRçŽDæŮGăzŭçŽDçijŮçăAæŮzăijR
detach()
    æŮzæşŤçğzėŽd'æŎL'ăũşă■YăĬĭçŽDæŮGæĬñçijŮçăAăşĆijŃ
ăzŭă;ĤçŤĭæŮrçŽDçijŮçăAæŮzăijRăzčæŽĕăĂCăyŃéĭcæŸrăyÄäyĭăĬĭ
    sys.stdout
ăyĽăĕŏæŤzçijŮçăAæŮzăijRçŽDăĭŃă■RiijŽ
```

```
>>> import sys
>>> sys.stdout.encoding
'UTF-8'
>>> sys.stdout = io.TextIOWrapper(sys.stdout.detach(), encoding=
↳'latin-1')
>>> sys.stdout.encoding
'latin-1'
>>>
```

ėĖŽæăũăAŽăRrėĈ;ăijŽăy■æŮ■ă;ăçŽDçzĽçñriijŃėĖŽėGŃăzĖăzĖæŸrăyžăzĖEăijŤçd'žėĂŃăũşăĂĈ

ëŏĭëŏž

I/OçşzçzşçŤşăyĂçşzăĽŮçŽDăşĆăñăæđDăzžėĂŃăĽŔăĂĈă;ăăŔăzžėŕŤçĭĂėĖŔėăŃăyŃéĭcėĖŽăyĭăŞ■ă

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

```

    äĬĭĭėĖŽăyĭăĭŃă■Ŕăy■riijŃio.TextIOWrapper
    æŸrăyÄäyĭçijŮçăAăşŃĖğççăAŮ-
nicodeçŽDæŮGæĬñăđ'ĐçŔĖăşĆijŃ
    io.BufferedWriter
    æŸrăyÄäyĭăđ'ĐçŔĖăzŃėĖŽăĽŮæŤŕæ■ŏçŽDăyėçijŞăĖşçŽDĬ/OăşĆijŃ
    io.FileIO
    æŸrăyÄäyĭăĭçd'žăŞ■ă;ĬçşzçzşăzŤăşĆăŮGăzŭăŔŔėĖŕçñççŽDăŎşăğŃæŮGăzŭăĂĈ
    äćđăĽăăĽŮæŤzăŔŸæŮGæĬñçijŮçăAăijŽăŭĽăŔĽăćđăĽăăĽŮæŤzăŔŸæĬĬĂyĽéĭcçŽD
io.TextIOWrapperăşĆăĂĈ
```

ăyĂėĽŃăĬėėŏşriijŃăĈŔăyĽéĭcă;Ńă■ŔėĖŽăăũėĂŽėĖĖŏĖĖŮăşđăĂğăĬijăĭėçŽŤ'æŎėăŞ■ă;Ĭăy■ăŔŃç
ăĭŃăėĈriijŃăėĈæđĬä;ăėŕŤçĭĂă;ĤçŤĭăyŃéĭcėĖŽăăũçŽDăĽăăĬŔăŤzăŔŸçijŮçăAçĬŃçĬŃăijŽăŔŤçŤşăzĂă

```
>>> f
<_io.TextIOWrapper name='sample.txt' mode='w' encoding='UTF-8'>
>>> f = io.TextIOWrapper(f.buffer, encoding='latin-1')
>>> f
<_io.TextIOWrapper name='sample.txt' encoding='latin-1'>
>>> f.write('Hello')
```

(continues on next page)

(continued from previous page)

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: I/O operation on closed file.
>>>
```

çzŞæđIĴăĜžĕŤŽăžEĭjŇăŽăăyžfçŽĎăŎşăĝŇăĂĭjăûşçzŔĕcñçăť'ăĭŔăžEăžűăĔŞĕŮăăžEăžŤăśCçŽĎăŮĜăă
detach() æŮžæşŤăĭjŽăŮăĭjĂæŮĜăžűçŽĎăIĴăéăűăśCăžűĕŤăŽđçñăžŇăśCĭĭjŇăžŇăŔŎăIĴăéăűăă

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

ăyĂæŮĕăŮăăĭjĂæIĴăéăűăśCăŔŎĭĭjŇăjăăŕşăŔăžĕçzŽĕŤăŽđçzŞæđIĴăűzăăLăăyĂăyĽăŮŕçŽĎăIĴăéăűăă

```
>>> f = io.TextIOWrapper(b, encoding='latin-1')
>>> f
<_io.TextIOWrapper name='sample.txt' encoding='latin-1'>
>>>
```

ărĭçŏăăşçzŔăŔŞăjăăĭjŤçđ'žăžEăŤžăŔŮçĭjŮçăAçŽĎăŮžæşŤĭĭjŇ
ăjEăŸŕăjăĕŤŸăŔŕăžĕăĽĭçŤĭĕŤŽçĝăăĽĂæIĴăĭĕăŤžăŔŮçĭjŮçăžűĕăŇăđ'ĐçŔĖăĂĂĕŤŽĕŕŕăIĴăăĽăăžĕăŔĽăă

```
>>> sys.stdout = io.TextIOWrapper(sys.stdout.detach(), encoding=
↳ 'ascii',
...                                     errors='xmlcharrefreplace')
>>> print('Jalape\u00f1o')
Jalape&#241;o
>>>
```

æşĽăĐŔăyŇăIĴăăŔŎĕçŞăĜžăyçŽĎăĭđASCIIăăŮçņĕ Āś æŸŕăĕCăjŤĕcñ ñ
ăŔŮăžççŽĎăĂĈ

7.17 5.17 ărĖăăŮĕĽCăĖŽăĔĕăŮĜăĭJăăŮĜăžű

éŮĕĕŮ

ăjăăCşăIĴăŮĜăĭJăăĭjŔăĽŤăĭjĂçŽĎăŮĜăžűăyăăĖŽăĔĕăŎşăĝŇăŽĎăăŮĕĽCăŤŕăăăăĂĈ

èġċaEşæÚzæaĹ

årEå■ÙèĹCæTŗæ■õçZt' æÕěaEŻăĖĖæŮĜăzŭçŻĎċijŞăEşăNză■şăRŗijNăĹNăċĊijŻ

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

çşzăijijçŻĎijNěĊĵad' şéĂŽēĴĜērzaRŮæŮĜæIJnæŮĜăzŭçŻĎ buffer
ăşđæĂĝæĹċērzaRŮăzNēĴĂĹŭæTŗæ■ōăĂĈ

èőĹëőŻ

I/OçşzçzşăzēăşĈçżĝçzŞăđĎçŻĎăĵċăijRăđĎăzżēĂNăĹRăĂĈ
æŮĜæIJnæŮĜăzŭæŸrēĂŽēĴĜăĹĹăyĂăyĹæNěæIJĹċijŞăEşçŻĎăzNēĴĂĹŭăĹăĵijRăŮĜăzŭăyĹăċđăĹăăyĂăy
buffer ăşđæĂĝæNĝăRŞărfzăzTçŻĎăzTşăCæŮĜăzŭăĂĈăċĈăđIJăĵçZt' æÕěēōĴēŮōăōĈçŻĎċĹăřşăijZçzTş

æIJnăRĹċĹCăĵNă■RăşTçđ'żçŻĎ sys.stdout ăRrēĈċIJNċĹŭæĹĹæIJĹċĈçĹ'zæōĹăĂĈ
ézŸēōđ' æĈĖăĖĵăyNijNsys.stdout æĂzæŸrăzææŮĜæIJnăĹăĵijRăĹŞăijĂçŻĎăĂĈ
ăĵEæŸrăċĈăđIJăăăĹĹăĖZăyĂăyĹĹIJăċĖAæĹŞă■řăzNēĴĂĹŭæTŗæ■ōăĹrăăĜăĜĖċŞăĜççŻĎĎŻæIJnçŻĎ

7.18 5.18 årEæŮĜăzŭæRĹrēĴrçņęăNĖċĖæĹRăŮĜăzŭăřzēşă

éŮőéćŸ

ăĵăæIJĹăyĂăyĹăřzăzTăzŌăŞ■ăĴçşzçzşăyĹăyĂăyĹăŭşæĹŞăijĂçŻĎI/OéĂŽéAşş(ærTăċĈæŮĜăzŭăĂAçş
ăĵăæĈşăřEăōĈăNĖċĖæĹRăyĂăyĹæZt' éŋŸăşĈçŻĎPythonæŮĜăzŭăřzēşăăĂĈ

èġċaEşæÚzæaĹ

ăyĂăyĹæŮĜăzŭæRĹrēĴrçņęăşNăyĂăyĹæĹŞăijĂçŻĎæŽōéĂŽæŮĜăzŭæŸrăy■ăyĂăăŭçŻĎăĂĈ
æŮĜăzŭæRĹrēĴrçņęăzĖăzĖæŸrăyĂăyĹċTşăŞ■ăĴçşzçzşăNĝăōZçŻĎæTt' æTŗijNçTĹăĹæNĝăzċæşRăyĹçş
ăċĈăđIJăĵçċřăŭĝæIJĹċĖZăzĹăyĂăyĹæŮĜăzŭæRĹrēĴrçņęĵijNăăăRřăzēĖĂŽēĴĜăĵċTĹ
open() ăĜăæTŗăĹăřEăĖŭăNĖċĖĂăyżăyĂăyĹPythonçŻĎæŮĜăzŭăřzēşăăĂĈ
ăĵăăzĖăzĖăRĹĹăĖAăĵċTĹċĖZăyĹæTt' æTŗăĂijçŻĎæŮĜăzŭæRĹrēĴrçņęăĴIJăyżçŋăyĂăyĹăRĈæTŗăĹăăzċæz

```
# Open a low-level file descriptor
import os
fd = os.open('somefile.txt', os.O_WRONLY | os.O_CREAT)
```

(continues on next page)

(continued from previous page)

```
# Turn into a proper file
f = open(fd, 'wt')
f.write('hello world\n')
f.close()
```

āĴšēñŸāsĈçŽDæŮĜāzūāržēsāècñāĖšēŮ■æĹŮèĀĖçāt'āĲçŽDæŮūāĀŽīījŅāžŦāsĈçŽDæŮĜāzūæŔŔèĒ
āēĈædĲēĴZāyĴāzūāy■æŸŕāĵāæĈşēēAçŽDçzŞædĲīījŅāĵāāŔŕāzēçzŽ open()
āĜĵæŦŕāījāēĀŞāyĀāyĴāŔŕéĀĻçŽD colsefd=False āĀĈæŕŦæĈīījŽ

```
# Create a file object, but don't close underlying fd when done
f = open(fd, 'wt', closefd=False)
...
```

èõlèõž

āĲĲUnixçşçzçşşāy■īījŅēĒŽçğ■āŅĖèçĒæŮĜāzūæŔŔèĒŕçñççŽDæĻæĲŕāŔŕāzēāĴĻæŮzāĴççŽDārĒāyĀā
āēĈçõæĀŞāĀĀāēŮæŌēā■Ůç■ĻāĀĈāyĴāĴŅæĴēèõšīījŅāyŅēĴæŸŕāyĀāyĴæŞ■āĲçõæĀŞççŽDäĴŅā■ŔīījŽ

```
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)
```

éĲĀèēĀēĜ■çÇzāījžērĈçŽDāyĀçÇzæŸŕīījŅāyĻéĴççŽDäĴŅā■ŔāzĒāzĒæŸŕāyžāžĒæījŦçd'žāĒĖçĴççŽD
open() āĜĵæŦŕççŽDāyĀāyĴçĻzæĀĝīījŅāzūāyŦāzşāŔŕéĀĈçŦĴāžŌāşžāžŌUnixççççççşşāĀĈ

æĊæđIĲăĵăæĊşârĒäyÄäyĭçşzæŮĠăzŭæŌëăŔċăĲĲçŦĭăĲĲăyÄäyĭăēŮæŌëăŮăzŭăyŊæĲZăĲçZĎăzċăĀăŔăŕă
 makefile() æŮŹăşŦăĀĊăĲĒæŸŕăæĊæđIĲăy■ēĀĊēZŖăŔŕçğzæđ'■æĀğçZĎēŕĭĲĲŊēĊċăyĻēĲçZĎēğċăĒşæ
 makefile() æĀğēĊĲæZŕ'ăēĲăyĲĲĊZăĀĊ

ăĲăăzşăŔŕăzēăĲçŦĭēŦZçğ■æĻăæĲŕăĲēăđĎēĀăăyÄäyĭăĻăŊăŔ■ĲĲŊăĒăēōyăzēăy■ăŔŊăzŌċŋăyĀăŋăă
 äĲŊăēĊĲĲŊăyŊēĲĲăĲĲŦĲđ'zăēĊăĲŦăĻZăzŷăyÄäyĭæŮĠăzŭăŕzēşĲĲŊăŏĊăĒăēōyăĲăēĲşăĠzăzŊēŦZăĻŮăŦŕăæ

```
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()
```

ărĲōăăŔŕăzēăŕĒäyÄäyĭăŭşă■ŸăĲĲçZĎæŮĠăzŭæŔŔēŦŕçŋēăŊēēĊĒăĻŔăyÄäyĭæ■ċăyŷçZĎæŮĠăzŭăŕzēă
 äĲĒæŸŕēēĀăşĭăĎŔçZĎæŸŕăzŭăy■æŸŕăĻĀăĲĲçZĎæŮĠăzŭăĲăĲĲŕēĊĲēċŋăŦŕăŊĲĲŊăzŭăyŦăşŔăzZçş
 (çĻŹăĻŊăŸŕăŭĻăŔĻăĻŕēŦZēŕŕăđ'ĎçŔĒăĀăæŮĠăzŭçzşăŕĲæĲăăzŭç■Ļç■ĻçZĎæŮŭăĀZ)ăĀĊ
 äĲĲăy■ăŔŊçZĎæş■ăĲçşzçşşăyĻēŦZçğ■ēăŊăyŷăzşæŸŕăy■ăyĀăăĲĲŊçĻŹăĻŊçZĎĲĲŊăyĻēĲçZĎăĲŊă■Ŕ
 æĻŖēŕ'ăzĒēŦZăzĻăđ'ZĲĲŊăĎŔăĀĲŕşæŸŕēŏĲăĲăēĒăĻĒæŦŊēŕŦēĠăŭşçZĎăŏđçŌŕăzċăĀĲĲŊçăŏăŦăŏĊēĊ

7.19 5.19 æĻZăzŷăyŦæŮŮæŮĠăzŭăŖŊæŮĠăzŭăđ'z

ēŮŌēĊŸ

ăĲăēĲĲăēĲăĲĲĲĲŊăzŔăĻ'ğēăŊæŮŮăĻZăzŷăyÄäyĭăyŦæŮŮæŮĠăzŭăĻŮçZŏăĲĲŊăzŷăyŊæĲZăĲçŦĭăŦăĻă

ēğċăĒşæŮŹăæăĻ

tempfile æĲăăĲŮăy■æĲĲăĲŦăĻăđ'ZçZĎăĠĲæŦŕăŔŕăzēăŏŊæĻŔēŦZăzŷăĻăăĀĊ
 äyŷăzĒăĻZăzŷăyÄäyĭăŊŦăŔ■çZĎăyŦæŮŮæŮĠăzŭăĲĲŊăŕŕăzēăĲçŦĭ tempfile.
 TemporaryFile ĲĲZ

```
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
```

æĻŮēĀĒĲĲŊăēĊæđIĲăĲăŮĲĲăŋċĲĲŊăĲăēŦŸăŔŕăzēăĊŔēŦZăăŭăĲçŦĭăyŦæŮŮæŮĠăzŭăĲĲZ

```
f = TemporaryFile('w+t')
# Use the temporary file
...
f.close()
# File is destroyed
```

TemporaryFile() çŽĎčňňäyÄäyläRCæTŗæYŗæŮĜäzŭælaaijRiiĴNéĂŽäyæIëèõsæŮĜæIJñælaaijRā
w+t ĩijŅāzŅēfZāLŭælaaijRā;fçTĪ w+b āĂĆ ēfZäylælaaijRāŔŅæŮŭæTŗæŅAērzaŠŅāEŽæŞ■ā;IJiĳŅāIJlēfZ
TemporaryFile() āRēād'ŮēfYŗæTŗæŅAēŭşāEĖç;õçŽĎ open()
āĜ;æTŗäyÄæäŭçŽĎāRCæTŗāĂĆæŕTæÇriĴ

```
with TemporaryFile('w+t', encoding='utf-8', errors='ignore') as f:
    ...
```

āIJlād'ġād'ŽæTŗUnixçşçzçşäyLiĳŅéĂŽēfĜ TemporaryFile()
āLZāzzçŽĎæŮĜäzŭēČ;æYŗāŅfāR■çŽĎiĳŅçTĴēĜşēfđçZōā;TéČ;æşæIJL'āĂĆ
āēČæđIJā;äæČşæLŞçār'ēfZäyléZŔāLŭiĳŅāŔŕäzēä;fçTĪ NamedTemporaryFile()
ælēäzçæZfāĂĆæŕTæÇriĴ

```
from tempfile import NamedTemporaryFile

with NamedTemporaryFile('w+t') as f:
    print('filename is:', f.name)
    ...

# File automatically destroyed
```

ēfZēĜŅiĳŅēcnæLŞaijÄæŮĜäzŭçŽĎ f.name āsđæĀġāŅĖāŔŅāzEērēäyt'æŮŭæŮĜäzŭçŽĎæŮĜäzŭāŔ
ā;Şā;äēIJÄēAārEæŮĜäzŭāŔ■aijæĀŞçzZāEŭāzŮāzççāAæIēæLŞaijÄēfZäylæŮĜäzŭçŽĎæŮŭāĂŽiĳŅēfZā
āŖŅ TemporaryFile() äyÄæäŭiĳŅçzŞæđIJæŮĜäzŭāEşēŮ■æŮŭaijZēcnēĜlāLlāLæZd'æŬLāĂĆ
āēČæđIJā;ääy■æČşēfZāZLāĂŽiĳŅāŔŕäzēäiĳæĀŞäyÄäyläEşēTōā■ŮāŔCæTŗ
delete=False ā■şāŔŕāĂĆæŕTæÇriĴ

```
with NamedTemporaryFile('w+t', delete=False) as f:
    print('filename is:', f.name)
    ...
```

äyžāZēāLZāzzäyÄäyläyt'æŮŭçZōā;TiiĳŅāŔŕäzēä;fçTĪ tempfile.
TemporaryDirectory() āĂĆæŕTæÇriĴ

```
from tempfile import TemporaryDirectory

with TemporaryDirectory() as dirname:
    print('dirname is:', dirname)
    # Use the directory
    ...

# Directory and all contents destroyed
```

ëöíëöß

TemporaryFile() åĀĀNamedTemporaryFile() åŠŃ
TemporaryDirectory() åĜ;æŦř åžŦérëæŸřad'ĐçŘEäyt' æŮüæŮĜäzŮçŽóå;ŦçŽĐæIJĀçóĀå■ŦçŽĐæŮü
åIJläyĀäyĽæŽŦ'ä;ŎçŽĐçžgĀĽñijŃä;ääŦřäzëä;ŧçŦĪ mkstemp() åŠŃ mkdtemp()
æĽæĀĽZāzžäyt' æŮüæŮĜäzŮåŠŃçŽóå;ŦāĀCærŦæCřijŽ

```
>>> import tempfile
>>> tempfile.mkstemp()
(3, '/var/folders/7W/7WZl5sfZEF0pljrEB1UMWE+++TI/-Tmp-/tmp7fefhv')
>>> tempfile.mkdtemp()
'/var/folders/7W/7WZl5sfZEF0pljrEB1UMWE+++TI/-Tmp-/tmp5wvcv6'
>>>
```

ä;EæŸřijŃëŧZāzŽāĜ;æŦřāzŮäy■äijŽāAžëŧZäyĀæ■ëçŽĐçóaçŘEäzEāĀC
ä;ŃäçCřijŃāĜ;æŦř mkstemp() äžĒäzĒāřsëŧŦāŽđäyĀäyĽāŎšāĝŃçŽĐŎSæŮĜäzŮæŦŦëŧçñëijŃä;äéIJĀëç
årŃNæāüä;äëŧŸéIJĀëçAëĜĽāūsæyĒçŘEëŧZāzŽæŮĜäzŮāĀC

éĀŽāyŷæĽëëöšiiijŃäyt' æŮüæŮĜäzŮåIJçşçzçşëzŸëöd' çŽĐä;■ç;öèçñāĽZāzžiiijŃærŦæC
/var/tmp æĽŮçşzäijijçŽĐāIJæŮzāĀC äyžāzEëŎüāŦŮçIJşāödçŽĐä;■ç;öiiijŃāŦřäzëä;ŧçŦĪ
tempfile.gettempdir() åĜ;æŦřāĀCærŦæCřijŽ

```
>>> tempfile.gettempdir()
'/var/folders/7W/7WZl5sfZEF0pljrEB1UMWE+++TI/-Tmp-'
>>>
```

æĽ'ĀæIJĽ'åŠŃäyt' æŮüæŮĜäzŮçŽyāĒşçŽĐāĜ;æŦřëC;āĒAëöyā;äéĀŽëŧĜä;ŧçŦĪāĒşçŦōā■ŮāŦCærŦř
prefix åĀĀsuffix åŠŃ dir æĽëëĜĽāöZāzĽçŽóå;ŦäzëāŦĽāŞ;år■ëĝĐāĽZāĀCærŦæCřijŽ

```
>>> f = NamedTemporaryFile(prefix='mytemp', suffix='.txt', dir='/tmp
↳ ')
>>> f.name
'/tmp/mytemp8ee899.txt'
>>>
```

æIJĀāŦŎëŧŸæIJĽ'äyĀçCžiiijŃär;årŦëC;äzëæIJĀāöĽ'āĒĽçŽĐæŮzāijŦä;ŧçŦĪ tempfile
æĽāāĽŮæĽæĀĽZāzžäyt' æŮüæŮĜäzŮāĀC åŃĒæŃnāzĒççŽā;ŞāĽ■çŦĪæĽüæŎĽæĪCëöŧëŮöäzëāŦĽāIJĽæŮĜäzŮ
ëçAæşĽæĐŦçŽĐæŸřäy■årŃçŽĐāzşāŦŦāŦŦëC;äijŽäy■äyĀæāüāĀCāZāæ■d'ä;ææIJĀäç;éŸĒëřz
āöŸæŮzæŮĜæaç æĽäzEëĝçæŽŦ'ad'ŽçŽĐçzEëĽCāĀC

7.20 5.20 äyŎäyşëāŃçñŦāŦççŽĐæŦřæ■öéĀŽäŧä

éŮöëçŸ

ä;äæCşëĀŽëŧĜäyşëāŃçñŦāŦççëŦzāĒZæŦřæ■öiiijŃāĒyāđŃāIJžæŽŦāŦsæŸŦāŃŸäyĀäzŽçāñāzŮëö;äd'ĜæĽS

èğcâEşæÚzæaĹ

år;çõaā;āāRřazēēĀŽēfĜā;fçTĪPythonāEĖç;õçŽĎI/OāĹaĹUāĹēāōNāĹRēfŽāyĹāzāĹāĹijNā;ĖāřzāžŌāy
pySerialāNĖ āĀC ēfŽāyĹāNĖçŽĎā;fçTĪēĹdāyŷçōĀāTĪijNāĖĹāōĹēčĖpySerialĹijNā;fçTĪçšāĹijāyNēĹēēfZā

```
import serial
ser = serial.Serial('/dev/tty.usbmodem641', # Device name varies
                    baudrate=9600,
                    bytesize=8,
                    parity='N',
                    stopbits=1)
```

èõĹāđ'ĜāRāřzāžŌāyāāNĖçŽĎēõĹāđ'ĜāŠNā\$ā;IJçšççz\$æYřāyāyĀæāūçŽĎāĀC
æřTāēCĹijNāIJĪWindowsçççz\$āyĹĹijNā;āāRřazēā;fçTĪ0, 1çĹēāĹçđ'žçŽĎāyĀāyĹēõĹāđ'ĜāĹēæĹ\$āĹĀēĀŽā
āyĀæUēçnřāRčæĹ\$āĹĀĹijNēCčāřsāRřazēā;fçTĪ read() ĹijNreadline() āŠN write()
āĜ;æTřēřzāĖŽæTřæāžĖāĀCā;NāēCĹijŽ

```
ser.write(b'G1 X50 Y50\r\n')
resp = ser.readline()
```

āđ'ğāđ'ŽæTřæČĖĖĖāyNĹijNçōĀāTçŽĎāyšāRčēĀŽāfāzŌæđ'āRŸāĹUāĀĹĖçōĀāTāĀC

èõĹēõž

år;çõāēāĹēĹcāyĹçIJNēĹūāĹēāĹĹçōĀāTĪijNāĖŪāōđāyšāRčēĀŽāfāæIJĹæŪūāĀŽāz\$æYřæNžēžçČççŽĎ
æŌĹēRā;āā;fçTĪçñāyĹæŪžāNĖāēC pySerial çŽĎāyĀāyĹāŌšāŽāæYřāōCæRŘā;ZāžĖāřzēnYçžğçĹzæĀ
(æřTāēCēūĖæŪŪĹijNāŌğāĹūāēĹĹijNçij\$āĖšāNžāĹūāŪŪĹijNāRāæĹNāRēōōçĹçĹ)āĀCāyĹāyĹā;NāRĹij
RTS-CTS æRāæĹNāRēōōĹijN ā;āāRĹēIJāēççŽ Serial() āĹjāēĀšāyĀāyĹ
rtscts=True çŽĎāRČæTřāšāRřāĀC āĖŪāōYæŪžæŪĖāçēĹdāyŷāōNāŪĎĹijNāŽāæđ'æĹSāIJēfŽēĜNā

æŪūāĹzēōřā;RæĹĀæIJĹæŪĹāRĹāĹRāyšāRčçŽĎI/OēČ;æYřāžNēfZāĹūāĹāĹijRçŽĎāĀCāŽāæđ'ĹijNçā
(æĹŪāæIJĹæŪūāĀŽæĹğēāNāŪĖāIJñçŽĎçijŪçāĀ/ēğççāĀæ\$ā;IJ)āĀC
āRēāđ'Ūā;Šā;āēIJāēçĀāĹZāžzāžNēfZāĹūçijŪçāĀçŽĎæNĖāzđ'æĹŪæTřæōāNĖççŽĎæŪūāĀŽĹijNstruct
āĹāāĹUāž\$æYřēĹdāyŷæIJĹçTĪçŽĎāĀC

7.21 5.21 āžRāĹUāNŪPythonāřzèšā

éŪōēčY

ā;āēIJāēçĀāřĖāyĀāyĹPythonāřzèšāāžRāĹUāNŪāyžāyĀāyĹāŪēĹCætĹĹijNāžēā;ĹāřĖāōČāĹāYāĹRāyĀ

èğcâEşæÚzæaĹ

āřzāžŌāžRāĹUāNŪæIJāæŽōēĀççŽĎāĀžæšTāřsæYřā;fçTĪ pickle
āĹāāĹUāĀCāyžāžĖāřĖāyĀāyĹāřzèšāāfĹāYāĹRāyĀāyĹæŪĖāžūāyĹĹijNāRřāzēēfZæāūāĀžĹijŽ

```
import pickle
```

```
data = ... # Some Python object
f = open('somefile', 'wb')
pickle.dump(data, f)
```

```
pickle.dump(data, f)
```

```
s = pickle.dumps(data)
```

```
pickle.load(s)
pickle.loads(s)
```

```
# Restore from a file
f = open('somefile', 'rb')
data = pickle.load(f)

# Restore from a string
data = pickle.loads(s)
```

_pickle

```
import pickle
f = open('somefile', 'wb')
pickle.dump(data, f)
f.close()
f = open('somefile', 'rb')
data = pickle.load(f)
f.close()

s = pickle.dumps(data)
data = pickle.loads(s)
```

```
>>> 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'}
>>>
```

ā;āēfYēČ;āzRāLŪāNŪāG;æTrijNčsziijNēfYæIJL'æŌēāRčiiNā;EæYřczŠædIJæTřæ■ōāzĚāzĚāřEāōČā

```
>>> import math
>>> import pickle.
>>> pickle.dumps(math.cos)
b'\x80\x03cmath\ncos\nq\x00.'
>>>
```

ā;ŠæTřæ■ōāR■āzRāLŪāNŪāZđælēčŽĐæŪāāĀZiiNāijŽāĚLāAĞāōZæL'ĀæIJL'čŽĐæzŘæTřæ■ōæŪāāĀ
æĪāāiŪāĀAčsžāŠNāG;æTřāijŽēGĪāĪāēNL'ēIJĀārijaĒēēfZæĪēāĀCārzažŌPythonæTřæ■ōēcnāy■āRŊæIJžāZĪā
æTřæ■ōčŽĐāĪā■YāRřēČ;āijZæIJL'ēŪōēcYrijNāZāyžæL'ĀæIJL'čŽĐæIJžāZĪēČ;āfĒēāzēōēēŪōāRŊāyĀyht

æşĪ

```
ā■ČāyGāy■ēēAārzáy■āfaāžžčŽĐæTřæ■ōā;fçTĪpickle.load() āĀČ
pickleāIJĪāĪāē; ;æŪūæIJL'āyĀāyĪāL'řā;IJçTĪārsæYřāōČāijžēGĪāĪāĪāē; ;çŽyāžTāĪāāiŪāž
ā;EæYřæSŘāyĪāĪRāžžāēČādIJçSēēAšpicklecžŽĐāūēā; IJāŌSçŘEiiijN
āžŪārsāRřāžēāĪZāžžāyĀāyĪāAūāĐRçŽĐæTřæ■ōārijēĜt'PythonæL'gèaŊēŽRāĐRæŊGāōžčŽĐçsžčž
āžāā■d' iijNāyĀāōžēēAāĪĪēfApickleāRĪāIJĪçŽyāžšāzŊēŪt' āRřāžēēōd' ērAārzáæŪžçŽĐēĝčædR
```

æIJL'āžZčsžādNčŽĐāržēsæYřāy■ēČ;ēcnāzRāLŪāNŪčŽĐāĀČēfZāzZēĀŽāyŷæYřēČcāzZā;ĪetŪād'Ūē
ærTāēČæL'ŠāijĀčŽĐæŪGāzūiiNč;ŠçzIJēfđæŌēiiNčžfçĪNiiNēfZçĪNiiNāāĪāyĝç■L'ç■L'āĀČ
çTĪæĪūēGĪāōZāzL'čsžāRřāžēēĀŽēfGæRŘā;Z _____getstate____()
āŠN _____setstate____() æŪzæşTælēçzTēfGēfZāzZēZŪāĪāĀČ
āēČædIJāōZāzL'āžEēfZāyđ'āyĪāŪzæşTiiijNpickle.dump()
āršāijZēfČçTĪ _____getstate____() èŪāRŪāzRāLŪāNŪčŽĐāržēsāāĀČ
çsžāijijčŽĐiiN_____setstate____() āIJĪāR■āzRāLŪāNŪæŪēcnēřČçTĪāĀČāyžāzEæijTçd'žēfZāyĪāūēā;IJāČ
āyŊēĪcæYřāyĀāyĪāIJāEĒēČĪāōZāzL'āžEāyĀāyĪçžfçĪNā;Eāz■çĐūāRřāžēāzRāLŪāNŪāŠNāR■āzRāLŪāNŪç

```
# 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)
```

ērTçlĀēfRēāNāyNēlćçŽĐāzRāLŪāNŪērTēlNāzčçāAīijŽ

```
>>> 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()
```

çĐūāRŌēĀĀGžPythonēğçædRāZlāzūēG■āRrāRŌāE■ērTēlNāyNīijŽ

```
>>> f = open('cstate.p', 'rb')
>>> pickle.load(f)
countdown.Countdown object at 0x10069e2d0>
T-minus 19
T-minus 18
...
```

äjāāRrāzēçIJNāLrçžŁçlNāRĀLāēGēŁzēLñçŽĐēG■çTšāzEīijNāzŌā;āçññāyĀæñqāzRāLŪāNŪāōCçŽĐāIJ

pickle ārzāžŌād'gādNçŽĐæTṛæ■ōçzŞædĐæfTæCä;ŁçTl array æLŪ numpy
ælqālŪāLZāzžçŽĐāžNēŁZāLŪæTṛçzĐæTlçŌGāzūāy■æYṛāyĀäylenYæTlçŽĐçijŪçāAæŪzāijRāĀĆ
æÇædIJä;æéIJĀēçAçğzāLlād'gēGRçŽĐæTṛçzĐæTṛæ■ōīijNā;āæIJĀāē;æYṛāĒLāIJlāyĀäyĽæŪGāzūāy■āṛEāĒ
(éIJĀēçAçññāyLæŪzāžŞçŽĐæTṛæNA)āĀĆ

çTšāžŌ pickle æYṛPythonçL'zæIJL'çŽĐāzūāyTēZĐçlĀāIJlæžRçāAāyLīijNæL'ĀæIJL'æÇædIJéIJĀēç
ä;NāçĆīijNāçÇædIJæžRçāAāRŸāLlāžEīijNā;āæL'ĀæIJL'çŽĐā■YāĆlæTṛæ■ōāRrēÇ;āijZēcñçāt'ālRāzūāyTāF
ālççZ;æLēēōīijNārzāžŌāIJlæTṛæ■ōāžŞāSŃā■YæaçæŪGāzūāy■ā■YāĆlæTṛæ■ōæŪīijNā;āæIJĀāē;ä;ŁçTlæZ
ēŁZāžZçijŪçāAæāijāijRæZt'æāGāGEīijNāRrāzēēcñāy■āRŃçŽĐēr■ēlĀæTṛæNāīijNāzūāyTāzşēÇ;ā;Lāē;çŽ

æIJĀāRŌāyĀçCžēçAæşlæDRçŽĐæYṛpickle æIJL'ād'gēGRçŽĐēĒ■ç;ōēĀL'ēāzāSŃāyĀāžZæçYæL'N
ārzāžŌæIJĀāyÿēçAçŽĐä;ŁçTlāIJzæZīijNā;āāy■éIJĀēçAāŌzæNĒāŁÇēŁZāyīijNā;EæYṛæÇædIJä;æççAāIJlæ
æIJĀāē;āŌzæşēçYĒäyĀäyN āōYæŪzæŪGæaç āĀĆ

8 çññāĒ■çñāīijŽæTṛæ■ōçijŪçāAāSŃād'ĐçRĒ

ēŁZāyĀçñāāyžēçAēōlēōžä;ŁçTlPythonād'ĐçRĒāRĐçğ■āy■āRŃæŪzāijRçijŪçāAçŽĐæTṛæ■ōīijNārTāç
āSŃæTṛæ■ōçzŞædĐēCçāyĀçñāāy■āRŃçŽĐæYīijNēŁZçñāāy■āijZēōlēōžçL'zæōŁçŽĐçŌUæşTēŪōēçYīijNē.

Contents:

8.1 6.1 èrZàEŽCSVæTřæ■ó

éUóécŸ

ä;äæÇşèrZàEŽäyÄäyłCSVæäijäijRçŽĐæŰGäzŰäĂĆ

èğcàEşæŰzæąŁ

årzäžŎåd'ğåd'ŽæTřçŽĐCSVæäijäijRçŽĐæTřæ■óèrZàEŽéŰóécŸijNéČ;årRäzèä;ŁçTł
csv äžŞäĂĆ ä;NäæČrijŽàAĞèö;ä;ääIJläyÄäyłàR■àRñstocks.csvæŰGäzŰäy■æIJL'äyÄäžŽèČaçèłäyČàIJæTřæ

```
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
```

äyNéİcàRŞä;ääsTçd'žæČä;TårEèŁZäžŽæTřæ■óèrZàRŰäyžäyÄäyłàĖČçzĐçŽĐäžRàŁŰrijŽ

```
import csv
with open('stocks.csv') as f:
    f_csv = csv.reader(f)
    headers = next(f_csv)
    for row in f_csv:
        # Process row
    ...
```

àIJläyŁéİççŽĐäžççäAäy■rijN row äijŽæŸřäyÄäyłàŁŰèąłàĂĆàŽæ■d'rijNäyžäžEèöŁéŰóæŞRäyłà■Űæö
row[0] èöŁéŰóSymbolijN row[4] èöŁéŰóChangeăĂĆ

çTšäžŎèŁŽçğ■äyNæăĞèöŁéŰóéĂžäyŸäijŽäijTçtŰæŰŰæŰErijNä;ääRřäzèèĂČèŽŚä;ŁçTłàŚ;årR■ăĖČçzĐä

```
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
    ...
```

ăöČăĖAèöyä;ää;ŁçTłàŁŰàR■ăèČ row.Symbol àŠN row.Change
äžæŽĖäyNæăĞèöŁéŰóăĂĆ éIJĂèçAæşłæĐRçŽĐæŸřèŁžäyłàRłæIJL'àIJłàŁŰàR■æŸřàRŁæşTçŽĐPythonæă
ä;ääRřèČ;éIJĂèçAăŁŰæTžäyNăŎşăğNçŽĐàŁŰàR■(ăèČàRĖİdăăĞèřEçñèăŰçñèæŽŁæ■čæŁRäyNăŁŞçžŁäž

årĖåd'ŰäyÄäyłéĂŁ'æNł'årśæŸřàRĖæTřæ■óèrZàRŰăŁřäyÄäyłà■ŰăĖyăžRàŁŰäy■ăŎžăĂĆàRřäzèèŁžæă

```
import csv
with open('stocks.csv') as f:
    f_csv = csv.DictReader(f)
    for row in f_csv:
        # process row
    ...
```

āĪĴēƧZāyĵĿL'æĪĴāy■ĳĴNā;āāRřāzēā;ƧĴĴlāĴŪāR■āŌžēōƧēŪōāŕRāyĀēāNĵŽDāŦŕā■ōāžEāĀĆāŕŦāēŦ
æĴŪēĀĒ row['Change']

āyžāžEāEŽāĒĒCSVæŦŕā■ōĳĴNā;āāz■ĴDūāRřāzēā;ƧĴŦĴsvāĴāĴŪĳĴNāy■ēƧĴēƧZāŪūāĀŽāĒĴāĴZāžā
writer āŕžēšāāĀĆā;NāēĆ:

```
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)
```

āēĆāēĴĴā;āæĪĴ'āyĀāyĴā■ŪāĒyāžRāĴŪĴŽDāŦŕā■ōĳĴNāRřāzēāĴŕēƧZāūūāĀŽĳĴ

```
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)
```

ēōlēōž

ā;āāžŦēŕēāĀžāŦŕāĳŦāĒĴēĀĴ'æNŦĴsvāĴāĴŪāĴĴāĴ'šæĴŪēģĴāēĴŦCSVæŦŕā■ōāĀĆā;NāēĆĳĴĴNā;āāRř

```
with open('stocks.csv') as f:
    for line in f:
        row = line.split(',')
        # process row
    ...
```

ä;ŁçTlèfZçg■æŰzâijRçZDäyÄäylçijžçCzâršæYřä;äaz■çDűéIJĀèeAāŌzād'DçRĒäyĀäzZæčYæL'NçZDç
ærTāeĆiijNāeĆædIJæšŘāzZā■ŰæōġāĀijècñâijTāRūāNĒāZt'rijNā;āäy■ā; Űäy■āŌzéZd'èfZāzZâijTāRūāĀC
āRēād'ŰriijNāeĆædIJäyÄäylècñâijTāRūāNĒāZt' çZDā■ŰæōġçřāuġāRnāIJL'äyÄäylèĀŰāRūriijNéCčāzŁçĪNāz

ézYèød'æČĚāEġäyNriijNcsv āzŠāRřerEāLnMicrosoft Ex-
celæL'Āä;ŁçTlçZDCSVçijŰçāAèġDāŁZāĀC èfZæŁŰèōyāzšæYřæIJĀäyÿèġAçZDā;čâijRriijNāzūäyTāzšâijZ
çDűēĀNriijNāeĆædIJä;āæšèçIJNcsvçZDæŰĠæāçriijNāršâijZāRŠçŌřæIJL'ā;Łād'Zçg■æŰzæšTārEāōČāzTçTl
ä;NāeĆiijNāeĆædIJä;āæČšerzāRŰäzētabāLEāL'sçZDæTřæ■ōriijNārřāzèèfZæāūāAžriijZ

```
# 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
    ...
```

āeĆædIJä;āæ■čāIJlërzāRŰCSVæTřæ■ōāzūārEāōČāzñè;ñæ■čäyžāŠ;āR■āĚČçzDriijNéIJĀèeAæšlæDRārZ
ä;NāeĆiijNāyÄäylCSVæâijâijRæŰĠāzūāIJL'äyÄäylāNĒāRnéĪdæšTæāĠerEçñeçZDāŁŰād't'èqNriijNçszâijijZ

```
StreetĀāAddress,Num-Premises,Latitude,Longitude 5412ĀāNĀāCLARK,10,
→41.980262,-87.668452
```

èfZæāūæIJĀçZŁâijZārījeĠr'āIJlāŁZāzžāyÄäylāŠ;āR■āĚČçzDæŰūāžġçTšāyÄäyl
ValueError âijČäyÿēĀNād'sèt'ēāĀC äyžāžEèġčāEšèfZēŰōécYriijNā;āāRřeČ;äy■ā; Űäy■āĚŁāŌzāfōæ■čā
ä;NāeĆiijNārřāzēāČRāyNéĪcèfZæāūāIJlélđæšTæāĠerEçñeäyŁä;ŁçTlāyÄäylæ■čāŁZēālē;ä;âijRæZŁæ■çriijZ

```
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
    ...
```

èfYæIJL'èĠ■èeAçZDäyĀçCzéIJĀèeAâijžerČçZDæYřriijNcsvāžġçTšçZDæTřæ■ōēČ;æYřā■Űçñeäyšçsz
āeĆædIJä;āeIJĀèeAāAžèfZæāūçZDçszādNè;ñæ■çriijNā;āāfĚēāzēĠlāūsæL'NāŁlāŌzāōđçŌrāĀC
äyNéĪcæYřäyÄäylāIJĪCSVæTřæ■ōäyŁæL'ġeāNāĚūāzŰçszādNè;ñæ■ççZDä;Nā■RriijZ

```
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))
    ...
```

āRēād'ŰriijNāyNéĪcæYřäyÄäylè;ñæ■čā■ŰāĚyāy■çL'zāōZā■ŰæōġçZDä;Nā■RriijZ

```

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)

```

éĀŽāyāælēēōīījNā;āāRrēČ;āzūāy■æČšēŁĠād'ŽāŌžēĀČēŽSēŁZāžZē;ñæ■céŮōécŸāĀĆ
āIJlāōdéŽēæČēĀēĤāy■īījNCSVæŪĠāzūēČ;æŁŪād'ŽæŁŪārŠæIJL'āžŽčijžād'sčŽDæTŗæ■ōīījNēcńčāt'āIRčŽL
āZāæ■d'īījNēZd'ēldā;āčŽDæTŗæ■ōčāōāōdāIJL'āŁlēŽIJæŸrāĠEčāōāŮāērřčŽDīījNāRēāŁZā;āāŁēēāzēĀČēŽ
æIJĀāRŌīījNāēČædIJā;āērzaRŪCSVæTŗæ■ōčŽDčŽōčŽDæŸrāAžæTŗæ■ōāŁēædRāŠNčžšēōāčŽDērīīj
ā;āāRrēČ;ēIJĀēēAčIJNāyĀčIJN Pandas āNĒāĀĆPandas
āNĒāRnāžEāyĀāyŁēlđāyŸæŪzāŁčŽDāĠ;æTŗāRń pandas.read_csv()
īījN āōČāRfāzēāŁāē;CSVæTŗæ■ōāŁrāyĀāyŁ DataFrame ārzēsāy■āŌzāĀĆ
čDūāRŌāŁl'čTlēŁZāyŁāržēsā;āārsāRfāzēčTšæŁRāRĎčg■ā;čāijRčŽDčžšēōāāĀAēŁĠæzd'æTŗæ■ōāzēāRŁæL
āIJl6.13ārRēŁCāy■āijZæIJL'ēŁZæāūāyĀāyŁā;Nā■RāĀĆ

8.2 6.2 èrzāĒJSONæTŗæ■ō

éŮōécŸ

ā;āæČšērzaĒĒJSON(JavaScript Object Notation)čijŪčāAæāijāijRčŽDæTŗæ■ōāĀĆ

èġčāĒšæŮzæāŁ

json ælāāIŮæRĠā;ŽāžEāyĀčg■ā;ŁčōĀā■TčŽDæŪzāijRælēčijŪčāAāŠNēġččāAJSONæTŗæ■ōāĀĆ
āĒūāy■āyđ'āyŁāyžēēAčŽDāĠ;æTŗæŸr json.dumps() āŠN json.loads()
īījN ēēAærTāĒūāzŪāžRāŁŮāNŪāĠ;æTŗāžšæČpicklečŽDæŌēāRčārSā;Ůād'ŽāĀĆ
āyNēlčāijTčđ'žāēČā;TārEāyĀāyŁPythonæTŗæ■ōčžšæđDē;ñæ■čāyžJSONīījŽ

```

import json

data = {
    'name' : 'ACME',
    'shares' : 100,
    'price' : 542.23
}

json_str = json.dumps(data)

```

āyNēlčāijTčđ'žāēČā;TārEāyĀāyŁJSONčijŪčāAčŽDā■Ůčņēāyšē;ñæ■čāŽđāyĀāyŁPythonæTŗæ■ōčžšæđDē

```
data = json.loads(json_str)
```

```
json.dump() ašN json.load() æIëcijÚçäAäŠNëgççäAJSONæTřæ■ōāĀĆä;NäeĆrijŽ
```

```
# 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)
```

ëöleöž

JSONcijÚçäAæTřæNẠçŽĐă\$žæIJnæTřæ■ōçšzādNäyž None iijN bool iijN int iijN float ašN str iijN äžēāRĹāNĒāRnēfZāžZçšzādNæTřæ■ōçŽĐlistsiiNtuplesāšNictionariesāĀĆāržāžŌictionariesiijNkeysēIJĀēēAæYřā■ŮçñēäyšçšzādN(ā■ŮāĒyāy■āzzā;TéIdā■ŮçñēäyšçšzādNçŽĐkeyāL.äyžāžEēAṭā;JSONēgĐēNĆrijNā;āāžTèrēāRĹcijÚçäAPythonçŽĐlistsāšNictionariesāĀĆēĀNäyTrijNāIJlwebāžTçTĪclNāžRāy■iijNēāuāsĆāržēšāēcncijÚçäAäyžāyĀäyĹā■ŮāĒyæYřāyĀäyĹæāGāGEāA

JSONcijÚçäAçŽĐæaijāijRāržāžŌPythonē■æšTēĀNāuśāGāāžŌæYřāōNāĒĹāyĀæāuçŽĐiijNēZd'āžEäyĀærTāēĆrijNTrueaijŽēcñæYāārDäyžtrueiijNFalseēcñæYāārDäyž-falseiijNēĀNNoneaijŽēcñæYāārDäyžnullāĀĆäyNēĪcæYřāyĀäyĹā;Nā■RiijNæijTçd'žāžEcijÚçäAāRŌçŽĐā■

```
>>> json.dumps(False)
'false'
>>> d = {'a': True,
...      'b': 'Hello',
...      'c': None}
>>> json.dumps(d)
'{"b": "Hello", "c": null, "a": true}'
>>>
```

æĀĆæđIJā;æērTçĪĀāŌžæĀæšēJSONēgççäAāRŌçŽĐæTřæ■ōrijNā;æēĀŽāyā;ĹéŽ;ēĀŽēfGçōĀā■TçŽĹçĹ'žāĹnæYřā;ŠæTřæ■ōçŽĐā;NāēŮçzŠæđĐāsĆæñāā;ĹæūsæĹŮēĀĒāNĒāRnād'gēGRçŽĐā■ŮæōtæŮūāĀĆäyžāžEēgçāEšēfZāyĹēŮōēcYiijNāRřāžēēĀĆēZŠā;fçTĪpprintēĹāāIŮçŽĐpprint() āĠ;æTřæĪēāžcæŽĹæŽōēĀŽçŽĐ print() āĠ;æTřāĀĆāōČaijŽæNĹçĒgkeyçŽĐā■Ůæī■ēāžāžRāžūāžēäyĀçg■æZt'āĹāç;ŌēgČçŽĐæŮžaijRē;ŠāGžāĀĆäyNēĪcæYřāyĀäyĹæijTçd'žāēČā;TæijČāžōçŽĐæĹŠā■rē;ŠāGžTwitteräyĹæRĪJçt'ççzŠæđIJçŽĐā;Nā■RiijŽ

```
>>> from urllib.request import urlopen
>>> import json
>>> u = urlopen('http://search.twitter.com/search.json?q=python&
↳ rpp=5')
>>> resp = json.loads(u.read().decode('utf-8'))
>>> from pprint import pprint
>>> pprint(resp)
```

(continues on next page)

(continued from previous page)

```
{'completed_in': 0.074,
'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'}
>>>
```

äyÄèLñæIëèøšijŃJSONèğççäAäijŽæāžæ■ōæRŘä;ŽçŽDæTřæ■ōāLŽāžzdictsæLŮlistsāĀĆ
æĀēCædIJä;äæČšèèAāLŽāžžāĒŮāžŮčšžādŇčŽDāržèšāijŇNāRřāžèçžŽ json.
loads() äijäēĀšobject_pairs_hookæLŮobject_hookāRĆæTřāĀĆ
ä;ŇæĀēČrijŇäyŇéIæYřæČä;TřæYäyĀyĪJSONā■ŮāĒyè;ñæ■cāyžäyĀyĪPythonāržèšā;Ňā■RijŽ

```
>>> 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)])
>>>
```

äyŇéIæYřæČä;TřæYäyĀyĪJSONā■ŮāĒyè;ñæ■cāyžäyĀyĪPythonāržèšā;Ňā■RijŽ

```
>>> class JSONObject:
...     def __init__(self, d):
...         self.__dict__ = d
...
>>>
>>> data = json.loads(s, object_hook=JSONObject)
>>> data.name
'ACME'
```

(continues on next page)

(continued from previous page)

```
>>> data.shares
50
>>> data.price
490.1
>>>
```

```
__init__()
    """Initialize the JSON object with the given data.
    The data can be a dictionary, a list, a string, or a file object.
    If the data is a string, it should be a JSON-formatted string.
    If the data is a file object, it should be a file containing a
    JSON-formatted string.
    """
```

```
    """Serialize the JSON object to a string.
    The data can be a dictionary, a list, a string, or a file object.
    If the data is a string, it should be a JSON-formatted string.
    If the data is a file object, it should be a file containing a
    JSON-formatted string.
    """
    return json.dumps(data, indent=4)
```

```
>>> print(json.dumps(data))
{"price": 542.23, "name": "ACME", "shares": 100}
>>> print(json.dumps(data, indent=4))
{
    "price": 542.23,
    "name": "ACME",
    "shares": 100
}
>>>
```

Traceback (most recent call last):

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

æĈæđIä;äæĈşăŹŔăĹŮăŃŮăŕŹèşăăđă;ŃiijŃă;ăăŔŕăžèæŔŔă;ŽăyĂăyĹăĜ;æŤŕiijŃăőĈĉŽĎë;ŞăĖëæŸŕ

```
def serialize_instance(obj):
    d = { '__classname__' : type(obj).__name__ }
    d.update(vars(obj))
    return d
```

æĈæđIä;äæĈşăŔ■èĤĜăĬèëŎŭăŔŮëĤŽăyĹăăđă;ŃiijŃăŔŕăžèëĤŽăăŭăĂŽiijŽ

```
# 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
```

ăyŃéĬæŸŕăĈă;Ťă;ĤĉŤĬèĤŽăžŽăĜ;æŤŕĉŽĎă;Ńă■ŔiijŽ

```
>>> 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 æĹăăĬŮèĤŸæĬĹ'ă;Ĺăđ'ŽăĖŭăžŮéĂĹ'éăžæĬèæŎĝăĹŭăŽŦă;ŎĉžĝăĹŋĉŽĎăŤŕă■ŮăĂĂĉĹ'žæőĹăĂŦ
ăŔŕăžèăŔĈèĂĈăőŸæŮžæŮĜăăĉèŎŭăŔŮæŽŦăđ'ŽĉžĖĎĹĈăĂĈ

8.3 6.3 èĝĉæđŔĉőĂă■ŤĉŽĎXMLæŤŕæ■ő

éŮőéĉŸ

ă;äæĈşăžŎăyĂăyĹĉőĂă■ŤĉŽĎXMLæŮĜăăĉăy■æŔŔăŔŮæŤŕæ■ăăĂĈ

åŖräzëä;ŁçŦĭ xml.etree.ElementTree æĭąąĭŰäzŎçŏÅă■ŦçŽĐXM-
 ŁæŨĞæąçäy■æŖŖåŖŨæŦŕæ■ŏăĀĆ äyžăEæĭjŦçd' žĭĭjŅăAŦĞëŏ;ă;ăæĈşëğçæđŖPlanet
 PythonăyŁçŽĐRSSæžŖăĀĆăyŅéĭçæŸŕçŽyăžŦçŽĐăžçăAĭĭjŽ

èƒRèaÑäyLéÍcçŽĎäzččăAĭĭjÑèĭSăĜžczŞæđIĲszăiĭjèƒŽæăũĭĭjŽ

```

    ħĭŁæŸĭçĎũĩĩŃæĈæđĬĴä;ăăĈşăAžēfZăyĂæ■ēĈĎăđ'ĐĉŘĖĩĩŃă;ăéĬĂèēAæŽfæ■
print() è■ăŘăĖĭăőŃăĹŔăĖŮăzŮæĬĴĹ'ēũĉĉŽĎăžŃăĈ

```

ãIJľãĹŁad'ŽãžTčTľćľŇãžRäy■ad'DčŘEXMLčijŮčãAæãijãijRčŽDæTřæ■óæYřãĹŁäyÿëğAçŽDãĂĆ
 äy■äzĚãZäyžXMLãIJľInternetäyĹéľcãũšçžŘècňãžŁæšŽãžTčTľãžŮæTřæ■óäžd'æ■ćijŇ
 âŔŇæŮũãőCãžšæYřäyĂçğ■ã■YăĆľãžTčTľćľŇãžRæTřæ■ócŽDäyÿçTľæãijãijR(æŕTăçĆã■Ůad'DčŘEijŇéššã
 æŮëäyŇæľčŽDěóľěőžãijŽãĚĹãAĞãőŽëržèĂĚãũšçžRăržXMLãšžçãĂæŕTèĹČçEšæĹŁ'ăžEãĂĆ

ãIJľãĹŁad'ŽæČĚãEĵäyŇijŇãĵŠăĵŁçTľXMLæľëäzĚäzĚã■YăĆľæTřæ■ócŽDæŮũãĂŽijŇãŕžãžTčŽDæŮĞ
 äĵŇæĆijŇäyĹéľcãĵŇã■Räy■čŽDRSSèőcéYĚæžŘçšžãijjãžŮäyŇéľčŽDæãijãijRijŽ

```
<?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()
find()
findtext()
channel/
item
title

```

```

doc.iterfind('channel/item')
item
title

```

```

ElementTree
tag
get()

```

```

>>> 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')
>>>

```

```

xml.etree.ElementTree
from lxml.etree import
parse

```

8.4 6.4 áćđéĠŕăĭjŔèġċæđŦăđ'ġăđŦXMLăŮĠăzŮ

éŮőécŸ

```

channel/
item
title

```

èġċăĖşşăŮzăăĹ

```

doc.iterfind('channel/item')
item
title

```

```

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

```

äyžāẸætNērTēfZāyIāG;æTrijNā;æIJÀēAāĒLæIJL'äyÄäyIād'gādNçŽĐXMLæŮĜāzūāĀĆ
 éĀŽāyyä;āāRřāzēāIJāTfāzIJç;ŚçñŽæLŮāĒñāĒsæTřæ■ōç;ŚçñŽäyLæL'čāLřēfŽæāũçŽĐæŮĜāzūāĀĆ
 ā;NāēCrijNā;āāRřāzēäyNē;XMLæāijāijRçŽĐēLiāLāāŞēāşŌäyĆēAŞēurāIŚæt'ijæTřæ■ōāzŞāĀĆ
 āIJlāEŻēfŽæIJñāzēçŽĐæŮūāĀŽrijNäyNē;æŮĜāzūāũşçzRāNĒāRñēũĒēfĜ100,000ēāNæTřæ■ōrijNçijŮçāA

```

<response>
  <row>
    <row ...>
      <creation_date>2012-11-18T00:00:00</creation_date>
      <status>Completed</status>
      <completion_date>2012-11-18T00:00:00</completion_date>
      <service_request_number>12-01906549</service_request_
↪number>
      <type_of_service_request>Pot Hole in Street</type_of_
↪service_request>
      <current_activity>Final Outcome</current_activity>
      <most_recent_action>CDOT Street Cut ... Outcome</most_
↪recent_action>
      <street_address>4714 S TALMAN AVE</street_address>
      <zip>60632</zip>
      <x_coordinate>1159494.68618856</x_coordinate>
      <y_coordinate>1873313.83503384</y_coordinate>
      <ward>14</ward>
      <police_district>9</police_district>
      <community_area>58</community_area>
      <latitude>41.808090232127896</latitude>

```

(continues on next page)

(continued from previous page)

```
<longitude>-87.69053684711305</longitude>
<location latitude="41.808090232127896"
longitude="-87.69053684711305" />
</row>
<row ...>
  <creation_date>2012-11-18T00:00:00</creation_date>
  <status>Completed</status>
  <completion_date>2012-11-18T00:00:00</completion_date>
  <service_request_number>12-01906695</service_request_
↪number>
  <type_of_service_request>Pot Hole in Street</type_of_
↪service_request>
  <current_activity>Final Outcome</current_activity>
  <most_recent_action>CDOT Street Cut ... Outcome</most_
↪recent_action>
  <street_address>3510 W NORTH AVE</street_address>
  <zip>60647</zip>
  <x_coordinate>1152732.14127696</x_coordinate>
  <y_coordinate>1910409.38979075</y_coordinate>
  <ward>26</ward>
  <police_district>14</police_district>
  <community_area>23</community_area>
  <latitude>41.91002084292946</latitude>
  <longitude>-87.71435952353961</longitude>
  <location latitude="41.91002084292946"
longitude="-87.71435952353961" />
</row>
</row>
</response>
```

åAĞeøö;ä;äæÇşåEŻäyÄäyİeDŽæIJnæİeæNL'çĖğāİŞæt'ijæLëåŞLæTřéGRæŎŞāLŮéCőcijŮāRŮçāAāĂĆā

```
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)
```

ēŁŻäyİeDŽæIJnăTřäyĂçŽĐeŮőécŸæYřăŎČäijŽăĚĹăřEæTř'äyİXMLæŮĞäzŭāLăe;ĵăĹăEĚă■Yäy■çĐŭ
āIJİæĹŞçŽĐæIJzăZİäyĹiijNăyžăžEēŁRëaŊēŁŻäyİçİŊăžRéIJĂēēĂçTİăĹř450MBăŭēāRşçŽĐăEĚă■YçĹ'žéŮřă
ăēĆăđIJă;ŁçTİăēCăyNăžčăĂřijŊçİŊăžRăRİēIJĂēēĂăŁőăTřäyĂçCžçCžijŽ

```
from collections import Counter
```

(continues on next page)

(continued from previous page)

```
potholes_by_zip = Counter()

data = parse_and_remove('potholes.xml', 'row/row')
for pothole in data:
    potholes_by_zip[pothole.findtext('zip')] += 1
for zipcode, num in potholes_by_zip.most_common():
    print(zipcode, num)
```

çzŞædIJæYřrijŽèŁŻäylçL'ŁæIJñçŽDžčçāAèŁRëāNæUũāRlėIJĀēēA7MBçŽDāEĖĀ■Y–ād'ğād'ğēŁCçžē

ėōlēōž

ėŁŻäyĀēŁCçŽDæŁĀæIJřaijŽä; İèŦŦ ElementTree æĹāāĹŦüäy■çŽDäyđ'äylæäyāŁČāŁšēČ;āĀĆ
çññäyĀiijŦiterparse() æŦŦzæşŦāĖĀēōyārŦXMLæŦŦGæaçēŁŻēāNācđēĠRæŞ■ā;IJāĀĆ
ā;ŁçŦĹæŦŦiijŦNā;āēIJĀēēAæRŘā;ZæŦŦGāzūāR■āŞŦāyĀäyĹāNĖāRñāyNēĹcāyĀçğ■æŁŦād'Žçğ■çşzādNçŽDā
start , end, start-ns āŞŦ end-ns āĀĆ çŦś iterparse()
āŁZāžžçŽDēŁ■āzčāŽĹaijŽāžğçŦŦā;čāēČ (event, elem) çŽDāĖČçzDřijŦ āĖŦüäy■
event æYřāyŁēŁřāzNāzūāŁŦēāĹāy■çŽDæşŘāyĀäyĹiijŦēĀŦ elem æYřçŽyāžŦçŽDŦXM-
ŁāĖČçŦ'āāĀĆā;NāēČiijŽ

```
>>> 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 āžNāzūāIJĹæşŘāyĹāĖČçŦ'āçññäyĀæñāēčēnāŁZāžžāžūāyŦēŁYæşāæIJĹ'ēčēnāRŞāĖĖāĖŦüāžŦæŦřæ■
ēĀŦ end āžNāzūāIJĹæşŘāyĹāĖČçŦ'āāuşçzŦRāōNāĹRæŦŦēčēnāŁZāžžāĀĆ
ār;čōāæşāæIJĹāIJĹā;Nā■Řāy■āijŦçđ'žriijŦ start-ns āŞŦ end-ns
āžNāzūēčēnçŦĹāēāđ'ĐçŘEXMLæŦŦGæaçāŞ;āR■çŦ'žēŦŦçŽDāçŦæYŦōāĀĆ

ėŁŻæIJñēŁČā;Nā■Řāy■iijŦ start āŞŦ end āžNāzūēčēnçŦĹāēčōāçŘĖāĖČçŦ'āāŞŦæāĠç■æāŁāĀĆ
æāŁāžčēāĹāžĖæŦŦGæaçēčēnēğçæđŘæŦŦūçŽDāşČæñāçzŞæđDřijŦ
ėŁYēčēnçŦĹāēāŁđ'æŦ■æşŘāyĹāĖČçŦ'ææYřāŘēāNzéĖ■āijāçzŽāĠ;æŦř
parse_and_remove() çŽDēŦŦřā;DāĀĆ āēČæđIJāNzéĖ■iijŦNāřsāŁŦ'çŦĹī yield
ēŦ■āŘēāRŞēŦČŦĹēĀĖēŁŦāZđēŁŻäyĹāĖČçŦ'āāĀĆ

āIJĹ yield āžNāŘŦŦçŽDäyNēĹcēŁŻäyĹē■āŘēāŁ■æYřā;Łā;ŦŦĹNāžRā■āçŦĹāđAāřSāĖĖĀ■YçŽDĖElement

```
elem_stack[-2].remove(elem)
```

ēfZāyīēŕ■āRēā;fā; ŪāzNāL'■çTš yield āžgçTšçZDāĒČçt' āāzŌāōČçZDçLūēLCçCzāy■āLāēZd' æŌL' ā
āAĠēō;āūšçzRæšæIJL'āĒūāōČçZDāIJræŪzāijTçTīēfZāyīāĒČçt' āāzEīijNēCčāzLēfZāyīāĒČçt' āāršēcñēTĀæ
āržēLCçCzçZDēf■āzčāijRēgçædRāSŊNāLāēZd' çZDæIJĀçzLæTlædIJāršæYřāyĀāyīāIJlæŪĠæaçāyLēnY
æŪĠæaçæāSçzSædDāzŌāgNēĠçzLæšæcñāōNæTt' çZDāLZāzžēfGāĀCār;çōāāçCæ■d' iijNēfYæYřēČ;éĀŽ
ēfZçg■æŪzæāLçZDāyžēçAçijžēZūāršæYřāōČçZDēfRēāNæĀgèČ;āžEāĀC
æLŠēĠāūsætNērTçZDçzSædIJæYřijNērzaRŪæTt' āyīāēŪĠæaçāLrāĒĒā■Yāy■çZDçL' LæIJnçZDēfRēāNēĀ
ā;EæYřāōČā■t' ā;fçTīāžEēūĒēfGāRŌēĀĒ60āĀ■çZDāĒĒā■YāĀC
āZāæ■d' iijNāçCædIJā;æZt' āĒšāfČāĒĒā■Yā;fçTīēGRçZDēfīijNēCčāzLācđēGRāijRçZDçL' LæIJnāōNēČIJ

8.5 6.5 āŕĒā■ŪāĒyē;ñæ■cāyžXML

éŪōécY

ā;āæČšā;fçTīāyĀāyīPythonā■ŪāĒyā■YāČlæTřæ■ōiijNāzūārĒāōČē;ñæ■cæLŔXMLæāijāijRāĀC

ēgčāĒşæŪzæāL

ār;çōā xml.etree.ElementTree āžSéĀŽāyçTīālēāAŽēgçædRāūēā;IJiijNāĒūāōđāōČāzšāRřāzēāL
ā;NāçČiijNēĀČēZŠāçCāyNēfZāyīāĠ;æTřijŽ

```
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
```

āyNēlçæYřāyĀāyīā;fçTīā;Nā■RiijŽ

```
>>> s = { 'name': 'GOOG', 'shares': 100, 'price':490.1 }
>>> e = dict_to_xml('stock', s)
>>> e
<Element 'stock' at 0x1004b64c8>
>>>
```

ē;ñæ■cçzSædIJæYřāyĀāyī Element āōđā;NāĀCārzažŌI/OæS■ā;IJiijNā;fçTī xml.
etree.ElementTree āy■çZD tostring() āĠ;æTřā;LāōzæYšāršēČ;ārĒāōČē;ñæ■cæLŔāyĀāyīā■ŪēL

```
>>> from xml.etree.ElementTree import tostring
>>> tostring(e)
b'<stock><price>490.1</price><shares>100</shares><name>GOOG</name></
↳stock>'
>>>
```

æĈæđĪä;äæĈşçŻæşŘäyĽăĚĈĉt'ăæûzâĽăăśđæĂğăĂijĭijŃăŔřăžěă;£çŤĪ set ()
æŮzæşŤĭjŽ

```
>>> e.set('_id', '1234')
>>> tostring(e)
b'<stock _id="1234"><price>490.1</price><shares>100</shares><name>
↳GOOG</name>
</stock>'
>>>
```

æĈæđĪä;äèĽŸæĈşăĽĪæŃAăĚĈĉt'ăçŽĐéąžăžŔĭjŃăŔřăžěèĂĈèŽŚæđĐéĂăăŸĂăŸĽ
OrderedDict æĽăăžçæŽĽăŸĂăŸĽæŽóéĂŽçŽĐă■ŮăĚŸăĂĈèŕŭăŔĈèĂĈ1.7ăŔŔèĽĈăĂĈ

èőĽèőž

ă;ŞăĽŽăžžXMLçŽĐæŮŭăĂŽĭjŃă;ăèćnéŽŔăĽŮăŔĽèĈ;æđĐéĂăă■ŮçņăŸşçşzăđŃçŽĐăĂĭăĂĈă;ŃăçĈĭ

```
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)
```

éŮőéćŸæŸŕăæĈæđĪä;äæĽŃăĽĽçŽĐăŮžæđĐéĂăçŽĐæŮŭăĂŽăŔŕèĈ;ăĭjŽççŕăĽŕăŸĂăžŽéžçÇèăĂĈă;ŃăçĈĭ

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

æşĽăĐŔăĽŕçĭŃăžŔçŽĐăŔŮéĭćéĈçăŸĽă;Ńă■ŔăŸ■ĭijŃă■Ůçņę âĂŸ<âĂŽ âŠŃ âĂŸ>âĂŽ
èćăæŽĽæ■çăĽŔăžĚ < âŠŃ >

äyÑéíçázĚäĭZăŔĈèĀĈĭijŇăęĈăđIJăĭăéIJĂëęAæLŇăLÍăŌzèĭňă■çèŁZăžZă■ŮčņęĭijŇăŔŕăzëăĭŁĉŦĭ xml.sax.saxutils äy■ĉŽĎ escape() äŠŇ unescape() äĜĭæŦŕăĀĈăĭŇăęĈĭijŽ

```
>>> from xml.sax.saxutils import escape, unescape
>>> escape('<spam>')
'&lt;spam&gt;'
>>> unescape(_)
'<spam>'
>>>
```

éŽď'ăžĚęĈĭăĹZăžzæ■çĉăŏĉŽĎĚĭŞăĜzăđ'ŮĭijŇëŁŸæIJLăŔĚăđ'ŮăyĂăyĹăŌşăZăæŌĭë■ŔăĭăăĹZăžž Element äŏđăĭŇëĀŇăy■æŸŕă■ŮčņęăyşĭijŇ éĈĉăŕşæŸŕăĭŁĉŦĭ■ŮčņęăyşĉžĐăŔĹăđĐéĂăăyĂăyĹăZŦ'ăđ'ĝĉ ěĀŇ Element äŏđăĭŇăŔŕăzëăy■ĉŦĭēĀĈèŽŚęĉăđŔXMLæŮĜæIJŇĉŽĎăĈĚăĚăyŇéĂŽēŁĜăđ'Žĉĝ■æŮzăăăžşăŕşæŸŕĕŦ'ĭijŇăĭăăŔŕăzëăIJăyĂăyĹénŸĉžĝăŦŕă■ŏĉžŞăđĐăyĹăŏŇăĹŔăĭăăĹĂæIJL'ĉŽĎăŞă■ăĭIJĭijŇăžŮ

8.6 6.6 ěĝĉăđŔăŠŇăŁŏăŦžXML

éŮŏéĉŸ

ăĭăăĈşĕŕzăŔŮăyĂăyĹXMLæŮĜăęĉĭijŇăŕzăŏĈăēIJĂăyĂăžZăŁŏăŦžĭijŇĉĐŮăŔŌăŕĚĉžŞăđIJăĚăZăđXM

ěĝĉăĚşæŮzăæĹ

ăĭŁĉŦĭ xml.etree.ElementTree æĹăăĭŮăŔŕăzëăĭĹăŏzæŸŞĉŽĎăđ'ĐĉŔĚēŁZăžZăžzăĹăăĀĈĉŇăyĂæ■æŸŕăzëēĂŽăyĉŽĎăŮzăĭjŔăĭēěĝĉăđŔēŁZăyĹăŮĜăęĉăĀĈăĭŇăęĈĭijŇăĂĜĕŏĭăĭăIJL'ăyĂăyĹă pred.xml ĉŽĎăŮĜăęĉĭijŇĉşzăĭĭjăyŇéíçēŁZăăŮĭijŽ

```
<?xml version="1.0"?>
<stop>
  <id>14791</id>
  <nm>Clark &amp; Balmoral</nm>
  <sri>
    <rt>22</rt>
    <d>North Bound</d>
    <dd>North Bound</dd>
  </sri>
  <cr>22</cr>
  <pre>
    <pt>5 MIN</pt>
    <fd>Howard</fd>
    <v>1378</v>
    <rn>22</rn>
  </pre>
  <pre>
    <pt>15 MIN</pt>
    <fd>Howard</fd>
```

(continues on next page)

(continued from previous page)

```
<v>1867</v>
<rn>22</rn>
</pre>
</stop>
```

äyÑéÍcæYřäyÄäyłŁ'çTÍ ElementTree æİēēřzāRŪēŁZäyłæŮĞæççāzūārřzāōČāAžäyÄäzZāŁōæTžçŽ

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

ād'DçŘĚçzŠæđIæYřäyÄäyłŁČŘäyÑéÍcèŁZæäüæŮřçŽĐXMLæŮĞäzūijŽ

```
<?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>
```


(continued from previous page)

```
>>> doc.find('content/{http://www.w3.org/1999/xhtml}html')
<Element '{http://www.w3.org/1999/xhtml}html' at 0x1007767e0>
>>> # Doesn't work
>>> doc.findtext('content/{http://www.w3.org/1999/xhtml}html/head/
↳title')
>>> # Fully qualified
>>> doc.findtext('content/{http://www.w3.org/1999/xhtml}html/'
... '{http://www.w3.org/1999/xhtml}head/{http://www.w3.org/1999/
↳xhtml}title')
'Hello World'
>>>
```

ä;ääRfrazëeÄZëfGärEäS;äR■çl'zéU'äd'DçRÉéÄZè;SāNĖëcĖÄyžäyÄäyläüëäĖüçszælēçōÄāNŨēfZäylē

```
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)
```

éÄZëfGäyNélcçŽDæŨžaijRä;fçTlëfZäylçsziiž

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

ëőléőž

èğçæđRāRñæIJL'āS;āR■çl'zéU't'çŽDXMLæŨGæaçäijZæfTè;ČçzAçRŘāĂĆ äyLélcçŽD
XMLNamespaces äzĖäzĖæYřāĖAëöyä;ää;fçTlçijl'çTĕāR■äzçæZfăōNæTt'çŽDURIārĖāĖŨāRŸā;Ũçl■ā;öç
ā;Ĺäy■āzȳçŽDæYřiiĴNāIJlāšzæIJñçŽD ElementTree
èğçæđRäy■æšqæIJL'äzzä;TĕĀTā;ĎëŌŭāRŨāS;āR■çl'zéU't'çŽDäfæAřāĂĆ
ä;EæYřiiĴNāçCæđIJä;ää;fçTlĭterparse() āG;æTřçŽDĕrfārsāRfrazëeŌŭāRŨæZt'äd'ŽāĖšzžŌāS;āR■çl'zé

```
>>> from xml.etree.ElementTree import iterparse
>>> for evt, elem in iterparse('ns2.xml', ('end', 'start-ns', 'end-
↳ns')):
...     print(evt, elem)
...
end <Element 'author' at 0x10110de10>
```

(continues on next page)

(continued from previous page)

```
start-ns ('', 'http://www.w3.org/1999/xhtml')
end <Element '{http://www.w3.org/1999/xhtml}title' at 0x1011131b0>
end <Element '{http://www.w3.org/1999/xhtml}head' at 0x1011130a8>
end <Element '{http://www.w3.org/1999/xhtml}h1' at 0x101113310>
end <Element '{http://www.w3.org/1999/xhtml}body' at 0x101113260>
end <Element '{http://www.w3.org/1999/xhtml}html' at 0x10110df70>
end-ns None
end <Element 'content' at 0x10110de68>
end <Element 'top' at 0x10110dd60>
>>> elem # This is the topmost element
<Element 'top' at 0x10110dd60>
>>>
```

æIJĀāRŌäyĂċĆzīijŊāēĆæđIJä;ăēēAāđ'ĐċŔĖċŽĐXMLæŪĠæIJñēŽđ'ăžĖēēAă;ċċŦĭlĀŦăĔūāzŪēñŸċžg
 āžžēōōă;ăæIJĀāē;æŸŦă;ċċŦĭ 1xml āĠă;ĤŦăžŠăĭēăžċæŽĤ ElementTree āĂĆ
 ä;ŊāēĆīijŊ1xml āŦŦăLŦċŦĭDTĐēŦērAæŪĠăæċăĀAæŽŦ'ăē;ċŽĐXPathæŦŦæŊĀăŦŊăyĀăžŽăĔūāzŪēñŸċž
 ēŦŽăyĀārŦēLĊăĔūāōđăŦŦæŸŦăŦŽă;ăăēĊă;ŦēōŦXMLēġċæđŦŦċġă;ōċōĀăŦăyĂċĆzăĂĆ

8.8 6.8 äÿŒǻĚșçșzǻđŊæȚræ■óǻžȘçŽǾžd'ǻžŠ

éŮóécŸ

ä:äăĈšăĬJăĬăĖščszăđNăTăă■ăžSăy■ăşëërăăĂăăđăĹăăĹŮăĹăăŽđ'eōăăTăĂĈ

èġčǎEşæŮźæąŁ

Pythonäy■èácd'žád'ŽèaÑæTřæ■óčŽDæăĜăĜEæŮzàiŋŘæŸřäyÄäyłcŤsăĚČčžDæđDæĹŘčŽDăžŘăĹŮă

```
stocks = [
    ('GOOG', 100, 490.1),
    ('AAPL', 50, 545.75),
    ('FB', 150, 7.45),
    ('HPQ', 75, 33.2),
]
```

ä;Iæ■ðPEP249iijÑéĀŽēfĠēfZçg■ā;ćaijRæRŘä;ŽæTṛæ■ōiijÑ
 âRfäzēä;LâôzæYŞçZDä;£çTĪPythonæāGāĠEæTṛæ■ōāzŞAPIâŞÑăĒşçşzādNæTṛæ■ōāzŞēfZēāÑāzd'āzŞăĀĆ
 æL'ĂæIJL æTṛæ■ōāzŞäyŁçZDæŞ■ä;IJēĆ;éĀŽēfĠSQLæşēērçēr■āRēæIēăōNăĹRăĀĆærRăyĀēăÑē;ŞăĒēē;

äyžāžEæijŦčd'žèrt'æYŦiijŦñā;āāRřāžēā;£çŦíPythonæāĞāĞEāžŞäy■çŽĐ sqlite3
æÍaqaíŦāĀĆ æÇædÍJā;āā;£çŦÍçŽĐæYřāyĀäyĺāy■āRŦçŽĐæŦřæ■ōāžŞ(æŦŦāēĆMySqlāĀPostgresqlæÍŦŦēĀ
ēŦYā;ŦāōŦēēĖçŽyāžŦçŽĐçññāyŦ'æŦžāŦaqaíŦāŦēāRŦā;ŽæŦřæŦāāĀĆ
äy■ēŦGçŽyāžŦçŽĐçijŦćŦŦāŦēāRčāĞāžŦōēĆ;æYřāyĀæāūçŽĐiijŦēŽĐ'āžEäyĀçĆçççççEā;ōāūōāŦŦāđ'Ŧā

cñňäyĂæ■ĕæYřęđđæŌéáŁrăTŗă■őázŞăĂĆéĂŽăÿÿă;ăĕeAæL'gèaŃ connect ()
 ǎĞ;æTrııjŃ cZŁóCăRŔă;ŻăŸĂăžZăTŗă■őázSăŖăăĂĂăÿzăIJzăĂAcŤlăŁuăŖăăĂĂărEcăAăŚŃăEűăzŰăĕĒ

```
>>> import sqlite3
>>> db = sqlite3.connect('database.db')
>>>
```

äyžāẒEāđ'ĐçŘEæTṛæ■ōījNāyNāyĀæ■ēā;āéIĀēēAāŁŻāzẒāyĀāyĭæyŷæāĜāĀĆ
äyĀæŪēā;āæIJL'āẒEæyŷæāĜījNēĆčāzŁā;āāřsāŘřāzēæL'gēāŃSQLæšēēřcēr■āŘēāẒEāĀĆæřTāēĆījŽ

```
>>> c = db.cursor()
>>> c.execute('create table portfolio (symbol text, shares integer,
↳ price real)')
<sqlite3.Cursor object at 0x10067a730>
>>> db.commit()
>>>
```

äyžāẒEāŘŚæTṛæ■ōāẒŞēāĭäy■æŘŚāĒēād'Žæĭæēōřā;TījNā;ŁçTĭčśzāijijäyNēĭcēŁŻæāũçŽĐēr■āŘēījŽ

```
>>> c.executemany('insert into portfolio values (?, ?, ?)', stocks)
<sqlite3.Cursor object at 0x10067a730>
>>> db.commit()
>>>
```

äyžāẒEæL'gēāŃæšŘāyĭæšēēřcērījNā;ŁçTĭāČRāyNēĭcēŁŻæāũçŽĐēr■āŘēījŽ

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

āēĆæđIJā;āæČşæŌēāRŪçTĭāŁũē;ŚāĒēā;IJāyžāŘĆæTṛæĭæL'gēāŃæšēēřcæŞ■ā;IJījNāŁĒēāzçāōāŁā;ā

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

ēōĭēōž

āIJĭæřTē;Čä;ŌçŽDçzĝāŁnāyŁāŠNæTṛæ■ōāẒŞāzđ'āẒŞæŸřēĭđāyŷçōĀā■TçŽĐāĀĆ
ä;āāŘĭēIJāæŘŘä;ŻSQLēr■āŘēāzūērČçTĭčŽyāẒTçŽĐāĭāĭŪāřsāŘřāzēæŽt'æŪřæŁŪæŘŘāŘŪæTṛæ■ōāẒEāĀ
ēŽ;ēřt'āēĆæ■đ'īijNēŁŸæŸřæIJL'äyĀāẒZæřTē;ČæçŸæL'ŃçŽDçzEēŁĆēŪōēcŸēIJĀēēAā;āéĀŘāyĭāŁŪāĜzāĆ

äyÄäylēŽ;çCzæYřæTřæ■ōāžSäy■çŽDæTřæ■ōāšŇPythonçszādNçŽt' æŌēçŽDæYāārDāĀĆ
 áržāžŌæŮēæIJçszādNřijŇéĀŽāyŷāRřāžēā;£çTl datetime ælqāiŮäy■çŽD datetime
 āōđā;ŇřijŇ æLŮēĀĒāRřēČ;æYř time ælqāiŮäy■çŽDçszçzšæŮēēŮt' æLšāĀĆ
 áržāžŌæTřā■ŮçszādNřijŇçLžāLnæYřā;£çTlāLřāRřæTřçŽDēGšēđ■æTřæ■ōřijŇāRřāžēçTl
 decimal ælqāiŮäy■çŽD Decimal āōđā;ŇæĪēēāçd'žāĀĆ
 äy■āžyçŽDæYřijŇāržāžŌäy■āRŇçŽDæTřæ■ōāžSēĀŇēĪĀĀĒŮā;SæYāārDēğDāLŽæYřāy■äyĀæāuçŽDřijŇā
 āRēād'ŮäyÄäylæŽt'āLāād'■æĪČçŽDēŮōēçYārsæYřSQLēr■āRēā■ŮçņäyšçŽDæđDēĀāĀĆ
 ā;āā■ČäyGäy■ēēAā;£çTlPythonā■ŮçņäyšæāijāijRāŇŮæS■ā;IJçņē(āēČ%)æLŮēĀĒ
 .format() æŮžæšTæĪēāLŽāžžē£ZæāuçŽDā■ŮçņäyšāĀĆ
 āēČæđIJāijāēĀšçžZē£ZāžZæāijāijRāŇŮæS■ā;IJçņēçŽDāĀijæĪēēGĪāžŌçTlæLūçŽDē;SāĒēřijŇēČčāžLā;āçŽ
 http://xkcd.com/327)āĀĆ æšēēřçēr■āRēäy■çŽDēĀŽēĒçņē ?
 æŇGçd'žāRŌāRřæTřæ■ōāžSā;£çTlāōČēGĪāuççŽDā■ŮçņäyšæZēæ■čæIJžāLŮřijŇē£ZæāuæŽt'āLāçŽDāōL'ā
 äy■āžyçŽDæYřijŇäy■āRŇçŽDæTřæ■ōāžSāRŌāRřāržāžŌēĀŽēĒ■çņēçŽDā;£çTlæYřāy■äyĀæāuçŽDā
 ? æLŮ %s řijŇē£YæIJL'āĒŮāžŮäyĀāžZā;£çTlāžEäy■āRŇçŽDçņēāRŮřijŇæřTāēČ:0æLŮ:1æĪēæŇGçd'žāRČ
 āRŇæāuççŽDřijŇā;āē£YæYřā;ŮāŌžāRČēĀČā;āā;£çTlçŽDæTřæ■ōāžSæĪqāiŮçŽYāžTçŽDæŮGæāčāĀĆ
 äyÄäylæTřæ■ōāžSæĪqāiŮçŽD paramstyle āsdæĀgāŇĒāRnāžEāRČæTřāijTçTlēcŌæāijçŽDā£æAřāĀĆ
 áržāžŌçōĀā■TçŽDæTřæ■ōāžSæTřæ■ōçŽDēržāEŽēŮōēçYřijŇā;£çTlæTřæ■ōāžSAPIēĀŽāyŷēĪdāyŷçōĀ
 āēČæđIJā;āēAād'ĐçRĒæŽt'āLāād'■æĪČçŽDēŮōēçYřijŇāžžēōōā;āā;£çTlæŽt'āLāēŇYçžgçŽDæŌēāRčřijŇæř
 çszāijij SQLAlchemy ē£ZæāuççŽDāžSāĒēōyā;āā;£çTlPythonçszāēēēāçd'žāyÄäylæTřæ■ōāžSēāřijŇ
 āžŮāyTēČ;āIJlēZRēŮRāžTāsCSQLçŽDæČēĀEřāyŇāōđçŌřāRĐçg■æTřæ■ōāžSççŽDæS■ā;IJāĀĆ

8.9 6.9 çijŮçăAāŠŇēğççăAā■AāĒ■ē£ZāLŮæTř

ēŮōēçY

ā;āæČšārEäyÄäylā■AāĒ■ē£ZāLŮā■ŮçņäyšēğççăAāēLŘāyÄäylā■ŮēŁČā■ŮçņäyšæLŮēĀĒāRēEäyÄäylā

ēğçăEşæŮzæāŁ

āēČæđIJā;āāRlæYřçōĀā■TçŽDēğççăAāēLŮçijŮçăAäyÄäylā■AāĒ■ē£ZāLŮççŽDāŌšāgŇā■ŮçņäyšřijŇā
 æĪqāiŮāĀČā;ŇāēČřijŽ

```

>>> # Initial byte string
>>> s = b'hello'
>>> # Encode as hex
>>> import binascii
>>> h = binascii.b2a_hex(s)
>>> h
b'68656c6c66f'
>>> # Decode back to bytes
>>> binascii.a2b_hex(h)
b'hello'
>>>
  
```

çszāijijçŽDāLšēČ;āRŇæāuāRřāžēāIJ base64 æĪqāiŮäy■æL;āLřāĀČā;ŇāēČřijŽ

```
>>> import base64
>>> h = base64.b16encode(s)
>>> h
b'68656C6C6F'
>>> base64.b16decode(h)
b'hello'
>>>
```

èõìèõž

åd'gëČlâLEæČĚâĚřäyNïijNéÅŽëfGä;fçTlâyLèfřçŽDâĜ;æTřæIèè;ñæ■čâ■AâĚ■èfZâLúæYřâ;ŁçóĀâ■T
 äyLéÍcäyd'çg■æLĀæIJřçŽDäyzèèAäy■âRŇâIJlâžŎâd'ğârRâEžçŽDâd'DçŘĚãĀĆ
 âĜ;æTř base64.b16decode() âŠŇ base64.b16encode()
 âRlèČ;æŞ■ä;IJâd'ğâEŽâ;čâijRçŽDâ■AâĚ■èfZâLúâ■Ůæř■ijŇ èĀŇ binascii
 ælââlŮäy■çŽDâĜ;æTřâd'ğârRâEžéČ;èČ;âd'DçŘĚãĀĆ

èfYæIJL'äyĀçČzéIJĀèèAæşlæDŘçŽDæYřçijŮčâAâĜ;æTřæL'ĀäžgçTşçŽDè;ŞâĜzæĀzæYřäyĀäyġâ■Ů
 âèČâdIJæČşâijžâLúäzèUnicodeâ;čâijRè;ŞâĜžijŇâ;æéIJĀèèAâčdâLäyYĀäyġéçlâd'ŮçŽDçTŇéÍæ■èéld'ãĀĆ

```
>>> h = base64.b16encode(s)
>>> print(h)
b'68656C6C6F'
>>> print(h.decode('ascii'))
68656C6C6F
>>>
```

âIJlègççâAâ■AâĚ■èfZâLúæTřæŮüijŇâĜ;æTř b16decode()
 âŠŇ a2b_hex() âRřäzèæŎèâRŮâ■ŮèLCæLŮunicodeâ■ŮçñèäyşâĀĆ
 ä;EæYřijŇUnicodeâ■ŮçñèäyşâfĚéqzäzĚäzĚâRlâŇĚâRŇASCIIçijŮčâAçŽDâ■AâĚ■èfZâLúæTřãĀĆ

8.10 6.10 çijŮčâAègççâABase64æTřæ■ó

éŮóécY

ä;æéIJĀèèAä;fçTlBase64æâijâijRègççâAæLŮçijŮčâAäžNèfZâLúæTřæ■óãĀĆ

ègçâEşæŮzæâĹ

base64 ælââlŮäy■æIJL'äyd'äyġâĜ;æTř b64encode() and b64decode()
 âRřäzèäyşâ;æègçâEşèfZäyġéŮóécYãĀĆ;NâèÇ;

```
>>> # Some byte data
>>> s = b'hello'
>>> import base64
```

(continues on next page)

(continued from previous page)

```
>>> # Encode as Base64
>>> a = base64.b64encode(s)
>>> a
b'aGVsbG8='

>>> # Decode from Base64
>>> base64.b64decode(a)
b'hello'
>>>
```

èõlèõž

Base64çijŮčăĀăžĒăžĚçŤlăžŎēlčăŔŖšă■ŮèŁĆçŽĐæŤŕæ■őæŕŤăçĈă■ŮèŁĈă■ŮçņęäÿšăŠŇă■ŮèŁĈæŤŕçæ■d'ăd'ŮiijŇçijŮčăĀăd'ĐçŔĚçŽĐēçŖšăĜžçzŖšăđIJæĀžæŸŕăÿĀăÿlă■ŮèŁĈă■ŮçņęäÿšăĀĆăçĈăđIJă;ăæĈşæŭŭăŔĹă;ŖçŤlBase64çijŮčăĀçŽĐæŤŕæ■őăŠŇUnicodeæŮĜæIJŇiijŇă;ăăŖĚăžæŭzăĹăăÿĀă

```
>>> a = base64.b64encode(s).decode('ascii')
>>> a
'aGVsbG8='
>>>
```

ă;ŖşēğçăĀBase64çŽĐæŮŭăĀŽiijŇă■ŮèŁĈă■ŮçņęäÿšăŠŇUnicodeæŮĜæIJŇēĈ;ăŔŕăžăä;IJăÿžăŔĈæŤŕăă;ĒæŸŕiijŇUnicodeă■ŮçņęäÿšăŔlèĈ;ăŇĚăŔŕŇASCIIă■ŮçņęăĀĆ

8.11 6.11 èŕžăĒžăžŇèĚŽăĹŮæŤŕçžĐæŤŕæ■ó

éŮőécŸ

ă;ăæĈşêŕžăĒžăÿĀăÿlăžŇèĚŽăĹŮæŤŕçžĐçŽĐçzŖšăđĐăŇŮæŤŕæ■őăĹŕPythonăĚĈçžĐăÿ■ăĀĆ

èğčăĒşæŮžæąĹ

ăŔŕăžăä;ŖçŤl struct æŭăăŮăđ'ĐçŔĚăžŇèĚŽăĹŮæŤŕæ■őăĀĆăÿŇèĹčæŸŕăÿĀăôŧçd'žăŖŇăžççăĀăŕĒăÿĀăÿIJPythonăĚĈçžĐăĹŮæŭăăĒăĒăÿĀăÿlăžŇèĚŽăĹŮæŮĜăžŭiijŇă struct ŕŕĒæŕŔăÿlăĚĈçžĐçijŮčăĀăÿžăÿĀăÿlçzŖšăđĐă;ŖăĀĆ

```
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'â;Lâd'Žçg■æŮzæſTæIëèrZâRŮèfZâyIæŮĜâzûâzûèfTâZđâyĂâyIâĚĈçzDâLŮèaIâĂĈ
éçŮâĚLijNâçĈæđIJâ;ăæLſçõŮâzèaIŮçŽĐâ;ćaijRâćđéĜRèrZâRŮæŮĜâzûiiijNâ;ăâRfäzèèfZæăûâAŽiijŽ

```
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
        ...
```

ăçĈæđIJâ;ăæĈſârEæTt'âyIæŮĜâzûâyĂæñæĂğèrZâRŮâLrâyĂâyIâ■ŮèĹĈâ■Ůçñçâyšây■iiijNçĐûâRŌâI

```
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
    ...
```

âyđ'çg■æĈĚâEğâyNçŽĐçzſæđIJéĈ;æŸrâyĂâyIâRfèfTâZđçTlæIèâĹZâzžèrèæŮĜâzûçŽĐâŌſăgNâĚĈçz

èóIèóž

âržâžŌéIJĀèçAçijŮçăAăſNègççăAăžNèfZâĹúæTtæ■óçŽĐçIŊâžRèAŊèIĀrijNéĂžâyÿaijŽa;fçTl
struct æĹaâIŮâĂĈâyžâžEăçræŸŌâyĂâyIæŮrçŽĐçzſæđDă;ſiiijNâRĹéIJĀèçAăĈRèfZæăûâĹZâzžâyĂâyI

(continued from previous page)

```
<callable_iterator object at 0x10069e6d0>
>>> for chk in chunks:
...     print(chk)
...
b'\x01\x00\x00\x00ffffff\x02@\x00\x00\x00\x00\x00\x00\x12@'
b'\x06\x00\x00\x00333333\x1f@\x00\x00\x00\x00\x00\x00"@'
b'\x0c\x00\x00\x00\xcd\xcc\xcc\xcc\xcc\xcc*\x9a\x99\x99\x99\x99YL@'
>>>
```

æĈăĵăæĹĀĕĝĀĵĴŃĀĹZăzăyĀăyĹăŔfēĹăzĉăŕzēsaĉZĎăyĀăyĹăŌŝăZăæŸŕăŏĈēĈĵăĒAĕŏyăĵĉŦĹăyĀăy
æĈăĎĪĴăăyăăĵĉŦĹēĹZĉĝăăĹĀăĪŕĵĴŃĕĈăzĹăzĉăăĀăŔfēĈĵăĵZăĈŔăyŃĕĬĕēĹZăăĵĵĴ

```
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)
```

ăĪĴăĜĵăŦŕ unpack_records() äyăăĵĉŦĹăzĒăŔĕăĎŰăyĀăĝăăŰzăŝŦ
unpack_from() āĀĈ unpack_from() árăzăŹŌăzŌăyĀăyĹăĎĝăĎŤăzŃĕĹZăĹŰăŦŕĉzĎăyăăŔŕăŔŰăzŃĕ
ăZăăyăăŏĈăyăăĵĴăzĝĉŦŝăzăĵŦĉZĎăyŦăŰăŕzēsaĉĹŰēĀĒēĹZēăŃăĒĒăŸăĎŦăĹŰăŝăăĴăĴĀĈ
ăĵăăŔĹēĪĴăĕĀĉZăŏĈăyĀăyĹăŰēĹĈăăŰĉĕăyŝ(ăĹŰăŦŕĉzĎ)ăŝŢăyĀăyĹăŰēĹĈăăŔĉĝzēĜŔĵĴŃăŏĈăĵĴă
æĈăĎĪĴăăyăăĵĉŦĹ unpack() æĹăăzăăZĹ unpack_from() ĵĵŢ
ăĵăĕĪĴăĕĀăĴŏăŦăzăĉăăĀăĹăĎĎĕĜŔĉZĎăŕŔĉZĎăĹĜĉĹĜăzēăŔĹēĹZēăŃăăŔĉĝzēĜŔĉZĎĕăĉŏŰ

```
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))
```

ēĹZĉĝăăŰzăăĹĒZĎăzĒăzĉăăĀĉĪŢŢăyĹăŌzăĹăĎŦăĬăĎŦŰĵĴŢĕŸăĴŰăĀZăĹăĎŦZĕĬăĎŦŰĉZĎăŰēăĴ
ăĎŦăĹŰăŦŕăăăzēăŔĹăĎĎĕĀăăŔĉZĎăĹĜĉĹĜăŕzēsaĉĀĈ æĈăĎĪĴăăăĜĒăĎĜăzŌĕŕăŔŰăĹŕĉZĎăyĀăyĹă
ăĵĴēăĴĉŌŕĉZĎăZŦăĜzēĹŦŝăĀĈ

ăĪĴĕĝĉăŢĒĕĉZĎăŰŰăĀZĵĴŢĴcollections æĴăăĴŰăyăăZĎăŝĵăŔăăĒĈĉzĎăŕzēsaĉĹŰēŏyăŸŕăăăĈŝē
ăŏĈăŔŕăzēēŏŦăĵăĉzZēĴŦăZĎăĒĈĉzĎĕŏĴĉŏăŝĎăăĜăŔăĉĝŕăĀĈăĴŢăĈĵĴ

```
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)
```

æĈæđĪĵăçŽĎċĪŇăžŔéĪĴăēĀăđ'ĎċŔĒăđ'gēĠŔċŽĎăžŇēŁŻăĹŪăŦŕă■ōīīĴŇăĵăăĪĴăēĵăĴçŦĪ
 numpy æĹăĹĪŪăĀĈăĴŇăēĈīīĴŇăĵăăŔŕăžēăŕĒăŷĂăŷĹăžŇēŁŻăĹŪăŦŕă■ōēŕăŔŪăĹŕăŷĂăŷĹçžŦăđĎăŇŪăŦŕç

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

æĪĴăŔŌăŔŔăŷĂçĈīīĴŇăēĈæđĪĵăæĪĴăēĀăžŌăűşçŦēçŽĎăŪĠăžŭăăĵăĵŔ(æĈăŽăĴĹĠĠăĵăĵŔīīĴŇăĹŔăçĂăŦēçĴĪŇĴŇPythonăŸŕăŷ■ăŸŕăűşçžŔăŔŔăĴăžĒçŌŕă■ŸçŽĎăĹăĹĪŪăĀĈăŽăăŷăžă■ăĹŕăŷĠăŷă■ăĴ

8.12 6.12 èŕăŕăŔŪăŦŇăēŪăŦŇăŔŕăŔŸéŦăžŇēŁŻăĹŪăŦŕă■ō

éŪōēćŸ

ăĵăæĪĴăēĀăŕăŕăŔŪăŦŇăēŔŇăŦŇăēŪăĹŪăĀăŔŕăŔŸéŦăēŕăĴŦéŦăŔŔăĴŦĎăđ'■ăĪĈăžŇēŁŻăĹŪăăĵăĵŔ

èğĉăĒşăŪăŕăĹ

struct æĹăĹĪŪăŔŕēĉŋĴŦăĪēċĵŪĉăĀ/èğĉĉăĀăĠăăžŌăĹ'ĂăĪĴĴşşăđŇĴŽĎăžŇēŁŻăĹŪăŦŦăŦŕă■ōçž
 æĪēăĴĉđ'žăŷĂăŷĹçžĎăĹŔăŷĂçşžăĹŪăđ'ŽēĴăŷăĴçŽĎĈçžŽĎéŦăŔŔăĴŦīīĴŽ

```
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) ],
]
```

çŌŕăĪĴăĀĠēŕăçĴăŷĹăŦŕă■ōēĉŋċĵŪĉăĀăĹŕăŷĂăŷĹăžēăŷŇăĹŪăđŦŕ'éĈăĴăĴăĠŇçŽĎăžŇēŁŻăĹŪăŪĠăžă

Byte	Type	Description
0	int	æŪĠăžăžăžçĉăĀīīĴĹ0x1234īīĴŇăŕŔĉŋŕīīĴŦ'
4	double	x çŽĎăĪĴăăŕŔăăĴīīĴĹăŕŔĉŋŕīīĴŦ'
12	double	y çŽĎăĪĴăăŕŔăăĴīīĴĹăŕŔĉŋŕīīĴŦ'
20	double	x çŽĎăĪĴăăđ'ğăăĴīīĴĹăŕŔĉŋŕīīĴŦ'

(continues on next page)

(continued from previous page)

```
+-----+-----+-----+
|28      | double | y çŽĎæIJĀād' ġāĀijiiĵĹāřŘçnrĳijL' |
+-----+-----+-----+
|36      | int     | äÿL' èğŠā;ćæŦřéĠRĳijĹāřŘçnrĳijL' |
+-----+-----+-----+
```

çť ġeũşçİĀād' t' éĆĹæŸřäÿĀçşzāĹŮçŽĎād' Žè;żā;ćèõřā;ŦĳijŇçijŮčāAæāijāijRāęĆäÿŇĳijŽ

```
+-----+-----+-----+
|Byte   | Type   | Description |
+=====+=====+=====+
|0       | int    | èõřā;ŦéŦĹāžęĳijĴNā■ŮèĹĆĳijL' |
|→      |        |             |
+-----+-----+-----+
|4-N     | Points | (X,Y) āĪřæāĠĳijŇāžēæŦõçĆzæŦřèāĴçd' ž |
|→      |        |             |
+-----+-----+-----+
```

äÿžāžĒāĒŽèĴZæāũçŽĎæŮĠžāũĳijŇā;āāŦřāžēā;ĴçŦĹāęĆäÿŇçŽĎPythonāžçčāĀĳijŽ

```
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))
```

ārĒæŦřæ■ćerzārŮāŽĎæĴçŽĎæŮūāĀŽĳijŇāŦřāžēāĴĴçŦĹāĠ;æŦř struct.unpack()
ĳijŇāžçčāĀā;ĴçŽÿāĳijĳijŇāşžæĴĴnāřsæŸřäÿĹéĴćāĒŽæş■ā;ĴçŽĎéĀĒāžŦāĀĆāęĆäÿŇĳijŽ

```
def read_polys(filename):
    with open(filename, 'rb') as f:
        # Read the header
        header = f.read(40)
        file_code, min_x, min_y, max_x, max_y, num_polys = \
```

(continues on next page)

(continued from previous page)

```
struct.unpack('<iddddi', header)
polys = []
for n in range(num_polys):
    pbytes, = struct.unpack('<i', f.read(4))
    poly = []
    for m in range(pbytes // 16):
        pt = struct.unpack('<dd', f.read(16))
        poly.append(pt)
    polys.append(poly)
return polys
```

ār;çōæŁŻäylāzčçāAāRřāzēāuēā;IJiijNā;EæYřéGŇéIcæuūāĪCāžEā;Ĺāđ'ŽērZāRŮāĀAèġcāNĚæTřæ■ōçž
éCċæIJāĤĚ■āzšād'łçZĀæĪCāžEçCžāĀCāZāæ■đ'ā;ĹæYčDūāzTèrēæIJĹāRēāyĀçġ■èġcāEşæŮzæşTāRřāzēçō

āIJāIJnārRèŁCæŌēāyNāelēçŽĎéCĹāĹEiijNāĹSāijŽéĀRæ■ēāijTčđ'žāyĀāylāēZt'āĹāāijYçġĀçŽĎēġcæ
çŽōāāGæYřāRřāzēçžZčĹNāžRāSŸæRŘā;ŽāyĀāylēnYčžġçŽĎæŮGāzūāāijāijRāNŮæŮzæşTrijNāzūçōĀāNŮ
æIJnārRèŁCæŌēāyNāelēçŽĎéCĹāĹEāzčçāAāžTèrēæYřæTt'æIJnāžēāy■æIJĀāđ'■æĪCæIJāénYčžġçŽĎā;Nā■
āyĀāōZēçAāžTçžEçŽĎéYĒērZæĹSāžnçŽĎéōlēōžéCĹāĹEiijNāRēāđ'ŮāzşēçAāRĈèĀČāyNāĒūāzŮçnāēŁCāE

ēçŮāĒĹiijNā;ŞērZāRŮā■ŮēŁCæTřæ■ōçŽĎæŮūāĀZiijNéĀŽāyāIJāŮGāzūāijĀāġNéCĹāĹEāijŽāNĚāR
ār;çōāstructæĹāāĪŮāRřāzēēġcāNĚēçZāžZæTřæ■ōāĹrāyĀāylāĒČçžĎāy■āŌzriijNāRēāđ'ŮāyĀçġ■ēāłçđ'žēçZçġ
ārśāČRāyNéIcēçZæāūiijŽ

```
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)
```

ēçŽéGŇæĹSāžnā;łçTĹāzEāyĀāylāēRŘēřāZĹāĪēēāłçđ'žæřRāylçžŞæđDā■ŮæōtrijNærRāylāēRŘēřāZĹāN
ā■YāCĹāIJāĤĚēCĹçŽĎāĤĚā■YçijŞāEşāy■āĀCāIJĹ __get__() æŮzæşTāy■iijNstruct.
unpack_from() āĢ;æTřēçnçTĹāĪēāzŌçijŞāEşāy■ēġcāNĚāyĀāylāĀijriijNçIJAāŌzāžEēcĹāđ'ŮçŽĎāĹEçŁ'Č

Structure çşzārşæYřāyĀāylāşžçāĀçşzriijNæŌēāRŮā■ŮēŁCæTřæ■ōāzūā■YāCĹāIJāĤĚēCĹçŽĎāĤĚā
StructField æRŘēřāZĹā;łçTĹāĀČ ēçŽéGŇā;łçTĹāzE memoryview()

iiijÑæĹŚāznāijŽāIJĹāŔŔŌēīcēfēçzEēðšēğçāōČæŸfçŦĹæĹēāzšāŸŽçŽĐāĂĆ

ä;ŁçŦĹēŦŽāyĹāzççăĀiiijÑä;ăçŎŔāIJĹāŕſēČ;ăōŽāzĹ'äyĀäyĹénŸāſĆæñăçŽĐçzŞæđĐāŕzēsăæĹēēāĹçđ'žāyĹēĹ

```
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)
```

äyÑēīcçŽĐä;Ŧā■ŔāĹŦçŦĹēŦŽāyĹçşzæĹēēŕzāŔŪāzŦāĹ■æĹŚāznāEŽāĔĔççŽĐād'Žē;žā;ćæŦŕæ■ōçŽĐād'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
>>>
```

ēŦŽāyĹā;ĹæIJĹ'ēūçiiijÑäy■ēŦĞēŦŽçğ■æŪzāijŔēŦŸæŸŕæIJĹ'äyĀāzŽçČēāzžçŽĐāIJŕæŪzāĂĆēēŪāĔĹiiij
ä;EæŸŕēŦŽāyĹāzççăĀēŦŸæŸŕæIJĹçČzēĞČēČŦiiijÑēŦŸēIJĀēçĀä;ŁçŦĹēĀĔæŦŦĠăōŽā;Ĺād'ŽāžŦāsČçŽĐçzE
StructFieldiiijÑæŦŦĠăōŽāĀŔçğzēĞŔç■Ĺ)ăĂĆ āŔēād'ŪiiijÑēŦŦāŽđçŽĐçzŞæđIJçşzāŦŦæăūçăōăōđāyĀă

ăzzä;ŦæŪūăĂŽāŔĹēçĀä;ăēĀĞāĹŕäžEāČŔēŦŽæăūăEŪä;ŽçŽĐçşzāōŽāzĹ'iiijÑä;ăāžŦēŕēēĂČēŽŚāyŦä;Łç
ăĔČçşzæIJĹ'äyĀäyĹçĹzæĀğāŕſæŸŕăōČēČ;ād'şēćŋçŦĹæĹēāŋăĔĔēōyād'Žä;ŎāsČçŽĐăōđçŎŔçzEĔČiiijŦāzŎ
äyÑēīcæĹŚæĹēäy;äyĹā;Ŧā■ŔiiijÑä;ŁçŦĹāĔČçşzçĹ■ă;ŏæŦzéĀăäyŦæĹŚāznçŽĐ Structure
çşziiijŽ

```
class StructureMeta(type):
    '''
    Metaclass that automatically creates StructField descriptors
    '''
    def __init__(self, clsname, bases, clsdict):
        fields = getattr(self, '_fields_', [])
        byte_order = ''
        offset = 0
        for format, fieldname in fields:
            if format.startswith(('<', '>', '!', '@')):
                byte_order = format[0]
                format = format[1:]
```

(continues on next page)

(continued from previous page)

```
        format = byte_order + format
        setattr(self, fieldname, StructField(format, offset))
        offset += struct.calcsize(format)
        setattr(self, 'struct_size', offset)

class Structure(metaclass=StructureMeta):
    def __init__(self, bytedata):
        self._buffer = bytedata

    @classmethod
    def from_file(cls, f):
        return cls(f.read(cls.struct_size))
```

ä;fcTlæŮrcŽD Structure çszijNä;ääRräzæäCRäyNélcèfZæäüäöZäzL'äyÄäylçzŞædDrijŽ

```
class PolyHeader(Structure):
    _fields_ = [
        ('<i', 'file_code'),
        ('d', 'min_x'),
        ('d', 'min_y'),
        ('d', 'max_x'),
        ('d', 'max_y'),
        ('i', 'num_polys')
    ]
```

æ■čæČä;äæL'ÄègAijNèfZæäüäEŽärşöÄ■Täd'ŽäzEäÄCæLSäznæüäLäçŽDçszæŮzæşT
from_file() èöl'æLSäznäIJläy■éIJÄèeAçşééAŞäzzä;TæTträæ■öçŽDäd'gärRäŞNçzŞædDçŽDæČĚäEtäyNä

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

äyÄæŮëä;ääijÄägNä;fcTlæzEäÄCçszijNä;äärsäRräzèèöl'äöČäRŸä;ŮæŽt'äLäæŽzèČ;äÄCä;NäeCijNä
äyNélcæYřärzäL■élcäEČçszçŽDäyÄäyläRçŽDæTžèfZijNæRRä;ŽäzEäyÄäylæŮrcŽDè;ĚäL'æRRèfřäZlæ

```
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)

```

aIJlëfZæôzæççäAäy■iijNNestedStruct æRRëfäZléçncTlæleäRäaLäaRëad'ÜäyÄäyIäöZäZL'äIJlæš
 äöCéÄZëfGärEäÖšägNäEëä■YcijSäEšëfZëaNäLGçL'GæS■äIJäRÖäöäJNäNÜçzZäöZçZDçzSædDçszädn
 æL'ÄäzëëfZçg■äLGçL'GæS■äIJäy■äijZäijTäRSäzzäJçZŽDëciäd'ÜçZŽDäEëä■Yäd'■äLüäÄCçZyäR■iijNäöC
 äRëad'ÜrijNäyZäZëYsä■céG■äd'■äöäJNäNÜrijNéÄZëfGäJçTlæšN8.10ärRëLÇäRñæäüçZDæLÄæIJrijN
 äJçTlæfZäyIäÜçZŽDäföæ■çL'LiijNäJääršäRfäzääCRäyNéIcéfZæäüçijÜäEZiijZ

```

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')
    ]

```

äzd' äžžæČŁëóůčŽDæŸřijŇăőČázšëČ;æŇL'čĚğécĎæIJšçŽDæ■čăyŷăũëă;IJijŇæĹŚăžňăóđéŽĚæŞ■ă;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
>>>

```

ăĹrçŽóăĹ■ăyžæ■ćijŇăyĂăylăđ'ĐçŘĚăóŽéTĚëőřă;TçŽDæăĚăđăũšçzŔăĚŽăë;ăžĚăĂĆă;ĚăŸřăęĆăđĹ
 æřTăęĆijŇăđ'Žë;žă;ćăŮĞăžăăŇĚăŔňăŔŸéTĚçŽDéČĹăĹĚăĂĆ

ăyĂçğ■ăŮžăăĹăŸřăĚžăyĂăylçşzăĹëăłçđ'žă■ŮëĹĆăŤŕăë■őijŇăŔŇăŮăăĚžăyĂăylăũëăĚăăĜ;ăŤŕăëĹ

```

class SizedRecord:
    def __init__(self, bytedata):
        self._buffer = memoryview(bytedata)

    @classmethod
    def from_file(cls, f, size_fmt, includes_size=True):
        sz_nbytes = struct.calcsize(size_fmt)
        sz_bytes = f.read(sz_nbytes)
        sz, = struct.unpack(size_fmt, sz_bytes)
        buf = f.read(sz - includes_size * sz_nbytes)
        return cls(buf)

```

(continues on next page)

(continued from previous page)

```
def iter_as(self, code):
    if isinstance(code, str):
        s = struct.Struct(code)
        for off in range(0, len(self._buffer), s.size):
            yield s.unpack_from(self._buffer, off)
    elif isinstance(code, StructureMeta):
        size = code.struct_size
        for off in range(0, len(self._buffer), size):
            data = self._buffer[off:off+size]
            yield code(data)
```

çşzæŰzæşT SizedRecord.from_file() æŶřäyÄäyġäüëäĖüüijŊçTġæġæzŌäyÄäyġæŰġäzūäy■ēřzä
èŁZäzşæŶřäġLäd'ŽæŰġäzūääijäijRäyçTġçŽDæŰzäijRäĀĆäġIäyžèġŞäĖēijŊăŌĆæŌēäRŰäyÄäyġäNĖäRňä
äRřéĀLçŽD includes_size äRĆæTřæŊĠăŌŽäžĖä■ŰēŁĆæTřæŶřäRēäNĖäRňäd't'ēČġäd'ġärRäĀĆ
äyNēġäŶřäyÄäyġäġNä■RæTžä;äæĀŌæäüä;ŁçTġäzŌäd'Žèġzä;ćæŰġäzūäy■ēřzäRŰä■TçNŋçŽDäd'Žèġzä;ćæ

```
>>> f = open('polys.bin', 'rb')
>>> phead = PolyHeader.from_file(f)
>>> phead.num_polys
3
>>> polydata = [ SizedRecord.from_file(f, '<i')
...               for n in range(phead.num_polys) ]
>>> polydata
[<__main__.SizedRecord object at 0x1006a4d50>,
<__main__.SizedRecord object at 0x1006a4f50>,
<__main__.SizedRecord object at 0x10070da90>]
>>>
```

äRřäzèçIJŊăĠzīijŊSizedRecord äŌdäġŊçŽDäĖĖäŌžèŁŶæşæIJL'ècŋèġçædRäĠzæġæġäĀĆ
äRřäzä;ŁçTġ iter_as() æŰzæşTæġæġ;äġřçŽŌçŽDijŊēŁZäyġæŰzæşTæŌēäRŰäyÄäyġçzŞædĎääijäijRä
Structure çşzäġIäyžèġŞäĖēäĀĆ èŁZæäüä■RäRřäzäġġŁçAġæ't'zçŽDäŌžèġçædRæTřæ■ŌijŊäġNäçĈijŽ

```
>>> for n, poly in enumerate(polydata):
...     print('Polygon', n)
...     for p in poly.iter_as('<dd'):
...         print(p)
...
Polygon 0
(1.0, 2.5)
(3.5, 4.0)
(2.5, 1.5)
Polygon 1
(7.0, 1.2)
(5.1, 3.0)
(0.5, 7.5)
(0.8, 9.0)
Polygon 2
(3.4, 6.3)
```

(continues on next page)

(continued from previous page)

```
(1.2, 0.5)
(4.6, 9.2)
>>>

>>> for n, poly in enumerate(polydata):
...     print('Polygon', n)
...     for p in poly.iter_as(Point):
...         print(p.x, p.y)
...
Polygon 0
1.0 2.5
3.5 4.0
2.5 1.5
Polygon 1
7.0 1.2
5.1 3.0
0.5 7.5
0.8 9.0
Polygon 2
3.4 6.3
1.2 0.5
4.6 9.2
>>>
```

āĖĖL'ĀĖIJL'ēfZāžZčzŠāRĻēṭāēlērījNāyNēlċāYrāyĀäyĭ
āĖĭāTṛčŽDāRēād'ŪāyĀäyĭāfōā■čĖL'ĻijŽ

read_polys()

```
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
```

ëóìèõž

èŁŻäyÄèŁĆàŘŚä;äāsŤčd'žāžÈeōyād'ŽénŸčžġčŽĎčijŮčĹNæŁÄæIJřijŇāŇĚæŇñæŘŘèĤřāŽĹijŇāžüèŁšš
čĎüèĀŇijŇāōČāžñèČ;äyžāžÈāŘŇāyÄäyĹčŁ'žāōŽčŽĎčŽōæĀĠāŁāĀČ

äyŁéİččŽĎāōđčŎřčŽĎäyÄäyĹäyžèèAçŁ'žāŁAæŸřāōČæŸřāšžāžŎæĠšèġčāŇĚčŽĎæĀİæČšāĀČā;šāyÄā
Structure āōđä;ŇèčŇāŁŽāžžæŮüüijŇ__init__() äžĚäžĚāŘĹæŸřāŁŽāžžāyÄäyĹāŮèŁČæŤřæŮčŽĎā
çŁ'žāŁŇčŽĎřijŇèŁŽæŮüāĀŽāžžæšāæIJL'āžžā;ŤčŽĎèġčāŇĚæŁŮèĀĚāĚūāžŮäyŎčžšæđĎčŽyāĚščŽĎæšā;
èŁŽæāüāĀŽčŽĎäyÄäyĹāŁāİJžæŸřā;āāŘřèČ;äžĚäžĚāŘĹāřžāyÄäyĹāŮèŁČèōřā;ŤčŽĎæšŘāyĀāŘŘèČĹāŁĒæ

äyžāžÈāōđčŎřæĠšèġčāŇĚāšŇæŁ'šāŇĚijŇĒİJĀèèAä;ŁçŤĹ
StructField æŘŘèĤřāŽĹčšžāĀČ çŤĹæŁūāIJĹ _fields_
äyāŁŮāĠžæİèçŽĎæřŘāyĹāšđæĀġèČ;äijŽèčŇè;ŇāŇŮæŁŘāyÄäyĹ StructField
æŘŘèĤřāŽĹijŇ āōČāřĒçŽyāĚščžšæđĎæijāijŘçāĀšŇāĀŘçġžāĀijāĤİāŸāŁřāŸāČĹçijšāŸāyāĀČāĒČçš;
StructureMeta āIJĹāđ'ŽāyĹčžšæđĎčšžèčŇāōžāžŁ'æŮüèĠĹāŁāŁŽāžžāžÈèŁŽāžžæŘŘèĤřāŽĹāĀČ
æŁšāžŇā;ŁçŤĹāĒČçšžčŽĎäyÄäyĹäyžèèAāŎšāžāæŸřāōČā;ŁāŁŮçŤĹæŁūéİđäyæŮžā;ŁçŽĎéĀžèŁĠāyÄäyĹæ

StructureMeta çŽĎäyÄäyĹāŁāŁāōæŽčŽĎāIJřæŮžāřšæŸřāōČāijŽāžžāōžāŮèŁČæŤřæŮèāžāžŘāĀ
äžšāřšæŸřèř'ijŇāèČæđIJāžžæĎŘčŽĎāšđæĀġæŇĠāōžāžÈäyÄäyĹāŮèŁČéāžāžŘ(<èāĹčđ'žā;Ŏā;āijŸāĒŁ
æŁŮèĀĒ >èāĹčđ'žénŸā;āijŸāĒŁ)ijŇ éČčāŘŎéİçæŁ'ĀæIJLāŮæōŤčŽĎéāžāžŘèČ;äžèèŁŽāyĹéāžāžŘāyžāĠĒ
æřŤæČřijŇā;āāŘřèČ;æIJLäyÄäyĹæřŤè;Čāđ'æİČčŽĎčžšæđĎijŇāřšāČŘāyŇéİçèŁŽæāüüijŽ

```
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') ]
```

äžŇāŁ'æŁšāžŇæŘŘāŁřèŁĠijŇmemoryview() çŽĎä;ŁçŤĹāŘřāžèäyōāŁ'æŁšāžŇæĀĤāĒāĒĒāŸçŽĹ
ā;šçžšæđĎāŸāIJĹāŇāèŮčŽĎæŮüāĀŽijŇmemoryviews āŘřāžèāŘāāŁāāŘŇāyĀāĒĒāŸçšāyŁāōžā
èŁŽāyĹčŁ'žæĀġæřŤè;ČāŁāōæŽijŇā;ĒæŸřāōČāĒšæšĹčŽĎæŸřāĒĒāŸçšāyōāŁ'æŁšāžŇæŮèŁČæŤřçžĎç
āèČæđIJā;āāIJĹäyÄäyĹāŮèŁČāŮçŇæyšæŁŮāŮèŁČæŤřçžĎäyĹæŁġèāŇāŁĠçŁ'Ġæšā;IJijŇā;äèĀžāyā
èĀŇāĒĒāŸçšāyōāŁ'æŁšāžŇæŮèŁČæŤřçžĎäyĹæŁġèāŇāŁĠçŁ'Ġæšā;IJijŇā;äèĀžāyā

èŁŸæIJL'āŁāđ'ŽçŽyāĒšçŽĎçŇæŁČāŘřāžèäyōāŁ'æŁšāžŇæŁ'āsŤèŁŽèĠŇèóìèõžçŽĎæŮžæāŁāĀČ
āŘČèĀČ8.13āřŘèŁČā;ŁçŤĹæŘŘèĤřāŽĹāđĎāžžāyÄäyĹçšžāđŇçšžçžšāĀČ
8.10āřŘèŁČæIJLæŽřāđ'ŽāĒšāžŎāžüèŁšèōāçŮŮāšđæĀġāĀijçŽĎèóìèõžřijŇāžüāyŤèüşNestedStructæŘŘèĤřā
9.19āřŘèŁČæIJLäyÄäyĹā;ŁçŤĹāĒČçšžæİēāŁİāġŇāŇŮçšžæŁŘāšŸçŽĎä;ŇāŮijŇāšŇ

StructureMeta çşzéİđâÿÿçŽÿäijijãĀĆ PythonçŽĎ ctypes
æžŘčãĀăŔŇæăüăžšăĹLæIJL'èüçijŇăôĈæŔŔăĹŽăžĒăržăôŽăžL'æŦŕæ■ôçzŞæđĎăĀăŦŕæ■ôçzŞæđĎăŦŇăĒŮ

8.13 6.13 æŦŕæ■ôçŽĎçŦŕăĹăäÿŎçzşëôăæŞ■ăĹĹ

éŮóécŸ

ăĵăéIJĀèçĀăđ'ĎçŔĒăÿĂăÿĹăĹăđ'ğçŽĎæŦŕæ■ôéŽĒăžüéIJĀèçĀèôăçôŮæŦŕæ■ôæĂăăŇăĒŮăĒŮăžŮçz

èğčăĒşæŮzæăĹ

ăržăžŎăžzăĵŦæŮL'ăŔĹăĹŦçzşëôăăĀăæŮüéŮŦ'ăžŔăĹŮăžčăŔĹăĒŮăžŮçzŽÿăĒşæĹĂæIJçŽĎæŦŕæ■ôăĹĒ
PandasăžŞăăĀĆ

ăÿžăžĒèŮŦ'ăĵăăĒĹăĵŞéŇăÿŇijŇăÿŇéİăĒŸŕăÿĂăÿĹăĵçŦĪPandasăİăăĹĒæđŔĒĹăĹăăŞăăşŎăÿĈçŽĎ
èĂĀéijăăŇăŦŮéĵçşzăĹĹçLŦ'æŦŕæ■ôăžŞ çŽĎăĹŇă■ŔăĀĆăIJăĹŦŦăĒçĒçŦçŦŦŮĜçŇăçŽĎæŮŮăĂŽijŇèçz

```
>>> import pandas

>>> # Read a CSV file, skipping last line
>>> rats = pandas.read_csv('rats.csv', skip_footer=1)
>>> rats
<class 'pandas.core.frame.DataFrame'>
Int64Index: 74055 entries, 0 to 74054
Data columns:
Creation Date 74055 non-null values
Status 74055 non-null values
Completion Date 72154 non-null values
Service Request Number 74055 non-null values
Type of Service Request 74055 non-null values
Number of Premises Baited 65804 non-null values
Number of Premises with Garbage 65600 non-null values
Number of Premises with Rats 65752 non-null values
Current Activity 66041 non-null values
Most Recent Action 66023 non-null values
Street Address 74055 non-null values
ZIP Code 73584 non-null values
X Coordinate 74043 non-null values
Y Coordinate 74043 non-null values
Ward 74044 non-null values
Police District 74044 non-null values
Community Area 74044 non-null values
Latitude 74043 non-null values
Longitude 74043 non-null values
Location 74043 non-null values
dtypes: float64(11), object(9)

>>> # Investigate range of values for a certain field
```

(continues on next page)

(continued from previous page)

```
>>> rats['Current Activity'].unique()
array([nan, Dispatch Crew, Request Sanitation Inspector],
      dtype=object)
>>> # Filter the data
>>> crew_dispatched = rats[rats['Current Activity'] == 'Dispatch_
↳Crew']
>>> len(crew_dispatched)
65676
>>>

>>> # Find 10 most rat-infested ZIP codes in Chicago
>>> crew_dispatched['ZIP Code'].value_counts()[:10]
60647 3837
60618 3530
60614 3284
60629 3251
60636 2801
60657 2465
60641 2238
60609 2206
60651 2152
60632 2071
>>>

>>> # Group by completion date
>>> dates = crew_dispatched.groupby('Completion Date')
<pandas.core.groupby.DataFrameGroupBy object at 0x10d0a2a10>
>>> len(dates)
472
>>>

>>> # Determine counts on each day
>>> date_counts = dates.size()
>>> date_counts[0:10]
Completion Date
01/03/2011 4
01/03/2012 125
01/04/2011 54
01/04/2012 38
01/05/2011 78
01/05/2012 100
01/06/2011 100
01/06/2012 58
01/07/2011 1
01/09/2012 12
>>>

>>> # Sort the counts
>>> date_counts.sort()
```

(continues on next page)

(continued from previous page)

```
>>> date_counts[-10:]
Completion Date
10/12/2012 313
10/21/2011 314
09/20/2011 316
10/26/2011 319
02/22/2011 325
10/26/2012 333
03/17/2011 336
10/13/2011 378
10/14/2011 391
10/07/2011 457
>>>
```

āŪřijŇčIJŇæūā■Ř2011āzt' 10æIJL7æŮěāržeĀAėjäāznæİëert' æŸřäylā;ĹāŁŻćŇčŽDæŮěā■ŘāŤLijA

ěőİěőŽ

PandasæŸřäyĀäylæŇæIJL'āĹĹād'ŽčL'záæĀğčŽDād'ğādŇāĜ;æŤřāžŠřijŇæĹSāIJĹēŁŽéĜŇäy■āŖřèČ;āzn
ä;EæŸřāŖĹèçAä;ăéİAèçAāŌzāĹEæđŖād'ğādŇæŤřæ■őéŽĚāŖĹāĀāřzæŤřæ■őāĹEçzDāĀAèőaçőŮāŖDçğ■

9 çňňäyČçňäijŽāĜ;æŤř

ä;ŁçŤĪ def èř■āŖěāőŽāzĹ'āĜ;æŤřæŸřæĹ'ĀæIJL'çĹŇāžŖčŽDāšžçāĀāĀĆ
æIJŇčňāçŽDçŽōæāĜæŸřèőšèğçäyĀāžŽæZt'āĹāénŸçžğāŠŇäy■āyŸèğAçŽDāĜ;æŤřāőŽāzĹ'äyŌä;ŁçŤĪæĹāijF
æŮĹ'āŖĹāĹŖčŽDāĚāőzāŇĚæŇňézŸēōđ'āŖČæŤřāĀāžzæĎŖæŤřéĜŖāŖČæŤřāĀāijzāĹūāĚšéŤōā■ŮāŖČ
āŖēād'ŮřijŇäyĀāžŽénŸçžğčŽDæŌğāĹūāŤAāŠŇāĹĹ'çŤĪāŽđērČāĜ;æŤřāijăéĀŠæŤřæ■őçŽDæĹĀæIJřāIJĹēŁŽ

Contents:

9.1 7.1 āŖŖæŌěāŖŮāzzæĎŖæŤřéĜŖāŖČæŤřçŽDāĜ;æŤř

éŮőécŸ

ä;ăæČşæđĎéĀäyĀäylāŖŖæŌěāŖŮāzzæĎŖæŤřéĜŖāŖČæŤřçŽDāĜ;æŤřāĀĆ

èğçāĚşæŮzæāĹ

äyžāžĚèČ;èőĹ'äyĀäylāĜ;æŤřæŌěāŖŮāzzæĎŖæŤřéĜŖçŽDä;■ç;őāŖČæŤřijŇāŖřāzēä;ŁçŤĪäyĀäyĪ*āŖČ

```
def avg(first, *rest):
    return (first + sum(rest)) / (1 + len(rest))
```

(continues on next page)

(continued from previous page)

```
# Sample use
avg(1, 2) # 1.5
avg(1, 2, 3, 4) # 2.5
```

āĲĲēĲZāyĲā;Nā■Rāy■ĲĲNrestāYřĲTśāL'ĀāĲĲ'āĲŪāzŪā;■Ĳ;ōāRĲāTřĲzĲāĲRĲĲZĲāĲĲĲzĲāĲĲĲDūāRĲ
āyžāZĲāŌēāRŪāzzāĲRāTřēĲRĲĲZĲāĲĲēTōā■ŪāRĲāTřĲĲNā;ĲĲTĲāyĲāyĲāzē**āĲĲāĲ't ĲZĲāRĲāTřā

```
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>')
```

āĲĲēĲZēĲNĲĲNattřāYřāyĲāyĲāNĲāRĲāL'ĀāĲĲ'ēĲnāĲāāĲēēĲZāĲēĲZĲāĲĲēTōā■ŪāRĲāTřĲZĲā■ŪāĲ
āēĲāĲĲā;āēĲYāyNāĲZā\$RāyĲāĲā;āTřēĲ;āRĲāŪāŌēāRŪāzzāĲRāTřēĲRĲĲZĲā;■Ĳ;ōāRĲāTřāŠNā

```
def anyargs(*args, **kwargs):
    print(args) # A tuple
    print(kwargs) # A dict
```

ā;ĲĲTĲēĲZāyĲāĲā;āTřāŪŪĲĲNāL'ĀāĲĲ'ā;■Ĳ;ōāRĲāTřāĲZēĲnāTĲāĲřāřāĲĲĲzĲāy■ĲĲNāL'ĀāĲĲ'āĲē

ēōĲēōZ

āyĲāyĲ*āRĲāTřāRĲēĲāĲZĲŌřāĲĲāĲā;āTřāŌZāZL'āy■āĲĲāRŌāyĲāyĲā;■Ĳ;ōāRĲāTřāRŌēĲĲĲNēĲN
**āRĲāTřāRĲēĲāĲZĲŌřāĲĲāĲĲāRŌāyĲāyĲāRĲāTřāĲĲāĲĲ'āyĲĲĲēĲĲāĲēĲāĲRĲĲZĲāYřĲĲNāĲĲ*āRĲā

```
def a(x, *args, y):
    pass

def b(x, *args, y, **kwargs):
    pass
```

ēĲZĲĲ■āRĲāTřāřāYřāĲŚāzñāL'ĀēĲ't ĲZĲāĲzāĲĲāĲēĲēTōā■ŪāRĲāTřĲĲNāĲĲāRŌēĲē7.2āřRēĲĲēĲYāĲ

aijzālŭāĖŝēTŕōā■ŬāRĆæTŕāIJāyĀāzZæZt'énYçžgāIJzāRĹāRŃæūāzŝāĹæIJL'çTĭlāĀĆ
äĹNāēČiijNāōCāznāRřāzēēčŋçTĭlæIēāIJlā;£çTĭl*argsāŠN**kwargsāRĆæTŕā;IJāyžēĹŝāĖēçŽDāĜ;æTŕāy■æRŠ

9.3 7.3 çZŽāĜ;æTŕāRĆæTŕāçđāŁāāĖČāŁæAŕ

éŬōécŸ

ä;āāĖZāē;äzĖāyĀāyĹāĜ;æTŕiijNçDŭāRŎæČŝāyžē£ZāyĹāĜ;æTŕçŽDāRĆæTŕāçđāŁāāyĀāzZēčĭad'ŬçŽD

èğčāĖŝæŬzæāĹ

ä;£çTĭlāĜ;æTŕāRĆæTŕæŝlēğçæYŕāyĀāyĹāĹāē;çŽDāŁđæŝTŕiijNāōČēČ;æRŔçđ'žçĭNāžRāŚYāzTèŕæĀŎ
äĹNāēČiijNāyNéĭcæIJL'āyĀāyĹēčŋæŝlēğçāzĖçŽDāĜ;æTŕiijŽ

```
def add(x:int, y:int) -> int:  
    return x + y
```

pythonèğçēĜĹāZĭlāy■aijZāržē£ZāzZæŝlēğçæūzāŁāāzžā;TçŽDēr■āzĹāĀĆāōCāznāy■aijZēčŋçŝādNāēĀ
çDŭēĀNŕiijNāržāžŎēČčāzŽēYĖēræzŔçāAçŽDāzžæIēēōŝārŝāĹæIJL'āyōāĹl'āTēāĀĆçŋnāyĹæŬzāūēāĖūāŠN

```
>>> help(add)  
Help on function add in module __main__:  
add(x: int, y: int) -> int  
>>>
```

ār;çōā;āāRřāzēä;£çTĭlāzžæDŔçŝzādNçŽDāržēŝaç;ZāĜ;æTŕæūzāŁāāŝlēğç(äĹNāēČæTŕā■ŬiijNā■Ŭçŋæ

èōlēōž

āĜ;æTŕæŝlēğçāŔĭā■YāĆĭāIJlāĜ;æTŕçŽD __annotations__
āŝđæĀğāy■āĀĆāĹNāēČiijŽ

```
>>> add.__annotations__  
{'y': <class 'int'>, 'return': <class 'int'>, 'x': <class 'int'>}
```

ār;çōāæŝlēğççŽDä;£çTĭlæŬzæŝTāŔŕēČ;æIJL'āĹĹād'Žçg■iijNā;ĖæYŕāōCāznçŽDāyžēēAçTĭlēĀTē£YæYŕ:
āZāāyžpythonāzūæŝæIJL'çŝzādNāčŕæYŎiijNéĀŽāyŷæIēēōŝāzĖāzĖēĀŽē£ĜēYĖēræzŔçāAāĹēŽ;çŝēēAŝā
ē£ZæŬūāĀZā;£çTĭlæŝlēğçārŝēČ;çzŽçĭNāžRāŚYæZt'ād'ŽçŽDæRŔçđ'žiiijNēōĭ'āzŬāznāRřāzēæ■ççāōçŽDä;£ç

ārĆēĀĆ9.20ārRēŁĆçŽDāyĀāyĹæZt'āŁāénYçžgçŽDäĹNā■RŕiijNæijTçđ'žāzĖāēČā;TāĹl'çTĭlæŝlēğçæIēāō

9.4 7.4 èŁTāZđād'ŽāyĹāĀijçŽDāĜ;æTŕ

éŬōécŸ

ä;āāyNæIJZæđDēĀāyĀāyĹāRřāzēē£TāZđād'ŽāyĹāĀijçŽDāĜ;æTŕ

èġċàEşæŮzæąŁ

äyžäzEèĈ;èŁTāZđāđ'ŽäyŁāĀijīijŃāĠ;æŤřĈZt' æŎēreturnäyĀäyŁāĒĈĉzĎārsèāŃāžEāĀĈă;ŃāēĈīijŽ

```
>>> def myfun():
...     return 1, 2, 3
...
>>> a, b, c = myfun()
>>> a
1
>>> b
2
>>> c
3
```

èőléőž

ār;ĉōāmyfun()ĉIJŃäyŁāŎžèŁTāZđāžEāđ'ŽäyŁāĀijīijŃāōđēŽĒäyŁæŸrāĒŁāŁZāžzāžEäyĀäyŁāĒĈĉzĎĉDĉD
èŁŽäyŁēr■æşŤĉIJŃäyŁāŎžæŕŤèĴĈāēĠæĀīijŃāōđēŽĒäyŁæŁSāžñā;ĲĉŤĴZĎæŸréĀŮāRūāĲēĉŤşæŁŔäyĀäy

```
>>> a = (1, 2) # With parentheses
>>> a
(1, 2)
>>> b = 1, 2 # Without parentheses
>>> b
(1, 2)
>>>
```

ā;ŞæŁSāžñērĈĉŤĲèŁTāZđäyĀäyŁāĒĈĉzĎĉZĎāĠ;æŤřĈZĎæŮŮāĀŽ
īijŃēĀŽāyŸæŁSāžñāijŽārEĉzŞæđIJĉŤŃāĀijĉzŽāđ'ŽäyŁāŔŸéĠŔīijŃārśāĈŔäyŁēĲĉĉZĎéĈĉæŮāĀĈ
āĒŮāōđēŁŽārśæŸŕ1.1ārŔēŁĈäy■æŁSāžñæL'Āèŕ'ĉŽĎāĒĈĉzĎēġĉāŃĒāĀĈēŁTāZđĉzŞæđIJāžşārŕāžēēŤŃāĀij
èŁŽæŮŮāĀŽèŁŽäyŁāŔŸéĠŔāĀijārśæŸrāĠ;æŤŕēŁTāZđĉZĎéĈĉäyŁāĒĈĉzĎæIJñēžñāžEīijŽ

```
>>> x = myfun()
>>> x
(1, 2, 3)
>>>
```

9.5 7.5 āōŽāžŁ'æIJŁ'éžŸēōđ'āŔĆæŤřĈZĎāĠ;æŤř

éŮōéćŸ

ä;ăæĈşāōŽāžŁ'äyĀäyŁāĠ;æŤŕæŁŮēĀĒæŮzæşŤīijŃāōĈĉZĎäyĀäyŁæŁŮāđ'ŽäyŁāŔĆæŤŕæŸŕāŔŕéĀŁĉŽ

èġċàEşæŨzæąŁ

ăőŽăzŁ'ăyĂăylăIJL'ăRřéĂL'ăRCăTřčŽďăĜĭăTřăYřéİďăyŷċőĂăTřčŽďİĭjŇčŽt'ăŔěăIJăĜĭăTřăőŽăzŁ

```
def spam(a, b=42):  
    print(a, b)  
  
spam(1) # Ok. a=1, b=42  
spam(1, 2) # Ok. a=1, b=2
```

ăĕĆăđIJéžYëöd'ăRřĆăTřăYřăyĂăylăRřăfőăTřčŽďăőžăŽĺăřTăĕĆăyĂăylăĹŨëăĺăĂăĕŽĖăRĹăĹŨëĂĖ

```
# Using a list as a default value  
def spam(a, b=None):  
    if b is None:  
        b = []  
    ...
```

ăĕĆăđIJăĭăăžŭăyăĕĆşăRřăĹZăyĂăylézYëöd'ăĂĭĭĭjŇëĂŇăYřăĆşăžĚăžĚăĭŇëřTăyŇăşŔăylézYëöd'

```
_no_value = object()  
  
def spam(a, b=_no_value):  
    if b is _no_value:  
        print('No b value supplied')  
    ...
```

ăĹŚăžŋăĭŇëřTăyŇëĤZăylăĜĭăTřĭjŽ

```
>>> spam(1)  
No b value supplied  
>>> spam(1, 2) # b = 2  
>>> spam(1, None) # b = None  
>>>
```

ăžTřčĖĕġĆăřşăRřăžăăRřĆőŔăĹřăĭăĕĂşăyĂăylŇoneăĂĭăşŇăyăăĭăăĂĭăyď'ċġăĕĈĖăĖĭăYřăIJL'ăŭőăĹ

ëőléőž

ăőŽăzŁ'ăyĕézYëöd'ăĂĭăRřĆăTřčŽďăĜĭăTřăYřăĹĹčőĂăTřčŽďİĭjŇăĖĸzĭăyăăžĚăžĖăĹăYřëĤZăylĭĭjŇ
ĕĕŨăĖĹĭĭjŇézYëöd'ăRřĆăTřčŽďăĂĭăžăĚăžĖăIJăĜĭăTřăőŽăzŁ'ċŽďăŨŭăĂŽĖĭŇăĂĭăyăĂăŋăăĂĆĕřTĸĹ

```
>>> x = 42  
>>> def spam(a, b=x):  
...     print(a, b)  
...  
>>> spam(1)  
1 42  
>>> x = 23 # Has no effect
```

(continues on next page)

(continued from previous page)

```
>>> spam(1)
1 42
>>>
```

æʃlæĐRāLřā;ŞæĹŚāznæŤzāRŸxçŽĐāĀijçŽĐæŮŭāĂZāržézŸēōd'āRĆæŤřāĀijāžŭæşæIJL'ā;śāŞ■īijNēā
āĒŭāñāīijNēzŸēōd'āRĆæŤřçŽĐāĀijāžŤērēæŸřāy■āRřāRŸçŽĐāržēsāīijNærŤāēĆNoneāĀTrueāĀFal
çL'zāĹŋçŽĐīijNā■ČāyGāy■ēēAāČRāyNēĹcēfZæāŭāEŽāzççāĀīijŽ

```
def spam(a, b=[]): # NO!
    ...
```

āēČæđIJā;ăēfZāzĹāAŽāžEīijNā;ŞézŸēōd'āĀijāIJlāĒŭāzŮāIJræŮžēcŋāfōæŤzāRŌā;āārEāijŽēAĠāĹrāŘ

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

ēfŽçġ■çzŞæđIJāžŤērēāy■æŸřā;ăæČşēēAçŽĐāĀČāyžāzEēAēāĒ■ēfŽçġ■æČĒāEŤçŽĐāRŚçŤşīijNæIJĀā
çĐŭāRŌāIJlāG;æŤřēĠNēĹcæçĀæşēāōČīijNāL■ēĹcçŽĐā;Nā■RāřsæŸřēfZæāŭāAŽçŽĐāĀĆ

āIJlætŤNērŤNoneāĀijæŮŭā;ŤçŤĹisæŞ■ā;IJçñæŸřā;ĹéĠ■ēēAçŽĐīijNāzşæŸřēfŽçġ■æŮžæāĹçŽĐāĒş
æIJLæŮŭāĂZād'gāōŭāīijŽçĹrāyNāyNēĹcēfZæāŭçŽĐēŤŽērīijŽ

```
def spam(a, b=None):
    if not b: # NO! Use 'b is None' instead
        b = []
    ...
```

ēfZāzĹāEŽçŽĐēŮōēcŸāIJlāžŌār;çōāNoneāĀijçāōāōđæŸřēcŋā;ŞæĹRFalseīijN
ā;EæŸřēfŸæIJLāĒŭāzŮçŽĐāržēsā(ærŤāēĆŤŤāžēāyžŌçŽĐā■ŮçñæyşāĀĀāĹŮēāĹāĀĀāĒČçzĐāĀĀā■ŮāĒy
āZāæ■d'īijNāyĹēĹcçŽĐāzççāĀāīijŽērřārEāyĀāzZāĒŭāzŮē;ŞāĒēāzşā;ŞæĹRæŸřæşæIJL'ē;ŞāĒēāĀĆærŤāēĆ

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

æIJĀāŔŌäyÄäylēŪōécŸæŕŦē;Ĉā;ōāēŽiijŊēĈĉāŕsæŸŕäyÄäylāĜ;æŦŕēIJĀēēAætŊŕŦæŖäyſāŔŕēĀL'āŔēŹæŪūāĀŽēIJĀēēAārŔāſĈĈŽĎæŸŕä;äāy■ēĈ;çŦſæŖäylēzŸēōd'āĀijæŕŦāēĈNoneāĀA
 0æŬŪēĀĒFalseāĀijæſēæŦŊŕŦçŦſæŬæŔŔä;ŽçŽĎāĀij(āŽäyžēſŽāžŽāĀijēĈ;æŸŕāŔŬæŖçŽĎāĀijſijŊæŸ
 āŽāæ■d'ſijŊä;āēIJĀēēAāĒūāzŪçŽĎēĝĉāEŖæŬzæāſLāžEāĀĈ

äyžāzEēĝĉāEŖæſŽäylēŪōécŸſijŊä;āāŔŕäzēāſŽāžžäyÄäylçŊſijŊæŸŭāāžŊçŽĎçĝAæIJL'āržēſāāōdä;Ŋſij
 āIJĀĜ;æŦŕēŊŒēſijŊä;āāŔŕäzēēĀŽēſĜæĉĀæŖēēĉſijæēĀŖāŔæŦŕāĀijēūŖēſŽäylāōdä;ŊæŸŕāŔēäyÄäyā
 ēſŽēŊŊçŽĎæĀſēſæŸŕçŦſæŬäy■āŔŕēĈ;āŌžāijæēĀŖēſŽäyl_no_valueāōdä;Ŋä;IJäyžē;ŖāĒēāĀĈ
 āŽāæ■d'ſijŊēſŽēŊŒēĀŽēſĜæĉĀæŖēēſŽäylāĀijārſēĈ;çāōāōŽæŖäyſāŔæŦŕæŸŕāŔēēĉſijæēĀŖēſŽæſēāzſ

ēſŽēŊŒāŕž object() çŽĎä;ſçŦſçIJŊäyſāŌžæIJL'çĈžäy■ād'ſäyſēĝAāĀĈobject
 æŸŕpythonäy■æŬŪſçſçŽĎāſçſzāĀĈ ä;āāŔŕäzēāſŽāžž object
 çſççŽĎāōdä;ŊſijŊä;EæŸŕēſŽāžŽāōdä;ŊæŖāžÄāžſāōdēŽēĈŦſād'ŊſijŊāŽäyžāōĈāžūæŖæIJL'āžžä;ŦæIJL'
 āžŖæŖæ■ō(āŽäyžāōĈæŖæIJL'āžžä;ŦçŽĎāōdä;Ŋā■ŪāĒŸſijŊä;āçŦŽēĜſēĈ;äy■ēĈ;
 ä;āāŦŕäyĀēĈ;āĀŽçŽĎāŕſæŸŕæſŦŕāŔſijŊæŸĝāĀĈēſŽäylāſŽāē;çſēāŔŬæſſçŽĎēēAæſĈſijŊāŽäyžæſ

9.6 7.6 āŌŽāžſāŊŒāŔæŬāſEēĀŦāĜ;æŦŕ

ēŪōécŸ

ä;āæĈſäyž sort() æŖ■ä;IJāſŽāžžäyÄäylā;ſçŖçŽĎāŽēŕĈāĜ;æŦſijŊä;Eāſſäy■æĈſçŦſ
 def āŌžāEŽäyÄäylā■ŦēāŊāĜ;æŦſijŊēĀŊæŸŕäyſæIJŽēĀŽēſĜæŖäyſāſſæ■æŪāijŔäzēāEēĀŦæŪžāij

ēĝĉāEŖæŬzæāſ

ā;ŖäyÄäžŽāĜ;æŦŕä;ſçſā■ŦſijŊāžĒäžĒāſſæŸŕēōāçſŪäyÄäylēāſē;ä;āijŔçŽĎāĀijçŽĎæŪūāĀŽſijŊāŕſ

```
>>> add = lambda x, y: x + y
>>> add(2, 3)
5
>>> add('hello', 'world')
'helloworld'
>>>
```

ēſŽēŊŒä;ſçŦſçŽĎλaddæāſē;ä;āijŔēūŖäyſſēſççŽĎæŦſæſIJæŸŕäyÄäyūçŽĎſijŽ

```
>>> def add(x, y):
...     return x + y
...
>>> add(2, 3)
5
>>>
```

λaddæāſē;ä;āijŔāŒŸāſſççŽĎä;ſçŦſſāIJæŖæŸŕæŖāŖāſſæŬāſŦŕæ■ōreduceç■ſſijŽ

```
>>> names = ['David Beazley', 'Brian Jones',
...         'Raymond Hettinger', 'Ned Batchelder']
>>> sorted(names, key=lambda name: name.split()[-1].lower())
```

(continues on next page)

(continued from previous page)

```
[ 'Ned Batchelder', 'David Beazley', 'Raymond Hettinger', 'Brian
↳ Jones' ]
>>>
```

èóíèőž

är;çøλλambdaèałè;ł;ąijRāĖĖèőyā;āāőžāzL'çőĀā■TāG;æTřijNā;EæYřāőČžDā;ŁçTłæYřæIJL'éŽŘāŁŮç
ā;āāRłèČ;æNĜāőŽā■Tāyłèałè;ł;ąijRřijNāőČžDāĀijāřsæYřæIJĀāRŮčŽDèŁTāŽđāĀijāĀčžšāřsæYřèřt'äy■è
āNĖæNñāđ'Žäyłèř■āRēāĀAæĪāzūèałè;ł;ąijRāĀAèŁ■āzčāzēāRŁāijČāyŷāđ'ĐçŘEç■L'ç■L'āĀč

ā;āāRřāzēāy■ā;ŁçTłλλambdaèałè;ł;ąijRāřsèČ;çijŮāĖŽāđ'ģéČĹāŁEpythoňāzččāĀāĀč
ā;EæYřijNā;ŠæIJL'āžžçijŮāĖŽāđ'ģéĜRèőačőŮèałè;ł;ąijRāĀijčŽDç\$■āřRāG;æTřæŁŮèĀĖēIJĀèēAçTłæŁūā
ā;āāřsāijŽçIJNāŁřλλambdaèałè;ł;ąijRčŽDèžná;šāzĖāĀč

9.7 7.7 āNĚāŘ■āG;æTřæ■TèŮāŘYéĜRāĀij

éŮóécŸ

ä;āçTłλλambdaāőžāzL'āžĖāyĀāyłāNĚāŘ■āG;æTřijNāzūāČšāIJĹāőžāzL'æŮūæ■TèŮāŁřæšŘāžŽāŘYéĜ

èğčāEşæŮzæał

āĖŁçIJNāyNāyNéĹčāzčçāAçŽDæTłŁæđIJijŽ

```
>>> x = 10
>>> a = lambda y: x + y
>>> x = 20
>>> b = lambda y: x + y
>>>
```

çŮřāIJĹāŁšéŮőā;ārijNā(10)āŠNĚb(10)èŁTāŽđçŽDçzŠæđIJæYřāzĀāzŁiijšāēČæđIJā;āèőđ'äyžçzŠæđIJæY

```
>>> a(10)
30
>>> b(10)
30
>>>
```

èŁŽāĖŮāy■çŽDāčēāēŽāIJĹāžŮλλambdaèałè;ł;ąijRāy■çŽDxæYřāyĀāyłèĜłçTšāŘYéĜRrijN
āIJĹèŁŘēāNæŮūçzŠāőŽāĀijrijNèĀNāy■æYřāőžāzL'æŮūāřsçzŠāőžrijNèŁŽèŮšāG;æTřçŽDézYèőđ'āĀijāRČā
āŽāæ■đ'rijNāIJĹèřČçTłèŁŽāyłλλambdaèałè;ł;ąijRčŽDæŮūāĀŽrijNççŽDāĀijæYřæŁģēāNæŮūçŽDāĀijāĀčā;N

```
>>> x = 15
>>> a(10)
25
```

(continues on next page)

(continued from previous page)

```
>>> x = 3
>>> a(10)
13
>>>
```

æċædIJä;äæĈşèŦ'æşŘäyġăŇĤăŘ■ăĜ;æŦřăIJġăŮŽăZL'æŮŭăřśæ■ŦèŮŭăĹăĀijġġŇăŦřăzēăřĒċĈăyġăŦĈ

```
>>> x = 10
>>> a = lambda y, x=x: x + y
>>> x = 20
>>> b = lambda y, x=x: x + y
>>> a(10)
20
>>> b(10)
30
>>>
```

èőléőž

ăIJġēĤŽēĜŇăĹŮăĜzæġēĈŽĎēŮŏēĈŸæŸřæŮřæĹŇăĹĹăŏžæŸŞĈĹĸŽĎēŦŽēřġġŇăĪĹ'ăžZæŮřæĹŇăŦřă
æŦĤăĈġġŇăĀŽēĤĜăIJġăŸĀăyġăĹ;ĲŦŦăĹŮăĹŮēăĹæŦŦăřġăŸ■ăĹZăžzăŸĀăyġlambdaēăĹē;ăġġŦăĹŮēăġġŇăŸăŮă

```
>>> funcs = [lambda x: x+n for n in range(5)]
>>> for f in funcs:
...     print(f(0))
...
4
4
4
4
4
4
>>>
```

ăġĒæŸřăŏđēŽĒæŦĹædIJæŸřēĤŘēăŇăŸřŋĈŽĎăĀġăŸžēĤ■ăžĈĈŽĎăIJăăŦŦăŸĀăŸăġăĀĈĈŦŦăĹĹăĹŦă

```
>>> funcs = [lambda x, n=n: x+n for n in range(5)]
>>> for f in funcs:
...     print(f(0))
...
0
1
2
3
4
>>>
```

ēĀŽēĤĜă;ĤĈŦġăĜ;æŦřēzŸēŏđ'ăĀġăŦĈæŦřă;ĈăġġŦġġŇlambdaăĜ;æŦřăIJġăŮŽăZL'æŮŭăřśēĈ;ĈzŦăŮŽăĹăŦă


```
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()
```

äy■ēfGrijŇāAĜēō;ä;äæČšçzŽEchoHandlerācđāŁäāyÄāyłāRřāzēæŎēāRŮāĚūāzŮēĚ■ç;őēĀL'ēāzčŽD
__init__ æŮzæšTāĀCærTāēČrijŽ

```
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)
```

ēfŽāzŁāfōæTzāRŎrijŇāĽSāznāršāy■ēIJĀēēAæY;āijRāIJřāIJITCPServerčšzāy■æūzāŁāāL■çijĀāzEāĀ
ä;EæYřā;āāE■āñāēfRēāŇçlŇāžRāRŎäijZæĽčšzāijijāyŇēĽčŽDēTŽēřrijŽ

```
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
→'
```

āĽIçIJŇētuālēāē;āČRā;ĽēŽ;āfōæ■čēfŽāyĽēTŽēřrijŇēŽd'āzEāfōæTz
socketserver āĽāIŮæžRāzččāAæĽŮēĀĚā;fçTlāšRāžZāēGæĀčŽDæŮzæšTāžŇād'ŮāĀC
ä;EæYřrijŇāēCādIJā;fçTlā partial() āřsēČ;ā;Ľē;žāēĽčŽDēgčāEšāĀTāĀTčzŽāōČāijāēĀŠ
ack āRČæTřçŽDāĀijālēāĽIāgŇāŇŮā■šāRřrijŇāēCāyŇrijŽ

```
from functools import partial
serv = TCPServer(('', 15000), partial(EchoHandler, ack=b'RECEIVED:
→'))
serv.serve_forever()
```

āIJlēfŽāyłā;Ňā■Rāy■rijŇ__init__() æŮzæšTāy■çŽDack-
āRČæTřācřæYŎæŮzāijRçIJŇāyŁāŎzā;ĽāIJL'ēūčrijŇāĚūāōdāršæYřācřæYŎackāyžāyĀāyłāijžāĽūāĚšēTōā■
āĚšāžŎāijžāĽūāĚšēTōā■ŮāRČæTřēŮōēcYæĽSāznāIJ7.2ārRēĽČæĽSāznāūšçzRēōlēōžēfGāžErijŇēržeĀĚāR
ā;Ľād'ŽæŮūāĀŽ partial() ēČ;āōđçŎřçŽDæTĽāđIJrijŇλāēālē;āijRāžšēČ;āōđçŎřāĀCærTāēČ

```

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))

```

èfZæäüâEZázšèĈ;ăôđĉŎřăŔŇæăüçŽĐæŦLæđIĲijŇăy■èfĜçŽyærŦèĂŇăüşăijŽæŸ;ă;ŮærŦè;ĈèĜĈèĈ
 èfZæŮüâĂZă;fçŦĲpartial() âŔřăžæŽt'âŁăçŽt'èğĈçŽĐæłè;ă;ăçŽĐæĐŔăŽ; (çžZæšŔăžZăŔĈæŦřécĐ

9.9 7.9 âŔĖâ■ŦæŮzæşŦçŽĐçşzè;ŋæ■căyžăĜ;æŦř

éŮóécŸ

ă;ăæIJL'ăyĂăyłéŽđ' __init__() æŮzæşŦăđ'ŮăŔłăôŽăzL'ăžĖăyĂăyłæŮzæşŦçŽĐçşzăĂCăyžăžĖçóĂ

èğĉăĖşæŮzæąŁ

âđ'ğăđ'ŽæŦřæĈĖăĖŦăyŇĲijŇăŔřăžæă;fçŦĲéŮ■ăŇĖæłĖăŔĖă■ŦăyłæŮzæşŦçŽĐçşzè;ŋæ■căĖŁŔăĜ;æŦřăĂ
 äy;ăyłă;Ňă■ŔĲijŇăyŇéłĉđ'žă;Ňăy■çŽĐçşzăĖĂĖőyă;fçŦĲĖĂĖăžæ■őăşŔăyłăłăĖłĖæŮzæąŁăĖĖőŮăŔŮă

```

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='sl1clv'):
    print(line.decode('utf-8'))

```

èfZăyłçşzăŔřăžæèĉăăyĂăyłæŽt'çóĂă■ŦçŽĐăĜ;æŦřăĖĖăžăçăžĖĲijŽ

```

def urltemplate(template):
    def opener(**kwargs):
        return urlopen(template.format_map(kwargs))
    return opener

# Example use
yahoo = urltemplate('http://finance.yahoo.com/d/quotes.csv?s={names}
    ↪ &f={fields}')

```

(continues on next page)

(continued from previous page)

```
for line in yahoo(names='IBM,AAPL,FB', fields='sl1clv'):  
    print(line.decode('utf-8'))
```

èóìèõž

åd'gëČlálĚæČĚãĚtăyŇiijŇă;ăæNëæIJL'äyÄäyġā■TæŮzæşTçşzçŽĐăŎşăZăæŸréIJĀèĕAă■ŸăČlăşŘăžZ
æŕTăĕČiijŇăôŽăžL'UrlTemplateçşzçŽĐăTŕăyĂçŽôçŽĐăŕşæŸŕăĒLăIJlăşŘăyġāIJŕăŮză■ŸăČlălăæĬăĀiijŇă

ă;ĤçTġăyÄäyġāĚĚĕČlăĜ;æTŕăĒĬŮĕĂĚĕŮ■ăNĚçŽĐăŮzæăĤĚĂžăyăiijŽăZt'ăijŸĕŽĚăyĂăžZăĂČçôĂă■
ăŕġăy■ĕĤĜăIJlăĜ;æTŕăĚĚĕČlăyĕăyĤăžĚăyĂăyĤĕčġăđ'ŮçŽĐăŕŸĕĜŕçŎŕăçČăĂČĕŮ■ăNĚăĚşĕTôçL'žçČžăŕşă
ăZăæ■đ'iijŇăIJlăĤsăžŇçŽĐĕğčăĚşæŮzæăĤăy■iijŇopener()ăĜ;æTŕĕôŕă;ŔăžĚ
templateăŔČăTŕçŽĐăĀiijŇăžŮăIJlăŎĕăyŇăĤĕçŽĐĕŕČçTġăy■ă;ĤçTġăôČăĂČ

ăžžă;TæŮŮăĂŽăŕĤĕĕAă;ăçčŕăĤŕĕIJĀèĕAçžZăşŘăyġāĜ;æTŕăçđăĤăĕčġăđ'ŮçŽĐçĤŮăĂăĤăæĤăĤçŽĐĕŮ
çŽăŕŕăŕĚă;ăçŽĐăĜ;æTŕĕ;Ňăæ■čăĤŕăyĂăyĤçşžĕĂŇĕĤĀiijŇĕŮ■ăNĚĕĂžăyăŸŕăyĂçş■æZt'ăĤăçôĂăŕ'Ăăş

9.10 7.10 äyĕĕčġăđ'ŮçĤŮăĂăĤăæĤăĤçŽĐăŽđĕŕČăĜ;æTŕ

éŮĕĕčŸ

ă;ăçŽĐăžçčăĂăy■ĕIJĀèĕAă;ĤĕŮăĤŕăžđĕŕČăĜ;æTŕçŽĐă;ĤçTġ(æŕTăĕČăžŇăžŮăđ'ĐçŔĚăŽġăĂăç■L'ă;Ě
ăžŮăyTă;ăĕŸĖĤĀĕĕAĕĕŕ'ăŽđĕŕČăĜ;æTŕăNĕæIJL'ĕčġăđ'ŮçŽĐçĤŮăĂăĀiijŇăžĕă;ĤăIJlăôČçŽĐăĚĚĕČlă

ĕğčăĚşæŮzæăĤ

ĕĤăyĂăŕŔĕĤăyžĕĕAĕĕŕĕõžçŽĐăŸŕĕČăžZăĜžçŎŕăIJlă;Ĥăđ'ŽăĜ;æTŕăžşăşŇăăĚăđŮăy■çŽĐăŽđĕŕ
ăyžăžĚăiijTçđ'žăyŎăŕŇĕŕTŷiijŇăĤsăžŇăĒĤăôŽăžL'ăĕČăyŇăyĂăyĤĖIJĀèĕAĕŕČçTġăŽđĕŕČăĜ;æTŕçŽĐăĜ;æTŕ

```
def apply_async(func, args, *, callback):  
    # Compute the result  
    result = func(*args)  
  
    # Invoke the callback with the result  
    callback(result)
```

ăôĕĕŽĚăyĤiijŇĕĤăŽăĕŕăžçčăĂăŕŕăžĕăĂžăžă;TæZt'ĕŇŸçžğçŽĐăđ'ĐçŔĚiijŇăNĚăNŇçžĤçĤŇăĂăĕĤçĤ
ăĤsăžŇăžĚăžĒăŕĤĖIJĀèĕAăĚşăşĤăžđĕŕČăĜ;æTŕçŽĐĕŕČçTġăĂăyŇĖĤăŸŕăyĂăyĤăiijTçđ'žăĂŎăăŮă;ĤçTġă

```
>>> 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
>>>

```

```

def print_result():
    """Print the result of the asynchronous operation"""
    result = apply_async(add, ('hello', 'world'), callback=print_result)
    print('Got: {}'.format(result))

```

```

class ResultHandler:

    def __init__(self):
        self.sequence = 0

    def handler(self, result):
        self.sequence += 1
        print('[{}] Got: {}'.format(self.sequence, result))

```

```

def handler():
    """Print the result of the asynchronous operation"""
    result = apply_async(add, ('hello', 'world'), callback=handler)
    print('Got: {}'.format(result))

```

```

>>> r = ResultHandler()
>>> apply_async(add, (2, 3), callback=r.handler)
[1] Got: 5
>>> apply_async(add, ('hello', 'world'), callback=r.handler)
[2] Got: helloworld
>>>

```

```

def make_handler():
    """Return a handler function"""
    sequence = 0
    def handler(result):
        nonlocal sequence
        sequence += 1
        print('[{}] Got: {}'.format(sequence, result))
    return handler

```

```

def make_handler():
    sequence = 0
    def handler(result):
        nonlocal sequence
        sequence += 1
        print('[{}] Got: {}'.format(sequence, result))
    return handler

```

```

def handler():
    """Print the result of the asynchronous operation"""
    result = apply_async(add, ('hello', 'world'), callback=handler)
    print('Got: {}'.format(result))

```

```

>>> 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'âRëad'ŮäyÄäylæZt'énŸçžğçŽDæŮzæşTrijNâRfäzëä;£çTlâ■RçlNæIëäöNæLŖâRŊNæäüçŽDäZ

```
def make_handler():
    sequence = 0
    while True:
        result = yield
        sequence += 1
    print(['{}'] Got: {}'.format(sequence, result))
```

âržäžŌä■RçlNrijNä;äéIJÄëeAä;£çTlâöČçŽD send() æŮzæşTä;IJäyžäZdërČäG;æTrijNæCäyNæL'Äç

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

ëöIëöž

âşžäžŌäZdërČäG;æTřçŽDè;fäzúëÄžäyýeČ;æIJL'âRrëČ;âRŸä;ŮëIdäyýad'■æIČäÄCäyÄeČlâLëäŌşäZ
âZäæ■d'iijNëruæšCæL'gëaŊâŠNäd'DçRëçžŞædIJäzNëŮt'çŽDæL'gëaŊçŌřäCäödeŽËäyLäüšçzRäyçad'säžE
éČčä;ääřšäŤÉëäžäŌžèğçäEşäeCä;TäŤlâ■ŸäŠNæAçäd'■çŽyâEşçŽDçLúæÄAäŤæAçäžEäÄC

èGşärŠæIJL'äyð'çğ■äyžèeAæŮžäijRæIëæ■TèŮäŠNäŤlâ■ŸçLúæÄAäŤæAçärijNä;ääRfäzëäIJläyÄäylär:
äyð'çğ■æŮžäijRçŽyæřTrijNëŮ■âNëæLŮëöyæŸræZt'âLäe;žéGRçžğäŠNëGłçDüäyÄçCžrijNäZäyžäöČäznä
äöČäznëŤŸeČ;èGłäLlæ■TèŮäæL'ÄæIJL'ëcnä;£çTlâLřçŽDâRŸéGRäÄCäZäæ■d'iijNä;æŮäéIJÄäŌžæNëäŤ

æeČædIJä;£çTlêŮ■âNërijNä;äéIJÄëeAçšlæDRâržéČčäžZâRfäŤæTžâRŸéGRçŽDæŞ■ä;IJäÄCäIJläyLé
nonlocal äçræŸŌër■âRëçTlæIëæNĞçd'žæŌëäyNæIëçŽDâRŸéGRäijZäIJlâZdërČäG;æTřäy■ëcnäŤæTžä

èÄNä;£çTlâyÄäylä■RçlNæIëä;IJäyžäyÄäyläZdërČäG;æTřäršæZt'æIJL'ëüçäžErijNäöČëüšëŮ■âNëæŮž
æşRçğ■æDRäZL'äyLæIëëöšrijNäöČæŸ;ä;ŮæZt'âLäçöÄæt'ArijNäZäyžæÄžäEšäršäyÄäyläG;æTřèÄNäüšä
äžüäyTrijNä;ääRfäzëä;LëGłçTšçŽDäŤæTžâRŸéGRëÄNæŮäéIJÄäŌžä;£çTl nonlocal
äçræŸŌäÄCè£Žçğ■æŮžäijRâTřäyÄçijžçCžäršæŸřçŽyâržäžŌäEüäžŮPythonæLæIJrëÄNëIæLŮëöyæřTè
âRëad'ŮëŤŸæIJL'äyÄäžZæřTè;ČëŽ;æGČçŽDëČlâLërijNærTäeCä;£çTlâžNâL'■éIJÄëeAèřČçTl
next() rijNäödeŽËä;£çTlæŮüëŤZäylæ■ëëld'ä;LäöžæŸšëcnäŤŸeöřäÄC
âr;çöäeČæ■d'iijNä■RçlNëŤŸæIJL'âEüäžŮçTlâd'DrijNærTäeCä;IJäyžäyÄäyläEĚèÄTäZdërČäG;æTřçŽDäöž

æeČædIJä;ääžEäžEäRlëIJÄëeAçžZäZdërČäG;æTřäijäeÄŠëcíäd'ŮçŽDäÄijçŽDërIrijNëŤŸæIJL'äyÄçğ■ä
partial() çŽDæŮžäijRäžšä;LæIJL'çTlâÄC äIJlæšæIJL'ä;£çTl partial()
çŽDæŮüäÄžrijNä;ääRrëČ;çžRäyýçIJNâLŖäyNëIçè£Žçğ■ä;£çTllambdaeäŤè;äijRçŽDäd'■æIČäžççäArijZ

```
>>> apply_async(add, (2, 3), callback=lambda r: handler(r, seq))
[1] Got: 5
>>>
```

âRfäzëâRČëÄC7.8ârRëLČçŽDâGäyŤçd'žä;NrijNæTžä;ääeCä;Tä;£çTl partial()
æIëæZt'æTžâRČæTřç■äR■æIëçöÄâNŮäyLëŤfäžççäAäÄC

9.11 7.11 áĚĚèĀĤāZdërĈāĜĭæŦř

éŮóécŸ

ā;Šā;āçijŮāĚŽā;ĤçŦĭāZdërĈāĜĭæŦřçŽDāzççāAçŽDæŮūāĀŽiijŊæŊĚāĤĈā;Ĺād'ŽārRāĜĭæŦřçŽDæLŦā
ā;āāyŊæIJZæL;āĹræšŘāyĭæŮzæŦæĭēēōŦ'āzççāAçIJŊāyĹāŌzæZŦ'āĈRæŸřāyĀāyĭæZōēĀŽçŽDæL'gēāŊāžĭ

èĝĉāĒşæŮzæāĹ

éĀŽēĤĜā;ĤçŦĭçŦŦşæĹRāZĭāŠŊā■RçĹŊāRřāzēā;Ĥā;ŮāZdërĈāĜĭæŦřāĒĚèĀĤāIJĭæšŘāyĭāĜĭæŦřāy■āĈ
āyžāžĒæijŦçd'žēŦ'æŸŌiijŊāĀĜēō;ā;āæIJĹ'āçĈāyŊæL'Āçd'žçŽDāyĀāyĭæL'gēāŊæšRçĝ■ēōāçōŮāzzāĹāçDŮ

```
def apply_async(func, args, *, callback):  
    # Compute the result  
    result = func(*args)  
  
    # Invoke the callback with the result  
    callback(result)
```

æŌēāyŊæĭēēōŦ'æĹSāzŋçIJŊāyĀāyŊāyŊéĭççŽDāzççāĀiijŊāōĈāŊĚāRŋāžĒāyĀāyĭ
Async çšzāŠŊāyĀāyĭ inlined_async èĈĒēēřāZĭiijŽ

```
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
```

ēĤZāyŦ'āyĭāzççāAçĹĜæōŦāĒĀēōyā;āā;ĤçŦĭyieldēr■āRēāĒĚèĀĤāZdërĈā■ēēĹd'āĈĀērŦāçĈiijŽ

```
def add(x, y):
    return x + y

@inline_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')
```

æĈædIJä;æĕŕĈĉŦĭ test () ĩijNä;äaijŽaĭ ŰäĽŕĉşzäijijæĈäyNĉŽĐēĭŞăĜzĭijŽ

```
5
helloworld
0
2
4
6
8
10
12
14
16
18
Goodbye
```

ä;äaijŽäŖŞĉŦŦĭijNĕŽd' äžĖĕĈäyĭĉĽ' žäĽŕĉŽĐēĕĖĕŕäŽĭäŞŦĭ yield
 ĕŕ■äŖĕäđ' ŰĭijNäĖŰäžŰäIJŕæŰžäžŰæşæIJĽ' äĜžĉŦŦŕäzzä;ŦĉŽĐäŽđĕŕĈäĜĭæŦŕ(äĖŰäŰŰđæŶŕäIJĭäŖŦŰäŖäŰŰŽäž

ĕŰĭĕŰž

æIJŋäŖŖĕĽĈäijŽäŰđäŰđäIJĭäIJĭĉŽĐæŦNĕŦŦä;ääĖŞäžŦŰäŽđĕŕĈäĜĭæŦŕäÄÄĉŦŦşæĽŖäŽĭäŞŦæŦĜäĽŰæŦĈŦŦ
 ĕĕŰäĖĽĭijNäIJĭĖIJÄĕĕÄä;ĭĉŦĭäĽŖäŽđĕŕĈĉŽĐäžĉĉäÄäy■ĭijNäĖŞĕŦŦĉĈäIJĭäžŦŰä;ŞäĽ'ĕŰäĉŰŰäŰĕä;IJäij
 ä;ŞĕŰäĉŰŰĕĜ■äŖŕæŰŰĭijNäŽđĕŕĈäĜĭæŦŕĕĉŕĕŦĈĉŦĭäĭĕĉžĝĉz■äđ'ĐĉŖĖĉzŞæđIJäÄĈäpplŷ_äsyſc()
 äĜĭæŦŕæijŦĉđ'žäžĖæĽ'ĝĕäNäŽđĕŕĈĉŽĐäŰđĕŽĖĕÄžĕ;ŞĭijN äŕĭĉŰäŰŰđĕŽĖæĈĖÄĭäy■äŰĈäŖŕĕĈ;äijŽæŽŦ'äĽä
 ĕŰäĉŰŰĈŽĐæŽĈäÄIJäyŦĕĜ■äŖŕæÄĭĕŰŕĕŰşĉŦŦşæĽŖäŽĭäĜĭæŦŕĉŽĐæĽ'ĝĕäNäĭäđNäy■ĕŕNĕÄNäŖĽäÄ
 äĖŰä;ŞäĭĕĕŰŰĭijNŷieldæŞ■ä;IJäijŽä;ĭäyÄäyĭĉŦŦşæĽŖäŽĭäĜĭæŦŕäžĝĉŦŦşäyÄäyĭäÄijäžŰäŽĈäÄIJäÄĈ
 æŦĕäyNäĭĕĕŕĈĉŦĭĉŦŦşæĽŖäŽĭĉŽĐ _____next__() æĽŰ send()
 æŰžæşŦŦŖĽäijŽĕŰ'äŰĈäžŦæŽĈäÄIJäđ'Đĉžĝĉz■æĽ'ĝĕäNäÄĈ
 æäžæ■ŰĕŦŽäyĭäÄĭĕŰŰĭijNĕŦŽäyÄäŖŖĕĽĈĉŽĐæyäŦĈäŖşäIJĭ inline_async()
 ĕĖĖĕŕäŽĭäĜĭæŦŕäy■äžĖäÄĈ äĖŞĕŦŦĉĈäŖşæŶŕĭijNĕĖĖĕŕäŽĭäijŽĕÄŖæ■ĕĕÄ■äŦŦĖĉŦŦşæĽŖäŽĭäĜĭæŦŕĉŽĐä
 yield ĕŕ■äŖĕĭijNäŖŖäyÄäŋäyÄäyĭäÄĈ äyžäžĖĕŦŽæŰäÄŽĭijNäĽŽäijÄäĝNĉŽĐæŰŰäÄŽäĽŽäžžäžĖäyÄä
 result ĕŶşäĽŰäžŰäŖŞĕĜŖĕĭĕäŦĭäĖĕäyÄäyĭ None äÄijäÄĈ

çDúâRÖâijÄâgNäyÄäylä;İçÖræŞ■ä;IijNäzÖeYşâLÜäy■âRÜâGzçzŞædIJâÄijâzüâRŞéÄAçzZçTşæLŘâZİli
 yield èr■âRëijN âIJlèfZéGNäyÄäyl Async çZDâõdä;NècnæÖëâRÜâLřāĀCçDúâRÖâ;İçÖrâijÄâgNæçĀæ
 apply_async() āĀC çDüèĀNijNèfZäylèõaçoŪæIJL'äylæIJĀeraâijCéCİāLĒæYřāõČāzüæşæIJL'ä;İçTİā
 put() æŪzæşTæİēāZdërČāĀC

èfZæŪüâĀZijNæYřæŪüâĀZèrèçzEèğçéGŁäyNāLřāZTāRŚçTşāzEāzĀāzLāzEāĀCäyza;İçÖrçnNā■şèf
 get() æŞ■ä;IJāĀC æÇædIJæTřæ■ōā■YāIJijNāõČäyĀāõZæYř put() āZdërČā■YæT;çZDçzŞædIJāĀCæÇædIJæşææIJL'æTřæ■ōijNéCčāzLāĒLæZČāAIJæŞ■ä;IJāzüç■L'ā;ĒçzŞæ
 èfZäylāĒüā;ŞæĀŌæāüāõdçÖræYřçTş apply_async() āG;æTřæİēāEşāõZçZDāĀC
 æÇædIJā;āāy■çZyāfaâijZæIJL'èfZāzLçèdæGçZDāzNæČEijNā;āāRřāzēā;İçTİ
 multiprocessing āzŞæİèèrTäyĀāyNijN âIJlā■TçNñçZDèfZçİNäy■æL'gēāNāijCæ■èèõaçoŪæŞ■ä;IijN

```

if __name__ == '__main__':
    import multiprocessing
    pool = multiprocessing.Pool()
    apply_async = pool.apply_async

    # Run the test function
    test()
    
```

āõdèZĒäyLä;āaijZāRŚçÖrèfZäylçIJşçZDārşæYřèfZæāüçZDijNā;EæYřèçAèğçéGŁäyĒæēZāĒüā;ŞçZİ
 āRĒād'■æİCçZDæŌgāLūætĀéZŘeŪRāLřçTşæLŘāZİāG;æTřèČNāRŌçZDä;Nā■RāIJæāGāGEāzŞāSŃç
 ærTāçĀijNāIJl contextlib äy■çZD @contextmanager
 èčĒèèrāZİā;İçTİāzEäyĀäylāzd'āzžèf zèğççZDæLĀāügijN éĀZèfGäyĀäyl yield
 èr■âRëârEèfZāĒēāSŃçzâijĀäyLäyNæŪGçõaçoRĒāZİçşYāRĒLāIJlāyĀetūāĀC
 āRĒād'ŪéİdāyÿætĀēāNçZD Twisted āNĒäy■āzşāNĒāRnāzEéİdāyÿçşzâijijçZDāEēēĀTāZdërČāĀC

9.12 7.12 èõŁéŪõéŪ■āNĒäy■āõZāzL'çZDāRŸéĠR

éŪõéçY

ä;āæČşèçAæL'fāsTāG;æTřäy■çZDæşRäyléŪ■āNĒijNāĒEäèõyāõČèC;èõŁéŪõāSŃāŁæTzāG;æTřçZDā

èğçāEşæŪzæāŁ

éĀZāyÿæİèèõşijNéŪ■āNĒçZDāEĒéCİāRŸéĠRārzaZŌād'ŪçTŃæİèèõşæYřāõNāĒéZŘeŪRçZDāĀC
 ä;EæYřijNā;āāRřāzēēĀZèfGçijŪāEŻèõŁéŪõāG;æTřāzüārEāĒüā;IJäyZāG;æTřāsdæĀğçzSāõZāLřéŪ■āNĒäy

```

def sample():
    n = 0
    # Closure function
    def func():
        print('n=', n)

    # Accessor methods for n
    def get_n():
        return n
    
```

(continues on next page)

(continued from previous page)

```
def set_n(value):
    nonlocal n
    n = value

# Attach as function attributes
func.get_n = get_n
func.set_n = set_n
return func
```

äyÑéÍæYřä;ŁçTÍçŽDä;Nā■Ř:

```
>>> f = sample()
>>> f()
n= 0
>>> f.set_n(10)
>>> f()
n= 10
>>> f.get_n()
10
>>>
```

èõìèõž

äyžāžEèrt' æYŌæyĚæěŽāóCă;Tăuěä;IçŽDřijÑæIJL'äyd'çCzéIJĚèAèğčéGLăyĂăyNăĂĆéçŮăĚĹij
ăčřæYŌăŘřäzèèòl'æĹSăzñçijŮăĚŽăĜ;æTřæİěăŁôæTžăĚĚéČlăRŸéĜRçŽDăĀijăĂĆ
ăĚŮăñajijNăĜ;æTřăšđăĂğăĚAèöyæĹSăzñçTlăyĂçğ■ă;ŁçôĂă■TçŽDæŮžajRăřEèóféŮôæŮžæşTçzSăóŽăĹ
èĚYăŘřäzèèĚŽăyĂæ■çŽDæĹl'ăsTřijÑèòl' éŮ■ăNĚăİææNşçşçŽDăóđă;NăĂĆă;ăèçAăĂŽçŽDăžĚăžĚă

```
import sys
class ClosureInstance:
    def __init__(self, locals=None):
        if locals is None:
            locals = sys._getframe(1).f_locals

        # Update instance dictionary with callables
        self.__dict__.update((key,value) for key, value in locals.
→items()

                                if callable(value) )

        # Redirect special methods
    def __len__(self):
        return self.__dict__['__len__']()

# Example use
def Stack():
    items = []
    def push(item):
```

(continues on next page)

(continued from previous page)

```
        items.append(item)

    def pop():
        return items.pop()

    def __len__():
        return len(items)

    return ClosureInstance()
```

äyÑéÍcæYřäyÄäyläžd'äzŠäijRäijŽerÍæÍæijTçd'žáoČæYřæČä;Tåüčä;IjçŽDiiž

```
>>> 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'ëüčçŽDæYřiižÑefŽäyläzčçäAæfRëaÑëtuaÍæäijŽæfTäyÄäylæŽöéÄŽçŽDçsžáoŽázL'èçAâfnâ;Lâd'

```
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)
```

äçCædIJæfŽæäüâAŽiižNä;ääijŽä;UâlRçszäiijäçCäyNçŽDçzŠædIJiiž

```
>>> from timeit import timeit
>>> # Test involving closures
>>> s = Stack()
>>> timeit('s.push(1);s.pop()', 'from __main__ import s')
```

(continues on next page)

Raymond HettingerárzázŒēfZāylēUőécYēō¿èōāāĜzāzEæZt'āŁăēZ¿;ăzēcŔEēgččŽDăTžēfZæŨzæāLăĀ
 èĀŇăyTăōCăŔlăēYŕçIJšăōđčšzçŽDăyĀăylăēĜăēĀłçŽDăZŁæ■ēĀŇăušijjNă¿ŇăēCijjŇçšzçŽDăyžēēAçŁ'zăA
 ăzūăyTă¿;ăēēAăAŽăyĀăžZăĒūăžŨçŽDăuēă¿IJăL'■ēC¿;ēōŁ'ăyĀăžZçŁ'zăēōŁăēŨzăşTçTşæŤŁ(ăŕTăēCăyŁēŁē
 ClosureInstanceăy■ēĜăăEŽēfĜçŽD__len__())ăōđčŎŕăĀĆ)

æĀzä; ŠäyŁeošrijŇaIŁéĒ■; ǫčŽDæUúāĀŽčzŽéU■āŇĒæúāŁāæŮzæsŤäijŽæIŁ'æŽt'ád'ŽčŽDáođčŤlāŁ
æŕŤæČä; áeIĀèèAéĠ■; ǫāEĒéČlčŁúæĀĀāĀĀłúæŮŕcijŠāEšāŇžāĀĀæyĒéŽd'čijŠā■ŸæŁŮāĒūāzŮčŽDāĶ

Contents:

ä;äæČšæŤžāRŸáržèsāāōdä;ŇčŽĐæLŠā■ræLŮæŸ;cd'žè;ŠāGžīījÑēōl'āōČžāznæZl'āĚuāRrèræĀgāĀĆ

ęAęTzāRŸäyÄäyłaöđä;ŇčŽDā■Ůçņäyšēāčd' žiijŇāRfēG■æŮřāōZāzL'āōČčŽD
 __str__() āŠŇ __repr__() æŮzæsTāĀČä;ŇāęČiiJŽ

```
class Pair:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def __repr__(self):
```

239

(continued from previous page)

```

    return 'Pair({0.x!r}, {0.y!r})'.format(self)

def __str__(self):
    return '({0.x!s}, {0.y!s})'.format(self)

```

```
__repr__() æŰzæʃT̪eʃT̪aZdäyÄäyłaõdäL̥N̥çŽDäzc̥çäAeal̥çd'z̥a;çaijR̥iijN̥eÄŽäy̥çT̪l̥eēG■æŰræd̥D̥
âEĖç;õçŽD̥ repr() âĖ;æT̪r̥eʃT̪aZd̥eʃŽäyła■Űçñ̥äy̥s̥iijN̥eũʃæL̥s̥az̥n̥ä;ɬçT̪l̥äzd̥'az̥šaijR̥eğç̥çēGL̥äŽl̥äY̥;çd'z̥
__str__() æŰzæʃT̪aʀEäõdäL̥N̥e;ñæ■çäy̥žäy̥Ääyła■Űçñ̥äy̥s̥iijN̥ä;ɬçT̪l̥ str() æL̥Ű
print() âĖ;æT̪r̥aijŽ̥eL̥š̥äG̥ž̥eʃŽäyła■Űçñ̥äy̥s̥äÄC̥ærT̪äeC̥iijŽ̥
```

```
>>> p = Pair(3, 4)
>>> p
Pair(3, 4) # __repr__() output
>>> print(p)
(3, 4) # __str__() output
>>>
```

[illegible]

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

èóìèőž

èĜłăőŻăzĹ' __repr__ () ăŠŇ __str__ () éĀŽăÿÿæŸřăĹăĕĭçŽĎăžăæČřĭĭŇăŽăăÿžăőČĕČĭçőĀăŇŮ
 äĹŇăĕČřĭĭŇăĕČăďĪăžĔăžĔăŘăĹăŸřăĹ'Šă■řĕĹăĜăăĹŮăŮăĹŮĕĹăĜăăăŘăÿłăőďăĹŇĭĭŇĕČăžĹĹĹŇăžŘăŚ
 __repr__ () ċŦšăĹŔĉŽĎăŮĜăĪŇă■ŮĉŇăÿšăăĜăĜĒăĀŽășŦăŸřéĪĂĕĕĂĕőĹ'
 eval (repr (x)) == x äÿžĉĪĴšăĀČăĕČăďĪăőďăĪĴăÿ■ĕĹĕĹŽăăăă■ŘăĀŽĭĭŇăžŦĕřăăĹŽăžžăÿĂăÿłăĪĴ
 < ăŠŇ > æŇŇĕŦăăĹĕăĀČăřŦăĕČřĭĭŽ

```
>>> f = open('file.dat')
>>> f
<_io.TextIOWrapper name='file.dat' mode='r' encoding='UTF-8'>
>>>
```

æĆæđIJ __str__() æšæIJL'èćnǎǒŽäzL'ijÑéĆčázĹăřšäijŽă;£çŦĭ __repr__()
æİěäžćæŽè;ŠăĜžăĂĆ

äYŁéİcǾŽĐ format () æŰzæſȚǾŽĐä;ŁǾȚȚİcİJNäYŁăŌzăĹŁæİJL'ëüćİİjNæäİjäİjŘăNŰăžćăĂ
 { 0 . x } âřžăȚǾŽĐæYřçññİäYŁăŘCăȚřǾŽĐxăſđăĂgăĂCăžăæ■d'İİjNăİJläYNeİcǾŽĐăG;æȚřäY■İİjN0ăăđéŽ
 self æİJñëžñİİjŽ


```
def __repr__(self):
    return 'Pair({0.x!r}, {0.y!r})'.format(self)
```

ä;IJäyžè£Žçg■āōđçŔřčŽĎäyÄäy!æŽ£äzčrijŇä;ääžšāŔřäzčä;£çŤí %
æŠ■ä;IJçñçijjŇāŕšāČŔäyŇéÍçè£ŽæürijŽ

```
def __repr__(self):
    return 'Pair(%r, %r)' % (self.x, self.y)
```

10.2 8.2 èĠlāōŽāzL'ā■ŮçñęäyšçŽĎæiijāijRāŇŮ

éŮóéčŸ

ä;äæČšéĀŽè£Ġ format() āĠ;æŤŕāŠŇā■ŮçñęäyšæŮzæšŤä;£ā;ŮäyÄäy!āŕžèšæèČ;æŤŕæŇAèĠlāōŽāzL'

èğčāEşæŮzæāL

äyžāzEèĠlāōŽāzL'ā■ŮçñęäyšçŽĎæiijāijRāŇŮrijŇæLŠāžñéIJĀèèAāIJÍçšzäyLéÍçāōŽāzL'
__format__() æŮzæšŤāĀČä;ŇāçČrijŽ

```
_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)
```

çŔŕāIJÍ Date çšççŽĎāōđä;ŇāŔřäzčæŤŕæŇAæiijāijRāŇŮæŠ■ä;IJäžErijjŇāçČāŔŇäyŇéÍçè£ŽæürijŽ

```
>>> d = Date(2012, 12, 21)
>>> format(d)
'2012-12-21'
>>> format(d, 'mdy')
'12/21/2012'
>>> 'The date is {:ymd}'.format(d)
```

(continues on next page)

(continued from previous page)

```
'The date is 2012-12-21'
>>> 'The date is {:mdy}'.format(d)
'The date is 12/21/2012'
>>>
```

èõléõž

`__format__()` æŰzæşŦçžŽPythonçŽĐā■ŰçñęäÿşæäijäijRāŃŰāŁşëČ;æRŔä;ŽäžEäÿÄäÿłéŠł'ā■RāÄ
èŁŽéĜŃéIJĀèçAçĬĀéĜ■äijžèrCçŽĐæŸræäijäijRāŃŰäžççāAçŽĐèġçæđRāũëä;IJāōNāĬłçŦşçşžèĜłāũsāEşşāōž
ä;NāçĈijNāRĈèĀCäÿNéĬçæĬèĜł datetime æłāłŰäÿ■çŽĐäžççāAijŽ

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

āržāžŌāEĚç;őçşžādŃçŽĐæäijäijRāŃŰæIJL'äÿÄäžZæāĜāĜEçŽĐçžæāōŽāĀĆ
āRŕäžæāRĈèĀĆ stringæłāłŰæŰĜæç èřt'æŸŌāĀĆ

10.3 8.3 èõł'áržèşæŦŕæŃÄäÿŁäÿNæŰĜçóaçŔEā■Ŕèõõ

éŰóécŸ

ä;äæČşèõł'ä;äçŽĐāržèşæŦŕæŃÄäÿŁäÿNæŰĜçóaçŔEā■Ŕèõõ(withèř■āŔë)āĀĆ

èġçāEşæŰzæąŁ

äÿžāžEèõł'äÿÄäÿłāržèşæŦEijāōž with èř■āŔëijNä;äéIJĀèçAāōđçŌř __enter__()
āŠŃ__exit__() æŰzæşŦāĀĆ ä;NāçĈijNèĀČèŽŠæCäÿŃçŽĐäÿÄäÿłçşžiiijNāōČèČ;äÿžæŁSäžñāŁZäžžäÿ

```
from socket import socket, AF_INET, SOCK_STREAM

class LazyConnection:
    def __init__(self, address, family=AF_INET, type=SOCK_STREAM):
        self.address = address
        self.family = family
        self.type = type
        self.sock = None
```

(continues on next page)

(continued from previous page)

```
def __enter__(self):
    if self.sock is not None:
        raise RuntimeError('Already connected')
    self.sock = socket(self.family, self.type)
    self.sock.connect(self.address)
    return self.sock

def __exit__(self, exc_ty, exc_val, tb):
    self.sock.close()
    self.sock = None
```

èŁŻäÿłçşzçŽĐăĖşēŤōçŁ'żçĆzăĬłăžŌăōČēăłçđ'žăžĖäÿĂäÿłç;ŚçzĬJēŁđăŌēĭĭŇă;ĖăŸřăĬlăğŇăŇŮçŽĐă
ēŁđăŌēçŽĐăžžçŇŇăŖŇăĖşēŮăŸřă;ŁçŤĬ with ěřăăŖēēĜłăĬlăŏŇăĬŖçŽĐĭĭŇă;ŇăēĆĭĭŹ

```
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
```

ěőłēőž

çĭĭŮăĖŽäÿŁäÿŇăŮĜçŏăçŖĖăŹłçŽĐäÿžēēĂăŌşçŖĖăŸřă;ăçŽĐăžççăĂăĭĭŹăŤ;ăĬŖ
with ěřăăŖēăĬŮăÿăăĬŖġēăŇăĂĆă;ŖăĜççŎŖ with ěřăăŖēçŽĐăŮăăĂŹĭĭŇăŖžēşăçŽĐ
__enter__() æŮžăşŤēçŇēğēăŖŖĭĭŇăŏŐçēŤăŹđçŽĐăĬĭĭ(ăēĆăđĬJăĬĬłçŽĐēŖĬ)ăĭĭŹēçŇēŤŇăĂĭççŽŽ
asăçŖăŸŎçŽĐăŖŸēĜŖăĂĆçĐăăŖŎĭĭŇwith ěřăăŖēăĬŮăĖŇēĬççŽĐăžççăĂăĭĭĂăğŇăĬŖġēăŇăĂĆ
ăĬJăăŖŎĭĭŇ__exit__() æŮžăşŤēçŇēğēăŖŖēŤžēăŇăÿĖçŖĖăŮēă;ĬJăĂĆ

ăÿăçŏă withăžççăĂăĬŮăÿăăŖŖŤşăžĂăžĬĭĭŇăÿĬēĬççŽĐăŎğăĬŮăŤĂēĆ;ăĭĭŹăĬŖġēăŇăŏŇĭĭŇăŖşçŏŮă
ăžŇăŏđăÿĬĭĭŇ__exit__() æŮžăşŤççŽĐçŇăÿĬăÿĬăŖĆăŤŖăŇēăŖŇăžĖăĭĭĬăÿÿçşăđŇăĂăĭĭĬăÿÿăĂĭăŖ
__exit__() æŮžăşŤēç;ēĜłăŮşăĖşăŏŹăĂŎăăŮăĬŖçŤĬēŁŻäÿłăĭĭĬăÿÿăĖăăĂŖĭĭŇăĬŮēĂēăŤçŤēăŏČăžŮă
ăēĆăđĬJ__exit__() ēŁŤăŹđ True ĭĭĭŇēĆçăžĬăĭĭĬăÿÿăĭĭŹēçŇăÿĖçŤĭĭŇăŖşăē;ăĆŖăžĂăžĬēĆ;ăşăăŖŖŖŤ
with ěřăăŖēăŖŎēĬççŽĐçĬŇăžŖççğçăăĬJăăĬăÿÿăĬŖġēăŇăĂĆ

ēŁŸăĬJĬăÿĂăÿłçžĖēĬĆēŮŏēçŸăŖşăŸř LazyConnection
çşzăŸřăŖăăĖăēŏÿăđ'Žäÿł with ěřăăŖēăĬăŤŇăēŮă;ŁçŤĬēŁđăŌēăĂĆ
ă;ĬăŸçĐŮĭĭŇăÿĬēĬççŽĐăŏŹăžĬăÿăăŸăăŇăăŖĬēç;ăĖăēŏÿăÿĂăÿłsocketēŁđăŌēĭĭŇăēĆăđĬJăăĬăĬJă;Łçç
with ěřăăŖēĭĭŇăŖşăĭĭŹăžğçŤşăÿĂăÿłăĭĭĬăÿÿăŤăăĂĆăÿăēŖă;ăăŖŖăžēăĆŖăÿŇēĬççēŤăăăăŏăŤăžăŇăÿĬē

```
from socket import socket, AF_INET, SOCK_STREAM

class LazyConnection:
```

(continues on next page)

(continued from previous page)

```
def __init__(self, address, family=AF_INET, type=SOCK_STREAM):
    self.address = address
    self.family = family
    self.type = type
    self.connections = []

def __enter__(self):
    sock = socket(self.family, self.type)
    sock.connect(self.address)
    self.connections.append(sock)
    return sock

def __exit__(self, exc_ty, exc_val, tb):
    self.connections.pop().close()

# Example use
from functools import partial

conn = LazyConnection(('www.python.org', 80))
with conn as s1:
    pass
    with conn as s2:
        pass
    # s1 and s2 are independent sockets
```

āIĴĉnāzNāyĽL'ŁāIĴnāy■iĴNLazyConnectionĉsāRřāzēēĉĉIJNāAŽæYřæ\$RāyĽēŁđæŌēāuēāŌCā
æřRānā __enter__() æŪzæşTæL'gēāNĉŽDæŪūāŽiĴNāŌČād'■āLūāLZāzzāyĀāyĽæŪřĉŽDēŁđæŌēāzūā
__exit__() æŪzæşTĉŌĀāTĉŽDāzŌæāLāy■āiĴzāGžæIJĀāRŌāyĀāyĽēŁđæŌēāzūāĒşēŪ■āŌČāĀĆ
ēŁŽēGŇĉĴ■āĴŌæIJL'ĉZēZĴĴRĒēgĉiĴNāy■ēŁGāŌČēĴ;āĒAēŌyāĴNāēŪā;ĴĴTĴ with
ēr■āRēāLZāzzād'ŽāyĽēŁđæŌēiĴNāřsāēCāyLēĴēāiĴTĉd'žĉŽDēČĉæūāĀĆ

āIĴēIJĀēēAĉŌāĉRĒāyĀāzŽēŁDæžRæřTāēČæŪGāzūāĀAĉ;ŚĉzIJēŁđæŌēāSŇēTĀĉŽDĉiĴŪĉĴNĉŌřāĉČāy■
ēŁŽāzŽēŁDæžRĉŽDāyĀāyĽāyžēēAĉL'zā;AæYřāŌČāznāŁēēāzēēĉāēL'NāLĴĉŽDāĒşēŪ■āLŪēGLæTĴ;æĴēĉāŌāŁ
ā;NāēČiĴNāēČēđIJā;āērūāēŚČāzēāyĀāyĽēTĀiĴNēČĉāzLā;āāŁēēāzĉāŌāŁāzNāRŌēGLæTĴ;āžEāŌČiĴNāRēāŁ
ēĀžēŁGāŌĉŌř __enter__() āSŇ __exit__() æŪzæşTāzūā;ĴĴTĴ with
ēr■āRēāRřāzēā;ŁāŌzæYŞĉŽDēAŁāĒēŁZāzŽēŪŌēēYiĴN āZāāyž __exit__()
æŪzæşTāRřāzēēŌ'ā;āæŪāēIJĀæNēāŁČēŁZāzŽāzēāĀĆ

āIĴĴ contextmanager æĴāāĴŪāy■æIJL'āyĀāyĽæāGāGĒĉŽDāyLāyNæŪGĉŌāĉRĒæŪzæāŁæĴāēĴiĴNā
āRŇæŪūāIJĴ12.6āřRēŁČāy■ēŁYæIJL'āyĀāyĽāřzæIJnēŁČĉd'zā;NĉĴNāžRĉŽDĉžĴĴĴNāŌL'āĒĴĉŽDāŁŌāTĴĴL'Ł

10.4 8.4 āLZāzzād'gēGŘāřzēsāæŪŪēŁČĴIAāĒĒāYæŪzæşT

éŪŌēēY

ā;āĉŽDĴĴNāžRēēAāLZāzzād'gēGŘ(āRřēČ;āyLĴZĴ;āyĴ)ĉŽDāřzēsāiĴNāřijēGr'ā■āĉTĴā;Łād'gĉŽDāĒēĒā■

èġċaEṣṣæŪzæaġĹ

årzäžÖäyžèeAæYřçTlæIëa;ŞæLŔçõĀā■TçŽDæTřæ■õçzŞædDçŽDçşzèĀÑelĀiijNā;āāRřazèéĀŽèĚĠçzŽ
__slots__ åsdæĀġæIëædĀād'ġçŽDāĠRārŞāōđāĹNæL'Āā■ăçŽDāĒĒā■YāĀCærTāeĆiijŽ

```
class Date:
    __slots__ = ['year', 'month', 'day']
    def __init__(self, year, month, day):
        self.year = year
        self.month = month
        self.day = day
```

ā;Şā;āāōŽāzL' __slots__ āRÖiijNPythonārşāijŽāyžāōđāĹNā;ġçTlāyĀçġ■æŽt'āĹaçt'ġāĠŞçŽDāĒĒā■YāāōđāĹNéĀŽèĚĠçzŽāyġāyġāĹLārRçŽDāZžāōŽād'ġārRçŽDæTřçzDæIëædDāžžiiNēĀNāy■æYřāyžæfRāyġāōđāĹN
āIJĹ __slots__ āy■āĹŪāĠççŽDāsdæĀġāR■āIJĹāĒĒā■YāāōđāĹNéĀŽèĚĠçzŽāyġāyġāĹTřçzDçŽDæNĠāōŽārRæāç
ā;ġçTlīslotsāyġāyġāy■āē;çŽDāIJræŪzārşæYřæLŞāžnāy■ēĆ;āĒ■çzŽāōđāĹNæūzāĹāæŪřçŽDāsdæĀġāžĒiijNā
__slots__ āy■āōŽāzL'çŽDēĆčāžZāsdæĀġāR■āĀC

èõlèõž

ā;ġçTlīslotsāRÖēĹCçIJAçŽDāĒĒā■YāijŽèuşā■YāĀlāsdæĀġçŽDæTřèĠRāŞNçşzādNæIJL'āĒşāĀC
āy■ēĚĠiijNāyĀēĹNāIëèõşiiNā;ġçTlāĹŔçŽDāĒĒā■YāĀzèĠRāŞNārĒæTřæ■ōā■YāĀlāIJġāyġāyġāĒēCçzDāy■ā
āyžāžĒçzŽā;āāyġāyġçŽt'ēġCèõd'ērĒiijNāĀĠççõĹā;āāy■ā;ġçTlīslotsçŽt'æŌēā■YāĀlāyġāyġāDateāōđāĹNiiN
āIJĹ64ā;■çŽDPythonāyĹēĹçèeAā■ăçTl428ā■ŪēĹCiiNēĀNāeCædIJā;ġçTlāžĒslotsiiNāĒĒā■Yā■ăçTlāyNéZ
āēCædIJçĹNāžRāy■ēIJĹāēeAāRŔNæŪūāĹZāžžād'ġēĠRçŽDæŪēāIJşāōđāĹNiiNēĆčāžĹēĚZāyġāřsēĆ;ædĀād'ġ

ār;çõqslotsçIJNāyĹāŌzæYřāyġāyġāĹĹæIJL'çTlçŽDçL'zæĀġiijNāĹĹād'ZæŪūāĀZā;āēĚYæYřāĹŪāĠRārş
PythonçŽDāĹĹād'ZçL'zæĀġēĈ;āĹĹetŪāžŌæŽōéĀZçŽDāşzāžŌā■ŪāĒyçŽDāōđçŌřāĀC
ārĒād'ŪiijNāōŽāzL'āžĒslotsāRŌçŽDçşzāy■āĒ■æTřæNāāyġāžZæŽōéĀZçşzçL'zæĀġāžĒiijNārTāeĆād'Zçz
ād'ġād'ZæTřæĈēĒĒāyNiiNā;āāžTērēāRĹāIJĹēĆčāžZçzRāyŷēcā;ġçTlāĹŔçŽDçTlā;IJæTřæ■õçzŞædDçŽDçş
(ærTāeĆāIJĹĹNāžRāy■ēIJĹāēeAāĹZāžžæşRāyġçşzçŽDāĠççZ;āyġāyġāōđāĹNāržèsa)āĀC

āĒşāžŌ __slots__ çŽDāyġāyġāyŷēġAēřrāNzæYřāōĈāRřazēā;IJāyžāyġāyġāĹAēĈĒāūēāĒūāIëēYşæ■ç
ār;çõqā;ġçTlīslotsāRřazèē;ĹāĹrēĚZæāūçŽDçŽççŽDiiNā;ĒæYřēĚZāyġāžūāy■æYřāōĈçŽDāĹIēāūāĀC
__slots__ æŽt'ād'ZçŽDæYřçTlæIëa;IJāyžāyġāyġāĒĒā■YāijYāNŪāūēāĒūāĀC

10.5 8.5 āĹĹçşzāy■ārĀēĈĒāsdæĀġāR■

éŪōécY

ā;āæĈşārĀēĈĒçşzçŽDāōđāĹNāyĹēĹççŽDāĀIJçġAæIJL'āĀIæTřæ■õiijNā;ĒæYřPythonēr■ēĹĀāžūæşāeIJL

èġċaEṣṣæŪzæaġĹ

PythonçĹNāžRāŞYāy■āŌzā;ĹetŪēr■ēĹĀçL'zæĀġāŌzārĀēĈĒæTřæ■õiijNēĀNæYřēĀŽèĚĠĠçzŽāyġāōŽ
çñnāyġāyġççāōZæYřāžžā;Tāžēā■TāyNāĹŞçzçL'āijĀād't'çŽDāR■ā■ŪēĈ;āžTērēæYřāĒĒā■YāēĈĹāōđçŌřāĀCærTā

```

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

```

Pythonázúäy■äijŽçIJšçŽĎēYzæ■cālŋāžžēōēUōāEĚēČlāŘ■çğrāĀĆä;EæYŕæÇæđIJä;æēŁŻāzŁāAŻēĆŕ
 āŖŊæŮūēŁYēēAæşlæĎŖāĽŕijŊā;ŁçŦlāyŊāĽŠçžŁāijĀād't'çŽĎçžēāōŽāŖŊæāūēĀĆçŦlāžŎēlāāiŮāŖ■āŠŊæ
 ä;ŊāēČiijŊāēČæđIJä;äçIJŊāĽŕæşŖāylæāāiŮāŖ■āžēā■ŦāyŊāĽŠçžŁāijĀād't'(æŕŦāēČ_socket)ijŊēČčāōČār
 çszāijijçŽĎijŊāēlāāiŮçžğāĽŋāĜ;æŦŕæŦāēČ sys._getframe()
 āIJlā;ŁçŦlçŽĎæŮūāĀŽārśā;ŮāĽāā■ārŖāēČāžEāĀĆ
 ä;æēŁYāŖŕēČ;äijŽēAĜāĽŕāIJłçszāōŽāzŁāy■ä;ŁçŦlāyđ'āylāyŊāĽŠçžŁāijĀād't'çŽĎāŚ;āŖ■āĀĆæŕŦā

```

class B:
    def __init__(self):
        self.__private = 0

    def __private_method(self):
        pass

    def public_method(self):
        pass
        self.__private_method()

```

ä;ŁçŦlāŖŊāyŊāĽŠçžŁāijĀāğŊāijŽārījēĜŕ'ēōēUōāŖ■çğrāŖYæĽŖāĚūāzŮā;čāijŖāĀĆ
 æŕŦāēČiijŊāIJlāĽ■ēłçŽĎçszBāy■iijŊçğAæIJŁāśđæĀğāijŽēčŋāĽēāĽŋēĜ■āŚ;āŖ■āyž
 _B__private āŠŊ _B__private_method āĀĆ ēēŁZæŮūāĀŽā;āāŖŕēČ;äijŽēŮōēŁZæāūēĜ■āŚ;āŖ■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

```

ēēŁZēĜŊiijŊçğAæIJŁāŖ■çğŕ __private āŠŊ __private_method
 ēčŋēĜ■āŚ;āŖ■āyž _C__private āŠŊ _C__private_method
 iijŊēēŁZāyļēūşçŁūçszBāy■çŽĎāŖ■çğŕæYŕāōŊāĚlāy■āŖŊçŽĎāĀĆ

èõléõž

äyŁéÍcæŘŘáŁŕæIJL'äyd'çġ■äy■āRŇçŽĎçijŮčāAçžęăōŽ(ā■TäyNāŁŠçžŁāŠNāRŇNäyNāŁŠçžŁ)æİēāŚ;āR
ād'gād'ŽæTŕēĀŇēĪĀrijNā;āāžTèrēēōĪ'ä;āçŽĎēĪđāĒnāĒśāR■çğŕäžēā■TäyNāŁŠçžŁāijĀād't'āĀCā;EæYŕijNāç
āžūäyTæIJL'āžŽāEĒēĪāśđæĀğāžTèrēāIJĪā■Rçşzäy■ēŽRēŮRètūæİēijNēCčāzŁæL'■ēĀCēŽŚā;ŁçTĪāRŇNäyNā

èŁYæIJL'äyĀçCžèçAæşĪæĐRçŽĎæYŕijNæIJL'æŮūāĀŽā;āăōŽāzŁ'çŽĎäyĀäyĪāRŸéĠRāŠNæşRäyĪāŁİç

```
lambda_ = 2.0 # Trailing _ to avoid clash with lambda keyword
```

èŁŽéĠNæŁŚāznāžūäy■ā;ŁçTĪā■TäyNāŁŠçžŁāL'■çijĀçŽĎāŌşāZæYŕāōČéAŁāĒ■èŕrēğčăōCçŽĎä;ŁçTĪā
(āçCā;ŁçTĪā■TäyNāŁŠçžŁāL'■çijĀçŽĎçŽōçŽĎæYŕäyžāžEēYşæ■čāŚ;āR■āEşçĪAēĀNäy■æYŕæNĠæYŌèŁŽ
éĀŽèŁGā;ŁçTĪā■TäyNāŁŠçžŁāRŌçijĀāRŕäžèçğčāEşèŁZäyĪēŮōéçYāĀC

10.6 8.6 āŁŽāžžāRŕçōāçRĒçŽĎāśđæĀğ

éŮōéçY

ä;āæČşçžŽæşRäyĪāōđä;ĪNattributeāçđāŁæéŽđ'èōŁéŮōäyŌāŁōæTžāžNād'ŮçŽĎāĒūāžŮād'DçRĒéĀžè;Ś

èğčāEşæŮzæāŁ

èĠāōŽāzŁ'æşRäyĪāśđæĀğçŽĎäyĀçğ■çōĀā■TæŮzæşTæYŕāŕEāōČāōŽāzŁ'äyžäyĀäyĪpropertyāĀC
ä;ĪNæČiijNäyNēĪççŽĎāžççāĀāōŽāzŁ'āžEäyĀäyĪpropertyiijNāçđāŁāāŕžäyĀäyĪāśđæĀğçōĀā■TçŽĎçşāđNæ

```
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")
```

äyŁēŕäžççāĀäy■æIJL'äyL'äyŁçŽyāĒşēĀTçŽĎæŮzæşTŕijNēŁŽäyL'äyĪæŮzæşTçŽĎāR■ā■ŮēČ;āŁĒēāžäy
çñnäyĀäyĪæŮzæşTæYŕäyĀäyĪ getter āĠ;æTŕiijNāōČā;Łā;Ů first_name

æLŔäyžäyÄäylåsdæÄgãÄĆ äĚüüzŬäyd'äylæŬzæşTçzŽ first_name åsdæÄgæüzåLääžE
 setter åŠŇ deleter åĠæTŕäÄĆ éIJÄèeAåijžerČçŽĐæŸřåŔlæIJL'åIJl first_name
 åsdæÄgècñåLŽåžzåŔŌiijŇ åŔŌéiççŽĐäyd'äylècĚéřåŽl @first_name.setter åŠŇ
 @first_name.deleter æL■èČçècñåŏŽåžL'äÄĆ

propertyçŽĐäyÄäylåĚšéTŏçL'žå;AæŸřåŏČçIJŇäyŁåŐžèü\$æŽŏéÄŽçŽĐattributeæşqazÄäžLäyd'æüiij
 ä;EæŸřèŏféŬŏåŏČçŽĐæŬååÄŽaijŽèĠåLlèğæŔS getter äÄAsetter åŠŇ deleter
 æŬzæşTäÄĆä;ŇæČiijŽ

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

åIJlåŏđçŎřäyÄäylpropertyçŽĐæŬååÄŽiijŇåžTåśĆæTŕæ■ŏ(æÇæđIJæIJL'çŽĐèŕl)äž■çĐüéIJÄèeAå■Ÿå
 åŽæ■d'iijŇåIJlgetåŠŇsetæŬzæşTäy■iijŇä;ääijŽçIJŇåLřårž _first_name
 åsdæÄgçŽĐæ\$■ä;IJiijŇèŁŽåž\$æŸřåŏđéŽĚæTŕæ■ŏäflå■ŸçŽĐåIJŕæŬžåÄĆ
 åŔëåđ'ŬiijŇä;ääŔŕèČ;èŁŸaijŽèŬŏäyžåžÄäžL __init__() æŬzæşTäy■èŏç;ŏäžE
 self.first_name èÄŇäy■æŸŕ self._first_name äÄĆ
 åIJlèŁŽäylå;Ňå■Ŕäy■iijŇåLŠåžñåLŽåžzäyÄäylpropertyçŽĐçŽŏçŽĐårşæŸřåIJlèŏç;ŏattributeçŽĐæŬååÄŽ
 åŽæ■d'iijŇä;ääŔŕèČ;æČşåIJlåLlågŇåŇŬçŽĐæŬååÄŽåžşèŁŽèqŇèŁŽçğ■çşådŇæçÄæşèäÄĆéÄŽèŁĠèŏç;
 self.first_name iijŇèĠåLlèŕČçTl setter æŬzæşTiiijŇ
 èŁŽäylæŬzæşTèĠŇèlçaijŽèŁŽèqŇåŔČæTŕçŽĐæçÄæşëiijŇåŔëåLŽårşæŸřçŽt æŐèèŏféŬŏ
 self._first_name äžEäÄĆ

èŁŸèČç;åIJlåüşå■ŸåIJlçŽĐgetåŠŇsetæŬzæşTåşžçqÄäyŁåŏŽåžLpropertyäÄĆä;ŇæČiijŽ

```
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)
```

ěóľěőž

äyÄäyĹpropertyāsđæĀğāĔūāōđārsæŸrāyĀçşzālŮçŽyāĔşçzŚāōŽæŰzæşŤçŽĐéZĚāŘĹāĀĆāēCæđIJāā
āřśāijŽāŘŚçŎřpropertyæIJñěžñçŽĐfgetāĀfsetāŠŇfdelāśđæĀğāřsæŸřçşzéGŇéİççŽĐæŽōéĀŽæŰzæşŤāĀĆ

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

éĀŽāyŷæĹěěōšīijNā;āāy■āijŽçŽŧ æŎēāRŮērČçŧĹfgetāĹŮēĀĔfsetīijNāōCāzñāijŽāIJĹěōĹēŮōpropertyçŽ
āŘĹæIJĹā;Şā;āçāōāōđēIJĀēçĀāřzattributeæĹğēāNāĔūāzŮécĹāđ' ŮçŽĐæŞ■ā;IJçŽĐæŮūāĀŽæĹ■āžTērē
æIJĹæŮūāĀŽāyĀāžŽāzŎāĔūāzŮçijŮçĹŇēr■ēĹĀ(æŧĹāēCJava)èĹGæĹēçŽĐçĹNāžRāŚŸæĀžēōđ' äyžæĹĀæIJĹ
æĹĀāžēāzŮāzñēōđ' äyžāzççāĀāžTērēāČRāyŇéĹçēĹZæūāāĔZīijŽ

```
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
```

äy■ēçĀāĔZēĹZçğ■æşææIJĹāĀŽāzā;ŤāĔūāzŮécĹāđ' ŮæŞ■ā;IJçŽĐpropertyāĀĆ
éēŮāĔĹīijNāōČāijŽēōĹ' ā;āçŽĐāzççāĀāRŸā;Ůā;ĹēGČèCĹīijNāzūāyŤēĹŸāijŽēĹūæČŚēŸĔēržēĀĔāĀĆ
āĔūāēñāijNāōČēĹŸāijŽēōĹ' ā;āçŽĐçĹNāžRēĹRēāNēŧūæĹēāRŸæĔçā;Ĺāđ' ŽāĀĆ
æIJĀāRŎīijŇēĹZæūōçŽĐēōçēōāāzūæşææIJĹāyçæĹēāzā;ŤçŽĐāē;āđ' ĎāĀĆ
çĹzāĹnæŸrā;Şā;āāžēāRŎæČşçzŽæŽōéĀŽattributeēōĹēŮōæūzāĹāécĹāđ' ŮçŽĐāđ' ĎçŘĔēĀžē;ŚçŽĐæŮūāĀŽ
ā;āāRřāžēārĔāōČāRŸæĹŖāyĀāyĹpropertyēĀŇæŮāēIJĀæŧzāRŸāŎşæĹēççŽĐāzççāĀāĀĆ
āŽāyžēōĹēŮōattributeçŽĐāzççāĀēĹŸæŸřāĹĹæŇĀāŎşæūāĀĆ

PropertiesēĹŸæŸrāyĀçğ■āōŽāzĹ' āĹĹæĀĀēōāçōŮattributeçŽĐæŰzæşŤāĀĆ
ēĹŽçğ■çşzādŇçŽĐattributesāzūāy■āijŽēcñāōđēŽĔçŽĐā■ŸāČĹīijŇēĀŇæŸřāĹĹēIJĀēçĀçŽĐæŮūāĀŽēōāçōŮ

```
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
```

Python uses `__init__` to initialize the object. The `radius` attribute is set to the value passed to the constructor. The `area`, `diameter`, and `perimeter` are properties that are calculated based on the `radius` attribute. The `area` property uses the formula $A = \pi r^2$, the `diameter` property uses the formula $d = 2r$, and the `perimeter` property uses the formula $P = 2\pi r$.

```
>>> c = Circle(4.0)
>>> c.radius
4.0
>>> c.area # Notice lack of ()
50.26548245743669
>>> c.perimeter # Notice lack of ()
25.132741228718345
>>>
```

Python uses `get` and `set` methods to access and modify attributes. The `get` method returns the value of the attribute, and the `set` method sets the value of the attribute.

```
>>> p = Person('Guido')
>>> p.get_first_name()
'Guido'
>>> p.set_first_name('Larry')
>>>
```

The `Person` class has two attributes: `first_name` and `last_name`. The `get_first_name` method returns the value of the `first_name` attribute, and the `set_first_name` method sets the value of the `first_name` attribute. The `Person` class also has a `__init__` method that initializes the `first_name` and `last_name` attributes.

```
class Person:
    def __init__(self, first_name, last_name):
        self.first_name = first_name
        self.last_name = last_name
```

(continues on next page)

(continued from previous page)

```
@property
def first_name(self):
    return self._first_name

@first_name.setter
def first_name(self, value):
    if not isinstance(value, str):
        raise TypeError('Expected a string')
    self._first_name = value

# Repeated property code, but for a different name (bad!)
@property
def last_name(self):
    return self._last_name

@last_name.setter
def last_name(self, value):
    if not isinstance(value, str):
        raise TypeError('Expected a string')
    self._last_name = value
```

éĜ■āđ'■āzčçăĀăijŽārijeĠt'èĠĈèĈĤāĀĀæŸŞăĠžēŤŽăŞŇăyŚéZNçŽDçÍŇăžRăĀĈăă;æúŁæĀřæŸřijŇéĀ
āŖřäzēāŖĈèĀĈ8.9ăŞŇ9.21ăŖĤèĈçŽDăĒĒăôžăĀĈ

10.7 8.7 ěŖĈçŤĭçŁúçşzæŰzæşŤ

éŰőécŸ

ăĵăæĈşăĬĴă■Ŗçşzäy■ěŖĈçŤĭçŁúçşzçŽDæşŖăylăûşçzŖècñèçĒçŽŰçŽDæŰzæşŤăĀĈ

èġĉăĒşæŰzæăĴ

äyžăžĒěŖĈçŤĭçŁúçşz(èŰĒçşz)çŽDăyĀăylăŰzæşŤijŇăŖřäzēă;ĚçŤĭ super()
ăĠĵăŤřijŇăŖŤăçĈijŽ

```
class A:
    def spam(self):
        print('A.spam')

class B(A):
    def spam(self):
        print('B.spam')
        super().spam() # Call parent spam()
```

super() __init__()
æŰzæşŤăy■çăăăĭçŁúçşzècñæ■ççăőçŽDăĴăġŇăŇŰăžĒijŽ

```
class A:
    def __init__(self):
        self.x = 0

class B(A):
    def __init__(self):
        super().__init__()
        self.y = 1
```

`super()` čŽDāRēād' ŪäyÄäylāyÿèğAçTlæsTāGžçŌrāIJlèçEçŽŰPythonçL'zæøLæŪzæsTçŽDäzççäAäy

```
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)
```

āIJlāyLéIcāzççäAäy■iijN__setattr__() çŽDāōđçŌrāNĚāRnāyÄäylāR■ā■ŪæçÄæšēāĀĆ
 æČædIJæšŘäylāsđæĀgāR■āzēäyNāLŠçžŁ()āijĀād' t' iijNārséĀŽēŁĜ super()
 ěČçTlāŌšāğNçŽD __setattr__() iijN āŘēāLŽçŽDērIāršāğTæt' ĺçžZāEĚēČlçŽDäzççŘEāržēsā
 self._obj āŌzād' DçŘEāĀĆ èŁŽçIJNāyLāŌzæIJL'çČzæDRæĀIrijNāZāäyžārščŌŪæšæIJL'æŸĺāijRçŽDæ
 super() äz■čDūāRfāzèæIJL'æTlçŽDāũēāIJāĀĆ

èőléőž

āōđéŽĚäyLiiijNād' gāōūārzāžŌāIJlPythonäy■æČä;Tæ■čçāōä;ŁçTl super()
 āGĵæTŗæŽŏeA■çšēāzNçTŽārSāĀĆ äĵæIJL'æŪūāĀŽäijŽçIJNāLřāCRäyNéIcèŁZæäũçŽt' æŌēērČçTlçLúçšçç

```
class Base:
    def __init__(self):
        print('Base.__init__')

class A(Base):
    def __init__(self):
        Base.__init__(self)
        print('A.__init__')
```

ārĵçōāārzāžŌād' gēČlāLēāzççäAēĀNēIĀēŁZāzLāAŽæšāzĀāzLēŪŏécŸiijNā;EæŸřāIJlæŽt' ād' ■æIČçŽĹ
 æřTāēČiijNēĀČèŽSāçCäyNçŽDæČĚāEĵiijŽ

```

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__')

```

æCædIJa;æfRèaÑèfZæôṭāzççăAâršaijŽâRŚçŒř
 ècnèrĈçTlāyḏ' æñaiijNăæCăyNæL' Åçḏ' žiijŽ

Base.__init__()

```

>>> c = C()
Base.__init__
A.__init__
Base.__init__
B.__init__
C.__init__
>>>

```

ârRèĈjāyḏ' æñæřĈçTl Base.__init__() æšqāzĂăžĹăİRăḏ' ĎriijNăjEæIJL' æUŭăĂŽăṛt' äy■æYřăĂĆ
 âRçäyĂæŮzéİcrijNăAĜèōĴă;ăăIĴlāzççăAăy■æ■ćæĹRăjŁçTl
 iijNçzŞædIJařsăĴĹăŏNçĴŌăžEiijŽ

super()

```

class Base:
    def __init__(self):
        print('Base.__init__')

class A(Base):
    def __init__(self):
        super().__init__()
        print('A.__init__')

class B(Base):
    def __init__(self):
        super().__init__()
        print('B.__init__')

class C(A, B):

```

(continues on next page)

(continued from previous page)

```
def __init__(self):
    super().__init__() # Only one call to super() here
    print('C.__init__')
```

èĚŘèĚÑèĚŽäyĽæŮřĽĽæIJñāŘŌĭijNā;āāijZāRŚçŌřæŔäyĽ __init__()
æŮzæşŤāŔĽāijŽècñèŕĈçŦĽäyĀæñāžĒĭijŽ

```
>>> c = C()
Base.__init__
B.__init__
A.__init__
C.__init__
>>>
```

äyžāžĒāijDæyĒāōĈçŽDāŌşçŔĒĭijNæĽŚāžñēIJĀèēAèĽşçĈzæŮŭēŮŦ'èğçéĜĽäyŦPythonæŸŕæĈcā;Ŧāōđ
āržāžŌā;āāōŽāzĽçŽDæŔäyĀäyĽçşzĭijŦPythonāijŽèōāçōŮāĜžäyĀäyĽæĽĀēŦşçŽDæŮzæşŤèğçæđŔēāžāžŔ(I
èĚŽäyĽMROāĽŮèāĽāŕşæŸŕäyĀäyĽçōĀā■ŦçŽDæĽĀæIJĽāşžçşçŽDçžĽæĀğēāžāžŔēāĽāĈcā;NāēĈĭijŽ

```
>>> C.__mro__
(<class '__main__.C'>, <class '__main__.A'>, <class '__main__.B'>,
<class '__main__.Base'>, <class 'object'>)
>>>
```

äyžāžĒāōđçŌŕçžġæĽĕĭijŦPythonāijZāIJMROāĽŮèāĽäyĽäzŌāŭēāĽŕāŔŕşāijĀāğNæşēæĽ;āşžçşzĭijNçZŦ'
èĀNèĚŽäyĽMROāĽŮèāĽçŽDæđĒēĀāæŸŕēĀžēĜäyĀäyĽççžĽæĀğāNŮçōŮæşŤæĽēāōđçŌŕçŽDāĈ
æĽŚāžñäy■āŌzæŭşçĽ'ŭèĚŽäyĽçōŮæşŤçŽDæŦŕā■ēāŌşçŔĒĭijNāōĈāōđēŽĒäyĽāŕşæŸŕāŔĽāžŭæĽĀæIJĽçĽçç

- ā■ŔçşzāijZāĒĽäžŌçĽŮçşžècñæĈĀæşē
- āđ'ŽäyĽçĽŮçşzāijZæāžæ■ōāōĈāžñāIJāĽŮèāĽäy■çŽDēāžāžŔècñæĈĀæşē
- āēĈæđIJāržäyNäyĀäyĽçşzā■ŸāIJläyđ'äyĽāŔĽæşŤçŽDēĀĽæNĦ'ĭijNēĀĽæNĦ'çññäyĀäyĽçĽçşž

èĀĀāōđēŦ'ĭijNā;æĽĀēēAçşēēAşçŽDārşæŸŕMROāĽŮèāĽäy■çŽDçşžēāžāžŔāijŽèōĽ'ā;āāōŽāzĽçŽDāž
ā;Şā;āā;ĕçŦĽŦŦsuper()āĜ;æŦŕæŮŭĭijŦPythonāijZāIJMROāĽŮèāĽäyĽçžġçz■æŔIJçŦcāyNäyĀäyĽçşzā
ārĽēēAæŔŔäyĽēĜ■āōŽāzĽçŽDæŮzæşŤçžşäyĀā;ĕçŦĽŦŦsuper()
āžŭāŔĽēŦççŦĽāōĈäyĀæñāĭijNēĈçāžĽæŌğāĽŭæŦAæIJĀçzĽāijŽēA■āŌēāōNæŦŦ'äyĽM-
ROāĽŮèāĽĭijNæŔŔäyĽæŮzæşŤāžşāŔĽāijŽècñèŕĈçŦĽäyĀæñāāĈ
èĚŽāžşæŸŕäyžāžĀāžĽāIJĽçññāžNäyĽā;Nā■Ŕäy■ā;äy■āijŽēŦççŦĽäyđ'æñā Base.
__init__() çŽDāŌşçāžāĈ

super()æIJĽäyĽāžđ'āžžāŔĈæĈĽçŽDāIJŕæŮzæŸŕāōĈāžŭäy■äyĀāōŽāŌzæşēæĽ;æşŔäyĽçşzāIJMRO
ā;āçŦŽēĜşāŔŕāžēāIJäyĀäyĽæşāæIJĽçZŦ'æŌēçĽŮçşççŽDçşžäy■ā;ĕçŦĽāōĈāĈcā;NāēĈĭijNēĀĈēŽŚæĈäyNē

```
class A:
    def spam(self):
        print('A.spam')
        super().spam()
```

āēĈæđIJā;æŦŦçĽĽĈçZŦ'æŌēā;ĕçŦĽēĚŽäyĽçşzārşāijZāĜžēŦŽĭijŽ

```
>>> 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;EæYřijŇæĈæđIJä;ää;ŁçTłăđ'ŽçzğæL'ŁçŽĐëřİçIJŇçIJŇäijŽăŔŚçTŖšăzĂăzŁřijŽ

```
>>> class B:
...     def spam(self):
...         print('B.spam')
...
>>> class C(A, B):
...     pass
...
>>> c = C()
>>> c.spam()
A.spam
B.spam
>>>
```

ä;ăăŔřăžčçIJŇăĹŕăIJłçśzAäy■ä;ŁçTł
 aôđéŽĚäyŁëřČçTłçŽĐæYřëuŖççszAæŕŇæŮăăĚŖççzçŽĐçszBäy■çŽĐ spam() æŮzæŖTăĂĈ
 èŁZăyŁçTłçszCçŽDMŖOăĹŮëăĹăŕŝăŔřăžăăŏŇăĚĹëğçéĠăyĚæčŽăžEřijŽ

```
>>> C.__mro__
(<class '__main__.C'>, <class '__main__.A'>, <class '__main__.B'>,
<class 'object'>)
>>>
```

ăIJłăŏŽăzŁ'æŮăăĚĚçszçŽĐæŮăăĂŽèŁZæăŮä;ŁçTł
 æYřă;ŁæŽŏéA■çŽĐăĂĈăŔřăžăăŔĈèĂĈ8.13ăŖŇ8.18ăŕŔèŁĈăĂĈ

çĐŮëĂŇřijŇçTŖšăžŎ super() âŔŕèČ;äijŽèřČçTłäy■æYřă;ăæČŖëçAçŽĐæŮzæŖTřijŇă;ăăžŤëŕééAřă;łäy
 éçŮăĚŁřijŇçăŏăĹăIJłçzğæL'Łă;Ŗççszäy■æL'ĂæIJL'çŽyăŖŇăŔ■ă■ŮçŽĐæŮzæŖTæŇæIJL'ăŔŕăĚijăŏžçŽĐăŔ
 èŁZăăŮăŔřăžčăŏăĹ super() èřČçTłäyĂäyĹéđçŽt'æŎëçŁŮçszæŮzæŖTæŮăy■äijŽăĠžéŤŽăĂĈ
 âĚŮăŇăřijŇæIJĂăç;çăŏăĹăIJĂéăŮăŝCçŽĐçszăŔŔă;ŽăžEèŁZăyŁæŮzæŖTçŽĐăŏđçŎřijŇèŁZăăŮçŽĐëŕłăIJł

ăIJłPythonçđ;ăŇžăy■ăŕžăžŎ super() çŽĐä;ŁçTłæIJL'æŮăăĂŽăijŽăijŤæĹëäyĂăžŽăzŁ'èŏăĂĈ
 âŕ;çŏăăĈæ■đ'řijŇæĈæđIJäyĂăĹĠéăžăĹĹ'çŽĐëřİřijŇă;ăăžŤëŕéăIJłä;ăæIJĂæŮŕăžčçăAäy■ä;ŁçTłăŏĈăĂĈ
 Raymond Hettingeräyžæ■đ'ăĚŽăžEäyĂçŕĠéđăyŷăë;çŽĐæŮĠçŇă äĂIJPythonăĂŽŝ super()
 Considered Super!ăĂİ řijŇ éĂŽèŁĠăđ'ġéĠŔçŽĐä;Ňă■ŔăŔŝăĹŝăžŇëğçéĠăžEäyžăžĂăzŁ
 super() æYřăđĂăë;çŽĐăĂĈ

10.8 8.8 ăĤşăÿæŁ'ăŢproperty

éŮőécÿ

ǎIǎǎ■Řčšzäy■iijǑnä;ǎæČšèeAæL'l'ásTǎóŽžäZL'ǎIǎǎ■Łűčšzäy■čŽDpropertyčŽDǎLšèČ;ǎǎČ

èġčǎẸșæŮźæǻŁ

èĂĈèŽŚaęĆäÿŦçŽĎäzčçăĂiijŦăőĈăőŽăzL'ăžĖäÿĂäÿłpropertyiijŽ

```
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")
```

äyNéIcæYřäyÄäyŁcd'zäŁNçsziijNăoČçzğæL'fëĞİPersonăzúæL'l'ăsTăžE
ăsđæĂğçŽĐăLşëČĭiijŽ

```
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)
```


æŒäyŊæİä;ŁçŦİēŁŻäyŁæŦŕçşziiŹ

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

æĈædİJä;ääzĖäzĖäŦŁæĈşæLŦ'āsŦpropertyçŽDæşŦäyÄäyŁæŦzæşŦiijŊéĈcäzŁäŦŕäzæäĈŦäyŊéİcèŁŻæ

```
class SubPerson(Person):
    @Person.name.getter
    def name(self):
        print('Getting name')
        return super().name
```

æŁŦŦæÄĖiijŊä;ääŦŁæĈşäŦŒäŦzsetteræŦzæşŦiijŊäŦŕşèŁŻäzŁäĖŻiiŹ

```
class SubPerson(Person):
    @Person.name.setter
    def name(self, value):
        print('Setting name to', value)
        super(SubPerson, SubPerson).name.__set__(self, value)
```

èŒİēŒ

âİJİāŦŦŕçşäy■æLŦ'āsŦäyÄäyŁpropertyâŦŕēĈ;äijŽäijŦtēŦüâŁäđ'Žäy■æŦŦşârşèĖŁçŽĐēŦŦŒēçŦiijŊ
âŽääyžäyÄäyŁpropertyâĖŦüâŦŒæŦŦ getterāÄĖsetter āŦŦ
deleter æŦzæşŦçŽĐēŽĖâŦŦİijŊēÄŦŦäy■æŦŦŦäyŁæŦzæşŦäĈ
âŽäæ■đ'iiŊŊä;şä;æLŦ'āsŦäyÄäyŁpropertyçŽDæŦŦüâÄŦiijŊä;æİJÄēæAâĖŁçäŦŦŦä;æŦŦŦŦēēæĖĖæŦŦŦäŦŦ

âİJİçŋŋäyÄäyŁä;ŊâŦŦŦäy■iijŊæL'ÄæİJŁçŽĐpropertyæŦzæşŦēĈ;èçŋēĖ■æŦŦŦäŦŦzäzŁ'āĈ
âİJİæŦŦäyÄäyŁæŦzæşŦäy■iijŊä;ŁçŦİläŦE super() æİēēŦĈŦİĈŁüçşçŽĐäŦŦđçŦŦāĈ
âİJİ setter âĖ;æŦŦäy■ä;ŁçŦİ super(SubPerson, SubPerson).
name.__set__(self, value) çŽĐēŦ■âŦŦæŦŦŦşæİJŁēŦŦŽçŽĐäĈ
äyžäzĖäĖŦæL'ŦçžžäzŊâL'■âŦŦzäzŁçŽĐsetteræŦzæşŦiijŊéİJÄēæAâŦŦæŦŦŦäŁüæİĈäijäēÄŦçžžäzŊâL'■âŦŦzäz
__set__() æŦzæşŦäĈ äy■ēŁĖiijŊēŦŦâŦŦŦēŁŻäyŁæŦzæşŦçŽĐäŦŦŦäyÄēĈĈä;ĐæŦŦŦä;ŁçŦİçşzâŦŦēĖŦŦēÄ
ēŁŻäzşæŦŦäyžäzÄäzŁæŁŦäzŋēæAä;ŁçŦİ super(SubPerson, SubPerson)
çŽĐäŦŦşäzāĈ

æĈæđIĴăăRĴæĈşéĜăăŏŽăžLăĔŭăŷăăŷĂăŷĴăŰăşŢĵĴŃĖĈăăRĴăĴĉŦĬ @property
 æIĴŋĕžŋăŸŕăŷăăđ'şĉŽĐăĂĈăŕŦăĖĈĵĴŃăŷŢĖĬĉŽĐăžĉĉăĂăŕşăŰăăşŦăŭĕăĴĴĵĴŽ

```
class SubPerson(Person):
    @property # Doesn't work
    def name(self):
        print('Getting name')
        return super().name
```

æĈæđIĴăăŕŦĉĬĂĕĖŖĕăŢăĴĴăŕŖşĉŎŕşetterăĜĴăŦŕăŦŦăŷĴăŰăĴăđ'şăžĖĵĴŽ

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

ăĴăăžŦĕŕĕăĈŖăžŢăĴăăŕŦ'ĕĖĜĉŽĐĕĈăăŭăĖŏăŦăžăžĉăĂĵĴŽ

```
class SubPerson(Person):
    @Person.name.getter
    def name(self):
        print('Getting name')
        return super().name
```

ĕĖŽăžĴăĖŽăăŖŎĵĴŢpropertyăžŢăĴăăŭşĉžŖăăŏŽăžLăĕĖĜĉŽĐăŰăăşŦăĴĴŽĕĉăđ'ăăĴăĕĖĜăĬĕĵĴŢĖĂŢĴĖĴ

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

ăĴĴĕĖŽăŷĴĉLăŕăĴŋĉŽĐĕĝĉăĖşăŰăăĴăŷăăĴăŷĴĴĴŢăŰăŷăăĴăđăşŦăĴĉŦĬăĖŽŦăăĖĂŽĉŦĬĉŽĐăŰăăĴăĴăŕăĈ
 Person ĉşăăŖăăĂĈ æĈæđIĴăăŷăăşĉşĕĖĂşăĴŕăžŦăŸŕăşĴăŷĴăşĉşăăŏŽăžLăžĖpropertyĵĴŢ
 ĕĈăăĴăăŖĬĕĴăŽĕĖĜăăŰăăŏŽăžLăĖĴăăĴĴpropertyăžŰăăĴĉŦĬ super()
 æĬăăŖĖăŎĝăĴăĬăĴăăĖĂşĉžŽăĴăăĖĬĉŽĐăăđĉŎŕăĂĈ

ăĂĵăăŰăăşĴăăĐŖĉŽĐăŸŕăŷăăĬĕăĵĴĉđ'žĉŽĐĉŋăŷăăĜăăăĴăăĴĖĖĖĖĉŋĉŦĬăĬăăĴăăŦăăşŦăăŷăăŷăă

```

# 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)

```

æIJĀāRŌāĀijā; ŪæşĴæDRçŽDæYřijNērzaĽrēŁZéGŅæŪiijNā; āāžTēřēaijŽāRŚçŌřā■RçśzāŅŪ
 setter āŠŅ deleter æŪzæşŦāĒūāōđæYřā; ĽçōĀā■ŦçŽDāĀĆ
 èŁZéGŅæijŦçđ'žçŽDègčāEşæŪzæāĽāRŅæūēĀĆçŦliijNā; EæYřāIJĴ PythonçŽDissueéāŦéĴc
 æĽēāŚĽçŽDāyĀāyĴbugiijŅæĽŪēōyāijŽā; ģā; ŪāřEæĴēçŽDPythonçĽĽæIJñāy■āGžçŌřāyĀāyĴæŽt' āĽāçōĀæt'

10.9 8.9 āĽZāžžæŪřçŽDçşzæĽŪāōđä; ŅāśđæĀğ

éŪōécŸ

ä;āæČşāĽZāžžāyĀāyĴæŪřçŽDæŅæIJĽāyĀāžZéciād' ŪāĽşèČ;çŽDāōđä; ŅāśđæĀğçşzādŅiijŅærŦāæĆç

èġċăEşæŮzæąĹ

ăĕĆăđĲăĵăăĈşăĹZăzžăyĂăyĴăĒĴăŮřċŽĐăőđăĴŊăşđăĂġĳĳŊăŊŕăzėėĂŽĕĴĠăyĂăyĴăŕŕĕĤŕăZĴĳşzċŽĐă

```
# 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]
```

ăyĂăyĴăŕŕĕĤŕăZĴĴăŕşăŸŕăyĂăyĴăőđċŎŕăžĒăyĴăyĴăăĴĈċŽĐăşđăĂġĕőĴėŮőăŞăĲĲ(get,
set, delete)ċŽĐċşzĳĳŊăĴĒĒăĴăĴăyž __get__()ăĂĂ__set__()
ăŞŊ __delete__() ĕĴZăyĴăyĴċĴăŕăĴċŽĐăŮzăşŤăĂĈ
ĕĴZăžZăŮzăşŤăŎĕăŕŮăyĂăyĴăőđăĴŊăĲĲăyžĕĴŞăĒĕĳĳŊăžŊăŔŎċŽăžŤċŽĐăŞăĲĲăőđăĴŊăžŤăşĈċŽĐă
ăyžăžĒăĲċŤĴăyĂăyĴăŕŕĕĤŕăZĴĳĳŊăŊĲĂăŕĒĕĴZăyĴăŕŕĕĤŕăZĴĴăőđăĴŊăĲĲăyžċşzăşđăĂġăŤăĴĴăyĂă

```
class Point:
    x = Integer('x')
    y = Integer('y')

    def __init__(self, x, y):
        self.x = x
        self.y = y
```

ăĴŞăĲăĕĴZăăăăĂZăŔŎĳĳŊăĴĒĂăĲĲăŕŕăzăŕŕĕĤŕăZĴăşđăĂġă(ăŕŤăĕĈăĴĴŮy)ċŽĐăőĴėŮőăĳĳŽĕĕŋ
__get__()ăĂĂ__set__()ăŞŊ __delete__()ăŮzăşŤăŞăŤĕŎăăĴŕăĂĈăĴŊăĕĳĳŊă

```
>>> p = Point(2, 3)
>>> p.x # Calls Point.x.__get__(p, Point)
2
>>> p.y = 5 # Calls Point.y.__set__(p, 5)
>>> p.x = 2.3 # Calls Point.x.__set__(p, 2.3)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "descrip.py", line 12, in __set__
    raise TypeError('Expected an int')
```

(continues on next page)

(continued from previous page)

```
TypeError: Expected an int
>>>
```

```
    @property
    def x(self):
        return self._x

    @x.setter
    def x(self, value):
        self._x = value

    @property
    def y(self):
        return self._y

    @y.setter
    def y(self, value):
        self._y = value
```

Decorators

```
def decorator(func):
    def wrapper(*args, **kwargs):
        # Do something before the function call
        return func(*args, **kwargs)
    return wrapper
```

```
class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def __str__(self):
        return f'Point(x={self.x}, y={self.y})'
```

```
    def __repr__(self):
        return f'Point(x={self.x}, y={self.y})'
```

```
# 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
```

```
    def __get__(self, instance, cls):
        if instance is None:
            return self
        else:
            return instance.__dict__[self.name]
```

```
# 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]
```

```
    def __get__(self, instance, cls):
        if instance is None:
            return self
        else:
            return instance.__dict__[self.name]
```

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

æRRèfrâZÍéĂŽăyyæYréCcăžZă;£çTíáLrèčĚēřăZÍæLŮăĚČšzçŽĎăd'găđNăæEăđúăy■çŽĎăyĂăylçžĎăyăylă;Nă■RrijNăyNéIcæYřăyĂăžZæŽt'énYçžgçŽĎăšzăžŌæRRèfrâZÍçŽĎăžčçăArijNăžúăŮL'ăRLăLřăyĂ

```
# 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
```

æIJăăRŌèçAæNĜăGžçŽĎăyĂçCzæYrijNăçCăđIJă;ăăRlæYřæČšçŏĂă■TçŽĎèĜlăŏŽăžL'æšRăylçšzçŽè£Žçg■æČĚăEțăyNă;£çTí8.6ărRèLCăžNçž■çŽĎpropertyæLĂæIJřăijZæŽt'ăLăăŏžæYšăĂĆă;ŠçlNăžRăy■æIJL'ă;Lăđ'ŽéĜ■ăđ'■ăžčçăĂçŽĎæŮăăĂŽæRRèfrâZÍăřsă;LăIJL'çTíăžE(æřTăçCă;ăæČșăIJă;ăăžčçăĂçŽĎă;Lăđ'ŽăIJăŮžă;£çTíæRRèfrâZÍæRRă;ZçŽĎăLșèČ;æLŮăĚĂăřEăŏCă;I

10.10 8.10 ä;£çTíăžúè£šèŏaçŏŮăśđæĂğ

éŮŏécŸ

ă;ăæČșărEăyĂăylăRlèřzăśđæĂğăŏŽăžL'æLřăyĂăylpropertyrijNăžúăyTăRlăIJlèŏ£éŮŏçŽĎæŮăăĂŽæL'ă;EăYřăyĂăŮèçnéŏ£éŮŏăRŌrijNă;ăăyNăæIJZçžšăđIJăĂijèçñçijšă■YèțuălèrijNăy■çTíăřRăñăçČ;ăŌžèŏă

èġċaEṣæŪzæaġL

ãŏŽāzLāyĀäyġāzūē£šāsđæĀğçŽDäyĀçġ■énŸæŦLæŪzæşŦæŸréĀŽē£Ġä;£çŦlāyĀäyġæŦŦē£ŦāZġcszġijŦ

```
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;äēIJĀēęAāČŦāyŦēİcē£ŽæăũāIJlāyĀäyġcszāy■ä;£çŦlāŏČġijŽ

```
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
```

äyŦēİcāIJlāyĀäyġāzđ'āžŠçŦŦācČäy■æijŦçđ'žāŏČçŽDä;£çŦlġijŽ

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

āzŤčzEēgĆāršā;āāijŽāRŚçŖræūLæAr Computing area āŠŇ Computing
perimeter āzĖāzĖāGžçŖrāyĀæñāāĆ

ěóíěőž

āĭLād'ŽæŮūāĀŽiijNāđDēĀāyĀāyĭāzūēſšēōaçōŮāśđæĀğçŽDāyžēēAçŽōçŽDæYřāyžāžEæRŘā■GæĀ
āĭNāēČiijNā;āāRřāžēēAĤāĖ■ēōaçōŮēſŽāžŽāśđæĀğāĀijriiNēŽd'ēIdā;āçIJšçŽDēIJĀēēAāōČāžñāĀĆ
ēſŽēGŇāijŤčd'žçŽDæŮzæāLārśæYřçŤlāĭēāōđçŖrēſŽæūçŽDæŤLæđIJçŽDriiN
ārĭāy■ēſGāōČæYřēĀžēſGāžēēIdāyēnYæŤLçŽDæŮzāijRā;ſçŤlāRŘēſrāŽlçŽDāyĀāyĭçſ;āēŽçL'zæĀğæĭē

æ■čāēČāIJlāĖūāzŮārRēLČ(āēČ8.9ārRēLČ)æL'ĀēōšçŽDēČčæūriiNā;ŠāyĀāyĭæRŘēſrāŽlēcñæŤĭāĖēāy
æſRræñæēōſēŮōāśđæĀğæŮūāōČçŽD __get__() āĀA__set__() āŠŇ __delete__()
æŮzæšŤārśāijŽēēñēgēāRŚāĀĆ āy■ēſGriiNāēČæđIJāyĀāyĭæRŘēſrāŽlāzĖāzĖāRlāōŽāzL'āžEāyĀāyĭ
__get__() æŮzæšŤçŽDērĭiijNāōČærŤēĀžāyççŽDāĖūæIJL'æŽt'āijšçŽDçzŠāōŽāĀĆ
çL'zāLñāIJriiNāRlāēIJL'ā;ŠēēñēōſēŮōāśđæĀğāy■āIJlāōđāĭNāžŤāsČçŽDā■ŮāĖyāy■æŮū
__get__() æŮzæšŤæL'■āijŽēēñēgēāRŚāĀĆ

lazyproperty çšzāL'çŤlēſŽāyĀçČziijNā;ſçŤl __get__()
æŮzæšŤāIJlāōđāĭNāy■ā■YāČlēōaçōŮāGžæĭēçŽDāĀijriiN ēſŽāyĭāōđāĭNā;ſçŤlçŽyāRŇçŽDāR■ā■Ůā;IJāyž
ēſŽæūāyĀāĭēriiNçzŠæđIJāĀijēcñā■YāČlāIJlāōđāĭNā■ŮāĖyāy■āzūāyŤāžēāRŖōārsāy■ēIJĀēēAāĖ■āŖēōaç
ā;āāRřāžēārĭērŤæŽt'æūsāĖēçŽDāĭNā■RræĭēēgĆāršçzŠæđIJiijŽ

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

ēſŽçg■æŮzæāLæIJL'āyĀāyĭārRçijžēŽuārśæYřēōaçōŮāGžçŽDāĀijēcñāLŽāžžāRŖōæYřārřāžēēēñāſōæŤ


```
>>> c.area
Computing area
50.26548245743669
>>> c.area = 25
>>> c.area
25
>>>
```

æĈædIJä;äæÑĖăĈĕĤZäyléUőécYriiÑéĈčázĹăRřázěä;ĤçTlăyĂçğ■ā;ăŕăşăĈčázĹénYæTĹçŽDăôđç

```
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
```

æĈædIJä;äă;ĤçTlăĤZäylçĹLăIJñriiNăřsăiJŽăRŚçŎřçŎřăIJlăŕăşăTzæŞ■ă;IJăũşçzRăy■ĕcŋăĖAĕőyăžĖiij

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

çDűĖĂñriiÑĖĤŽçğ■ăŰzæăĹæIJĹăyĂăylçijžçĈzărşăYřæĹĂæIJĹgetăŞ■ă;IJĕĈ;ăĤĖĕăžĕcŋăőŽăRŚăĹăřăşă
getterăĤ;æTřăyĹăŎžăĖĤĕĤZäylĕuşăžNăĹ■ĈőĂă■TçŽDăIJlăôđă;Nă■ŰăĖyăy■ăşĕăĹ;ăĂijçŽDăŰzæă
æĈædIJăĈşĕŎăRŰăZt'ăd'ŽăĖşăžŎpropertyăSŇăRřçőăçRĖăşđăĂğçŽDăĤăăAřriiNăRřázěăRĈĕĂĈ8.6ărR

10.11 8.11 çőĂăŇŰæTřæ■őçzŞæđDçŽDăĹiăğNăŇŰ

éUőécY

ă;ăăĖŽăžĖăĹăđ'ŽăžĖăžĖĖçTlă;IJăTřæ■őçzŞæđDçŽDçşzriiNăy■ăĈşăĖŽăđ'ăđ'ŽçĈĕăžžçŽD
__init__()ăĤ;æTř

èġċăĖşăŮzăăĹ

ăŔăzéăĹĴăyĂăyĴăşżçşzăy■ăĖŽăyĂăyĴăĖŋĴĴçŽĐ __init__() ăĢăĤŕĴĴŽ

```
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)
```

çĐúăŔŎăĴăĵăçŽĐçşçzġăĹ'ĤăĢĤăĤŽăyĴăşżçşz:

```
# 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
```

ăĴĤçĴĴăĤŽăżŽçşçzŽĐçđ'żăĴŦĴĴŽ

```
>>> 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
```

ăĖĴăđĴĤăĤŸăĴşăŦŕăŦăĴăĤşăŦŦăă■ŮăŔĴăĤŦĴĴŦăŔăzéăŕĖăĤşăŦŦăă■ŮăŔĴăĤŦŕăŦçĴĴăyżăŦđăĴŦăşđăĤ

```
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;äefYëČ;ärEäy■āIJĭ _fields äy■čŽDāŘ■çğrāŁāāĖēāĹrāsđæĀğäy■āŎziijŽ

```

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:

```


@abstractmethod èfYèČ;æşlëğćéIzëĀAæŨzæşTăĀAçşzæŨzæşTăŞŃ
properties ăĂĆ äĳăăRİeIJĀăfİeĖAeĖFZăylæşlëğćĉt' gëİăăIJăĜ;æTăŕăŏZăZL'ăL'■ăşşăRİijZ

èóìèőž

ä:äaRräzëä;ɬçTlécDāoŽZāLçŽDæL;èsaçszælēæL'gëaŋNæZt'éÄŽçTlçŽDčszadNæcÄæšëijNä;NäçCijŽ

(continues on next page)

(continued from previous page)

```
if isinstance(x, collections.Iterable):
    ...

# Check if x has a size
if isinstance(x, collections.Sized):
    ...

# Check if x is a mapping
if isinstance(x, collections.Mapping):
```

är;çõqABCsâRřäzčëöl' æĹŚäznâĴĹæŮzäĴçŽDâAŽčszădNăcĂæšëiijNă;EæYřæĹŚäznâIJlăzčçăAăy■æĹ
ăZăăyžPythonçŽDăIJnêťĹæYřăyĂéŮlăĹlăĂAçijŮçĹNêř■ēĹAiiijNăĚŮçŽôçŽDăřsæYřçzŽă;ăæŽťăd'ŽçAĵæt'z
ăijžăĹŮçszădNăcĂæšëæĹŮëöl'ă;ăăzčçăAăRŸăĴŮæŽťăd'■æĹĹiijNêřŽæăăăAŽæŮăăijCăžŌēĹ■æIJnăśCăIJ

10.13 8.13 ăódçŎřæŮæĹăđNçŽDçszădNçžæĹš

éŮóécŸ

ăĵăæČšăŏŽăZĹæšŘăžZăIJlăśđæĂğêťNăĂijăyĹéĹcæIJĹ'éŽŘăĹŮçŽDăŮřæ■ôçzšăđDăĂĆ

èğçăEşşæŮzæăĴĹ

ăIJĹēŽăyĹéŮóécŸăy■iijNă;ăēIJĂēçAăIJlăřzæšŘăžZăŏđăĴNăśđæĂğêťNăĂijăŮŮēřŽăqNăcĂæšëăĂĆ
æĹ'ĂăžăăĵăēçAēĴăŏŽăZĹăśđæĂğêťNăĂijăĴĴæŮřiiijNêřŽçğ■æČĚăEĵăyNăEIJăăē;ăĴçŮĹăRŘēřăŽĹăĂĆ
ăyNéĹčçŽDăzčçăAăĴçŮĹăRŘēřăŽĹăŏđçŎřăžEăyĂăyĴçşçzçşçszădNăŠNêťNăĂijēĹNêřAăqEăđŮiijŽ

```
# 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)

```

ěĚŽăžŽčšzârſæŸřä;ăēĀăĹŽăžžčŽĐæŤřæ■ōăĹăđŇæĹŮčšzăđŇčšzčžščŽĐăšžčăĂăđĐăžžăĹăĹăĹăŮăĂĈăŸŇăĹăĹăſæŸřăĹŤăžžăăđăŽĚăăōŽăžĹčŽĐăŤřăĐčğ■ăŸ■ăŤŇčŽĐæŤřæ■ōčšzăđŇiijŽ

```

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

```

čĐăăŤŤŤăŤčŤĹčŤĹčŽăžŽčŤĹăăōŽăžĹæŤřæ■ōčšzăđŇiijŇæĹŤăžžăăđăŽăžĹăŸăŸăŸčšzăđŇiijŽ

```

class Stock:
    # Specify constraints
    name = SizedString('name', size=8)
    shares = UnsignedInteger('shares')
    price = UnsignedFloat('price')

```


(continued from previous page)

```
def __init__(self, name, shares, price):
    self.name = name
    self.shares = shares
    self.price = price
```

çDúãRÕætÑerTefZäyİçşzçŽDāsđæĀğēŦNāĀijçzēæİšüijNāRřāRŚçŎřāřzæšŘäžŽāsđæĀğçŽDēŦNāĀijēŁİ

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

ēŁŸæIJL'äyĀäžZæŁĀæIJřāRřäzēçŏĀāNŮäyŁēİççŽDäzççăĀüijNāĒüäy■äyĀçğ■æŸřā;ŁçŦİçşzèçĒēēřăŽİ

```
# 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
```

ãŘead'ŮäyÄçg■æŮzáijRæYřä;ŁçTlãĚČşziiZ

```
# 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
```

ěóľěőž

æIJñeŁĆä;ŁçTlãžEä;Ład'ŽénYčžgæLĀæIJřiiJNãÑĚæNñæRRèřřãZlãĀæuũăĚěčşzãĀAsuper()
 çŽDä;ŁçTlãĀAçşzècĚěěřãZlãŠNãĚČşzãĀČ äy■ãRřèÇ;ãIJlèŁŽéGNäYĀäyĀèřèçzEäšTãijĀæİëèőšiiJNä;EæY
 ä;EæYřiiJNæŁSãIJlèŁŽéGNèŁYæYřèEæRŘäYĀäyNãGäyŁeIJĀèEæşŁæDRçŽDçCzãĀČ

éçŮăĚĹiiJNãIJĹ Descriptor äşžçşzäy■ä;ääijŽçIJNãĹræIJL'äyŁ
 __set__() æŮzáşTiiJNã■'æşqæIJL'çŽyãžTçŽD __get__() æŮzáşTãĀČ
 æČædIJäYĀäyŁæRRèřřãžEäžĚæYřãžŎãžTãšCãõđä;Nã■ŮăĚyäy■eŎũăRŮæşRäyŁãşđæĀgãĀijçŽDèřiiJNéC
 __get__() æŮzáşTãĀČ

æLĀæIJL'æRRèřřãZlçşzéČ;æYřãşžãžŎæuũăĚěčşzæİëăõđçŎřçŽDãĀČæřTæČ
 Unsigned äŠN MaxSized èEæèũşăĚũăžŮčžgæLŁeGŁ Typed çşzæuũăĚěăĀČ
 èŁŽéGNãĹL'çTlãd'ŽçžgæLŁæİëăõđçŎřçŽyãžTçŽDãŁşèÇ;ãĀČ

æuũăĚěčşzçŽDäYĀäyŁæřTè;ČéŽ;çŘEèğççŽDãIJræŮZæYřiiJNèřČçTl

super() ħĠġæTřæŮüüijNăĵăăžŭăŷ■çšëeAşçl' ŭçnşëeAërČčTlăŞlăŷlăĚŭăĵŞçşzăĂĆ
 äĵăěIJăĕeAeŭşăĚŭăžŭŮçşzçzŞăŘĹăŘŌæL'■èČĵæ■čçăŏçŽDăĵĚçTlŭijNăžşăřsæŸřăĚĚăžăŘĹăĵIJăL'■èČĵăžğçT
 äĵĚçTlŭçşzçèçĚëērăŽlăŞNăĚČçşzēĂŽăŷŷăŘřăžççŏĂăNŮăžççăĂăĂĆăŷĹĹĹăŷd'ăŷlăĵNă■Řăŷ■ăĵăăĵĴăŘŚ

```
# Normal
class Point:
    x = Integer('x')
    y = Integer('y')

# Metaclass
class Point(metaclass=checkedmeta):
    x = Integer()
    y = Integer()
```

æL'ĂæIJL'æŮžæşTăŷ■üijNçşzçèçĚëērăŽlăŮžæăĹăžTërěæŸřæIJăÇAĵæt'zăŞNăIJăénŸæŸŌçŽDăĂĆ
 éçŮăĚĹijNăŏČăžŭăŷ■ăĵĹĕŷŮăžzăĵăĚŭăžŮăŮřçŽDăĹĂæIJřijNærTăçČăĚČçşzăĂĆăĚŭăñăĵijNèçĚëērăŽlă
 æIJăĂŘŌijNèçĚëērăŽlăĚŸèČĵăĵăŷžæŭăăĚëçşzçŽDăŽĚăžçæĹĂæIJăĹăăŏđçŌřăŘNăăŭçŽDăŢĹăđIJ

```
# 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

```

èŁŻçğ■æŮžâijRăőŽăzL'çŽDçsžèŭšăzNăL'■çŽDæŢŁæđIJăŸĂæăũijÑèĂŃăŸŦæL'ğèąŃéĂšăžęăijŽæŽt'ă

èõç;õäÿÄäÿlçõÄ■TçŽĐçşzädNásđæÄğçŽĐäÄijñijNèçĚēřāŽlæŪzāijRēēAæfTāzNāL■çŽĐæūūāĚēçşzçŽĐçŌřāIJlā;āāzTēřēāzEāzÿèGłāūsērzaōNāzEæIJnèŁĆāĚlėĆlāEĚāōzāzEāRğñijş^_^

10.14 8.14 áõđçŌřèGłāōŽāzL'áožāŽÍ

éŬóécŸ

ä;äæČşáođçŌřäÿÄäÿlèGłāōŽāzL'çŽĐçşzæĚēāġæNşāĚĚç;õçŽĐāōžāŽlçşzāŁşèČ;ñijNæfTāēĆāLŬēāġāŠN

èğčāEşæŪzæąŁ

collections áõŽāzL'āzEā;Łād'ŽæŁ;èśāāşžçşñijNā;Şä;äæČşèGłāōŽāzL'áožāŽlçşzçŽĐæŪūāĚāōčæfTāēĆā;äæČşèõl'ä;āçŽĐçşzæTřæNĀēf■āzçñijNēĆčāřsèõl'ä;āçŽĐçşzçzğæL'£
collections.Iterable ā■şāRñijŽ

```
import collections
class A(collections.Iterable):
    pass
```

äÿ■ēfGä;āēIJĀēēAáođçŌř collections.Iterable
æL'ĀæIJL'çŽĐæŁ;èśāæŪzæşTñijNāRēāŁŽāijŽæŁēēTŽ:

```
>>> 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;āāRĚēAáođçŌř __iter__() æŪzæşTāřsäÿ■āijŽæŁēēTŽāzE(āRCèĀČ4.2āŠN4.7āřRèŁĆ)āĀĆ

ä;āāRřāzēāĚĚlērTçİĀāŌžāōđä;NāNŪäÿÄäÿlāržèşāñijNāIJlėTŽēřræRRçd'žäÿ■āRřāzēæL;āĹRēIJĀēēAāō

```
>>> 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__
>>>
```

äÿNēlćæŸřäÿÄäÿlçõÄ■TçŽĐçd'žä;NñijNçzğæL'£èGłāÿŁēlćSequenceæŁ;èśāçşñijNāzūäÿTáođçŌřāĚČ

```
class SortedItems(collections.Sequence):
    def __init__(self, initial=None):
        self._items = sorted(initial) if initial is not None else []
```

(continues on next page)

(continued from previous page)

```
# Required sequence methods
def __getitem__(self, index):
    return self._items[index]

def __len__(self):
    return len(self._items)

# Method for adding an item in the right location
def add(self, item):
    bisect.insort(self._items, item)

items = SortedItems([5, 1, 3])
print(list(items))
print(items[0], items[-1])
items.add(2)
print(list(items))
```

ǎŔřăžěçIJŇÁĹŕiijŇSortedItemseũşæŽőěĂŽçŽĎăžŔĂĹŮæşăăžĂăžĹăyd' æăüiijŇæŦŕæŇAæĹ'ĂæIJĹ'ăyŷç
èçŽéĜŇéİcă;ŷçŦĹăĹŕăžEbisect æĹăăĹŮiijŇăőĈæŸŕăŷĂăŷĹăIJăŐşăžŔĂĹŮeăĹăŷ■æŔŞăĖăăĖĈçŦ'ăçŽ.

èóìèőž

ä;fçTl collections äy■çŽDæL;ësaaşžçsžāRřāzēçāōāflā;äëĠlāōŽāzL'çŽDāōžāZlāōđçŌřāžEæL'ĂæL
ä;äçŽDëĠlāōŽāzL'āōžāZlāi;Žæzāēūşād'gēČlāLĖççsžādNæçĂæşēēlJĀēēAñijŇāēCāyŇæL'Ăçd'žii;Ž

```
>>> 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 äy■āŁŁād'ŽæŁ;èšaqšzäijŽäyžäyÄäzŽäyÿëğAáožáZÍæŞ■ä;IJæRŘä;ŽézŸèöd'čŽĎãođĆ
èŁŽæüüäYÄæIëä;ääRÍéIJÄëçAáođçŎřéĆčäzŽä;ääIJÄæĎšăEr'èúččŽĎæŰzæşŢă■şăRřăĂĆăAĞèèŁ;ä;ăçŽĎçşz
collections.MutableSequence iijŃăçCăyŃiijŽ

```
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)

```

æĆædIJä;ääLZåzz Items çŽDåóä;NiiJNä;äaijŽåRŠçŎřåŏČæŦræŊAåGääzŎæL'ĂæIJL'çŽDæäyåŦČå
äyŊéIcæYřä;ŦçŦlæijŦçd'žiiJŽ

```

>>> 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
>>>

```

æIJñârRèLCârĲæYřârZPythonæL;èsaçsžâŁšèĲ;çŽĎæŁŽçăŰâijTçŎL'ăĂĆnumbers
 æĲaaiŰæRĲăĲZăžEăyĂăyĲçsžâijijçŽĎeũ\$æT'æTřçsžâĎNçŽyăĔşçŽĎæL;èsaçsžâĎNéZEăRĲăĂĆ
 âRřăžăăRĲèĂĆ8.12ârRèLCæĲæĎĎÉĂăæŽt'ăĎ'ŽèĲăŏŽăžĲ'æL;èsaçsžsžsžăĂĆ

10.15 8.15 ásdæĲğçŽĎăžčçŘĚèŏéŰŏ

éŰŏéćŲ

ăĲăæĲşârEæşŘăyĲăŏĎăĲNçŽĎăsdæĲğèŏéŰŏăžčçŘĚăĲăRăĲĚéĲăRăyĂăyĲăŏĎăĲNăy■ăŎžiiĲNçŽŏçŽĎă

èğçăĲşæŰzæăĲ

çŏĂă■TæĲèèrt'iiĲNăžčçŘĚæYřăyĂçğ■çijŰçĲNăĲăaiĲRiiĲNăŏĲârEæşŘăyĲæş■ăĲĲèĲçğzçzŽăRăĎ'ŰăyĂ
 æĲĲçŏĂă■TçŽĎăĲăijRăRřèĲ;æYřăĲRăyNéĲèĲŽæăũiiĲŽ

```
class A:
    def spam(self, x):
        pass

    def foo(self):
        pass

class B1:
    """çŏĂă■TçŽĎăžčçŘĚ"""

    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
```

ăĲĲæĎĲĲăžĔârşăyĎ'ăyĲæŰzæşTĲĲĲæĲăžčçŘĚiiĲNéĲçăžĲăĲŘĲéĲŽæăũăĲŽârşêuşăĎ'şăžĔăĂĲăĲăEæY
 éĲçăžĲăĲăĲçĲĲĲ __getattr__() æŰzæşTæĲŰŏŏyæĲŰæŽt'ăĲăžŽiiĲŽ

```
class B2:
    """ăĲăĲçĲĲĲ__getattr__çŽĎăžčçŘĚiiĲNăžčçŘĚæŰzæşTæĲŰŏŏyæĲŰæŽt'ăĲăžŽiiĲŽ"""

    def __init__(self):
```

(continues on next page)

(continued from previous page)

```
self._a = A()

def bar(self):
    pass

# Expose all of the methods defined on class A
def __getattr__(self, name):
    """
    → "èĚžÿłæŮzæşŦăĬJlěőĚéŮőčŽĎattributeäy■ā■ŸăĬJlčŽĎæŮŭăĂŽècñèřČčŦĭ
    the __getattr__() method is actually a fallback method
    that only gets called when an attribute is not found"""
    return getattr(self._a, name)
```

__getattr__ æŮzæşŦăŸăĬJlěőĚéŮőattributeäy■ā■ŸăĬJlčŽĎæŮŭăĂŽècñèřČčŦĭijNă;ĚčŦĭæijŦčd'ž

```
b = B()
b.bar() # Calls B.bar() (exists on B)
b.spam(42) # Calls B.__getattr__('spam') and delegates to A.spam
```

ăŖead'ŮäyĂäyłäzččŖĚă;Nă■ŖăŸŕăőđčŎŕăzččŖĚăĭăăijŖĭijNă;NăĚĆĭijŽ

```
# 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)
```

ă;ĚčŦĭĚžÿłäzččŖĚčşzæŮŭijNă;ăăŖĭĬJĂĚĚAčŦĭăőČăĭăNĚĚčĚäyNăĚŭăzŮčşză■şăŖĭijŽ

```

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

```

éÅžèƒGèĠåőŽăZŁ'ásđæĀğèőŁéŮóæŰzæşȚiijŃă;ăăRřăžčȚlăy■ăRŃæŰzăijRèĠåőŽăZŁ'ăžčĉŘĚçśžèàŃ

èóíèőž

ăžčĉŘĚçśžæIJŁ'æŮűăĂŽăRřăžëä;IJăyžçžğæŁƒçŽĐæŽŁăžčæŰzæąŁăĂĆăŃăęĆiijŃăyĂăyłçőĂă■ȚçŽĐ

```

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;ƒçȚlăžčĉŘĚçŽĐërİiijŃăřsæŸřăyŃéłçèƒŽæăüiijŽ

```

class A:
    def spam(self, x):
        print('A.spam', x)
    def foo(self):
        print('A.foo')

class B:
    def __init__(self):
        self._a = A()
    def spam(self, x):
        print('B.spam', x)
        self._a.spam(x)

```

(continues on next page)

(continued from previous page)

```
def bar(self):
    print('B.bar')
def __getattr__(self, name):
    return getattr(self._a, name)
```

ā;ŠāōđčŌrāzččŘĚāíāijRæŮüijNēfYæIJL'āžZčzĚēLĆéIJĀēēAæšlæDRāĀĆ
éēŮāĒĹijN__getattr__() āōđēŽĚæYřäYĀäylāŔŌād'ĠæŮzæšTijNāRlæIJL'āIJlāsđæĀgäy■ā■YāIJlæŮ
āZāæ■d'ijNāēCāđIJāzččŘĚčsāōđā;NæIJnēznæIJL'ēfZāylāsđæĀgčZĎērliijNēCčāzLāy■āijŽēgēāRŠēfZāylā
āRēād'ŮijN__setattr__() āŠN__delattr__() éIJĀēēAēcíāđ'ŮčZĎē■TæšTælēāNžāLEāzččŘĚāōđ
_obj čZĎāsđæĀgāĀĆ āyĀäylēĀZāyycZĎčzēāōZæYřāRlāzččŘĚēCčāzZāy■āzēāyNāLŠčžf
_āijĀād't'čZĎāsđæĀg(āzččŘĚčsāRlæŽt'ēIJšēcnāzččŘĚčsčZĎāĒnāĒsāsđæĀg)āĀĆ
ēfYæIJL'āyĀčCžéIJĀēēAæšlæDRčZĎæYřijN__getattr__()
ārzāžŌād'gēČlāLEāzēāRŇāyNāLŠčžf(____)āijĀāgNāŠNčzŠār;čZĎāsđæĀgāzūāy■ēĀĆčTlāĀĆ
ærTāēČijNēĀČēZŠāēCāyNčZĎčsžijZ

```
class ListLike:
    """__getattr__
    →ārzāžŌārŇāyNāLŠčžfāijĀāgNāŠNčzŠār;čZĎāŮzæšTæYřäy■ēČ;čTlčZĎiijNēIJĀēēAāyĀäylāy1
    →"""

    def __init__(self):
        self._items = []

    def __getattr__(self, name):
        return getattr(self._items, name)
```

āēCāđIJæYřāLZāzžāyĀäyListLikeāržēšāijNāijZāRŠčŌrāōCæTřæNĀæZōēĀŽčZĎāLŮēāíæŮzæšTijN
āJĚæYřā■āy■æTřæNĀlen()āĀAāĒČčt'āæšēāL'č;■L'āĀĆā;NāēČijZ

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

āyžāžĚēōl'āōCæTřæNĀēfZāžZæŮzæšTijNā;āāfĒēāzæL'NāLlčZĎāōđčŌrēfZāžZæŮzæšTāzččŘĚijZ

```
class ListLike:
    """__getattr__
    →ārzāžŌārŇāyNāLŠčžfāijĀāgNāŠNčzŠār;čZĎāŮzæšTæYřäy■ēČ;čTlčZĎiijNēIJĀēēAāyĀäylāy1
    →"""
```

(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]

```

11.8ärRèŁĆèŁŸæIJL'äyÄäyŁaIJŁèŁIJŁÍNæŰzæşŤerČçŤÍçŎřácČäy■ä;ŁçŤÍázčçŘEçŽDä;Nā■ŘāĂĆ

10.16 8.16 ăJÍčśzäy■ăŏŽăzL'ăd'ŽăyŁæđĐéĂăăŽÍ

éŰőécŸ

ä;ăæČşăŏđçŎřäyÄäyŁçśzïijŇéŽd'ăžEä;ŁçŤÍ __init__()
æŰzæşŤăd'ŰïijŇèŁŸæIJL'ăŰüázŰæŰzăijRăŔřăzeăŁiăğNăŇŰăŏČăĂĆ

èğcăEşæŰzæăŁ

äyžăžEăŏđçŎřăd'ŽăyŁæđĐéĂăăŽÍïijNă;ăéIJĂèçAă;ŁçŤÍăŁŕçşzæŰzæşŤăĂĆă;NăçĆïijŽ

```

import time

class Date:
    """æŰzæşŤăyÄïijŽă;ŁçŤÍčśzæŰzæşŤ"""
    # Primary constructor
    def __init__(self, year, month, day):
        self.year = year
        self.month = month
        self.day = day

    # Alternate constructor
    @classmethod
    def today(cls):

```

(continues on next page)

(continued from previous page)

```

t = time.localtime()
return cls(t.tm_year, t.tm_mon, t.tm_mday)

```

çŹt' æŒëŕČŹŤlčŝzæŮzæŝŤā■ŝāŔŕiijNäyNéÍcæŸŕä;£çŤlčd'žä;NŕiijŽ

```

a = Date(2012, 12, 21) # Primary
b = Date.today() # Alternate

```

èóìèőž

çŝzæŮzæŝŤçŽDäyÄäyſäyžèeAçŤléĀŤŕſæŸŕăŏŽăzL'ăd'ŽăyſæđDěĀăăŽlăĀĈăăŔŮăyÄäyſ
class ä;IJäyžçñňäyÄäyſăŔĈæŤŕ(cls)ăĀĈ ä;ăăžŤèŕeæŝſæĐŔăĹŕăžEè£ŽăyſçŝzècŋçŤlæİăĹŽăžžăžűè£ŤăŽđă

```

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 āĹŽăžžăy■èŕČŹŤlinitæŮzæŝŤçŽDăŏđă;N

éŮŏécŸ

ă;ăăČŝăĹŽăžžăyÄäyſăŔđă;NŕiijNă;EæŸŕăyNæIJŽçzŤè£ĜæĹğèăN __init__()
æŮzæŝŤăĀĈ

èğčăEŝæŮzæſĹ

ăŔŕăžèeĂŽè£Ĝ __new__() æŮzæŝŤăĹŽăžžăyÄäyſæIJăĹĹăğNăŨŮçŽDăŏđă;NăĀĈă;NăeĈèĂĈèŽŖă

```

class Date:
    def __init__(self, year, month, day):
        self.year = year
        self.month = month
        self.day = day

```

äyNéÍcæiijŤçd'žăeĈă;Ťăy■èŕČŹŤl __init__() æŮzæŝŤæİăĹŽăžžè£ŽăyſDateăŏđă;NŕiijŽ

```

>>> d = Date.__new__(Date)
>>> d
<__main__.Date object at 0x1006716d0>
>>> d.year
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>

```

(continues on next page)

(continued from previous page)

```
AttributeError: 'Date' object has no attribute 'year'
>>>
```

çzŞæđIJăRřæžčIJŇăĹřijŇèŁŻăyĤDateăôđăĹŇčŽĐăśđæĂğyearèŁŸăy■ăŸăIJĹijŇæĹĂăžăă;ăéIJĂèçAă

```
>>> data = {'year':2012, 'month':8, 'day':29}
>>> for key, value in data.items():
...     setattr(d, key, value)
...
>>> d.year
2012
>>> d.month
8
>>>
```

èõlèõž

ă;ŞăĹŚăžŇăIJăR■ăžRăĹŮăřzèşăăĹŮèĂĖăôđçŎřăşRăyĹçşzăŮzăşTăđĐéĂăăĜ;ăTřăŮúéIJĂèçAçzTè
__init__() æŮzăşTăĹèăĹZăžžăřzèşăăĂĆăĹŇăçCiiŇŇăřžăžŎăyĹéĹčçŽĐDateăĹèèôšijŇæIJĹăŮăăŽă;ă
today() ĩijŽ

```
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
```

ăRŇăăũijŇăIJă;ăăR■ăžRăĹŮăŇŮJSONăTřă■ŎăŮăăžğçTşăyĂăyĹăçCăyŇčŽĐă■ŮăĖyăřzèşăijŽ

```
data = { 'year': 2012, 'month': 8, 'day': 29 }
```

ăçCăđIJă;ăăçşăřĖăôĈç;Ňă■căĹRăyĂăyĤDateçşzăđŇăôđăĹŇijŇăRřæžă;ĲçŤĹăyĹéĹčçŽĐăĹăIJăřăĂĆ

ă;Şă;ăéĂŽèŁĜèŁŻçğ■éĹđăyŷèĝĐăŮzăijRăĹèăĹZăžžăôđăĹŇčŽĐăŮăăĂŽijŇăIJĂăç;ăy■èçAçŽt' æŎèăă
ăRăăĹZçŽĐĹijŇăçCăđIJèŁŻăyĹçşză;ĲçŤĹăžĖ __slots__ ăĂăproperties ăĂăde-
scriptors æĹŮăĖŮăžŮénŸçžğăĹăIJřçŽĐăŮăăĂŽăžççăĂăřşăijŽăđ'săŤĹăĂĆ
èĂŇèŁZăŮăăĂŽă;ĲçŤĹ setattr() æŮzăşTăijŽèŏĹă;ăçŽĐăžççăĂăRŸăĹŮăŽt'ăĹăéĂŽçŤĹăĂĆ

(continued from previous page)

```
'''
__slots__ = ()

def __setitem__(self, key, value):
    if not isinstance(key, str):
        raise TypeError('keys must be strings')
    return super().__setitem__(key, value)
```

èfZázZçszā■TçNñā;£çTlétuæIēæšæIJL'āzzā;TæDRāzL'ijNāzNāōđāyLæCædIJā;āāŌzāōđā;NāNŪāzzā
āōČāznæYřçTlæIēéĀZè£Gād'ŽçzğæL'£æIēāŠNāĒūāzŪæYāārDārZèšæuūāĒēā;£çTlçŽDāĀĆā;NāēČiijŽ

```
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'
```

èfZāyIā;Nā■Rāy■iijNāRřāzèçIJNāLřæuūāĒēçszèušāĒūāzŪāušā■YāIJlçŽDçsz(æřTāēĆdictāĀAdefaultd
çzŠāRĹLāRŌāršèČ;āRŠæNēæ■cāyāŁšæTlāzĒāĀĆ

èõlèõž

æuūāĒēçszāIJlæāGāĠēāžŠāy■ā;Łād'ŽāIJræŪzéČ;āGžçŌřè£GiiijNéĀŽāyēČ;æYřçTlæIēāČRāyLéIcéČ
āōČāznāzšæYřād'ŽçzğæL'£çŽDāyĀāyIāyžèēAçTlēĀTāĀĆæřTāēČiijNā;Šā;āçijŪāĒZç;ŠçzIJāzççāAæUūāĀŽ
ā;āāijŽçzRāyā;£çTl socketserver ælāāIŪāy■çŽD ThreadingMixIn
æIēçzŽāĒūāzŪç;ŠçzIJçŽyāĒēçszāčđāŁāād'Žçz£çIŌNæTřæNāāĀĆ
ā;NāēČiijNāyNéIcæYřāyĀāyIād'Žçz£çIŌNçŽDXML-RPCæIJ■āŁāiijŽ

```
from xmlrpc.server import SimpleXMLRPCServer
from socketserver import ThreadingMixIn
class ThreadedXMLRPCServer(ThreadingMixIn, SimpleXMLRPCServer):
    pass
```

ārNæUūāIJlāyĀāzŽād'ğādNāzŠāŠNæAĒēđūāy■āzšāijŽāRŠçŌřæuūāĒēçszçŽDā;£çTl iijNçTlēĀTāRŌNæ
ārZāžŌæuūāĒēçsz iijNæIJL'āGāçĆzéIJĀèēAçōrā;RāĀĆēēŪāĒLæYřriijNæuūāĒēçszāy■ēČ;çŽt æŌēēcāō

àĖŮæñāijŃæũāĖĖċšzæšqæIJL'èĠlāũŝçŽDçŁŮæĀAāfæAřijŃāzšāřsæŸřèrt'āóČāznāzŮæšqæIJL'āóŽāzL'
 __init__()
 æŮzæšŤijŃāzŮāyŤæšqæIJL'āóđäĭŃāsdæĀğāĀĆ
 èŁŽāzšæŸřāyžāzĀāzŁæŁŚāznāIJlāyŁéÍcæŸŌçāóāóŽāzL'āžĖ__slots__ = () āĀĆ
 èŁŸæIJL'āyĀçğāāóđçŌřæũāĖĖċšzçŽDæŮzāijRāřsæŸřā;ŁçŤÍçšzèċĖéērāŽÍrijŃāçCāyŃæL'Āçd'žijŽ

```

def LoggedMapping(cls):
    """çññāzŃçğ■æŮzāijŖiijžä;ŁçŤÍçšzèċĖéērāŽÍ"""
    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 {} = {}'.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
  
```

èŁŽāyĭæŤŁæđIJèũšāzŃāL'■çŽDæŸřāyĀæũççŽDrijŃèĀŃāyŤāy■āĖ■éIJĀèçAā;ŁçŤÍāđ'ŽçžgæL'ŁāžĖāĀ
 āŖCèĀĆ8.13ārRèŁCæšççIJŃæŽr'ād'ŽæũāĖĖċšzāšŃçšzèċĖéērāŽÍçŽDäĭŃā■RāĀĆ

10.19 8.19 āóđçŌřçŁŮæĀAāřzèšqæŁŮèĀĖçŁŮæĀAæIJž

éŮóéćŸ

äĭāæČšāóđçŌřāyĀāyĭŁŮæĀAæIJžæŁŮèĀĖæŸřāIJlāy■āŖŃçŁŮæĀAāyŃæL'gèāŃæš■äĭIJçŽDāržèšāij

èğcāĖşæŮzæāŁ

āIJāĭŁāđ'ŽçÍŃāžŖāy■ijŃæIJL'āžŽāržèšāijŽæāžæ■ŁŮæĀAçŽDāy■āŖŃæĭæL'gèāŃāy■āŖŃçŽDæš

```

class Connection:
    """æŽŏéĀŽæŮžæŁiijŇăĕ;ăd'ŽăÿłăĹd'æŮëŗăŘëiijŇæŤĹçŎĞă;ŎăÿŇ~~"""

    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'

```

èĤŽæăăăĒŽæIJĹăĹăd'ŽçijžçĈziiŇĒēŮăĒĹæŸřăžçăĀăd'Ĺăd'æĬĈăžĒiijŇăĕ;ăd'ŽçŽĎăĹăžăžăĹăd'æŮ
 ăŽăăÿžăÿĂăžŽăÿÿèĢĀçŽĎăŖăĬJæřŤăçĈread()ăĂwrite()æřŘăňăæĹğĕăŇăĹăĈĈ;éIJĂĕĕĂăĹğĕăŇăĕĈĂă
 äÿĂăÿłăŽŤăĕ;çŽĎăĹăđăŖŤăŸřăÿžæřŘăÿłçĹŮăĂăăŏŽăžĹăÿĂăÿłăřžĕăĭijŽ

```

class Connection1:
    """æŮřæŮžæŁăăĤăĤăĤăřžæřŘăÿłçĹŮăĂăăŏŽăžĹăÿĂăÿłçŖšž"""

    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):

```

(continued from previous page)

```
print('writing')

@staticmethod
def open(conn):
    raise RuntimeError('Already open')

@staticmethod
def close(conn):
    conn.new_state(ClosedConnectionState)
```

äyÑéÍcæYřä;ŁçTłæijTčd'žiiž

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

èóìèőž

ăĉĆădIJăzčĉăAăy■ăGžĉŎřăd'łăd'ŽčŽDăĭăžăúăĽăd'ăŮ■ĕr■ăRĕĉŽDĕrłiijNăzčĉăAăřsăijŽăRŸă;ŮĕŽ;ăžĕ
ĕŁŽĕGŇĉŽDĕğĉăEşăŮžăăĽăYřăřEăřRăyłĉŁŭăĂăĽ;ăRŮăGžăĭăăőŽăžĽăĽăRăyĂăyłĉsžăĂĈ

ĕŁŽĕGŇĉIJNăyĽăŎžăIJĽĉĆžăĕGăĂhijNăřRăyłĉŁŭăĂăăřžĕśăĕĈ;ăRłăIJĽĕÍŽăĂăŮžăşTiiijNăžăúă
ăőđĕŽĚăyŁiiijNăĽĂăIJĽĉŁŭăĂăăřăăăřĕĈ;ăRłă■YăĆĭăIJĭ Connection
ăőđă;Năy■ăĂĈăIJĭăşžĉsžăy■ăăőŽăžĽĉŽD NotImplementedError
ăYřăyžăžĚĉăăĭă■RĉsžăăđĉŎřăžĚĉŽyăžTĉŽDăŮžăşTăĂĈĕŁŽĕGŇă;ăăĽŮĕăőyĕŁYăĈşă;ŁçTł8.12ăřRĕĽĈ

èö;ĕőăăĭăijRăy■ăIJĽăyĂĉğ■ăĭăăijRăRŋĉŁŭăĂăăĭăăijRiiijNĕŁŽăyĂăřRĕĽĈĉőŮăYřăyĂăyłăĽĭă■ăăĽ

10.20 8.20 éĀŽèŁĠā■ŮčņēäÿšërČčŤlāržèśaqæŮzæşŤ

éŮóécŸ

äĵäæIJL'äÿÄäÿlā■ŮčņēäÿšäĵcāijŖčŽĎæŮzæşŤāŖ■čğřijŇæČšéĀŽèŁĠāőČërČčŤlæşŖäÿlāržèśaqæŽĎáržā

èğčāEşşæŮzæqĹ

æIJĀčóĀā■ŤčŽĎæČĚāEğřijŇāŖřäžèäĵčŤlĭ `getattr()` ĩĵŽ

```
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)
```

āŖēād'ŮäÿĀčğ■æŮzæşŤæŸŖäĵčŤlĭ `operator.methodcaller()` ĩĵŇäĹŇæČřijŽ

```
import operator
operator.methodcaller('distance', 0, 0)(p)
```

āĵŠäĵäéIJĀèēAéĀŽèŁĠčŽÿāŖŇčŽĎāŖČæŤŖād'ŽæñæërČčŤlæşŖäÿlæŮzæşŤæŮřijŇäĵčŤlĭ
`operator.methodcaller` āŖšäĵĹæŮžäĵčžĚāĀČ æŖŤæČäĵäéIJĀèēAæŌšāžŖäÿĀčşžāĹŮčŽĎčČřijŇāŖ

```
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))
```

èóíèőž

ərČçTlāyĀäyŁæŪzæʃTāōđéZĚäyŁæYřayd' éČlčNŋčnNæŞ■ā;IJiijNčnŋäyĀæ■ēæYřæšēæL' ĸāsđæĀgġiijNč
 āZāæ■d' iijNäyžāZĚərČçTlāēšRāyŁæŪzæʃTiiijNā;āāRřazēēēŪāĒĒēĀZēŁG getattr()
 æĒēæšēæL' ĸāŁrēŁZāyŁāsđæĀgġiijNčDūāRŌāE■āŌzāzēāĠ;æTřæŪzāijRērČçTlāōČā■şāRřāĀČ
 operator.methodcaller() āŁZāzzāyĀäyŁāRřērČçTlāržēsāiijNāzūāRŊNæŪūāRŘā;ZæL'ĀæIJL'āf
 čDūāRŌēČçTlčZDæŪūāĀZāRĒēIJĀēēĀāRĒāōđā;NāržēsāiijāēĀŚčzZāōČā■şāRřiijNārTāēČġiijZ

```
>>> p = Point(3, 4)
>>> d = operator.methodcaller('distance', 0, 0)
>>> d(p)
5.0
>>>
```

éÅžèfGæŨzæsŦår■çğr■ŨçņęäÿsæiëèřČťlæŨzæsTéÅžäÿyăĜžçŎřăIJléIĀëèAælaæNš
case èr■ārĒæLŨāōđčŎřèōfÉŨōèĀĒælaajrČžDæUūăĂžăĂĈ
ârĈèĂĈăÿNăÿĀârĒèĹĈèŎūăRŨăŽt'ăđ'ŽénŸçžğă;Nă■RăĂĈ

10.21 8.21 aóđçÖřèó£éŮóèĂĚæíaǻijŘ

éŮőécÿ

ä;äëeAåd'DčŘEçTśad'géGRäy■aŘNčszadNčŽDǎřzèsaçzDǎĹŘčŽDǎd'■aeIcæTřæ■oçzŞædDiiJNǣrRäyÄ
ærTǣeCiiJNéA■aŎEävÄävǣaŚǎ;čczSædDiiJNčDüaŘŎǣāzæ■oæfRǎvǣlĹčCčzčŽDčŽvǎžTčĹuǣÄAæL'gëaN

èġčǎẸșæŮźæǻŁ

æʒéGÑéAǵǻLŕçŽĐēŬōécYǎIJlçijŮčlNécEǻşşäy■æYřǻĹŁæZőeA■çŽĐrijNæIJL æŮũǻǺŽaijZæđDǻžžǻ
ǻAǵěõ; ä;ǻëeAǻEZǻyÄǻýłeałcd'zǻTrǻ■ealē; ; ãijRçŽĐclNǻžRriijNéCčázĽǻ;ǻǻRrēČ; ēIJǻëeAǻőZǻžL'ǻeCǻyN

```
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):
    pass
```

(continues on next page)

(continued from previous page)

```
pass

class Mul(BinaryOperator):
    pass

class Div(BinaryOperator):
    pass

class Negate(UnaryOperator):
    pass

class Number(Node):
    def __init__(self, value):
        self.value = value
```

çĐúâŔŌâlŦçŦlêŁŻăžŹçşzæđĐăžžâŦŊăĖŮæŦŕæ■ōçzŞæđĐiijŊăĕCăyŊæL'Ăçd'žiižŽ

```
# 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)
```

èŁŻæăăâĀŹçŽĐĕŮŏĕĲæŸŕăŕzăžŌăŕŔăyŦeăŦĕŦăijŔiijŊăŕŔăŋăĕĲĕĕĀĕĜ■æŰŕăŏŽăžL'ăyĂĕĀ■iijŊăŕ
èŁŻĕĜŊăĻSăžŋăŦçŦlêŁŦĕŦŮŏĕĂĖăŦăijŔăŔŕăžĕĕŦăĻŦĕŁŻæăăĕŹĐçŽŏçŽĐiijŽ

```
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__))
```

ăyžăžĒăŦçŦlêŁŻăyŦçşžiiŊŊăŔŕăžăžăŏŽăžL'ăyĂăyŦçşzçžğæL'ŦăŏĲăžăŷăyŦăŏđçŌŕăŔĐçğ■
visit_Name() æŰžăşŦiijŊăĖŮăy■NameæŸŕnodeçşzăđŊăĂĲ
ăŦŊăĕĲiijŊăĕĲăđĬăŦăăĲşăşĲĕăŦĕŦăijŔçŽĐăĂiijŊŊăŔŕăžĕĕŁŻæăăĒĒiijŽ

```
class Evaluator(NodeVisitor):
    def visit_Number(self, node):
        return node.value

    def visit_Add(self, node):
        return self.visit(node.left) + self.visit(node.right)
```

(continues on next page)

(continued from previous page)

```
def visit_Sub(self, node):
    return self.visit(node.left) - self.visit(node.right)

def visit_Mul(self, node):
    return self.visit(node.left) * self.visit(node.right)

def visit_Div(self, node):
    return self.visit(node.left) / self.visit(node.right)

def visit_Negate(self, node):
    return -node.operand
```

ä;ŁçŦŦçd'žä;ŦiijŽ

```
>>> e = Evaluator()
>>> e.visit(t4)
0.6
>>>
```

ä;IJäyžäyÄäyläy■āŦŦçŽDä;Ŧā■ŦiijŦäyŦéŦāōŽäzL'äyÄäylçszāIJläyÄäylæāLäyŁéŦārEäyÄäylèāŦä;ā

```
class StackCode(NodeVisitor):
    def generate_code(self, node):
        self.instructions = []
        self.visit(node)
        return self.instructions

    def visit_Number(self, node):
        self.instructions.append(('PUSH', node.value))

    def binop(self, node, instruction):
        self.visit(node.left)
        self.visit(node.right)
        self.instructions.append((instruction,))

    def visit_Add(self, node):
        self.binop(node, 'ADD')

    def visit_Sub(self, node):
        self.binop(node, 'SUB')

    def visit_Mul(self, node):
        self.binop(node, 'MUL')

    def visit_Div(self, node):
        self.binop(node, 'DIV')

    def unaryop(self, node, instruction):
        self.visit(node.operand)
```

(continues on next page)

(continued from previous page)

```
self.instructions.append((instruction,))

def visit_Negate(self, node):
    self.unaryop(node, 'NEG')
```

ä;fcTÍcd'žä;NriiŽ

```
>>> s = StackCode()
>>> s.generate_code(t4)
[('PUSH', 1), ('PUSH', 2), ('PUSH', 3), ('PUSH', 4), ('SUB',),
 ('MUL',), ('PUSH', 5), ('DIV',), ('ADD',)]
>>>
```

ëöléöž

álŽaijAägNçŽDæUúäÄZä;ääRřèČ;aijŽaEZad'géGRčŽDif/elseèr■āRēæIēāōđçŎriiŇ
èfŽéGŇèōféUōēĀĒæIāaijRčŽDāē;ād'ĐārśæŸréĀŽèfĜ
æIēēŌūāRŮčŽyāžTčŽDæŮzæšTiiŋNāzūāL'čTléĀŠā;ŠæIēéA■āŎĒæL'ĀæIJLčŽDēLCčCžiiŋŽ

```
def binop(self, node, instruction):
    self.visit(node.left)
    self.visit(node.right)
    self.instructions.append((instruction,))
```

èfŸæIJLäyĀçCžéIJĀèēAæŇGāGžçŽDæŸriiŋNèfŽçg■āLĀæIJřázšæŸřāōđçŎřāĒūāžŮēr■ēIĀy■switch
ærTāēČriiŇāçCādIJä;āæ■čāIJlāEZäyĀäyIHTTPæqEæđūriiŇä;ääRřèČ;aijŽaEZèfŽæāūāyĀäyIērūāsČāLEāR.

```
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
```

èōféŮōēĀĒæIāaijRāyĀäyIçijžçCžārśæŸřāōČāyēēĜ■ā;IètŮēĀŠā;ŠriiŇāçCādIJæTřæ■ōçžŠæđDāŋNāēŮ
æIJLæŮūāÄZäiŋŽēūĒèfĜPythonçŽDēĀŠā;ŠæūsāžēçŽRāLŮ(āRČēĀČ
getrecursionlimit())āĀČ

āRřāžēāRČçĚg8.22ārRēLCriiŇNāL'čTlçTšæLRāŽlæLŮèf■āžčāŽlæIēāōđçŎřéIdéĀŠā;ŠéA■āŎĒçōŮæšT

āIJlēușēgçæđRāŠŇçijŮērŠçZyāĒšçŽDçijŮçIŇāy■ā;fcTléōféŮōēĀĒæIāaijRæŸréIdāyŷāyŷēgAçŽDāĀČ
PythonæIJñēžnçŽD ast æIāāIŮāAijā;ŮāĒšæšlāyŇriiŇNāRřāžēāŎžçIJNçIJNæžRčāĀāĀČ
9.24ārRēLCæijTçd'žāžEäyĀäyIāL'čTl ast æIāāIŮæIēād'ĐçRĒPythonæžRāžççāAçŽDä;Nā■RāĀČ

ěĂĈěŽŚăĉĆăŸŇăžĉăĀĭĭŇéĀăŎĚăŸĂăŸlèăĭèĭĭăĭĭŖĉŽĎăăŚĭĭŽ

```
class UnaryOperator(Node):
    def __init__(self, operand):
        self.operand = operand

class BinaryOperator(Node):
    def __init__(self, left, right):
        self.left = left
        self.right = right

class Add(BinaryOperator):
    pass

class Sub(BinaryOperator):
    pass

class Mul(BinaryOperator):
    pass

class Div(BinaryOperator):
    pass

class Negate(UnaryOperator):
    pass

class Number(Node):
    def __init__(self, value):
        self.value = value

# A sample visitor class that evaluates expressions
class Evaluator(NodeVisitor):
    def visit_Number(self, node):
        return node.value

    def visit_Add(self, node):
        return self.visit(node.left) + self.visit(node.right)

    def visit_Sub(self, node):
        return self.visit(node.left) - self.visit(node.right)

    def visit_Mul(self, node):
        return self.visit(node.left) * self.visit(node.right)

    def visit_Div(self, node):
        return self.visit(node.left) / self.visit(node.right)

    def visit_Negate(self, node):
        return -self.visit(node.operand)
```

(continues on next page)

(continued from previous page)

```
if __name__ == '__main__':
    # 1 + 2*(3-4) / 5
    t1 = Sub(Number(3), Number(4))
    t2 = Mul(Number(2), t1)
    t3 = Div(t2, Number(5))
    t4 = Add(Number(1), t3)
    # Evaluate it
    e = Evaluator()
    print(e.visit(t4)) # Outputs 0.6
```

æĈædIIætNæĈŮăŖĈæŋăd'æŭŖĈăzLăyLèĤřĈŽĎEvaluatorăřŖăijŽăd'ŖăĤLăijŽ

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

ĉŖăIIăĤŖăzŋĈăĤăăĤăăĤăăNăyLéĤĉĈŽĎEvaluatorăijŽ

```
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)
```

ăĤăăăăĤŖăăNăijNăřăyăăăijŽăĤéĤŽăăĤăijŽ

```
>>> a = Number(0)
>>> for n in range(1, 100000):
...     a = Add(a, Number(n))
...
>>> e = Evaluator()
>>> e.visit(a)
4999950000
>>>
```

æĈæđIĴä;æĕŸæĈşæûzâŁăăĔüăzŮëĠăôŽăzL' éĂzè;ŚăzşæşæéŮőécŸiijŽ

```
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
    ...
```

äyŇéİcæŸřċôĂă■ŤċŽĎæŧŇèřŤiijŽ

```
>>> e = Evaluator()
>>> e.visit(t4)
Add: <__main__.Add object at 0x1006a8d90>
left= 1
right= -0.4
0.6
>>>
```

èőİeőž

èĚŽăyĂăřŔëĹĈæĹŚăznæijŤċđ'žăžĚċŤşæĹŔăŽĹăŠŇă■ŔċĹŇăĴċĹŇăžŔæŎğăĹúæŧAæŮzéİċĈŽĎăijžăđ'ğ
éAĤăĔĔ■éĂŞă;ŞċŽĎăyĂăyĹéĂŽăyŷæŮžæşŤæŸŕă;ĤċŤĹăyĂăyĹæăĹæĹŮéŸşăĹŮċŽĎæŤŕæ■ŎċžŞæđĎăĂĈ
ă;ŇăĕĈiijŇăûşăžĕăijŸăĔĹċŽĎéA■ăŎĚċôŮæşŤiijŇċňăyĂæŋăċŕăĹŕăyĂăyĹéĹĈċĈzæŮŭăŕĚăĔăŎŇăĔĕăă
æŮžæşŤċŽĎæăyăĤĈæĂİèŭŕăŕşæŸřèĚŽăăăĂĈ

ăŔĕăđ'ŮăyĂăyĹéIĴăĕAċŔĚĕğċĈŽĎăŕşæŸřċŤşæĹŔăŽĹăy■yieldĕŕ■ăŔĕăĂĈă;ŞċŕăĹŕyieldĕŕ■ăŔĕăŮŭiij
ăyĹéİċĈŽĎă;Ňă■Ŕă;ĤċŤĹéĚŽăyĹæĹĂæIĴŕæĹăžċæŽĤăžĚĕĂŞă;ŞăĂĈă;ŇăĕĈiijŇăžŇăĹ'■æĹŚăznæŸřèĚŽăăă

```
value = self.visit(node.left)
```

ċŎŕăIĴă■cæĹŔyieldĕŕ■ăŔĕiijŽ

```
value = yield node.left
```

ăŏĈăijŽăŕĚ node.left èŤăŽđċžŽ visit() æŮžæşŤiijŇċŮăŔŎ visit()

æÚzæşTërÇşTlëCçäyIèLCçCzçZyāžTçŽĐ visit_Name() æÚzæşTāĀĆ yield-
æŽCæŰuārEçlNāžRæŌgāLūāZlèol' āGžçzŽerÇşTlëĀĒiijNā;ŞæL'gèaŊāōNāRŌiijNçzŞæđIJāijŽetŊāĀijçzŽv
çIJNāōNēŁZāyĀārRèLCiijNā;āāzşèōyæČşāŌzāræL'çāĒūāōČæşæIJL'yieldèr■āRèçŽĐæÚzæāLāĀĆā;E
āçNāçĆiijNāyžāzEæūLéZd' éĀŞā;ŞiijNā;āāŁĒēāzèçAçzt' æLd'āyĀāyIæāLçzŞæđĐiijNāçCæđIJāy■ā;ŁçTlçTşæ
āōđéŽĒāyLiiNā;ŁçTlīyieldèr■āRēāRfāzèèol'ā;āāĒŽāGžēlđāyāyæijČāžōçŽĐāzççāAīijNāōCæūLéZd'āžEéĀŞā;

10.23 8.23 āçŁçŌrāijTçTlæTřæ■ōçzŞæđĐçŽĐāĒĀ■YçōāçŘĒ

éŰōécŸ

ā;āçŽĐçlNāžRāLZāzžāžEāçLād'ŽāçŁçŌrāijTçTlæTřæ■ōçzŞæđĐ(æřTāçCæāSāĀĀāZçāĀĀèğČārşèĀĒæ

èğçāĒşæŰzæāL

āyĀāyIçōĀā■TçŽĐāçŁçŌrāijTçTlæTřæ■ōçzŞæđĐāçNā■RārşæYřāyĀāyIæāŞā;ççzŞæđĐiijNāRŊāžşèLC
èŁŽçğ■æČĒāĒēāyNīijNāRfāzèèĀČèZŞā;ŁçTlīweakrefāžŞāy■çŽĐāijsāijTçTlāĀĆāçNāçĆiijŽ

```
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
```

èŁŽçğ■æYřæČşæŰzāijRāĒĀèōyparentéIŽézYçzLæ■cāĀĆāçNāçĆiijŽ

```
>>> root = Node('parent')
>>> c1 = Node('child')
>>> root.add_child(c1)
>>> print(c1.parent)
```

(continues on next page)

(continued from previous page)

```
Node('parent')
>>> del root
>>> print(c1.parent)
None
>>>
```

èóìèőž

åŁłçŒřąjȚȚȦčŽĐæȚřæ■őçzŞæđĎăİJlPythonäy■æÿřäyĂäylâŁŁæçÿæL'ŇçŽĐěŮőécÿñijŃăZăäyžæ■čây
äŁŇăęĆėĂĈēŽŚăęĈăyŇăžčĉăĄñjŻ

```
# 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
```

äyÑéÍcæĹSäznä;ŁçŦlêŁZäyŁazččAæIěaAŽäyĂăžZadČaIJ;ăZdæŦüerŦelŦiiJŽ

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

āRřäzēčIJNāLřriiŇŇæIJAāRŌäyĀäyłčŽDāLāēZd' æUūæL' Šā■řēr■āRēæšæIJL' āGžčŌřāĀCāŌšāZāæYřPy
 ā; ŠayĀäyłřzēsacŽDaijTčTlæTrāRŸæLŔ0čŽDæUūāĀZæL' ■aijŽčnNā■šāLāēZd' æŌL' āĀCēĀNāržāžŌā; łčŌř
 āZāæ■d' iijNāIJlāyŁēlĀcā; Nā■Rāy■æIJAāRŌéČlāLEřiiŇŇČLūēŁČčCžāŠNā■l' ā■RēŁČčCžāžŠčZyæNēæIJL' āřza

PythonæIJL'âRëad'ŨçŽDădČăIJ,ăZđæŤuăZÍlæİëäyŞeŨléŞLărfă,İçŖăijŤçŤİçŽDüijNă;EăYřă,ăæřyèİJ
âRëad'Ũă;ăæfYăRřăzëăL'NăLİçŽDëğăRŚăoČiijNă;EăYřăzččăAcIJNăyŁăŖă;ŁăNńiijŽ

```
>>> import gc
>>> gc.collect() # Force collection
Data.__del__
Data.__del__
>>>
```

æĆæđIĲā;ĲčŎřāijTçTĲčŽĐāržēsæĠlāũsēfYāōŽāzL'āžEēĠlāũsčŽĐ
 __del__() æŰžæşTrijNēČčāzLāijŽēŎl'æČĒāEġāRŲā;ŰæŽt'çşşçşTāĂĆ
 āAĠēŎġ;ă;ăăČRăyNēĲcēfZæăũçzŽNodeăŎŽāzL'eĠlāũsčŽĐ __del__() æŰžæşTrijŽ

```
# 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 self.parent
    del self.children
```

ēfŽçġ■æČĒāEġāyNrijNādČāIĲ;āŽđæTŰæřyēfIĲéČ;ăy■ăijŽāŎžāŽđæTŰēfZăylāržēsacŽĐrijNēfYăijŽārīj
 æĆæđIĲā;ăērTçIĲāŎžēfRēāNāŎČăijŽāRŚçŎřijNData.__del__
 æŰLæAřæřyēfIĲăy■ăijŽāĠžçŎřāžE,çTŽēĠşāIĲă;ăăijžāLŰăEġā■YāŽđæTŰæŰŰrijŽ

```
>>> a = Node()
>>> a.add_child(Node())
>>> del a # No message (not collected)
>>> import gc
>>> gc.collect() # No message (not collected)
>>>
```

ăijsăijTçTĲlæŰLēŽd'āžEăijTçTĲlā;ĲčŎřçŽĐēfZăylēŰŏēcYrijNæIJnēt'læĲēŏŝrijNăijsăijTçTĲlārşæYřayĂăy
 ä;ăăRřāžēēĂŽēfĠ weakref æĲēăLŽāzžăijsăijTçTĲlĂĆă;NăēĆrijŽ

```
>>> import weakref
>>> a = Node()
>>> a_ref = weakref.ref(a)
>>> a_ref
<weakref at 0x100581f70; to 'Node' at 0x1005c5410>
>>>
```

ăyžāžEēŏēfēŰŏăijsăijTçTĲlæL'ĂăijTçTĲčŽĐāržēsărijNă;ăăRřāžēăČRăĠ;æTřayĂæăŰăŎžērČçTĲlăŏČă■şăR

çTšazŎaŎšāgNārfzèšaçŽDāijTçTlèðqæTṛæšqæIJL'ácđāLāiijNéCčāzLāršāRfāzēāŎzāLāēZd'āōČāžEāĀČāLāNā

```
>>> print(a_ref())
<__main__.Node object at 0x1005c5410>
>>> del a
Data.__del__
>>> print(a_ref())
None
>>>
```

éĀŽèŁĜèŁZéĜNāijTçd'žçŽDāijsāijTçTlāLĀæIJriijNā;āaijŽāRŚçŎrāy■āE■æIJL'āLçŎrāijTçTlēUōécY
ä;āèŁYèČ;āRČèĀČ8.25ārRèŁČāĒšāžŎāijsāijTçTlçŽDāRēād'ŪāyĀāyĭāLāNā■RāĀČ

10.24 8.24 èŏl'çśzæTṛæNĀærfTèŁČæŞ■āIJ

éUōécY

ä;āæČšèŏl'æšRāyĭçśzçŽDāŏđāLāNæTṛæNĀæāGāĜEçŽDærfTèŁČèŁRçŏŬ(ærfTāēČ>=,!=,<=,<ç■L')iijNā;E

èġčāEşæŪzæāL

PythonçśzārærfRāyĭærfTèŁČæŞ■ā;IJēČ;éIJĀēçAāŏđçŎrāyĀāyĭçL'zæŏŁæŪzæşTæĭæTṛæNĀāĀČ
äL'NāçČāyžāžEæTṛæNĀ>=æŞ■ā;IJçñēriijNā;āéIJĀēçAāŏŽāzL'āyĀāyĭ _____ge____()
æŪzæşTāĀČār;çŏāāŏŽāzL'āyĀāyĭæŪzæşTæşāāzĀāzLēUōécYriijNā;EāçČæđIJēçAā;āāŏđçŎræL'ĀæIJL'ārřè

èčĒēēřāZĭfunctools.total_orderingārşæYřçTlāĭēçŏĀāNŪēŁZāyĭād'DçŘEçŽDāĀČ
ä;ŁçTlāŏČæĭēēčĒēēřāyĀāyĭæĭēriijNā;āāŘĭéIJĀāŏŽāzL'āyĀāyĭ _____eq____()æŪzæşTiiijN
ād'ŪāLāāĒūāzŪæŪzæşT(__lt__, __le__, __gt__, or __ge__)āy■çŽDāyĀāyĭā■şāRřāĀČ
çDūāRŎēčĒēēřāZĭaijŽēĜĭāLĭāyžā;āāqāāĒĒāĒūāŏČærfTèŁČæŪzæşTāĀČ

ä;IJāyžāLāNā■RriijNæŁSāžnæđDāzžāyĀāzZæŁĤā■RriijNçDūāRŎçzZāŏČāžnāçđāLāāyĀāzZæŁĤēŬriijNæ

```
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_foot 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

```

```

    def __eq__(self, other):
    def __lt__(self, other):

```

```

# 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'

```

èõléõž

 ãĖũãõđ total_ordering èċĖëĕrãŽlãžšæšæċĈãžŁċĕđċğŸãĂĈ
ãõĈãřsæŸřãõŽãžŁ'ãžĖäÿĂäÿlãžŌæřRäÿlæřTè;ĈæŤræŃAæŰžæšŤãĹræŁ'ĂæIJL'ėIJĂĕĖAãõŽãžŁ'ċŽĎãĖũãžŮ
æřŤãĖĈã;ããõŽãžŁ'ãžĖ __le__() æŰžæšŤiijŃĕĈãžŁãõĈãřsĕċŋċŤlæĭæđĎãžžæŁ'ĂæIJL'ãĖũãžŰċŽĎĖIJĂĕ
ãõđĕŽĖäÿŁãřsæŸřãIJĹċšžĕĜŃĕĭċãĈRäÿŃĕĭċĕřŽæãũãõŽãžŁ'ãžĖäÿĂãžŽĈŁ'žæõŁæŰžæšŤiijŽ

```
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
```

 ã;ŞċĎŮiijŃã;ăĕĜlãũsãŌžãĖŽãžšã;ŁãõžæŸŞiijŃã;ĖæŸřã;ĤċŤĭ @total_ordering
ãŖãžĕċõĂãŃŰãžċċãAŋiijŃã;ŤãžRĕĂŃäÿ■äÿžãŚċãĂĈ

10.25 8.25 ãĹŽãžžċijŞã■Ÿãõđã;Ń

éŮõĕċŸ

 ãIJlãĹŽãžžãÿĂäÿlċšžċŽĎãřžĕsæŰũiijŃãĖĈæđIJãžŃãŁ'■ã;ĤċŤĭãŖŃæãũãŖĈæŤřãĹŽãžžĕĖĜĕĤŽäÿlãřžĕs
ã;ăæĈşĕĤŤãŽĎãõĈċŽĎċijŞã■ŸãijŤċŤĭãĂĈ

èġċãĖşæŰžæãĹ

 ĕĤŽċğ■ĖĂŽäÿÿæŸřãŽãäÿžã;ãäÿŃæIJŽċŽÿãŖŃãŖĈæŤřãĹŽãžžċŽĎãřžĕsæŰũã■Ťã;ŃċŽĎãĂĈ
ãIJlã;Łãđ'ŽãžŞäÿ■ĖĈ;æIJL'ãõđĕŽĖċŽĎã;Ńã■ŖiijŃæřŤãĖĈ logging
æĭããĭŮiijŃã;ĤċŤĭċŽÿãŖŃċŽĎãŖ■ċğřãĹŽãžžċŽĎ logger ãõđã;ŃæřÿĕĭIJãŖlæIJL'äÿĂäÿlãĂĈã;ŃãĖĈiijŽ

```
>>> import logging
>>> a = logging.getLogger('foo')
>>> b = logging.getLogger('bar')
>>> a is b
False
>>> c = logging.getLogger('foo')
>>> a is c
True
>>>
```

äÿžãžĖĕ;ãĹŖĕĤŽæãũċŽĎæŤĹæđIJiijŃã;ăĖIJĂĕĖAã;ĤċŤĭäÿĂäÿlãŚŃċšžæIJŋĕžŃãĹĖãijĂċŽĎãũĕãŌĈãĜ

çĐũăŔŌăĀŽăÿĂăÿlæŦNërŦijŇăjăaijŽăŖŚçŎřèűşázŊăL'■éĆcäÿlæŮěăŁŮărzèsaçŽďăĹZăżzèąŇăÿžæỲr

èóíèőž

(continues on next page)

(continued from previous page)

```
print('Initializing Spam')
self.name = name
```

__init__() æfRæñæÇ;äijZëcñerÇçTlrijNäy■çöæfZäyIäöðä;NæYräRëcñçijSä■YäZËäÄÇä;NäçCrijZ

```
>>> s = Spam('Dave')
Initializing Spam
>>> t = Spam('Dave')
Initializing Spam
>>> s is t
True
>>>
```

èfZäyIæLÜèöyäy■æYrä;äæÇçèAçZDæTlædIrijNäZäæ■d'èfZçg■æÜzæçTäzúäy■äRräRÜäÄÇ

äyLéIæLŠäznä;ççTlälRäZËäijsäijTçTlèöææTrijNärzäZÖädČaIJ;äZðæTüæIëèöšæYrä;LæIJL'äyöäL'çZ
ä;çSæLŠäznäæIæNääöðä;NçijSä■YæUürijNä;äæRrëÇ;äRlæÇsäIJçlNäZRäy■ä;ççTlälRäöČäznæUüæL'■äflä■
äyÄäyI WeakValueDictionary äöðä;NäRlæijZäflä■YéCçäZZäIJlâEüäöČaIJæÜzèfYäIJlëcñä;ççTlçZDä
äRëäLZçZDërlrijNäRlëçAäöðä;Näy■äE■ëcñä;ççTlälZËrijNäöČärsäZÖä■UäËyäy■ëcñçgzeZd'äZËäÄÇègČärš

```
>>> 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)
[]
>>>
```

ärzäZÖäd'gëČlälEçlNäZRëÄNäüšrijNëfZéGñäZççäAäüšçzRäð'ççTlälZËäÄÇäy■èfGëfYæYräIJL'äyÄä
éçÜäËLæYrëfZéGñä;ççTlälRäZËäyÄäyIäEläsÄäRÿéGRrijNäZüäyTäüëäÖČaG;æTÿëüšçszæTlâIJläyÄ

```
import weakref

class CachedSpamManager:
    def __init__(self):
        self._cache = weakref.WeakValueDictionary()

    def get_spam(self, name):
        if name not in self._cache:
            s = Spam(name)
            self._cache[name] = s
        else:
```

(continues on next page)

(continued from previous page)

```
s = self._cache[name]
return s

def clear(self):
    self._cache.clear()

class Spam:
    manager = CachedSpamManager()
    def __init__(self, name):
        self.name = name

    def get_spam(name):
        return Spam.manager.get_spam(name)
```

èŁŻæăüçŽĐèřřăžččăĂæŽť æÿĚæŽřřijŇăžűăÿŤăžšæŽť çĂăť æ' žřijŇæĹŤăžňăŤăžěăčďăĹăæŽť âď ŽčŽĐčřij
èŁŸæĹĹ'ăÿĂčĆžăřăŤăřřijŇæĹŤăžňăæŽť éĹŤăžĚčšžčŽĐăôďăĹăŇŮčžŽčŤĹæĹŮřřijŇčŤĹæĹăĹăĹăôžăŸŤăž

```
>>> a = Spam('foo')
>>> b = Spam('foo')
>>> a is b
False
>>>
```

æĹĹăĹăčğ■æŮžăřăŤăžěăŸŤăčŤĹæĹăüçŽæăăĂžřřijŇčňăÿĂăÿĹæŸřăřĚčšžčŽĐăŤăŮăŮăôăŤžăÿ
čňăžŇčğ■ăřăŤăřřijŇæĹŤăžňăæŽť éĹŤăžĚčšžčŽĐ __init__() æŮžăŤăĹăŽăĹăžăÿĂăÿĹăřăčăÿřřijŇăôčăÿ■ččččăŮăĹă

```
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
```

čĐăăŤăŮăôăŤžăřřijŤăžččăŸčôăčŤăžăžččăĂřřijŇăĹčŤĹ Spam._new()
æĹăĹăŽăžăôďăĹăřřijŇăŇăÿ■æŸčŽť æŮčččččŤĹ Spam() æďĐăĂăăĹăĹăřřijŽ

```
# -----æĹĹăŤăŮăôăŤžăřřijŇăŮăôăčăŮăžăăĹ-----
↪ ---
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

```

æIJĀāRŌēfZæūçZĎæŪzæāLārśāũščzRēũšād' šāē;āžEāĀĆ
 çijŠā■ŸāŠNāĒŪāzŪæđDēĀāæIqāijRēfŸāRřāzēā;ŁçTĪ9.13ārRēLĆāy■çZĎāĒČçśzāōđçŌřçZĎæZt'āijŸēZĒāy

11 çñňäzlçnáiiijZāĒČçijŪćlŃ

ē;řāzŭāijĀāRŚécEāššāy■æIJĀçzRāĒÿçZĎāRčād't'çēĒārsæŸřāĀIJdonāĀŽt repeat your-
 selfāĀlāĀĆ āzšārsæŸřēřt'iiĴNāzā;TæŪūāĀZā;Šā;āçZĎćlŃāžRāy■ā■ŸāIJlénŸāžēēĠād'■(æLŪēĀĒæŸřēĀ.
 āIJlPythonā;Šāy■iiijNēĀZāyēČ;āRřāzēēĀŽēfĠāĒČçijŪćlŃNēlēēçĀEşēŁZçśzéŪōécŸāĀĆ
 çōĀēĀNēlĀāzNiiĴNāĒČçijŪćlŃNārsæŸřāĒşāžŌāLZāžzæŞ■ā;IJæžRāzčçāĀ(ærTāēĆāŁōæTzāĀAçTšæLŘæLŪ
 āyžēēĀæLĀæIJřæŸřā;ŁçTĪēčĒēēřāZĪāĀAçśzēčĒēēřāZĪāŠNāĒČçśzāĀĆāy■ēŁĠēŸŸæIJL'āyĀāžZāĒŪāzŪæLĀ
 āNĒēNñç■;āR■āržēsāāĀā;ŁçTĪ exec() æLġēāNāžčçāĀāžēāRĀřzāĒēēČlāĠ;æTřāŠNçśzçZĎāR■ārĎæL
 æIJñçāçZĎāyžēēĀçZōçZĎæŸřāRŚād'ġāōūāzNçz■ēŁZāžZāĒČçijŪćlŃNæLĀæIJřiiĴNāzŭāyTçzZāĠzāōđā;Næ

Contents:

11.1 9.1 ālJlāĠ;æTřāyŁæŭzāŁāāNĒēčĒāZl

éŪōécŸ

ā;āæČşāIJlāĠ;æTřāyŁæŭzāŁāāyĀāyĪāNĒēčĒāZlīijNāćđāŁāécĪād'ŪçZĎæŞ■ā;IJād'ĎçŘĒ(ærTāēĆāŪēā

ēġçāEşæŪzæāĹ

āēĆādIJā;āæČşā;ŁçTĪēčĪād'ŪçZĎāžčçāĀāNĒēčĒāyĀāyĪāĠ;æTřiiĴNāRřāzēāōZāzL'āyĀāyĪēčĒēēřāZĪāĠ

```
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
```

äyÑéÍæYřä;ŁçTíèčĚéčřăZÍçŽDă;Nă■RřijŽ

```
>>> @timethis
... def countdown(n):
...     '''
...     Counts down
...     '''
...     while n > 0:
...         n -= 1
...
>>> countdown(100000)
countdown 0.008917808532714844
>>> countdown(10000000)
countdown 0.87188299392912
>>>
```

ěóíěőž

äyÄäyłèčĚéčřăZÍłřsæYřäyÄäyłăG;æTřřijNăóČæŎěăRŮäyÄäyłăG;æTřřä;IJäyžăRCæTřřázúèŁTăZđäyÄäy
ă;Šă;ăăČRäyÑéÍèçŁæăăăĚŽřijŽ

```
@timethis
def countdown(n):
    pass
```

èùšăČRäyÑéÍèçŁæăăăĚŽăĚăăóđæTŁæđIJæYřäyÄäyłăçŽDřijŽ

```
def countdown(n):
    pass
countdown = timethis(countdown)
```

éąžă;Łèř'äyÄäyNřijNăĚĚç;óçŽDèčĚéčřăZÍłřTăçČ @staticmethod,
@classmethod, @property ăŎšçŘĚázšæYřäyÄäyłăçŽDăĂČ

ä; NäeĆiijNäyNéIcèfZäyd' äyläzççäAçL' GæøtæYřç■L' äzüçZDiiijZ

```
class A:
    @classmethod
    def method(cls):
        pass

class B:
    # Equivalent definition of a class method
    def method(cls):
        pass
    method = classmethod(method)
```

åIJläyLéIcçZD wrapper() åG;æTřäy■iijN' ècĚéērāZlāEĚēČlāōZāzL' āzEäyĀäyIā;ŁçTl
*args åŠN **kwargs æIēæŌēāRŪāzææĎRāRCæTřçZDāG;æTřāĀĆ
åIJlēfZāyIāG;æTřēGŇéIcērČçTlāzEāŌŝāgNāG;æTřāzūārEāĚūçzŞæĎIJēfTāZDiiijNäy■ēfGā;æēfYāRfāzææūz
çDūāRŌēfZāyIāŪřçZDāG;æTřāNĚēcĚāZlēcñā;IJāyžçzŞæĎIJēfTāZDæIēāzçæZŁāŌŝāgNāG;æTřāĀĆ
éIJĀēēAaijzērČçZDæYřēcĚéērāZlāzūāy■aijZāŁæTřāŌŝāgNāG;æTřçZDāRCæTřç■āR■āzēāRLēfTāZ
ā;ŁçTl *args åŠN **kwargs çZōçZDārśæYřçāōāfIāzā;TāRCæTřēČ;ēČ;éĀĆçTlāĀĆ
èĀNēfTāZdçzŞæĎIJāĀijāşzæIJnéČ;æYřērČçTlāŌŝāgNāG;æTř func(*args,
**kwargs) çZDēfTāZdçzŞæĎIJiijNāĚūāy■funcārśæYřāŌŝāgNāG;æTřāĀĆ
ālZāijĀāgNā■ēāzæcĚéērāZlçZDæŪūāĀZiijNāijZā;ŁçTlāyĀāzZçōĀā■TçZDā;Nā■RæIēērt' æYŌiijNær
äy■ēfGāōdēZĚāIJzæZřā;ŁçTlæŪūiijNēfYæYřæIJLāyĀāzZçzEēLĆēŪōēcYēēAæşlæĎRçZDāĀĆ
ærTāēČāyLéIcā;ŁçTl @wraps(func) æşlēğçæYřā;LéG■ēēAçZDiiijN
āōČēČ;āfIçTřāŌŝāgNāG;æTřçZDāĚČæTřæ■ō(äyNäyĀārRēLČaijZēōşāLr)iijNæŪræLŊçzRāyyaijZāf;çTēē
æŌēāyNālēçZDāGāyIārRēLČæLŞāznāijZæZr' āŁāæūşāĚēçZDēōşēğçēcĚéērāZlāG;æTřçZDçzEēLĆēŪōēcY

11.2 9.2 āLZāzzēcĚéērāZlāæŪūāfIçTřZāG;æTřāĚČāŁæAř

éŪōēcY

ā;āāEŻāzEäyĀäyIēcĚéērāZlā;IJçTlāIJlæşRāyIāG;æTřäyLiiijNā;EæYřēfZāyIāG;æTřçZDēG■ēēAçZDāĚ

ēğçāEşæŪzæāŁ

āzzā;TæŪūāĀZā;āāōZāzL'ècĚéērāZlçZDæŪūāĀZiijNēČ;āzTērēā;ŁçTl functools
āzŞäy■çZD @wraps ècĚéērāZlāIēæşlēğçāzTřāsCāNĚēcĚāG;æTřāĀĆā;NāēĆiijZ

```
import time
from functools import wraps
def timethis(func):
    '''
    Decorator that reports the execution time.
    '''
    @wraps(func)
    def wrapper(*args, **kwargs):
```

(continues on next page)

(continued from previous page)

```
start = time.time()
result = func(*args, **kwargs)
end = time.time()
print(func.__name__, end-start)
return result
return wrapper
```

äyÑéíæŁŚäzñä;£çŦíè£ŽäyłècñăÑĚèĉĚăŘŎçŽĎăĜ;æŦřăžűæĉĂæšëăóĈçŽĎăĚĈă£æAřijŽ

```
>>> @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'>}
>>>
```

èóìèőž

ăIJłijŮăĚžèĉĚéērăŽíçŽĎăŮűăĂŽăđ'■ăĹűăĚĈă£æAřæŸřăyĂäyłéİđăyyéĜ■èĉAçŽĎéĈłăĹĚăĂĈă£æA
@wraps ijŦ éĈcázĹă;ăaijŽăŘŚçŎřècñèĉĚéērăĜ;æŦřăyĉăđ'šăžĒăĹ'ĂæIJĹæIJĹçŦíçŽĎă£æAřăĂĈăřŦăĉ
@wraps âŘŎçŽĎăŦĹăđIJæŸřăyÑéíçè£ŽæăűçŽĎijŽ

```
>>> countdown.__name__
'wrapper'
>>> countdown.__doc__
>>> countdown.__annotations__
{}
>>>
```

@wraps æIJĹăyĂäyłéĜ■èĉAçĹ'žă;AæŸřăőĈèĈ;èóĹ'ă;ăéĂŽè£ĜăśđăĂğ
__wrapped__ çŽt' æŎèèłéŮŏècñăÑĚèĉĚăĜ;æŦřăĂĈă;ŦăĉĈ:

```
>>> countdown.__wrapped__(100000)
>>>
```

__wrapped__ áśđăĂğè£ŸèĈ;èóĹ'ècñèĉĚéērăĜ;æŦřă■ççăŏæŽt' éIJšăžŦăśĈçŽĎăŘĈăŦřç■;ăŘ■ă£æA

```
>>> from inspect import signature
>>> print(signature(countdown))
(n:int)
>>>
```

äyÄäylä;LæŽðéA■çŽĐéŮóécŸæŸræĂŌæäüèóI'èçĚéěřăŽlăŌžçŽt'æŌěăd'■ăLŭăŌşăğNăĜ;æTřçŽĐăRČ
 âĕĆăđIJăCşèĜlăûsăL'NăLlăóđçŎřçŽĐĕřlĕIJăĚĕAăAžăđ'gĕĜRçŽĐăüĕă;IJiijNăIJăĀĕ;ăřşçóĂă■TçŽĐă;ĕçŮ
 @wraps èçĚéěřăŽlăĂĆ éĂŽèĚĜăžTăśĆçŽĐ __wrapped__
 âşđăĂĝèóĚéŮóăLřăĜ;æTřç■;ăR■ăĚăæAřăĂĆăŽt'ăđ'ŽăĚşăžŎç■;ăŘ■çŽĐăĚăóăăRřăžăăRČăĂĆ9.16ăřRĕL

11.3 9.3 èĝcéŽd'äyÄäylèçĚéěřăŽl

éŮóécŸ

äyÄäylèçĚéěřăŽlăûşçžRă;IJçŤlăIJlăyÄäylăĜ;æTřäyLiiijNă;ăăCşăŞđ'ėŤĂăóĈiijNçŽt'æŌěèóĚéŮăŌşăğ

èĝcăĚşăŮžăăĹ

ăAĜèó;èçĚéěřăŽlăŸřéĂŽèĚĜ @wraps (ăRČăĂĆ9.2ăřRĕLC)æĹăăóđçŎřçŽĐiijNĕĆcăžLă;ăăRřăžăăĂŽ
 __wrapped__ âşđăĂĝăĹèèóĚéŮăŌşăğNăĜ;æTřiijŽ

```
>>> @somedecorator
>>> def add(x, y):
...     return x + y
...
>>> orig_add = add.__wrapped__
>>> orig_add(3, 4)
7
>>>
```

èóĹéŮž

çŽt'æŌěèóĚéŮăIJăNĚèçĚçŽĐăŌşăğNăĜ;æTřăIJlĕřĈĕřŤăĂăĀĚĚçIJăăŠNăĚüăžŮăĜ;æTřăŞ■ă;IJăŮ
 ä;ĚăŸræĹŚăžñĕĚŽĕĜNçŽĐăŮžăăĹăžăĚăžĚăĂĆçŤlăžŎăIJlăNĚèçĚăŽlăy■ă■ççăŏă;ĕçŤlăžĚ
 @wraps æĹŮĕĂĚçŽt'æŌěèó;ç;ŏăžĚ __wrapped__ âşđăĂĝçŽĐăĈĚăĚăĂĆ

ăĕĆăđIJăIJL'ăđ'ŽăylăNĚèçĚăŽlăiijNĕĆcăžĹèóĚéŮă __wrapped__
 âşđăĂĝçŽĐăăNăyžăŸrăy■ăRřĕćĎçşĕçŽĐiijNăžŤĕřĕĖAăăĚ■ĕĚăăüăăĂžăĂĆ
 âIJlPython3.3ăy■iijNăóĈăijŽçŤĕĕĚĜăĹ'ĂăIJL'çŽĐăNĚèçĚăŚĈiijNăřŤăĕĈiijNăĂăĜăĖĆă;ăăIJL'ăĕĆăyNçŽĐ

```
from functools import wraps

def decorator1(func):
    @wraps(func)
    def wrapper(*args, **kwargs):
```

(continues on next page)

(continued from previous page)

```
        print('Decorator 1')
        return func(*args, **kwargs)
    return wrapper

def decorator2(func):
    @wraps(func)
    def wrapper(*args, **kwargs):
        print('Decorator 2')
        return func(*args, **kwargs)
    return wrapper

@decorator1
@decorator2
def add(x, y):
    return x + y
```

äyÑéíæĹŚäzñåĲPython3.3äyÑæŧÑèŕŦijŽ

```
>>> add(2, 3)
Decorator 1
Decorator 2
5
>>> add.__wrapped__(2, 3)
5
>>>
```

äyÑéíæĹŚäzñåĲPython3.4äyÑæŧÑèŕŦijŽ

```
>>> add(2, 3)
Decorator 1
Decorator 2
5
>>> add.__wrapped__(2, 3)
Decorator 2
5
>>>
```

æĲĲĀŖŌèĕAèŕŧ'çŽDæŸŕijŊāzūäy■æŸŕæĹ'ÄæĲĲ'çŽDèĕĒëĕŕāŽléČ;ä;ŧçŦíāžE
@wraps ijŊāZāæ■d'èŧŽéĜŊçŽDæŸzæqĹāzūäy■āĒléČléĀČçŦíāĀČ
çĹ'zāĹŋçŽDijŊāĒĒç;ōçŽDèĕĒëĕŕāŽÍ @staticmethod āŠŊ @classmethod
āŕśæşqæĲĲ'éAŧā;ŧèŧZäyŧçzèāōŽ (āōČäzñæĹĹāŌşāğŊāĜ;æŦŕā■ŸāČĲĲĲāśdæĀğ __func__
äy■)āĀČ

11.4 9.4 āōŽāzĹ'äyĀäyĲāyēāŖČæŦŕçŽDèĕĒëĕŕāŽÍ

éŬŌéćŸ

ä;ăæČşāōŽāzĹ'äyĀäyĲāŕŕāzèæŌēāŖŬāŖČæŦŕçŽDèĕĒëĕŕāŽÍ

èğçàEşæŮzæąĹ

æĹSäznçTĹäyÄäyĹäĹNā■RèřęçzEéYRèřřäyNæŌěârŮârĹCæTřçŽDäd'DçŘEèĹĜçĹNāĀĆ
āAĜèőĹäĹāæĈşâEŻäyÄäyĹèçĒěčřāZĹiijNçzŽāĜĹæTřæûzâĹāæŮěâĹŮâĹşèĈĹiijNāŘNæŮûâĒAèőyçTĹæĹûæN
äyNéĹcæYřèřZäyĹèçĒěčřāZĹçŽDăőZăzĹ'âŞNăĹçTĹçd'žăĹNiiž

```
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!')
```

āĹĹçIJNętuæĹëriijNęĹZçĝ■ăôđçŌřçIJNäyĹăŌzâĹĹâd'■æĹĈiijNăĹEæYřæäyâĹĈæĀĹæĈşâĹĹçőĀâ■TăĀĆ
æĹJĀâd'ŮâşĈçŽDăĜĹæTř logged() æŌěârŮârĹCæTřăzûârEăőĈăznăĹIJçTĹâĹJĹâĒĒěĈĹçŽDèçĒěčřāZĹâĜĹæ
âĒĒăşĈçŽDăĜĹæTř decorate() æŌěârŮäyÄäyĹâĜĹæTřäĹJäyžârĹCæTřriijNçDăârŌâĹJĹâĜĹæTřäyĹéĹcæY
èĹZéĜNçŽDăEşèTőçĈzæYřâNĒèçĒăZĹæYřârRăzèäĹçTĹâiijăéĀŞçzŽ logged()
çŽDăĹCæTřçŽDăĀĆ

èőĹéőž

ăőZăzĹäyÄäyĹæŌěârŮârĹCæTřçŽDăNĒèçĒăZĹçIJNäyĹăŌzæřTèĹĈâd'■æĹĈäyžèçAæYřăZäyžăzTăşĈ

```
@decorator(x, y, z)
def func(a, b):
```

(continues on next page)

```
pass
```

ěčĚěřāZlād'DčŘĚěfGčlNěušāyNělččŽDěřČčTlāYřčL'æTlčŽD;

```
def func(a, b):
    pass
func = decorator(x, y, z)(func)
```

decorator(x, y, z) čŽDěfTāZđczSædIJāfĚēāzæYřāyĀāylāRřerČčTlārZèšāijNāōČæŌěāRŪāyĀ
āRřāzēāRČèĀČ9.7āRŘèLČāy■āRēād'ŪāyĀāylāRřæŌěāRŪāRČæTřčŽDāNěččĚāZlā;Nā■RāĀČ

11.5 9.5 āRrèGlaōŽāzL'āsdæĀgčŽDěčĚěřāZl

éŬóécY

äjäæČšāEžāyĀāylēcĚěřāZlālēāNěččĚāyĀāylāGjæTřrijNāzūāyTāĚAēōyčTlāLūæRŘā;ZāRČæTřāIJlē

ěğčāEšæŪzæāL

ājTāĚēāyĀāylēōēUōāGjæTřrijNā;fčTl nonlocal ælčāfōæTzāEĚčlāRŸéGRāĀČ
čDúāRŌēfZāylēōēUōāGjæTřřecnāIJāyžāyĀāylāsdæĀgčNāĀijčzZāNěččĚāGjæTřāĀČ

```
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!')
```

äyÑéÍcæYřäzď'äzŠčŮřácČäyŇçŽďä;ŁçTłäŁŇä■ŘiijŽ

```

>>> 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
>>>
```

èõìèõž

```
    ẽƒŽäýÄärŘèŁĆçŽĎǺĚšéŤôçĆzǻIJǻžŎèõŁéŮôǻĜ;æŤř(ǻĉĆ      set_message()
ǻŠŇ      set_level()      )iijŇǻõČǻžñèćǻ;IJǻýžǻśđæǻĜèŤŇçžŽǻŇĚèćĚǻŽǻǻǻĈ
æŤřǻýłèõŁéŮôǻĜ;æŤřǻĚǻèõýǻ;ŁçŤǻnonlocal æĬèǻǻôæŤžǻĜ;æŤřǻĚĚĈĬçŽĎǺŘŸéĜŤǻǻĈ
```

```
    ẽƒŸæIJL'ǻýǻǻýłǻžd'ǻžžǻǻŘĈæĈŁçŽĎǺIJřæŮžæŸřèõŁéŮôǻĜ;æŤřǻiijŽǻǻIJǻđ'ŽǻśĈèćĚéèřǻǻŽǻéŮŤ'ǻiijǻæŠ■
@functools.wraps æšǻéĝĉ)ǻǻĈ æĬŇǻæĈiijŇǻǻǻĜèõĬǻ;ǻǻiijŤǻĚèǻǻĤǻđ'ŮǻýǻǻýłèćĚéèřǻǻŽǻiijŇǻæŤǻæĈ9.2ǻ
@timethis iijŇǻǻĈŘǻýŇéĬèƒŽæǻǻiijŽ
```

```
@timethis
@logged(logging.DEBUG)
def countdown(n):
    while n > 0:
        n -= 1
```

ǻ;ǻǻiijŽǻǻŘŚçŎŤèõŁéŮôǻĜ;æŤřǻĬǻæŮĝæIJL'æŤǻIijŽ

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

ǻ;ǻèƒŸǻiijŽǻǻŘŚçŎŤǻ■ǻ;ŁèćĚéèřǻǻŽǻǻĈŘǻýŇéĬèƒŽæǻǻǻžèçŽŸǻǻ■çŽĎǺŮžǻǻŘŚæŎŚæŤĬiijŇǻæŤǻǻđIJǻž

```
@logged(logging.DEBUG)
@timethis
def countdown(n):
    while n > 0:
        n -= 1
```

èƒŸèĈ;éǻŽèƒĜǻ;ŁçŤǻlambdæǻǻéĬĬǻ;ǻiijŘǻžçĉǻǻǻĬèèŎŤ'èõŁéŮôǻĜ;æŤřçŽĎèƒŤǻǻŽđǻý■ǻǻŇçŽĎèõĬǻǻõŽǻǻ

```
@attach_wrapper(wrapper)
def get_level():
    return level

# Alternative
wrapper.get_level = lambda: level
```

ǻýǻǻýłǻæŤǻĬĈéŽĬçŘĚèĝĉçŽĎǺIJřæŮžǻǻřǻæŸřǻǻžǻžŎèõŁéŮôǻĜ;æŤřçŽĎéçŮæǻǻǻ;ŁçŤǻǻǻĈǻĬŇǻæĈiijŇ

```
@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
```

èĚZäylæŮzæſTäzſâRřèĈ;æ■čäyŷäüëä;IJiijŇä;ĚäL'■æRŘæŸrâŏĈâĚĚéazæŸræIJĀād' ŮāsĈçŽĎèĈĚéērāZ
 âĚĈädIJĀŏĈçŽĎäyLéİcèĚŸæIJL'âRĚad' ŮçŽĎèĈĚéērāZĪ(æŕTāĚCäyLéİcæRŘâĹŕçŽĎ
 @timethis ä;Ňâ■R)iiijŇéĈčäzLâŏĈäijŽéŽRèŮRâžTſâſĈâſdæĀgiiijŇä;Ěä;ŮäĚŏæTžâŏĈäzñæſqæIJL'äzzä;T
 èĀŇéĀŽèĚGä;ĚçTlèŏĚéŮŏâG;æTſrâſèĈ;éĀĚâĚ■èĚZæäüçŽĎâſĀéŽRæĀgāĀĈ

æIJĀâRŎæRŘäyĀçĈziiijŇèĚZäyĀârRèĹĈçŽĎæŮzæäĹäzſâRřäzëä;IJäyž9.9ârRèĹCäy■èĈĚéērāZĪçſçŽ

11.6 9.6 äyĚâRřéĀL'âRĈæTſçŽĎèĈĚéērāZĪ

éŮŏécŸ

ä;äæĈſâĚZäyĀäyġèĈĚéērāZĪiijŇæŮçâRřäzëäy■äijäâRĈæTſçzŽâŏĈiiijŇæŕTāĚĈ
 @decorator iiijŇ äzſâRřäzëäijäéĀſâRřéĀL'âRĈæTſçzŽâŏĈiiijŇæŕTāĚĈ
 @decorator(x, y, z) äĀĈ

èğçâĚſæŮzæäĹ

äyŇéİcæŸr9.5ârRèĹCäy■æŮèäĚŮèĈĚéērāZĪçŽĎäyĀäyġäĚŏæTžçĹL'æIJñiiijŽ

```
from functools import wraps, partial
import logging

def logged(func=None, *, level=logging.DEBUG, name=None, _
↳message=None):
    if func is None:
        return partial(logged, level=level, name=name, _
↳message=message)

    logname = name if name else func.__module__
    log = logging.getLogger(logname)
    logmsg = message if message else func.__name__

    @wraps(func)
    def wrapper(*args, **kwargs):
        log.log(level, logmsg)
        return func(*args, **kwargs)

    return wrapper

# Example use
```

(continues on next page)

(continued from previous page)

```
@logged
def add(x, y):
    return x + y

@logged(level=logging.CRITICAL, name='example')
def spam():
    print('Spam!')
```

ãŕäzëçIJÑãĹŕiijÑ@logged èçĒëĕŕãŽĹãŕŕäzëãŕÑæŮüäy■äyëãŕĈæŦŕæĹŮäyëãŕĈæŦŕäĈ

ëõĹëõž

èĚŽëĜÑæŔŔãĹŕçŽĎèĚŽäyĹëŮöëçŸäŕsæŸŕéĀŽäyÿæĹ'ÀèŕŦ'çŽĎçijŮçĹĹNäyĀèĜŦ'æĀĜëŮöëçŸäĈ
ä;ŞæĹSäznä;ĤçŦĹëçĒëĕŕãŽĹçŽĎæŮüãĀŽiijÑãĎ'ġëĈĹãĹĒçĹĹNãžŔãSŸäžãæĈŕäžĒëçĀäzĹäy■çzŽãöĈäznäijäéĀ
ãĒüãöđäzŌæĹĀæIJŕäyĹæĹëëöšiiijÑæĹSäznãŕŕäzëãöŽäzĹ'äyĀäyĹæĹ'ĀæIJĹ'ãŕĈæŦŕéĈ;æŸŕãŕŕéĀĹçŽĎèçĒ

```
@logged()
def add(x, y):
    return x+y
```

ä;ĒæŸŕiijÑèĚŽçġ■ãĒZæşŦäzŮäy■çñëãŕĹæĹSäznçŽĎäzãæĈŕiijÑæIJĹ'æŮüãĀŽçĹĹNãžŔãSŸäŕŸëöŕãĹäã
èĚŽëĜÑæĹSäznãŕŔã;ããşŦçĎ'žäžĒæçĈä;ŦäžëäyĀèĜŦ'çŽĎçijŮçĹĹNëçŌæäijæĹëãŕÑæŮüæzæüşæşæIJĹ'æNñã

äyžäžĒçŔĒëġçäzççãĀæŸŕäçĈä;Ŧäüëä;IJçŽĎiijÑä;äéIJĀëçĀéĹäyÿçĒşæĈĹ'èçĒëĕŕãŽĹæŸŕäçĈä;Ŧä;IJçŦ
ãŕzäžŌäyĀäyĹãĈŕäyÑëĹçèĚŽæäüçŽĎçöĀ■ŦçĒĒëĕŕãŽĹiijŽ

```
# Example use
@logged
def add(x, y):
    return x + y
```

èĚŽäyĹŕçĈŦĹãžŔãĹŮëüşäyÑëĹçç■Ĺ'äzŮiijŽ

```
def add(x, y):
    return x + y

add = logged(add)
```

èĚŽæŮüãĀŽiijÑëçñèçĒëĕŕãĜ;æŦŕäijŽëçñã;ŞãĀŽçññäyĀäyĹãŕĈæŦŕçŽŦ'æŌëäijäéĀŞçzŽ
logged èçĒëĕŕãŽĹãĈ äŽäæ■d'iijÑlogged() äy■çŽĎçññäyĀäyĹãŕĈæŦŕäŕsæŸŕëçñãÑĒëçĒãĜ;æŦŕæIJñë
èĀŦãŕžäžŌäyĀäyĹäyÑëĹçèĚŽæäüæIJĹ'ãŕĈæŦŕçŽĎèçĒëĕŕãŽĹiijŽ

```
@logged(level=logging.CRITICAL, name='example')
def spam():
    print('Spam!')
```

ŕçĈŦĹãžŔãĹŮëüşäyÑëĹçç■Ĺ'äzŮiijŽ

ǎĹiǎgNērČćŦĪ logged() āĞ;æŦřæŮırijNěcńāNĚēcĚāĞ;æŦřázúæšqæIJL'ājāēĀŠēfZæIēāĀĆ
āŽāē■d'āIJlēcĚēēřāZĹāEĲijNāōČāfĚēązæYřārřéĀĽčŽDāĀĆēfZāyĹāR■ēfGāIēāijŽēfńā;fāĒūāzŮāŘĆæŦř
ázúāyŦrijNā;EēfZāzZāŘĆæŦřēcńāijāēĀŠēfZæIēāŘŌrijNēcĚēēřāZlēęAēfŦāZđāyĀāylæŌēārŬāyĀāylāĞ;æ'
āyžāEēfZæāūāĀŽrijNāĽSāznā;fçŦĹāzĚāyĀāylæĽĀāũgrijNārřæYřālŦčŦĪ functools.
partial āĀĆ āōČāijŽēfŦāZđāyĀāylæIJlāōNāĚĹāĹiǎgNāNŮčŽĐēĞēznrijNēŽd'āžĚēcńāNĚēcĚāĞ;æŦřād'
ārřāzēārŘēĀĆ7.8ārŘēĽĆēŌūārŬāēZt'ad'Ž partial() æŮzæsTcŽĐçšēēfĚāĀĆ


```
def decorate(func):
    # If in optimized mode, disable type checking
    if not __debug__:
        return func
```

inspect.signature() `inspect.signature(spam)`

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

`inspect.signature(spam).bind_partial(x=1, y=2)`

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

`inspect.signature(spam).bind(x=1, y=2, z=3)`

`inspect.signature(spam).bind(x=1, y=2, z=3)`

```
>>> bound_values = sig.bind(1, 2, 3)
>>> bound_values.arguments
OrderedDict([('x', 1), ('y', 2), ('z', 3)])
```

(continues on next page)

```
>>>
```

ä;ŁçŦłēŁZäyŁæŸäârĎæŁŚäznâRřäzčä;Łè;žæĬçŽĎăôđçŎřæŁŚäznčŽĎăijžăĹŭçşzăđNăčĂæşëijŽ

```
>>> for name, value in bound_values.arguments.items():
...     if name in bound_types.arguments:
...         if not isinstance(value, bound_types.arguments[name]):
...             raise TypeError()
...
>>>
```

äy■ēŁĜēŁZäyŁæŸzæŁēŁŸæĬĹçČžârRçŦçŦĭijNăôČăržăžŎæĬĹēžŸēôđ'ăĂijçŽĎăRČæŦřăžüäy■éĂ
ærŦæČäyNēĬčçŽĎăžččăĂârRăzčæ■čăyŷăüēăĬĭijNărçŏăitemşçŽĎçşzăđNăŸřéŦŽēřççŽĎijŽ

```
>>> @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]
>>>
```

æĬĬĂârŎäyĂçČžæŸřăĚşăžŎēĂČçŦłēčĚēēřăŽĭăRČæŦřăŦŦăĜ;æŦřæşlēğčăžNēŮřçŽĎăžĹ'èôžăĂČ
ăĬNăēČĭijNăyžăžĂăžĹäy■ăČŦăyNēĬčēŁZăăüăĚŽăyĂäyłēčĚēēřăŽĭăĬæēşēæĹ;ăĜ;æŦřăy■çŽĎăşlēğčăŦçĭijş

```
@typeassert
def spam(x:int, y, z:int = 42):
    print(x, y, z)
```

äyĂäyŁăRřēČ;çŽĎăŎşăŽăæŸřăēČæđĬă;ŁçŦłăžĚăĜ;æŦřăRČæŦřæşlēğçĭijNēČčăžĹăřşēčnéŽŦăĹŭăžĚă.
ăēČæđĬăşlēğçēčēčŦĭăĬăĂŽçşzăđNăčĂæşēârşäy■ēČ;ăĂŽăĚŷăžŮăžNăČĚăžĚăĂČēĂNăyŦ
@typeassert äy■ēČ;ăĚ■çŦłăžŎă;ŁçŦłæşlēğčăĂŽăĚŷăžŮăžNăČĚçŽĎăĜ;æŦřăžĚăĂČ
ēĂNă;ŁçŦłăyĹēĬççŽĎēčĚēēřăŽĭăRČæŦřçĂŦæt'žæĂĝăđ'ĝăđ'ŽăžĚĭijNăžşæŽřăĹăéĂŽçŦĭăĂČ

ârřăzčăĬĬPEP 362ăžčăŦĹ inspect æĬăăĬŮäy■æĹ;ăĹŦæŽřăđ'ŽăĚşăžŎăĜ;æŦřăRČæŦřăřžēşççŽĎăŁă

èġċăĖşăŮzăăĹ

ăyżăŹĖărĖċĖĖĕřăŹĹăŮŹăŹĹ'ăĹŖăŷĂăŷĹăŮđăĹŊĭŷŊăĵăĖĹĂĖĕĂĉăăŹĹăŮĈăăŮđĉŮŹăŹĖ
__call__()ăŖŊ__get__()ăŮŹăşŤăĂĈăĹŊăĖĈĭŷŊăŷŊĖĹĉŹŹăŹĉĉăĂăŮŹăŹĹ'ăŹĖăŷĂăŷĹĉşŷĭŷŊăŮŮĈă

```
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)
```

ăĵăăŖŖăŹĕărĖăŮĈăĵŖăĂŹăŷĂăŷĹăŹŮĖĂŹĉŹŹăĈĖĖĕĕřăŹĹăĹĕăĵĉŤĹĭŷŊăĹĹĹĉşŷĖĜŊĖĹĉăĹŮăđ'ŮĖĹĉĖĈĵăŖŖă

```
@Profiled
def add(x, y):
    return x + y

class Spam:
    @Profiled
    def bar(self, x):
        print(self, x)
```

ăĹĹăŹđ'ăŹŖŹŮŖăĈĈăŷăŹŹăĵĉŤĹĉđ'ăŹăŊĭŷŹ

```
>>> 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
```

èõléõž

årÈèçÈéèràŽlãðŽázL'æL'RçşzéĂŽăÿÿæYřăĹçõĂă■TçŽDăĂĆă;EæYřèŁŽéĜÑèŁYæYřæIJL'ăÿĂăžZçE
éèŮăĚĹiijNă;ŁçTl functools.wraps() åĜjæTřçŽDă;IJçTlèuřázNăL'■èŁYæYřăÿĂăăiijNăřÈèçná
ăĚŮăñăiijNéĂŽăÿÿăĹLãðžæYřşăijŽăŁ;èĝEăÿŁéİçŽD _____get____()
æŮžæşTăĂĆăĈăĈăĬJă;ăăŁ;çTèăőCrijNăŁIăNăăĚŮăžŮăžççăAăÿ■ăRŸăE■ăñăèŁRèăNăiijN
ă;ăăiijŽăRŠçŮřă;Şă;ăăŌžèřČçTlèçnèçÈéèřăðđă;NăŮžæşTăŮăăĜžçŮřăĹLăĖGăĂłçŽDèŮèçYăĂĆăĹNăĖCrij

```
>>> s = Spam()
>>> s.bar(3)
Traceback (most recent call last):
...
TypeError: bar() missing 1 required positional argument: 'x'
```

ăĜžéTřZăŌşăŽăæYřă;ŞăŮžæşTăĜjæTřăIJlăÿĂăÿŁçşžăÿ■èçnáşşæŁ;æŮiijNăőČăžñçŽD
_____get____() æŮžæşTăĹIă■őæRŘèŁřăŽlă■RèððèçnèřČçTlriijN
ăIJl8.9ăřRèŁĆăuřççRèðşèŁřèŁGăRŘèŁřăŽlă■RèððăžEăĂĆăIJlèŁŽéĜÑiijN_____get____()
çŽDçŽõçŽDæYřăĹZăžžăÿĂăÿŁçžŞăðŽæŮžæşTăřžèşă(æIJĂçžĹăiijŽçžŽèŁŽăÿŁæŮžæşTăiijăĂşselfăRĆăTřă)

```
>>> s = Spam()
>>> def grok(self, x):
...     pass
...
>>> grok.__get__(s, Spam)
<bound method Spam.grok of <__main__.Spam object at 0x100671e90>>
>>>
```

_____get____() æŮžæşTăYřăÿžăžEçăðăŁçžŞăðŽæŮžæşTăřžèşăç;èçná■ççăðçŽDăĹZăžžăĂĆ
type.MethodType() æŁNăĹlăĹZăžžăÿĂăÿŁçžŞăðŽæŮžæşTăĹă;ŁçTlăĂĆăRlăIJL'ă;ŞăðđăĹNèçná;ŁçT
ăĈăĈăĬJèŁŽăÿŁæŮžæşTăYřăIJłçşžăÿŁéİçăĹèèðŁéŮiijN éĆçăžĹ _____get____() äÿ■çŽDin-
stanceăRĆăTřăiijŽèçnèð;ç;ðæĹRNoneăžŮçŽt'æŌèèŁTăŽđ Profiled äðđă;NăIJnèžnáĂĆ
èŁŽăăiijçŽDèřlăĹSăžnăřăşăRřăžæRŘăŮŮăðČçŽD ncalls áşđăĂĝăžEăĂĆ

ăĈăĈăĬJă;ăăĈşéAŁăĚ■ăÿĂăžZăŮăžšiiijNăžşăRřăžèèĂĆèŽŞăRèăđ'ŮăÿĂăÿĹă;ŁçTlèŮ■ăNĚăŞN
nonlocal âRŸéĜRăðđçŮřçŽDèçÈéèřăŽlriijNèŁŽăÿĹăIJl9.5ăřRèŁĆăIJL'èðşăĹřăĂĆăĹNăĖCrijŽ

```
import types
from functools import wraps

def profiled(func):
    ncalls = 0
    @wraps(func)
    def wrapper(*args, **kwargs):
        nonlocal ncalls
        ncalls += 1
        return func(*args, **kwargs)
    wrapper.ncalls = lambda: ncalls
    return wrapper
```

(continues on next page)

(continued from previous page)

```
# Example
@profiled
def add(x, y):
    return x + y
```

ncalls
add.ncalls()

```
>>> add(2, 3)
5
>>> add(4, 5)
9
>>> add.ncalls()
2
>>>
```

11.10 9.10 äyžčśzǎŠŇéíŽæĀAæŮzæšTæRŘä;ŽèĚéěřǎŽí

éŮóécŸ

ä;ǎæČšžZčśzæĹŮéíŽæĀAæŮzæšTæRŘä;ŽèĚéěřǎŽíǎǺĆ

èġčǎEşæŮzæǎĹ

čžŽčśzæĹŮéíŽæĀAæŮzæšTæRŘä;ŽèĚéěřǎŽíǎǺĲŁčŏǺǎ■TčŽďijŇǎy■èĚĜèĚAçǎŏǎĲèĚĚéěřǎŽíǎǺĲ
@classmethod æĹŮ @staticmethod äžŇǎĹ■ǎǺĆǎ;ŇǎĚĆijŽ

```
import time
from functools import wraps

# A simple decorator
def timethis(func):
    @wraps(func)
    def wrapper(*args, **kwargs):
        start = time.time()
        r = func(*args, **kwargs)
        end = time.time()
        print(end-start)
        return r
    return wrapper
```

```
# Class illustrating application of the decorator to different
↳ kinds of methods
class Spam:
    @timethis
    def instance_method(self, n):
```

(continues on next page)

(continued from previous page)

```
        print(self, n)
        while n > 0:
            n -= 1

    @classmethod
    @timethis
    def class_method(cls, n):
        print(cls, n)
        while n > 0:
            n -= 1

    @staticmethod
    @timethis
    def static_method(n):
        print(n)
        while n > 0:
            n -= 1
```

ěĚěěřăŔŎčŽDčšzăŠŇéÍŽæĀAæŪzæşŦăŔŕæ■čăyŷăũčăĭJĭijŇăŔlăy■ēĤĜăćďăĽăăžĚćíăď'ŮčŽĎěőæŮ

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

ěőĹěőž

ăĕĆăďĬJăĭăæĽĽěĚěěřăŽÍčŽĎéąžăžŔăĚŽéŦŽăžĚăřšăijŽăĜžéŦŽăĀĆăĭŇăĕĆĭijŇăAĜěőĭăĭăăĈŔăyŇéĬć

```
class Spam:
    @timethis
    @staticmethod
    def static_method(n):
        print(n)
        while n > 0:
            n -= 1
```

éĆčăžĽăĭăĕŦĈşŦĬēĤŽăŷĹéÍŽæĀAæŪzæşŦăŮăřšăijŽæĽěéŦŽĭijŽ

```
>>> Spam.static_method(1000000)
Traceback (most recent call last):
```

(continues on next page)

(continued from previous page)

```
File "<stdin>", line 1, in <module>
File "timethis.py", line 6, in wrapper
start = time.time()
TypeError: 'staticmethod' object is not callable
>>>
```

```

        ěŮőécŸaIJlāžŎ                                @classmethod                                āšŇ                                @staticmethod
        āōđēŽĚäyŁāžŭäy■aijŽāŁZāžžāŖŕçŽŦ æŌēēŕČçŦlçŽĎāŕžēsaijŇ
        èĀŇæŸŕāŁZāžžçŁŦžæŌŁçŽĎæŖŖēēŕāŽlāŕžēsā(āŖČēĀČ8.9ārŖēŁČ)āĀČāŽāæ■dŦāŦšāŦæŕŦçlĀāIJlāĒūāžŮēč
        çāōāŦlēŦŽçg■ēčĒēēŕāŽlāGžçŌŕāIJlēcĒēēŕāŽlēšçŦäy■çŽĎçñnäŸĀäyŦāŦçŦŕāŕfāžēāŦōādŦēŦŽäyŦēŮőécŸāĀČ
        āŦšāŦĒsāžñāIJlæŁŦēsāāšžçszäy■āōŽāžŁŦçszæŮžæsŦāšŇēlŦæĀĀæŮžæsŦ(āŖČēĀČ8.12ārŖēŁČ)æŮūŕijŦ
        āŦŇāēŦŕijŇŇāēČādIJāŦāēČsāōŽāžŁŦäyĀäyŦæŁŦēsāçszæŮžæsŦijŇŇāŕfāžēāŦçŦlçszāijijäyŇēlčçŽĎāžčçāŦŕijŽ
    
```

```
from abc import ABCMeta, abstractmethod
class A(metaclass=ABCMeta):
    @classmethod
    @abstractmethod
    def method(cls):
        pass
```

```

    @IJlèfZæotjzçăĂăy■iijN@classmethod          èu$          @abstractmethod
    äy'd'èÄËcZDëažazRæYræIJL'ëöscł'úcZDriijNăeĆædIJä;äerÇă■cáoCăznçZDëažazRărsäi;ZăGzëTZăĂĆ

```

11.11 9.11 èĈĖėřăZlăyžèćńăŃĖĉĖĖĜjæTřăćďăŁăăRĆæTř

éŮőécŸ

ä;äxČšǎlJlčĚēēřǎZlǎy■czŽēčnǎNĚčĚǎĜ;æTrǎcđǎLǎēcǎđ' ŮčŽĐǎRĆæTrǐj;Nǎ;EǎYřǎy■ēČ;ǎ;sǎS■ēfZ

èğčǎẸșæŮźæǻŁ

ǎRǎřěä;ŁćTłǻĚřěTǫǎ■ŮǎRĆæTrǎĲěčŻěćńǎŇěčĚǎĜ;æTrǎćďǎŁǎěćĲǎďŮǎRĆæTrǎǺćĚǺćěŻŚǎyŇéĲć

```
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
>>>

```

èóìèőž

éĂŽèŁĠèċĚēēŕăŽłæłēçzŻēćnăŇĚēċĚăĠ;æŦŕăċďăŁăăŦĊæŦŕçŽďăĂŽæşŦăżúăy■ăyÿèğĂăĂĆ
 ăŕ;çôăăċĈæ■ď'ijŇăIJL'æŮŭăĂŽăőĈăŦŕăžēéĂġăĚ■ăyĂăžŽéĠăď'■ăžċăĂăĂĆă;ŦăċŦijŇăċĈăđIJă;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')

```

éĆčăžŁă;ăăŦŕăžēăŦĚăĚŭéĠăđĐăĹŦŕēŁŽăăŭijŽ

```

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
```

èŁŻçġ■āōđçŎřæŰzæāŁāzŊæŁ'ĀāzēēāŊāŁŰēĀŽījŊāĪĪāžŎāījžāŁūāĒşēŦōā■ŰāŔĈæŦřāŁāōzæŶşēćŋā
 *args āŤŊ **kwargs āŔĈæŦřçŽĎāĠ;æŦřāŷ■āĀĈ éĀŽèŁĠāŁçŦĪāījžāŁūāĒşēŦōā■ŰāŔĈæŦřījŊāōĈēćŋā
 āžūāŷŦæŎēāŷŊāēāzĒāzĒāŁçŦĪāĪŦā;ŽçŽĎā;■ç;ōāŤŊāĒşēŦōā■ŰāŔĈæŦřāŎžēŦçŦĪēŁŽāŷĪāĠ;æŦřæŰūīj
 āžşāŕŝæŶřēŦŦījŊāōĈāzūāŷ■āījŽēćŋçžşāĒēāĪŦ **kwargs āŷ■āŎzāĀĈ

èŁŶæĪĪĪāŷĀāŷĪēŽŁçĈzāŕŝæŶřāēĈā;ŦāŎzād'ĎçŔĒēćŋæūzāŁāçŽĎāŔĈæŦřāŷŎēćŋāŊĒēćĒāĠ;æŦřāŔĈæ
 āŁŊāēĈījŊāēĈāđĪēćĒēēŕāŽĪ @optional_debug ā;ĪĪçŦĪāĪĪāŷĀāŷĪāūşçzŔæŊēæĪĪĪāŷĀāŷĪ
 debug āŔĈæŦřçŽĎāĠ;æŦřāŷŁæŰūāījŽæĪĪĪēŰōēćŶāĀĈ èŁŽēĠŊæŁŤsāžŋāćđāŁāāžĒāŷĀæ■ēāŔ■ā■ŰāēĈĀæ

āŷĪēĪćçŽĎæŰzæāŁēŁŶŶāŔŕāzēæŽŦ'āōŊçŁŎāŷĀçĈzījŊāZāāŷžçş;æŶŎçŽĎçĪŊāžŔāŤŶāžŦēŕāŔŤçŎŕāž

```
>>> @optional_debug
... def add(x, y):
...     return x+y
...
>>> import inspect
>>> print(inspect.signature(add))
(x, y)
>>>
```

éĀŽèŁĠāŁçŦĪāījžāŁūāĒşēŦōā■ŰāŔĈæŦřāŷŎēćŋāŊĒēćĒāĠ;æŦřāŔĈæ

```
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
```

éĀŽèĤĜèĤŽæũçŽĎăĤœŤžijŇăŇĚèĤĚăŔŎçŽĎăĜĭæŤŕç■ĭăŔ■ăŕsèĈĭæ■ççăŏçŽĎăŸĭçđ'ž
debug âŔĈæŤŕçŽĎă■ŸăĬĬăžĒăĂĈăĭŇăĕĈijŽ

```
>>> @optional_debug
... def add(x, y):
...     return x+y
...
>>> print(inspect.signature(add))
(x, y, *, debug=False)
>>> add(2, 3)
5
>>>
```

âŔĈèĂĈ9.16ârŔèĤĈèŎûâŔŮæŽŧ'ăđ'ŽăĚşăžŎăĜĭæŤŕç■ĭăŔ■çŽĎăĤăæĀŕăĂĈ

11.12 9.12 äĭĤçŤĭèĈĒēēŕăŽĭæĻŧ'ăĒĒçşżçŽĎăĤşèĈĭ

éŮŏéćŸ

äĭăæĈşéĂŽèĤĜăŔ■çĬĬæĻŮèĂĒéĜ■ăĒŽçşżăŏŽăžĻçŽĎăşŔĒĈĭăĻĒæĭăăĤœŤžăŏĈçŽĎăŇăŸžijŇăĭĒ

èğĉăĒşæŮžæąĻ

èĤŽçğ■ăĈĒăĒŧăŔŕèĈĭæŸŕçşżèĈĒēēŕăŽĭæĬĬăēĭçŽĎăĭĤçŤĭăĬĬæŽŕăžĒăĂĈăĭŇăĕĈijŇăŸŇéĭĈæŸŕăŸĂă
__getattr__ çŽĎçşżèĈĒēēŕăŽĭijŇ âŔŕăžæĻŧşă■ŕăŮăăĤŮijŽ

```
def log_getattribute(cls):
    # Get the original implementation
    orig_getattribute = cls.__getattr__

    # Make a new definition
    def new_getattribute(self, name):
        print('getting:', name)
        return orig_getattribute(self, name)

    # Attach to the class and return
    cls.__getattr__ = new_getattribute
    return cls

# Example use
@log_getattribute
class A:
    def __init__(self, x):
        self.x = x
    def spam(self):
        pass
```

ăŸŇéĭĈæŸŕăĭĤçŤĭæŤĻăđĬijŽ


```
>>> a = A(42)
>>> a.x
getting: x
42
>>> a.spam()
getting: spam
>>>
```

èõìèõž

çşzèçĖėėřăŽléĂŽăÿăŕŕăzēă;IJăÿžăĖũăžŮénŸçžğæŁĂæIJŕæŕŤæĆăũăăĖēæŁŮăĖĈçşžçŽDăÿĂçğ■ėİđăæŕŤæĆiijNăÿLéİçđ'žă;Năÿ■çŽDăŕăđ'ŮăÿĂçğ■ăôđçŎŕă;ŁçŤlăĹŕçžğæŁŕiijŽ

```
class LoggedGetattribute:
    def __getattr__(self, name):
        print('getting:', name)
        return super().__getattr__(name)

# Example:
class A(LoggedGetattribute):
    def __init__(self, x):
        self.x = x
    def spam(self):
        pass
```

èĤŽçğ■æŮžæăĹăžşēăNă;ŮēĂŽiijNă;ĒæŸŕăÿžăžĒăŎžçŖĒēğçăŏĆiijNă;ăăŕşăĤĖēăžçşēēĂşæŮžæşŤŕĈăăžēăŖĹăĖũăăŏĈ8.7ăŕŖēĹĆăžNçž■çŽDçžğæŁŕçşşēŕĒăĂĆ æşŖçğ■ćĹNăžăÿŁæĹēèŏşiiijNçşzèçĖėėřăŽlăŮžæăăžăăÿžăžŮăžũăÿ■ă;ĹètŮ super() âĠ;æŤŕăĂĆ

ăĕĆăđIJă;ăçşžæĈşăIJăÿĂăÿĹçşžăÿĹéİćă;ŁçŤlăđ'ŽăÿĹçşzèçĖėėřăŽlăiijNéĆăžĹăŕşéIJăĕĕĂæşĹăĎŖăÿNéă;NăĕĆiijNăÿĂăÿĹēçĖėėřăŽlăăiijŽăŕĒăĖũēçĖėėŕçŽDăŮžæşŤăŏNăŤŕ'æŽĤæ■ćăĹŖăŕăÿĂçğ■ăôđçŎŕiijNăĖNăŖăÿĂăÿĹēçĖėėřăŽlăŖăŕĹăŸŕçŏĂă■ŤçŽDăIJăăĖũēçĖėėŕçŽDăŮžæşŤăÿ■ăũăăĹăçĆžéćĹăđ'ŮēĂžē;ŠăĂĖéĆăžĹĤēĤăŮăĂžēçĖėėřăŽlăăŕşéIJăĕĕĂæŕŤ;ăIJăĹēçĖėėřăŽlăBçŽDăĹ■éİćăĂĆ

ă;ăĕĤŸăŕŕăzēăŽđēă;ăÿĂăÿN8.13ăŕŖēĹĆăŕăđ'ŮăÿĂăÿĹăĒşăžŎçşzèçĖėėřăŽlçŽDăIJĹçŤlçŽDă;Nă■Ŗ

11.13 9.13 ä;ŁçŤlăĖĈçşžæŎğăĹăăôđă;NçŽDăĹŽăžž

éŮŏéćŸ

ă;ăæĈşēĂŽēĤĠăŕŸăôđă;NăĹŽăžžæŮžăiijŖăĹēăôđçŎŕă■Ťă;NăĂăçijŞă■ŸăĹŮăĖũăžŮçşžăiijçŽDă

èğçăĒşæŮžæăĹ

PythonćĹNăžŖăŞŸēĆ;çşēēĂşiiijNăĕĆăđIJă;ăăŏŽăžĹăžĒăÿĂăÿĹçşžiiijNăŕşēĆ;ăĈŖăĠ;æŤŕăÿĂăăũçŽDă

```
class Spam:
    def __init__(self, name):
        self.name = name

a = Spam('Guido')
b = Spam('Diana')
```

æĆædIJä;äæČšèĠåőŽăZL'èĚŽăylæ■ēēld'rijNă;ăăRřăžěăőŽăZL'ăyĂăylăĚČşşăăžűēĠåűsăőđčŐř
 __call__() æŰzæşŢăĂĆ
 äyžăĚæijŢčd'žrijNăĀĠèö;ă;ăäy■æČşăžă;ŢăžžăĹŽăžžèĚŽăylşşčŽĎăđăĹNrijŽ

```
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')
```

èĚŽăăüçŽĎèrijNşŢlăĹăRlèČ;èřČŢlèĚŽăylşşčŽĎéIŽæĂĀæŰzæşŢrijNèĂNăy■ēČ;ă;ĚčŢlèĂŽăyç

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

çŐřăIJlrijNăĀĠăĚăČă;äæČşăőđčŐřă■ŢăĹNăĹărijRrijĹăRlèČ;ăĹŽăžžăŢřăyĂăđăĹNşŽĎçşrijLrijNăőđč

```
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')
```

éĆčázĹSpamçşzârşâRĭèČĭăĹZăzzăTřäyĂçŽDăóďăĹNăžEĭijNăejTčd'žăĕĆăyNĭijŽ

```
>>> a = Spam()
Creating Spam
>>> b = Spam()
>>> a is b
True
>>> c = Spam()
>>> a is c
True
>>>
```

æIJĀăRŎĭijNăAĞèőĹăĭăæČşăĹZăzz8.25ârRèĹĆăy■éĆčæăüçŽDçijŞă■ŸăóďăĹNăĂĆăyNéĭćăĹSăznăRă

```
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
```

çDŭăRŎăĹSăzşæĭěæĭNèřTăyĂăyNĭijŽ

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

èóíéőž

ǎĹ'čŤlǎĚČšzǎđđŔǎđ'Žčğ■ǎđđǎĹNǎĹZǎžžǎĹǎĹjŔéĂžǎŷŷēēAǎŕŤǎŷ■ǎ;ŕčŤlǎĚČšzčŽĐǎŮžǎĹŔǎĹjŮ
ǎĂĜēōĹǎ;ǎǎŷ■ǎ;ŕčŤlǎĚČšzġġNǎ;ǎǎŔŕēČ;éIJǎēēAǎŕĚčšzéŽŔēŮŔǎIJǎšŔǎžŽǎŷēǎŎCǎĜ;ǎŤŕǎŔŎéĹcǎĂ
ǎŕŤǎēCǎŷžǎžĚǎđđŔŎŕǎŷĂǎŷĹǎ■ŤǎĹŦġġNǎ;ǎǎ;ǎǎŔŕēČ;ǎġjŽǎČŔǎŷŦéĹēŔŽǎǎŷǎĚŽġġjŽ

```
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
```

ǎŕ;čđǎǎ;ŕčŤlǎĚČšzǎŔŕēČ;ǎġjŽǎŷĹ'ǎŔĹǎĹŕǎŕŤē;ČénŮčžğČžčŽĐǎĹĂǎIJŦġġNǎ;ĚǎŮŕǎđŔčŽĐǎžččǎĂ
ǎŽŦ'ǎđ'ŽǎĚšǎžŎǎĹZǎžžčġjŠǎ■ŮǎđđǎĹNǎĂĂǎġjšǎġjŤčŤĹč■ĹǎĚǎđžġġjNŕŕŷǎŔČēĂČ8.25ǎŔŔēĹCǎĂČ

11.14 9.14 ǎ■ŤēŎŭčšzčŽĐǎđđǎĂğǎŎŽǎžĹ'éǎžǎžŔ

éŮđéčŮ

ǎ;ǎǎČšēĜĹǎĹēōŕǎ;ŤǎŷĂǎŷĹčšzǎŷ■ǎđđǎĂğǎŦNǎŮžǎšŤǎđŽǎžĹ'čŽĐéǎžǎžŔġġN
čĐŮǎŔŎǎŔŕǎžēǎĹ'čŤlǎđŎCǎĹēǎĂžǎĹĹǎđ'Žǎš■ǎ;IJġġĹǎŕŤǎēCǎžŔǎĹŮǎNŮǎĂĂǎŷŮǎǎŕĐǎĹŕǎŤŕǎēǎžšč■Ĺ'č

èğčǎĚšǎŮžǎǎĹ

ǎĹ'čŤlǎĚČšzǎŔŕǎžēǎĹĹǎđžǎŮščŽĐǎ■ŤēŎŭčšzčŽĐǎđŽǎžĹ'ǎŔǎǎŔǎĂCǎŷŦéĹcǎŮŕǎŷĂǎŷĹǎĹNǎ■ŔġġN

```
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()

```

OrderedDict`æ■TèÕûâLřiiĴŃ çTŠæLŔçŽDæIJL' āžRāŔ■çğřāžŌā■ŪāĒÿäÿ■æŔŔāŔŪāĠžæİē
 ``_order äÿ■āĀČæŹæūçŽDēŔçšzäÿ■çŽDæŪzæşTāŔŕāžēēĀŽēŔĠād'Žçğ■æŪzāijŔæİēä;ŔçŤlāŏČāĀČ
 äĴŃāēČŕijŃäÿŃēİcæŸŕäÿĀäÿlçŏĀā■ŤçŽDçşzĕijŃä;ŔçŤlēŹäÿlæŌŠāžŔā■ŪāĒÿæİēāŏđçŌŕāŕĒäÿĀäÿlçşzāŏđæ

```

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

```

æĴŖšāžŃāĴlāžd'āžŠāijŔçŌŕācČäÿ■ætŃērŤäÿĀäÿŃēŹäÿlStockçşzĕijŽ

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

ěóľěőž

æIJñĚĹĆäyÄäyĹăĚšĕŤōçĆzārśæŸrOrderedMetaāĚČšzäy■ăōŽăzĹ'çŽĎ “ __pre-
 pare__()“ æŮzæşŤăĀĆ ěĚžăyĹăŮzæşŤăijŽăIJĹăijĂăğŇăōŽăzĹ'çşzăŞŇăōČçŽĎĹŮçşzçŽĎæŮŮăĂŽĕcŋæĹ'ğ
 æĹŚăznĕřŽĕĜŇĕĂŽĕĚĜĕĤăŽďăžĚăyÄäyĹăOrderedDictĕĂŇăy■æŸrăyÄäyĹăŽōĕĂŽçŽĎ■ŮăĚyĭijŇăŔřăžĕă
 âĕĆăđIJăjăæČşăđĎĕĂăĕĜĹăŮşçŽĎçşză■ŮăĚyăržĕşăĭijŇăŔřăžĕăĹăŮžæŸşçŽĎæĹ'ŮăşŤĕĚăyĹăĹşĕČjă

```
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)
```

äyŇĕĹăĹŚăznăĕŤŇĕřŤĕĜ■ăđ'■çŽĎăōŽăzĹ'ăijŽăĜžçŎřăžĂăzĹăČĚăĚĭijŽ

```
>>> class A(metaclass=OrderedMeta):
...     def spam(self):
...     pass
...     def spam(self):
...     pass
```

(continues on next page)

(continued from previous page)

```
...
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "<stdin>", line 4, in A
  File "dupmethod2.py", line 25, in __setitem__
    (name, self.clsname))
TypeError: spam already defined in A
>>>
```

```
__new__()
class
dict
d = dict(clsdict)
```

```
as_csv()
```

```
class Stock(Model):
    name = String()
    shares = Integer()
    price = Float()
```

```
as_csv()
```

11.15 9.15

æŮœćŸ

æŮœćŸ

èğçàĒşæŮzæąĹ

“metaclass”

```
from abc import ABCMeta, abstractmethod
class IStream(metaclass=ABCMeta):
    @abstractmethod
    def read(self, maxsize=None):
        pass
    @abstractmethod
```

(continues on next page)

(continued from previous page)

```
def write(self, data):  
    pass
```

çĐüèĀŇīījŇāĪĪlèĠāōŽāzĻāĒĈçśzäy■æĻŚāznèfŸāŔŕāzèæŔŔäĳZāĒüāzŪçŽĐāĒşéŤōā■ŪāŔĈæŤŕīījŇā

```
class Spam(metaclass=MyMeta, debug=True, synchronize=True):  
    pass
```

äyžāŸEäĳfāĒĈçśzæŤŕæŇAèfŽāžZāĒşéŤōā■ŪāŔĈæŤŕīījŇāĳāāfĒēāzçāōāfĪāĪĪĪ
__prepare__() , __new__() āŖŇ __init__() æŪzæşŤäy■
éĈĳäĳfçŤĪāījžāĻūāĒşéŤōā■ŪāŔĈæŤŕāĀĈāŕsāĈŔāyŇéĪcèfŽæūīījŽ

```
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)
```

èóíèőž

çžŽāyĀāyĪāĒĈçśzæūzāĻāāŔŕéĀĻāĒşéŤōā■ŪāŔĈæŤŕéĪĀèçAāĳāāōŇāĒĪāījĐæĠĈçśzāĻZāžžçŽĐæĻĀā
āZāāyžèfŽāžZāŔĈæŤŕāījŽècŇāījæĀŖçžZæŕŔāyĀāyĪçŽyāĒşçŽĐæŪzæşŤāĀĈ

__prepare__() æŪzæşŤāĪĪæĻĀæĪĪĻçśzāōŽāzĻāījĀāğŇæĻğèāŇāĻ■ēçŪāĒĒcŇèŕĈçŤĪīījŇçŤĪæĪāĻZ.
éĀŽāyŷæĪèçōŕīījŇèfŽāyĪæŪzæşŤāŔĪæŸŕçōĀā■ŤçŽĐèfŤāZđāyĀāyĪā■ŪāĒyæĻŪāĒüāzŪæŸāŕĐāŕžèşāāĀĈ
__new__() æŪzæşŤècŇçŤĪæĪèāōđäĳŇāŇŪæĪĀçžĻçŽĐçśzāŕžèşāāĀĈāōĈāĪĪçśzçŽĐāyžäĳşècŇæĻğèāŇāō
__init__() æŪzæşŤæĪĀāŔŌècŇèŕĈçŤĪīījŇçŤĪæĪæĻğèāŇāĒüāzŪçŽĐāyĀāžZāĻāğŇāŇŪāūèäĳĪāĀĈ

āĳşæĻŚāznædĐéĀāāĒĈçśzçŽĐæŪūāĀŽīījŇéĀŽāyŷāŔĪèĪĀèçAāōŽāzĻāyĀāyĪ
__new__() æĻŪ __init__() æŪzæşŤīījŇāĳEäy■æŸŕāyđ'äyĪèĈĳāōŽāzĻāĀĈ
āĳEæŸŕīījŇāçĈæđĪĪĪĀèçAæŌèāŔŪāĒüāzŪçŽĐāĒşéŤōā■ŪāŔĈæŤŕçŽĐèŕīījŇèfŽāyđ'äyĪæŪzæşŤāŕşèçAā
ézŸèōđ'çŽĐ __prepare__() æŪzæşŤæŌèāŔŪāžzæĐŔçŽĐāĒşéŤōā■ŪāŔĈæŤŕīījŇāĳEæŸŕāījŽāfĳçŤèāō

æL'ÄäzëäRlæIJL'ä;ŞëfZäzZëcİad'ŪçZDäRCæTṛäRrëÇ;äijZä;säŞ■äLrçsžäS;äR■çl'zéŪt'çZDäLZäzæŪüä;äa
 __prepare__() æŪzæşTṛÄĆ
 éÄZëfGä;fçTlāijžāLūāĖşéTōā■ŪāRCæTṛiijNāIJlçsžçZDäLZäzžëfGçlNäy■æLŠäzñāfĖēāzēÄZëfGäĖş
 ä;fçTlāĖşéTōā■ŪāRCæTṛēĖ■ç;ōäyÄäyġāĖČçsžèfYāRfäzèègĖä;IJärzçsžäRÝéGRçZDäyÄçg■æZäzčæ

```
class Spam(metaclass=MyMeta):
    debug = True
    synchronize = True
    pass
```

ārĖēfZäzZäsdæÄgāōZäzL'äyžäRCæTṛçZDäe;ād'DäIJlāžŌāōČzāñäy■äijZæsäæşŞçsžçZDäR■çgrçl'zéŪt'
 èfZäzZäsdæÄgāzĖäzĖāRlāzŌāsdāžŌçsžçZDäLZäzžëfYūæōṛiijNēÄNäy■æYrçsžäy■çZDēr■āRēæL'gēāNéYūā
 āRēād'ŪiijNāōČzāñāIJl__prepare__() æŪzæşTäy■æYrāRrāzèècñèōfēŪōçZDiiijNāZäyžëfZäyġæŪzæşT
 ä;ĖæYrçsžäRÝéGRāRlëÇ;āIJlāĖČçsžçZD__new__() äŠN__init__() æŪzæşTäy■āRrègAāÄĆ

11.16 9.16 *argsäŠN**kwargsçZDäijžāLūāRCæTṛç■çäR■

éŪōécY

ä;äæIJL'äyÄäyġāGjæTṛæLŪæŪzæşTṛiijNāōČä;fçTl'argsäŠN**kwargs;IJäyžäRCæTṛiijNēfZæäüä;fäçŪ
 ä;ĖæIJL'æŪüāÄZä;äæČşæçÄæŞëäijäēÄŞëfZæġçZDäRCæTṛæYräy■æYräşRäyġä;äæČşëçAçZDçsžādNāÄĆ

ègçÄĖşæŪzæāL

ärzäzä;TæŪL'āRlāLræŞ■ä;IJäGjæTṛērČçTlç■çäR■çZDēŪōécYiijNä;äēČ;āžTērēä;fçTl
 inspect æġāLŪäy■çZDç■çäR■çL'zæÄgāÄĆ æLŠäzñæIJäyžèçAāĖşæşlāy'd'äyġçsžiiijZSignature
 äŠN Parameter āÄÇäyNēlçæYräyÄäyġāLZäzžāGjæTṛäL'■ēlççZDäz'd'äžŠäçNā■RiijZ

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

äyÄæŪēä;äæIJL'äžĖäyÄäyġç■çäR■ärzèšäiijNä;ääršäRfäzèä;fçTlāōČçZD bind()
 æŪzæşTṛäçLāōžæYŞçZDärĖāōČçzŠāōžāLr *args äŠN **kwargs äyġāŌžāÄĆ
 äyNēlçæYräyÄäyġçōÄ■çZDäijTçd'žiiijZ

```
>>> def func(*args, **kwargs):
...     bound_values = sig.bind(*args, **kwargs)
```

(continues on next page)

(continued from previous page)

```
...     for name, value in bound_values.arguments.items():
...         print(name,value)
...
>>> # Try various examples
>>> func(1, 2, z=3)
x 1
y 2
z 3
>>> func(1)
x 1
>>> func(1, z=3)
x 1
z 3
>>> func(y=2, x=1)
x 1
y 2
>>> func(1, 2, 3, 4)
Traceback (most recent call last):
...
  File "/usr/local/lib/python3.3/inspect.py", line 1972, in _bind
    raise TypeError('too many positional arguments')
TypeError: too many positional arguments
>>> func(y=2)
Traceback (most recent call last):
...
  File "/usr/local/lib/python3.3/inspect.py", line 1961, in _bind
    raise TypeError(msg) from None
TypeError: 'x' parameter lacking default value
>>> func(1, y=2, x=3)
Traceback (most recent call last):
...
  File "/usr/local/lib/python3.3/inspect.py", line 1985, in _bind
    '{arg!r}'.format(arg=param.name))
TypeError: multiple values for argument 'x'
>>>
```

ãŕŕãzẽçIJŇãGžãİëñjÑëĂŽẽĝĞãŕĖç■;ãŕ■ãŠŇãijãëĂŠçŽĐãŔĆãŦŕçzŠãõŽẽtũãİëñjŇãŔŕãzẽãijžãŁũãĜ;
äyŇëİćãŸŕäyĂäyİãijžãŁũãĜ;ãŦŕç■;ãŕ■ãŽŦ'ãĖũã;ŞçŽĐã;Ňã■ŔãĂĆãIJİãzççãAäy■ñjŇãĹSãzñãIJİãşçç
__init__() æŰžãşŦñjŇçĐũãŔŖœĹSãzñãijžãŁũãĹĂæIJĹçŽĐã■ŔçşzãĤĖãzãŔŔã;ŽäyĂäyİçĹ'žãõŽçŽ

```
from inspect import Signature, Parameter

def make_sig(*names):
    parms = [Parameter(name, Parameter.POSITIONAL_OR_KEYWORD)
              for name in names]
    return Signature(parms)

class Structure:
```

(continues on next page)

(continued from previous page)

```
__signature__ = make_sig()
def __init__(self, *args, **kwargs):
    bound_values = self.__signature__.bind(*args, **kwargs)
    for name, value in bound_values.arguments.items():
        setattr(self, name, value)

# Example use
class Stock(Structure):
    __signature__ = make_sig('name', 'shares', 'price')

class Point(Structure):
    __signature__ = make_sig('x', 'y')
```

äyÑélcæYřä;ŁçTlèŁŻäyŁ Stock ŁşŁŻĎĎd'žä;NřijŽ

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

èóléőž

āJlāŁŚāznēIJĀēēAæđDāzēēĀŽçTlāĜ;æTřāžŠāĀAçijŮāEžēēĒēēāZlāŁŮāōđçŎřāžčçŘEçŽĎæŮūāĀZ
*args āŠŇ **kwargs çŽĎä;ŁçTlæYřä;ŁæŽóéA■çŽĎāĀĆ
ä;EæYřijŇēŁZæūçŽĎāĜ;æTřæIJL'äyĀäylçijžçČzāřsæYřä;Šä;æČšēAāōđçŎřēĜlāūsçŽĎāŘCæTřæčĀéłN
ēŁZæŮūāĀZæŁŚāznāRřāžēēĀŽēŁĜāyĀäylç■;āŘ■āřzēsælēçōĀāŇŮāōČāĀĆ

āJlāIJĀāRŎçŽĎäyĀäylæŮzæāŁāōđä;Ňäy■řijŇæŁŚāznēŁYāRřāžēēĀŽēŁĜä;ŁçTlèĜlāōŽāzL'āĒČşzæł

```
from inspect import Signature, Parameter

def make_sig(*names):
    parms = [Parameter(name, Parameter.POSITIONAL_OR_KEYWORD)
              for name in names]
    return Signature(parms)

class StructureMeta(type):
    def __new__(cls, clsname, bases, clsdict):
        clsdict['__signature__'] = make_sig(*clsdict.get('_fields',
→ []))
```

(continues on next page)

(continued from previous page)

```
        return super().__new__(cls, clsname, bases, clsdict)

class Structure(metaclass=StructureMeta):
    _fields = []
    def __init__(self, *args, **kwargs):
        bound_values = self.__signature__.bind(*args, **kwargs)
        for name, value in bound_values.arguments.items():
            setattr(self, name, value)

# Example
class Stock(Structure):
    _fields = ['name', 'shares', 'price']

class Point(Structure):
    _fields = ['x', 'y']
```

```
__signature__
inspect
```

```
>>> import inspect
>>> print(inspect.signature(Stock))
(name, shares, price)
>>> print(inspect.signature(Point))
(x, y)
>>>
```

11.17 9.17

éÚóécŸ

äjäçŽDćlŃăžŔăŃĖăŔăŃăŸĂăŸlăŁăd'ğçŽDçşzçğæL'£ă;ŞçşziŃŃă;ăăŸŃăIJZăijzăLŮăL'ğăăŃă\$ŔăžZçij

èğčăEşşăŮzæąŁ

```
type
__new__
__init__
```

```
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)
```

āŕĕäyǺçġ■æŸřijŇăőŽázL' __init__() æŰzæşŦiijŽ

```
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
```

äyžāŹĒä;ŕçŦlĕŹäyĴāĔĈşziiŇŇä;ăĖĂžäyÿĕĖAārĒăőĈæŦç;ăĴŕăĴŕäyĂäyĴēăŭçžğçĴŭçşzăőŽázL' äy■ijŇçĴ

```
class Root(metaclass=MyMeta):
    pass

class A(Root):
    pass

class B(Root):
    pass
```

ăĔĈşşzçŽĎäyĂäyĴāĔĖşĕŦōçL'žçĈzæŸŕăőĈăĔĖăőŷă;ăăĴĴăőŽázL'çŽĎæŰŭăĂŽæĈĂæşĕçşzçŽĎăĔĖăőžă.
__init__() æŰzæşŦäy■ijŇ ä;ăăŦŕăžēăçĴĴē;zæĴçŽĎæĈĂæşĕçşză■ŰăĔŷăĂĀçĴŭçşzç■Ĵç■ĴăĂĈăžŭäyŦŕ
ăŽăă■d'ijŇäyĂäyĴēăĖăđŭçŽĎăđĎăžžĕĂĔăŕşĕĈ;ăĴĴăđ'ğăđŇçŽĎçžğæL'ŕă;Şçşzäy■ĖĂŽĕŕĜçzŽäyĂäyĴēăŭ

ă;ĴJäyžäyĂäyĴāĔŭă;ŞçŽĎăžŦçŦĴăçŇă■ŦrijŇäyŇĕĴăőŽázL'ăžĒäyĂäyĴāĔĔĈşziiŇŇăőĈăijŽæŇŞçzĴăžză;Ŧ

```
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
```

ă;ĴJäyžæŽŦ'ĕŇŸçžğăŦŇăőđçŦĴçŽĎăçŇă■ŦrijŇäyŇĕĴăĴĴL'äyĂäyĴāĔĔĈşziiŇŇăőĈçŦĴăĴēăĈĂæŦŇĕĜ■ĕ;Ŧ

```
from inspect import signature
import logging

class MatchSignaturesMeta(type):
```

(continues on next page)

(continued from previous page)

```
def __init__(self, clsname, bases, clsdict):
    super().__init__(clsname, bases, clsdict)
    sup = super(self, self)
    for name, value in clsdict.items():
        if name.startswith('_') or not callable(value):
            continue
        # Get the previous definition (if any) and compare the
        ↪ signatures
        prev_dfn = getattr(sup, name, None)
        if prev_dfn:
            prev_sig = signature(prev_dfn)
            val_sig = signature(value)
            if prev_sig != val_sig:
                logging.warning('Signature mismatch in %s. %s !
                ↪= %s',
                                value.__qualname__, prev_sig,
                ↪ val_sig)

# Example
class Root(metaclass=MatchSignaturesMeta):
    pass

class A(Root):
    def foo(self, x, y):
        pass

    def spam(self, x, *, z):
        pass

# Class with redefined methods, but slightly different signatures
class B(A):
    def foo(self, a, b):
        pass

    def spam(self, x, z):
        pass
```

æĈædIJă;æĕŦRèaÑèĕZæôĭăzĉĉăAĭijÑăřsăijŽă;ŮăĽrăyÑéĬĕĕZæăŭĉŽĎĕ;ŞăĜžĉzŞædIJĭijŽ

```
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)
```

ĕĕŽĉĝ■ē■ĉăSĽăĕAĕrăřzăŽŌă■ŦēŌŭăyĂăžŽă;ôăĕŽĉŽĎĉĬNăžRbugæŸră;ĽăIJĽĉŦĬĉŽĎăĂĈă;NăĕĈĭj
éĈăžĽă;Şă■ŦĉşzæŦzăRŸăŦĈăŦŕăŦ■ă■ŮĉŽĎăŮăăĂŽăřsăijŽĕŦĈĉŦĬăĜžĕŦŽăĂĈ

èõlèõž

àIJläd' gädNéicáRŠáržèsäçŽDçlNäzRäy■iijNéÄŽäyÿärEçsžçŽDäöŽäzL' æTç; àIJlãĚČsžäy■æŎğáLúæYř
ãĚČsžäRřäzèçŽSæŎğçsžçŽDäöŽäzL' iijNè■æSŁçijŮçlNäžžäSŸæšŘäzŽæšæIJL' æšlæĎRāLřçŽDāRřèç; àĠ

æIJL' äžžāRřèç; äijŽer' iijNāČRēfŽæäüçŽDēTŽerřāRřäzèèÄŽèfGçlNāžRāLEædRāüèāĚūæLŮIDEāŎžā
ä; EæYřiijNāçCædIJä; āāIJlædDāžžäyÄäyſææEædūæLŮāG; æTřäžŠä; ŽāĚūäzŮäžžä; fçTl iijNéCčäzLä; äæšāāL
āŽāæ■d' iijNāržäžŎèfŽçg■çsžādNçŽDçlNāžRiijNāçCædIJāRřäzèāIJlãĚČsžäy■æAŽæçĀætNæLŮèöyāRřäzè

àIJlãĚČsžäy■æĀL' æNl' éG■æŮřāöŽäzL' _____new____() æŮžæšTēfYæYř
____init____() æŮžæšTāRŮāEšäžŎä; äæČsæĀŎæäüä; fçTlçzŠædIJçsžāĀČ _____new____()
æŮžæšTāIJçsžāLŽäžžāzNāL' ■ècnèrČçTl iijNéÄŽäyÿçTlāžŎéÄŽèfGæšŘçg■æŮžäijRiijLæfTāçCéÄŽèfGæT
èĀN _____init____() æŮžæšTæYřāIJçsžècnāLŽäžžāzNāRŎècnèrČçTl iijNā; Šä; æēIAèçAāōNæTř' ædDāžžçsž.
àIJlæIJāĀRŎäyÄäyſä; Nā■Räy■iijNèfŽæYřæfEèçAçŽD iijNāŽäyžāōČä; fçTlāžE super()
àĠ; æTřæIēæRIJçt' cāzNāL' ■çŽDäöŽäzL' āĀČ āōČāRlèç; àIJlçsžçŽDäöä; NècnāLŽäžžāzNāRŎiijNāžūäyTçZ

æIJāĀRŎäyÄäyſä; Nā■RēfYæijTçd' žāžEPythonçŽDāG; æTřç; äR■āržèsäçŽDä; fçTlāĀČ
āōdēŽĚäyL iijNāĚČsžärEæfRäyſāRřèrČçTlāōŽäzL' æTç; àIJlāyÄäyſçsžäy■iijNæRIJçt' cāl' ■äyÄäyſāōŽäzL' iijL
çDūāRŎéÄŽèfGä; fçTl inspect.signature() æIèçōĀā■TçŽDæfTè; ČāōČāžñçŽDèrČçTlç; äR■āĀČ

æIJāĀRŎäyÄçCž iijNāžčçāAäy■æIJL' äyÄèāNä; fçTlāžE super(self, self)
āžūäy■æYřæŎšçL' LēTŽerřāĀČ ä; Šä; fçTlāĚČsžçŽDæŮūāÄŽiijNæLŠäžñèçAæŮūāLžèōrā; RāyÄçCžāršæY
self āōdēŽĚäyLæYřäyÄäyſçsžāržèsāĀČ āŽāæ■d' iijNèfŽæIæf■āRēāĚūāōdāršæYřçTlæIēāržæL' ä; ■āžŎçz
self çLŮçsžçŽDäöŽäzL' āĀČ

11.18 9.18 äžèçijŮçlNæŮžäijRāōŽäzL'çsž

éŮèéçY

ä; āāIJlāEŽäyÄæōtāžççāA iijNæIJÄçzLéIJÄèçAāLŽäžžäyÄäyſæŮřçŽDçsžäržèsāĀĀČä; äèĀČèŽSārEçsžç
āžūäyTä; fçTlāG; æTřæfTāç exec() æIēæL' gèāNāōČiijNä; EæYřä; äæČšāržæL' äyÄäyſæŽt' āLāāijYéŽĚçZ

èğçāEšæŮžæāL

ä; āāRřäzèä; fçTlāG; æTř types.new_class() æIēāLiāgNāNŮæŮřçŽDçsžäržèsāĀĀČ
ä; äēIJÄèçAāAŽçŽDāRlæYřæRŘä; ŽçsžçŽDāR■ā■ŮāĀAçLŮçsžāĚČçzDāĀāĚšçTōā■ŮāRČæTřiijNäžèāRŁ

```
# stock.py
# Example of making a class manually from parts

# Methods
def __init__(self, name, shares, price):
    self.name = name
    self.shares = shares
    self.price = price
def cost(self):
    return self.shares * self.price
```

(continues on next page)

(continued from previous page)

```
cls_dict = {
    '__init__': __init__,
    'cost': cost,
}

# Make a class
import types

Stock = types.new_class('Stock', (), {}, lambda ns: ns.update(cls_
    dict))
Stock.__module__ = __name__
```

èŁŻçġæŰzâijRâijŽædĐâžžäyĂäylæŽôéĂŽçŽĐçšzâržèšâijŇâžũäyŤæŇLčĚğă;ăçŽĐæIJšæIJŽâuëă;IJi

```
>>> s = Stock('ACME', 50, 91.1)
>>> s
<stock.Stock object at 0x1006a9b10>
>>> s.cost()
4555.0
>>>
```

èŁŻçġæŰzæşŤäy■rijŇäyĂäylæŤè;ČéŽ;çŘĚğççŽĐâIJæŰzæŸŤâIJlërČçŤlăôŇ
types.new_class() âŕž Stock.__module__ çŽĐèŤŇâĀijăĂĆ
æŤŤæŇăă;šăyĂäylçšžèçŇăôŽăžL'ăŖŎrijŇăôČçŽĐ __module__
ăšđæĂğăŇĚăŖŇăôŽăžL'ăôČçŽĐæŤăăŮăŖ■ăĂĆ èŁŽäylăŖ■ăŮçŤlăžŎçŤšæĹŖ
__repr__() æŰzæşŤçŽĐè;šăĜžăĂĆăôČăŖŇæăũăžšèçŇçŤlăžŎă;Ĺăđ'ŽăžšijŇæŤŤăç
pickle âĂĆ âŽăæ■'rijŇäyžăžĚèŎ'ă;ăăĹăžžçŽĐçšzæŸŤâIJæ■ççăôăĂĭçŽĐrijŇă;ăéIJăèĚăçăôăŤièŁŽäy

ăĚČăđIJăăčšăĹăžžçŽĐçšžéIJăèĚăŷĂäylăy■ăŖŇçŽĐăĚČçšžrijŇăŖŤăžééĂŽèŁĜ
types.new_class() çŇŇäyL'äylăŖĆæŤŤăijăéĂšçžăôČăĂĆăŇăçŤrijŽ

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

çŇŇäyL'äylăŖĆæŤŤèŁŸăŖŤăžéăŇĚăŖŇăĚăžŮçŽĐăĚšéŤôă■ŮăŖĆæŤŤăĂĆæŤŤăçŤrijŇäyĂäylçšžçŽĐăô

```
class Spam(Base, debug=True, typecheck=False):
    pass
```

éĆčăžĹăŖŤăžéăŤŤăĚŮçŤžèŤšăĹŖăçĂyŇçŽĐ new_class() èŤČçŤlă;ăâijRijŽ


```
Spam = types.new_class('Spam', (Base,),
                        {'debug': True, 'typecheck': False},
                        lambda ns: ns.update(cls_dict))
```

```
new_class()
__prepare__()
update()
lambda ns: ns.update(cls_dict))
```

namedtuple

```
namedtuple()
collections.namedtuple()
```

```
>>> Stock = collections.namedtuple('Stock', ['name', 'shares',
        ↳ 'price'])
>>> Stock
<class '__main__.Stock'>
>>>
```

```
namedtuple()
exec()
```

```
import operator
import types
import sys

def namedtuple(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
```

```
>>> Point = namedtuple('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
>>>
```

```
Stock = type('Stock', (), cls_dict)
```

```
import types
metaclass, kwargs, ns = types.prepare_class('Stock', ()), {'metaclass': type})
```

354

11.19 9.19 aJlãóŽázL'çŽDæUúãÄŽãLiãġNãÑŮçszçŽDæLRãSŸ

éUóécŸ

äjäæČšãIJłçszècñãóŽázL'çŽDæUúãÄŽãLiãġNãÑŮäyÄéČlãLEçszçŽDæLRãSŸiijNèÄÑäy■æŸrèeAç

èġčãEşæŮzæqL

ãIJłçszãóŽázL'æUúãrãsæL'ġèãNãLiãġNãÑŮæLŮèóç;óæŞ■ä;IJæŸrãĚČçszçŽDäyÄäyłãĚyãdNãžTçTlãIJ
èŁŽæUúãÄŽã;ããRřãžèæL'ġèãNäyÄžŽéclãd'ŮçŽDæŞ■ä;IJãÄĆ

äyNélcæŸřäyÄäyłã;Nã■RřijNãL'çTlèŁŽäyłæÄlèûrælèãŁŽãžçszçäiijjãžŮ
collections ælããlŮäy■çŽDãS;ãR■ãĚČçzDçŽDçszçiijŽ

```
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)
```

èŁŽæóřãžçčãAãRřãžèçTlælèãóŽázL'çóÄã■TçŽDãšžãžŮãĚČçzDçŽDæTřæ■óçzŞædDřijNãeČäyNæL'Äç

```
class Stock(StructTuple):
    _fields = ['name', 'shares', 'price']

class Point(StructTuple):
    _fields = ['x', 'y']
```

äyNélcæijTçd'žãóČãeČã;Tãuëã;IJijŽ

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

ěóľěőž

ěĚZäYÄärRèLCäy■īijŇčšz StructTupleMeta ěŌŭāRŪāLŕčšzāsdæĀğ _fields
äy■čŽĎāsdæĀğāR■āŭāLŪēāīijŇčĎŭāRŌārĒāōČāzñē;ñæ■cæLŔčŽyāzTčŽĎāRfēōēŪōčL'zāōŽāĚČčzĎæ;
operator.itemgetter() āLŽāzžāyÄäyľēōēŪōāŽlāG;æTŕīijŇčĎŭāRŌproperty()
āG;æTŕārĒāĚŪē;ñæ■cæLŔäyÄäyľāsdæĀğāĀĆ

æIJñèLCæIJĀēŽ;æGČčŽĎēČlāLĒæYŕčšēēAšāy■āRŇčŽĎāLlāgNāNŪæ■ēēld'æYŕāzĀāzLæŪŭāĀŽāRš
StructTupleMeta äy■čŽĎ __init__() æŪzæšTāRlāIJlāfRäyľčšzēcñāōŽāzL'æŪēcñēŕČčTlāyĀæñā.
cls āRČæTŕāršæYŕēČčäyľēcñāōŽāzL'čŽĎčšzāĀĆāōēēŽĒäyLīijŇāyLēfŕāzččāAā;ččTlāzĒ
_fields čšzāRŸēGRælēāflā■YæŪŕčŽĎēcñāōŽāzL'čŽĎčšzīijŇ
čĎŭāRŌčzŽāōČāĒæŭzāLāyĀčČzæŪŕčŽĎäyIJēēfāĀĆ

StructTuple čšzā;IJäyžāyÄäyľæŽōēĀŽčŽĎāšžčšzīijŇā;ŽāĚŭāzŪā;ččTlāĀĒælēčžgæL'fāĀĆ
ěĚZäyľčšzāy■čŽĎ __new__() æŪzæšTčTlālēēēdĎēĀāæŪŕčŽĎāōdā;NāĀĆ ěĚZēGŇā;ččTl
__new__() āžŭāy■æYŕā;LāyŷēgAīijŇāyžēēAæYŕāZāāyžæLšāzñēēAāfōæTzāĚČčzĎčŽĎēŕČčTlč;āR■īij
ā;ččŪāLšāzñāRŕāzēāČRæŽōēĀŽčŽĎāōdā;NēŕČčTlēČčæāŭāLŽāzžāōdā;NāĀĆāŕšāČRäyNēlēēēZæāŭīijŽ

```
s = Stock('ACME', 50, 91.1) # OK
s = Stock(('ACME', 50, 91.1)) # Error
```

ěŭš __init__() äy■āRŇčŽĎæYŕīijŇ__new__() æŪzæšTāIJlāōdā;NēcñāLŽāzžāzNāL'■ēcñēgēāRš
čTšāzŌāĚČčzĎæYŕāy■āRfāfōæTzčŽĎīijŇæL'ĀāzēāyĀæŪēāōČāzñēcñāLŽāzžāzĒāršāy■āRfēČ;āržāōČāĀŽā
__init__() äijŽāIJlāōdā;NāLŽāzžčŽĎæIJĀāRŌēcñēgēāRšīijŇ
ěĚZæāŭčŽĎēŕlāLšāzñāršāRŕāzēāĀŽæLšāzñæČšāĀŽčŽĎāžĒāĀĆēēZāžšæYŕāyžāzĀāzL
__new__() æŪzæšTāŭščžŔēcñāōŽāzL'āžĒāĀĆ

ār;čōāæIJñèLCā;Lčš■īijŇēfYæYŕēIJĀēēAā;āēČ;āzTčzĒčāTŕēzīijŇæŭšāĚēæĀlēĀČPythončšzæYŕāēČā
ěĚYæIJL'āršæYŕāĚČčšzāšŇčšzčŽĎāRĎäyľāy■āRŇčŽĎæŪzæšTčl'ŭčŇšāIJlāzĀāzLæŪŭāĀŽēcñēŕČčTlāĀĆ

PEP 422 æRŔā;ZāžĒäyÄäyľēgčāĒæIJñèLCēŪōēēYčŽĎāŕēād'ŪäyĀčg■æŪzæšTāĀĆ
ā;ĒæYŕīijŇæLlæ■cāLŕæLšāĒēēZēēZæIJñāžēčŽĎæŪŭāĀŽīijŇāōČēēYæšāēcñēēGčžšāšŇæŌēāRŪāĀĆ
ār;čōāēČæ■d'īijŇāēČædIJā;āā;ččTlčŽĎæYŕPython 3.3æLŪēZt'énYčŽĎčL'LæIJīijŇēČčāzLēēYæYŕāĀijā

11.20 9.20 āL'čTlāG;æTŕæšlēgčāōdčŌŕæŪzæšTēG■ē;ī

ěŪōēčY

ā;āāŭščžŔā■ēēfGæĀŌæāŭā;ččTlāG;æTŕārČæTŕæšlēgčīijŇēČčāzLā;āāRfēČ;āijŽæČšāL'čTlāōČælēāō
ā;ĒæYŕā;āy■čāōāōŽāzTēŕēæĀŌæāŭāŌzāōdčŌŕīijLæLŪēĀĒāLŕāžTēāNā;ŪēĀŽāy■īijL'āĀĆ

èġċăEşæŮzæąĹ

æIJnărRèLCçŽDæLĂæIJræYřăşzăžŌăyĂăylçôĂă■TçŽDæLĂæIJřijŇéCčărsæYřPythonăĚAęőyăRĆæT

```
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
```

äyŇéİcæYřæĹSăznçnnăyĂă■ċŽDărĭerTřijŇă;ĚçTĭăĹrăžEăyĂăylăĚCçsăŠŇæRŘèřăŽĭijŽ

```
# 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

```

(continued from previous page)

```
'''
def __new__(cls, clsname, bases, clsdict):
    return type.__new__(cls, clsname, bases, dict(clsdict))

@classmethod
def __prepare__(cls, clsname, bases):
    return MultiDict()
```

äyžāEä;fçTlëfZäyŁçszijŇä;ääRräzëäČRäyŇéİçèfZæăũăEŻiijŽ

```
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)
```

äyŇéİçæYřäyÄäyŁäzd'äžŠçd'žä;ŇäİçéİŇëfAăóČèČ;æ■čçăőčŽDăũă;IJiijŽ

```
>>> s = Spam()
>>> s.bar(2, 3)
Bar 1: 2 3
>>> s.bar('hello')
Bar 2: hello 0
>>> s.bar('hello', 5)
Bar 2: hello 5
>>> s.bar(2, 'hello')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "multiple.py", line 42, in __call__
    raise TypeError('No matching method for types {}'.
↪format(types))
TypeError: No matching method for types (<class 'int'>, <class 'str'
↪'>)
>>> # Overloaded __init__
>>> d = Date(2012, 12, 21)
```

(continues on next page)

(continued from previous page)

```
>>> # Get today's date
>>> e = Date()
>>> e.year
2012
>>> e.month
12
>>> e.day
3
>>>
```

èõlëõž

āleçŽ;āleëõšijNçŽyāržāžŌéĀŽāyŷçŽDāžççāAēĀNāūsæIJnèLĆā;ŁçTlāLřāžEā;Lād'ŽçŽDēTæšTāžçç;ā;EāYřijNāōČā'èČ;èōl'æLŠāžnæūsāĒēçRĒēğçāĒČçšāŠNæRRēřāŽlçŽDāžTāsČāuēā;IJāŌšçRĒijNāžūēČ;āLāæūsāržēfZāžZæČāŁççŽDā'řēsāāĀČāZāæ'd'ijNārščōŪā;āāžūāyāijŽçnNāšāŌžāžTçTlāIJnèLĆāōČçŽDāyĀāžZāžTāsČæĀIæČšā'āijŽā;śāŠāLřāĒūāōČæūL'āRĀLāLřāĒČçšāĀAæRRēřāŽlāŠNāG;æTřæš

æIJnèLĆçŽDāōđçŌřāyçŽDāyžèeAæĀIèurāĒūāōđæYřā;LçōĀāTçŽDāĀĆMultipleMetaāĒČçšā;ŁçTlāōČçŽD __prepare__() æŪzæšT æIææRRā;ŽāyĀāylā;IJāyž MultiDictāōđā;NçŽDēGlāōZāZL'āŪāĒyāĀČēfZāylēušæZōéĀŽāŪāĒyāyāyĀæāuçŽDæYřijNMultiDict āijŽāIJlāĒČçt'āèçnèō;ç;ōçŽDæŪūāĀZæčĀæšæYřāRēāuščžRāYāIJlījNāçCæđIJāYāIJlçŽDMultiMethod āōđā;NāyāRĀLāžūāĀĆ

MultiMethod āōđā;NéĀŽēfGæđDāžžāžŌçšžādNç;āRāāLřāG;æTřçŽDæYāārDæIææTūēZEæŪzæšTāIJlēfZāylæđDāžžēfGçlNāyijNāG;æTřæšlēğçèçnçTlāIææTūēZEēfZāžZç;āRçDūāRŌæđDāžžēfZāylæYēfZāylēfGçlNāIJl MultiMethod.register() æŪzæšTāyāōđçŌřāĀĆēfZçgæYāārDçŽDāyĀāylāĒšēTōçL'žçČæYřāržāžŌād'ŽāylæŪzæšTijNæL'ĀæIJL'āRČæTřçšžādNéČ;āfĒē

āyžāžEèōl' MultiMethod āōđā;NālæāNšāyĀāylēřČçTlījNāōČçŽD__call__() æŪzæšTēçnāōđçŌřāžEāĀĆ ēfZāylæŪzæšTāžŌæL'ĀæIJL'æŌšÉŽd' slefçŽDāRČæTřāyæđDāžžāyĀāylçšžādNāĒČçžDijNāIJlāĒĒēČlmapāyæšæL'ēfZāylæŪzæšTijNçDūāRŌēřČçTlçŽyāžTçŽDæŪzæšTāĀČāyžāžEèČ;èōl' MultiMethodāōđā;NāIJlçšžāōŽāZL'æŪūāççāōæSā;IJlījN__get__() æYřāfĒēāžā;ŪāōđçŌřçŽDāĀĆāōČèçnçTlāIææđDāžžæççāōçŽDçžSāōZæŪzæšTāĀČæřTāçCijŽ

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

āyæfGæIJnèLĆçŽDāōđçŌřēfYæIJL'āyĀāžZēŽRāLūijNāĒūāyāyĀāylæYřāōČāyēČ;ā;ŁçTlāĒšēTōā


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

äzšëöÿæIJL'äĖüāzŮčŽDæŮzæšTèČ;æúžāŁăēfZçĝ■æTŕæŇAĭijŇNä;EæYŕăōČéIJĀēēAäyĀäyĽăōŇăĚĽäy■
éŮōécYāIJĽāzŎāĚšéTōā■ŮāŔCæTŕçŽDăGžçŎŕæYŕæšqæIJL'éqžāžŔçŽDăĀČā;ŠăōČëŭ\$ä;■ç;ôāŔCæTŕæŭŭāĽ
éCčä;ăçŽDăŔCæTŕăŕšăijZăŔYă;ŮăŕTè;ČæŭŭāzšăžEĭijŇēfZæŮŭăĀŽă;ăäy■ă;Ůäy■ăIJĽ
__call__() æŮzæšTäy■ăĚĽăŎžăĀŽäyĽăŎšăžŔăĀČ

ăŔŇăăŭăŕžăžŎçzĝæŁăžšæYŕæIJL'éZŔăĽŮçŽDĭijŇNä;ŇăēČĭijŇçšzăijijäyŇéĽcēfZçĝ■ăžççăĀăŕšăy■ēČ;

```
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)
```

ăŎšăZăæYŕăZăäyž x:A æšĽëĝčäy■ēČ;æĽŔăĽšăŇzéĚ■ă■Ŕçšzăōđă;ŇĭijĽăŕTăēCBçŽDăōđă;ŇĭijĽ'ĭijŇă

```
>>> s = Spam()
>>> a = A()
>>> s.foo(a)
Foo 1: <__main__.A object at 0x1006a5310>
>>> c = C()
>>> s.foo(c)
Foo 2: <__main__.C object at 0x1007a1910>
>>> b = B()
>>> s.foo(b)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "multiple.py", line 44, in __call__
    raise TypeError('No matching method for types {}'.
↪format(types))
```

(continues on next page)

(continued from previous page)

```
TypeError: No matching method for types (<class '__main__.B'>,)
>>>
```

```
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[tuple(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
```

```
class Spam:
    @multimethod
    def bar(self, *args):
        # Default method called if no match
        raise TypeError('No matching method for bar')

    @bar.match(int, int)
    def bar(self, x, y):
        print('Bar 1:', x, y)

    @bar.match(str, int)
```

(continues on next page)

(continued from previous page)

```
def bar(self, s, n = 0):  
    print('Bar 2:', s, n)
```

æRŘèfrãZÍæÚzæqLãRÑæũázšæIJL'ãL'■éíææRŘãLřçŽĐéŽŘãLũiiJLäy■æTřæÑAãĚšéTřoã■ÚãRĆæTřãš
æL'ĂæIJL'ăžŇçL'íĚČ;æYřázšç■L'çŽĐřijŇæIJL'ăě;æIJL'ăIŘřijŇăžšèőyæIJĂăě;çŽĐăLđæşTăřsæYřăIJă
ăy■ēĚGăIJL'ăžŽçL'zæőLæĈĚăĚtăyŇēĚYæYřæIJL'æĐŘăžL'çŽĐřijŇæřTăĚĆăşžăžŎăġăijRăŇzéĚ■çŽĐæÚz
ăy;ăyġă;Ňă■ŘřijŇ8.21ăřRèLĆăy■çŽĐèőĚéŮőēĂĚăġăijRăRřăžēăĚőăTžăyžăyĂăyġă;ĚçTġæÚzæşTéĚ■ē;çŽ
ă;ĚæYřijŇēŽđ'ăžĚēĚŽăyġăžēăđ'ŮřijŇēĂžăyăy■ăžTēřăă;ĚçTġæÚzæşTéĚ■ē;řijLăřsçőĂă■TçŽĐă;ĚçTġăy■ă
ăIJPythonçđ'ġăŇžăřžăžŎăőđçŎřæÚzæşTéĚ■ē;çŽĐèőġēőžăũşçžRçTşăġăũşăžĚăĂĈ
ăřžăžŎăijTăRŞēĚŽăyġăžL'èőžçŽĐăŎşăZăřijŇăRřăžēăRĆēĂĈăyŇGuido van
RossumçŽĐēĚŽçřĚă■ŽăőçřijŽ Five-Minute Multimethods in Python

11.21 9.21 éAŁăĚ■ēG■ăđ'■çŽĐăşđæĂġæÚzæşT

éŮőéçY

ă;ăăIJġçşăy■ēIJăĚēĂēĚăđ'■çŽĐăőŽăžL'ăyĂăžZæL'ġēăŇçŽyăRŇēĂžē;ŞçŽĐăşđæĂġæÚzæşTřijŇæřT

èġçĂĚşæÚzæqL

èĂĈèŽŞăyŇăyĂăyġçőĂă■TçŽĐçşzijŇăőĈçŽĐăşđæĂġçTşăşđæĂġæÚzæşTăŇĚēĈĚřijŽ

```
class Person:  
    def __init__(self, name , age):  
        self.name = name  
        self.age = age  
  
    @property  
    def name(self):  
        return self._name  
  
    @name.setter  
    def name(self, value):  
        if not isinstance(value, str):  
            raise TypeError('name must be a string')  
        self._name = value  
  
    @property  
    def age(self):  
        return self._age  
  
    @age.setter  
    def age(self, value):  
        if not isinstance(value, int):
```

(continues on next page)

(continued from previous page)

```
raise TypeError('age must be an int')
self._age = value
```

ãRràzèçIJNãLrriijNäyžazEãóðçÕrãsdæÄgãÄijçZDçszãdNæcÄæšæeLŠäznãEŽãžEã;Lãd'ŽçZDëG■ãd'■ã
ãRlèeAä;ääzèãRÕçIJNãLrçszãijijèfZæãüçZDãžçãÄijNä;æc;ãžTèræcÇsãLdæsTãÕžçõÄãNŨãõCãÄC
äyÄäyIãRrèãNçZDæŨzæsTæYrãLZãžzäyÄäyIãG;æTçTlãIèãõŽãzL'ãsdæÄgãzüèfTãZdãõCãÄCã;NæCrijZ

```
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
```

èõlèõž

æIJnèLČæLŠäznæijTçd'žãEÈéČlãG;æTæLŨèÄÈéŨ■ãNÈçZDäyÄäyIèG■èeAçL'zæÄgrijNãõCãznã;Lã
typed_property() çIJNäyLãÕžæIJL'çCžéŽ;çRÈègçijNãEüãóðãóCæL'ÄãAŽçZDãžEãžEãrsæYrãyžã;äç
ãZãæ■d'iijNä;ŠãIJläyÄäyIçszäy■ä;fçTlãõCçZDæŨüãÄZiijNæTlædIJèüšãrEãõCèGÑelççZDãžçãÄæT;ãLr
ãr;çõãsdæÄgçZD getter ãŠN setter æŨzæsTèøféŨõãžEæIJnãIJrãYëGRæc name ,
expected_type äžèãRL storate_name iijNèfZäyIã;Læ■çäyüijNèfZãžZãRÝëGRçZDãÄijäijZãfIã■Y

æLŠäznèfYãRràzèã;fçTl functools.partial() ælèçl■çl■æTzãRÝäyNèfZäyIã;Nã■RrijNã;LæIJL

```
from functools import partial

String = partial(typed_property, expected_type=str)
Integer = partial(typed_property, expected_type=int)

# Example:
```

(continues on next page)

(continued from previous page)

```
class Person:
    name = String('name')
    age = Integer('age')

    def __init__(self, name, age):
        self.name = name
        self.age = age
```

āĖūāōđā;āāRřāzēāRŠçŎřijNēŁŻēĜŇçŽĎāzčçāĀēũ\$8.13āRŘēŁCāy■çŽĎçšzādŇçšzçzšæRŘēřāZlāzčçā

11.22 9.22 āōŽāzL'āyŁāyNāŮĜçōaçŘĒāZlçŽĎçŎĀā■TæŮzæşT

éŮōécŸ

ä;āæČşēĜlāūsāŎzāōđçŎřāyĀāylāēŮřçŽĎāyŁāyNāŮĜçōaçŘĒāZlçijNāzēäŁæ;ŁçŦlwithēř■āRēāĀĆ

èĝcāEşæŮzæąŁ

āōđçŎřāyĀāylāēŮřçŽĎāyŁāyNāŮĜçōaçŘĒāZlçŽĎāIJāçŏĀā■ŦçŽĎāŮzæşTārsæŸřā;ŁçŦl
contextlib ælāāIŮāy■çŽĎ @contextmanager èčĚēēřāZlāĀĆ
āyNēlĀæŸřāyĀāylāōđçŎřāzĒāzčçāĀāIŮēōāæŮūāŁşèČ;çŽĎāyŁāyNāŮĜçōaçŘĒāZlçĬNā■ŘijŽ

```
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
```

āIJlāĜ;æŦřtimethis() äy■ijNŸyield āzNāL■çŽĎāzčçāĀāijŽāIJlāyŁāyNāŮĜçōaçŘĒāZlāy■ā;IJāy
__enter__() æŮzæşTæL'ĝēāŇrijŇ æL'ĀæIJL'āIJl yield āzNāRŎçŽĎāzčçāĀāijŽā;IJāyž
__exit__() æŮzæşTæL'ĝēāŇāĀĆ āēČæđIJāĜçŎřāzĒāijČāyŸrijNāijČāyŸāijŽāIJlyield-
ēr■āRēēĆčēĜNāŁZāĜzāĀĆ

āyNēlĀæŸřāyĀāylāēŽŦ āŁāénŸçžĝāyĀçČzçŽĎāyŁāyNāŮĜçōaçŘĒāZlçijNāōđçŎřāzĒāLŮēāłāřzēsāyŁ

```
@contextmanager
def list_transaction(orig_list):
    working = list(orig_list)
    yield working
    orig_list[:] = working
```

èĚŽæōțăzçăAçŽĎä;IJçTĭæYřăzzä;TărzâĹŮëaĭçŽĎăĔōæTřâRĭæIJĹ'â;ŞæL'ĂæIJĹ'ăzçăAèĚŘëaŇăōŇæĭ
ăyŇéĭcæĹSăzŇæĭcæijTçd'žăyĂăyŇijŽ

```
>>> 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]
>>>
```

èōĭèōž

éĂŽăyŷæCĚăĔăyŇijŇăcĎđIJècAăEZăyĂăyĭăyĹăyŇæŮĜcōaçŘĚăŽĭijŇă;ăéIJăècAăōŽăzĹ'ăyĂăyĭc
__enter__() āSŇăyĂăyĭ __exit__() æŮzæşTĭijŇăcCăyŇæĹ'Ăçd'žĭijŽ

```
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))
```

ăr;çōaèĚŽăyĭăzşăy■éŽ;ăEZĭijŇă;ĔæYřcŽyæřTè;ČăEZăyĂăyĭcōĂă■TçŽĎä;ĤçTĭ
@contextmanager æşĭèğççŽĎăĜ;æTřèĂŇéĭĂèĚYæYřcĭ■æYĭăzRăSşăĂĆ

(continued from previous page)

```
...     print(b)
...
>>> test()
14
>>>
```

ěōlēōž

āōđēŽĚäyŁáržāžŎ exec() çŽDæ■ççāōä;ŁçTlæYřæfTè;ČěŽçŽDāĀĆad'gād'ŽæTřæČĚāĚtäyNā;Šā;äē
exec() çŽDæŮūāĀŽiijN èŁYæIJL'āRēād'ŮæŽt'āē;çŽDèğčāĚşæŮzæāLiiJLærTāēČēcĚēčāZlāĀĀéŮ■āNĚā

çDūēĀNrijNāēČædIJä;āāz■çDūēĚAä;ŁçTl exec() iijNæIJnèŁČāLŮāGžāžĚäyĀāžZāēČä;Tæ■ççāōä;Łç
ézYēōd'æČĚāĚtäyNiiJNexec() äijZāIJlērČçTlēĀĚāsĀēČlāŠNāĚlāsĀēNČāZt'āĚĚāL'gēāNāžççāĀāĀČçDū
äijāēĀŠçžŽ exec() çŽDāsĀēČlēNČāZt'æYřæNūètIāōđēŽĚāsĀēČlāRŸéGRçžDæLRçŽDäyĀäylā■ŮāĚyāĀ
āZāæ■d'iijNāēČædIJ exec() āēČædIJæL'gēāNāžĚāŁōæTzæŞ■ä;IJiijNèŁŽçg■āŁōæTzāRŎçŽDçzŞædIJáržā
äyNélcæYřāRēād'ŮäyĀäylæijTçd'žāōČçŽDä;Nā■RiijŽ

```
>>> def test1():
...     x = 0
...     exec('x += 1')
...     print(x)
...
>>> test1()
0
>>>
```

äyLélcāžççāĀēGNrijNā;Šā;äērČçTl locals() èŮāRŮāsĀēČlāRŸéGRæŮiijNā;äēŮā;ŮçŽDæYřäi
exec() çŽDāsĀēČlāRŸéGRçžDäyĀäylæNūètIāĀĆ ēĀŽēŁGāIJlāžççāĀæL'gēāNāRŎāōāæşēēŁZäylā■ŮāĚyā

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

āžTçzĚēğČārşæIJāRŎäyĀæ■ēçŽDè;ŞāGžiiijNéŽd'ēIdā;āārE loc
äy■ēčnāŁōæTzāRŎçŽDāĀijæL'NāLlētNāĀijçžŽxiiJNāRēāLZxāRŸéGRāĀijæYřäy■äijZāRŸçŽDāĀĆ

āIJlā;ŁçTl locals() çŽDæŮūāĀŽiijNā;äēIJĀēĚAæşlæDRæŞ■ä;IJēāžāžRāĀĆærRāñāōČēcñērČçTlç
locals() äijŽēŮūāRŮāsĀēČlāRŸéGRāĀijäy■çŽDāĀijāžūēĚçŽŮā■ŮāĚyāy■çŽyāžTçŽDāRŸéGRāĀĆ
ērūæşlæDRēğČārşäyNäyNélcēŁZäylærTçTēNçŽDè;ŞāGžçzŞædIJiijŽ


```
>>> 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}
>>>
```

æʃlæDRæIJĀāRŌäyĀæñæřČčTl locals() çŽDæŮŭāĀŽxçŽDāĀijæYřæĆä;TēcñèçEçZŮæŌLçŽDā
ä;IJäyž locals() çŽDäyĀäyĭæZfäzçæŮzæqLijNä;āāRřæžæ;ŁçTlā;æĜĥaũsçŽDā■ŮāĚyijNāzŭârEāō
exec() āĀĆä;NāçCijZ

```
>>> def test4():
...     a = 13
...     loc = { 'a' : a }
...     glb = { }
...     exec('b = a + 1', glb, loc)
...     b = loc['b']
...     print(b)
...
>>> test4()
14
>>>
```

ād'gēČlāLEæČĚāEřäyNijNèŁZçg■æŮzāijRæYřä;ŁçTl exec() çŽDæIJĀä;şāōđeuřāĀĆ
ä;āāRřæIJĀèçAāĭĭerAāĚlāsĀāSŊāsĀēČlā■ŮāĚyāIJlāRŌēlçazčçāAēōĤēŮōæŮŭāũsçzRēcñāLlāgNāNŮāĀĆ
èŁYæIJL'äyĀçCzrijNāIJlā;ŁçTl exec() äzNāL■rijNä;āāRřæČ;éIJĀèçAéŮōäyNèĜĥaũsæYřāRřæIJL'āĚ
ād'gād'ZæTřæČĚāEřäyNā;Şā;āèçAēĀČèZSā;ŁçTl exec() çŽDæŮŭāĀZrijN
èŁYæIJL'āRēād'ŮæZř'āç;çŽDēğçāEşæŮzæqLijNæřTāçCèçĚēčřāZlāĀAēŮ■āNĚāĀAāĚČçsrijNæLŮāĚŭāz

11.24 9.24 èğçædŘäyŌāLEædŘPythonæžŘçāA

éŮōécY

ä;āæČşāEŽèğçædŘāzŭāLEædŘPythonæžŘäzççāAçŽDçlNāzŘāĀĆ

èğçāEşæŮzæqL

ād'gēČlāLEçlNāzŘāSŮçşēçAşPythonèČ;ād'şēōaçŌŮāLŮāL'gēāNā■Ůçñæyřā;ćāijRçŽDæžŘäzççāAā

```

>>> 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
>>>

```

āř;çőāæĆæ■d'īijŃast ælāaiUèĈ;ècñĉTlæiēārEPythonæžŘčăAçijŮërŚæĹRăÿĂăylâRfècñăĹEæđŘčŽĐ

```

>>> 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=[]])"
>>>

```

āĹEæđŘæžŘčăAæăŚéIJĀèĕAă;æĖĹăũsæZt'ād'ŽčŽĐă■ēăžăīijŃăōĆæŸřĉŤsăÿĂçşzăĹŮASTēĹĈĉĈžčŽĐ
āĹEæđŘēĤŽăžŽēĹĈĈžæIJĀçőĂă■ŤčŽĐæŮžæşŤăřsæŸřăōŽăžĹ'ăÿĂăÿĹēōĤéŮōēĂĖçşžīijŃăōđĉŎřăĹăđ'Ž
visit_NodeName() æŮžæşŤīijŃŃodeName() āŃzéĚ■ēĈčăžŽă;ăæĐšăĖt'èũĉčŽĐēĹĈĉĈžăĂĈăÿŃéĹcæ

```

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)

```

æĆæđIJă;ăĕĲRĕăŇĕĲZăyĲĲĲŇăzŔĲĲŇă;ăăĲjZăĲŮăĲŕăyŇĕĲĕĲZăăĲĲZĎĲŖăĲzĲĲjZ

```

Loaded: {'i', 'range', 'print'}
Stored: {'i'}
Deleted: {'i'}

```

æIJăŕŔŔĲĲŇASTăŔŕăzĕĕĂZĕĲĲ compile() âĲ;æŦŕăĲĕĲĲŮĕŕŖăzŮăĲĲĕăŇăĂĲăŇăĕĲĲjZ

```

>>> exec(compile(top, '<stdin>', 'exec'))
0
1
2
3
4
5
6
7
8
9
>>>

```

ěóľěőž

ăĭȘăĭăĕĈĭăđ'șăĹĒăđŘăĕžŘăĕžĉăĂăžűăžŎăÿ■ēŎăŔŬăĕăĕĀŕĉŽĎăŬăăĂŽĭĭŬăĭăăŕŕŝēĈĭăĒŽăĹăđ'Žăžă
ăĹăŬăĕĈĭĭŬăĈĭŬăĕŕĤĉŽŝĉŽŎĉŽĎăĭăĕĂŝăÿĂăžŽăžĉăĂăĈĹăĜăŏĤăĹŕĉŝăĭĭĭ
exec() ăĜĭăĤŕăÿ■ĭĭŬăĭăăŔŕăžēăĒĹăŕĒăŏĈēĭă■ĕăĹŔăÿĂăÿĹASTĭĭŬă
ĉĎŭăŔŎĕĝĈăŕŝăŏĈĉŽĎĉĭĒĈĉĬŬăŏĈăĹŕăžĤăŸŕăĂŎăăŭăĂŽĉŽĎăĂĈ
ăĭăĕĕŸăŔŕăžēăĒŽăÿĂăžŽăŭēăĒŭăĹēăŝĕĬŬăŝŔăÿĹăĹăĹŬăĈŽĎăĒĹēĈăĕžŔĉăĂĭĭŬăŬăžűăÿĤăĬĬă■đ'ăŝžĉăĂăÿ
ēĬĂăĕĕĂăŝĹăĎŔĉŽĎăŸĭĭŬăŬăĕĈăđĬĂăĭăĉŝĕĕĂŝĕĜăĤăŭŝăĬĬăžŝăĤĕĭĭŬăĭăĕĕŸăĕĬăđ'ŝĕĜăăĒŽASTăĹēăă
ăÿŬăĹēĬăŸŕăÿĂăÿĹēĈĒēĕŕăŽĹăŬă■ŔĭĭŬăŔŕăžēăĂŽĕĕĜĕĜă■ŨŕĕĝĉăđŔăĜĭăĤŕăĭŝăžŔĉăĂăăĂă
ēĜăăĒŽASTăžűēĜă■ŨŕăĹăžăžăĜĭăĤŕăžĉăĂăŕžĕŝăĕĹăŕĒăĒĹăŝăĒēŏĕĕŨŏăŔŸĕĜŔĕŽăÿžăĜĭăĤŕăĭŝăĬĬăŬăĈ

```
# namelower.py
import ast
import inspect

# Node visitor that lowers globally accessed names into
# the function body as local variables.
class NameLower(ast.NodeVisitor):
    def __init__(self, lowered_names):
        self.lowered_names = lowered_names

    def visit_FunctionDef(self, node):
        # Compile some assignments to lower the constants
        code = '__globals = globals() \n'
        code += '\n'.join("{0} = __globals['{0}']".format(name)
                           for name in self.lowered_names)
        code_ast = ast.parse(code, mode='exec')

        # Inject new statements into the function body
        node.body[:0] = code_ast.body

        # Save the function object
        self.func = node

# Decorator that turns global names into locals
def lower_names(*namelist):
    def lower(func):
        srclines = inspect.getsource(func).splitlines()
        # Skip source lines prior to the @lower_names decorator
        for n, line in enumerate(srclines):
            if '@lower_names' in line:
                break

        src = '\n'.join(srclines[n+1:])
        # Hack to deal with indented code
        if src.startswith((' ', '\t')):
            src = 'if 1:\n' + src
        top = ast.parse(src, mode='exec')
```

(continues on next page)

(continued from previous page)

```
# Transform the AST
cl = NameLower(namelist)
cl.visit(top)

# Execute the modified AST
temp = {}
exec(compile(top, '', 'exec'), temp, temp)

# Pull out the modified code object
func.__code__ = temp[func.__name__].__code__
return func
return lower
```

äyžāžĒä;fçTlêfZäyłāzčçăĀiijNă;ăăRfrazēăČRäyNéİcèfZæăûăĒZiijŽ

```
INCR = 1
@lower_names('INCR')
def countdown(n):
    while n > 0:
        n -= INCR
```

èčĚéērăZlăijŽârĒ countdown() âĜ;æTřéĜ■ăĒZäyžçşzâijijäyNéİcèfZæăûă■RiijŽ

```
def countdown(n):
    __globals = globals()
    INCR = __globals['INCR']
    while n > 0:
        n -= INCR
```

ăIJlæĂğèČ;æıNèrTäy■iijNăôČăijŽèol'âĜ;æTřèfRëqNăłn20%

çŎřăIJlriijNă;ăæYřäy■æYřæČşäyžă;ăæL'ĂæIJL'çŽDăĜ;æTřéČ;ăLăäyLêfZäyłèčĚéērăZlăSćiijşæLŮèôy
ă;ĒæYřriijNêfZă■'æYřărzăžŎäyĂăžŽénYçžğæLĂæIJrærTăeCASTæŞ■ă;IJăĂAæžRçăAæŞ■ă;IJç■L'ç■L'çŽDă
æIJnèLCăRŮăRëad'ŮäyĂäyłăIJlActiveStateäy■ad'DçŘĒPythonă■ŮèLCçăAçŽDçnăeLCçŽDăRřç
ă;ĒçTlĀSTæYřäyĂäyłæZt'ăLăénYçžğçČzçŽDăLĂæIJriijNăžûäyTăžşæZt'çôĂă■TăžZăĂČăRČèĂČäyNéİcäy

11.25 9.25 æŊĒĒğçPythonă■ŮèLCçăA

éŮóécŸ

ă;ăæČşéĂŽèfĜârĒä;ăçŽDăzčçăAăR■çijŮërSæLŘă;ŎçžğçŽDă■ŮèLCçăAæİæşèçIJNăôČăžTăşČçŽDă

èğçăĒşæŮzæăĹ

dis æłăăiŮăRfrazèèçñçTlăİèè;ŞăĜžăžă;TPythonăĜ;æTřçŽDăR■çijŮërSçzŞædIJăĂČă;NăeČriijŽ

```
>>> def countdown(n):
...     while n > 0:
...         print('T-minus', n)
...         n -= 1
...     print('Blastoff!')
...
>>> import dis
>>> dis.dis(countdown)
...
>>>
```

èõléõž

âĴšăăæĈşèeAşşéeAŞşăăçŽĐċlNăžRăžTăşĆçŽĐèŁRèaŃæIJžăLŭçŽĐæŮŭăĂŽiijŃdis
æĴaăĴUæŸřăĴLæIJL'çTĴçŽĐăĂĈæřTăeĈăeĈădIJăăæĈşerTçĴĂçRĒèğçæĂğèĈĴçL'žăĴAăĂĈ
ècŋdis() âĴĴæTřèğçădRçŽĐăŮşăğŃăŮèŁĆçăAăeĈăyŃæL'Ăçd'žiiž

```
>>> 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"
>>>
```

ăeĈădIJăăæĈşèĴăŭşèğçèĴLèŁŽæŮžăžçăAĴiijŃăăeIJĂèeAăĴçTĴăyĂăžZăIJĴ opcode
æĴaăĴŮăyăăŮŽăžL'çŽĐăyŷeĴRăĂĈăĴŃăeĈiijŽ

```
>>> c = countdown.__code__.co_code
>>> import opcode
>>> opcode.opname[c[0]]
>>> opcode.opname[c[0]]
'SETUP_LOOP'
>>> opcode.opname[c[3]]
'LOAD_FAST'
>>>
```

ăeĴæĂĴçŽĐæŸřiijŃăIJĴdis æĴaăĴŮăyăăžŭæşăeIJL'ăĴĴæTřèŮĴăăžžèçijŮċlŃæŮžăijRăĴLăŮžæŸşçŽĐ.
ăyăeĴĴiijŃăyŃeĴççŽĐçTşæLŖăŽĴăĴĴæTřăRăžèărĒăŮşăğŃăŮèŁĆçăAăžRăLŮèĴăæĴæLŖ
opcodes âŞŃăŖĈæTřăĂĈ

```
import opcode

def generate_opcodes(codebytes):
    extended_arg = 0
    i = 0
    n = len(codebytes)
    while i < n:
```

(continues on next page)

(continued from previous page)

```
op = codebytes[i]
i += 1
if op >= opcode.HAVE_ARGUMENT:
    oparg = codebytes[i] + codebytes[i+1]*256 + extended_arg
    extended_arg = 0
    i += 2
    if op == opcode.EXTENDED_ARG:
        extended_arg = oparg * 65536
        continue
else:
    oparg = None
yield (op, oparg)
```

ä;fçTlæŰzæşTæCäyNijŽ

```
>>> for op, oparg in generate_opcodes(countdown.__code__.co_code):
...     print(op, opcode.opname[op], oparg)
```

èfŽçg■æŰzäijRä;LärSæIJL'äzzçšëeAŞijNä;ääRräzëäL'çTláoCæZæ■cäzzä;Tä;ääČşëeAæZæ■ççŽD
äyNélcæLSäzñçTläyÄäyłçd'žä;NæleæijTçd'žæT'äyłèfGçlNijŽ

```
>>> 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;ääRräzëäČRèfZæäüèÄ■äd'gæNZeöl'ègçéGLäZlæTæžČäÄCä;EæYriijNärzäžŎcijŰâEŽæZt'énYçžgai
äzŰäznâRrëČ;çIJşçŽDélJÄëeAéG■âEŽä■ŰèLCçăAāÄCæIJñèLCæIJÄâRŎçŽDëČlâlEæijTçd'žäžEèfZäylæ
this code on [ActiveState](#)

12 çññå■AçñäïïjŽælaaiUäyÓåÑĚ

ælaaiUäyÓåÑĚæYřäzzä;Täd'ğädNçlNāžRçŽDæäyåŁÇiijNāřsèŁdPythonåŁ'èçĚlNāžRæIJnèžnāžšæYřä

Contents:

12.1 10.1 ædDāzzäyÄäylælaaiUçŽDāsĆçžgāÑĚ

éUóécŸ

ä;äæČšårEä;äçŽDäzčçäAçzDçzGæLRçTšå;Låd'ŽåLEāsĆælaaiUædDæLRçŽDāÑĚäĀĆ

èğçåEşæÚzæaŁ

årAèçĚæLRåÑĚæYřä;ŁçõĀå■TçŽDāĀĆåIJæŮGäzúçşzçzşäyŁçzDçzGä;äçŽDäzčçäAïijNāžúçåõäŁæřl
ä;NāçĈiijŽ

```
graphics/  
  __init__.py  
  primitive/  
    __init__.py  
    line.py  
    fill.py  
    text.py  
  formats/  
    __init__.py  
    png.py  
    jpg.py
```

äyÄæUçä;ääAŽåŁräžEèŁŽäyĀçĆziijNä;ääžTèřèèÇ;åd'şæL'gèaŊåRDçğ■importèr■åRëiijNæCäyŊiijŽ

```
import graphics.primitive.line  
from graphics.primitive import line  
import graphics.formats.jpg as jpg
```

èóléõž

åõŽäzL'ælaaiUçŽDāsĆæñaçzŞædDåršåČRåIJæŮGäzúçşzçzşäyŁäzzçñNçŽõä;TçzŞædDäyÄæåüåõžæY
æŮGäzū__init__.pyçŽDçŽõçŽDæYřèeAåÑĚåRñäy■åRŊèŁRèaŊçžgāŁñçŽDāÑĚçŽDåréĀŁçŽDāLiāgNāŊ
äy;äylä;Nā■RiijNæCædIJä;äæL'gèaŊäžEèr■åRëimport graphicsiijŊ æŮGäzúgraph-
ics/__init__.pyårEèçñåríjåĚë,āžžçñŊgraphicsåS;åR■çl'žèŮt'çŽDåEĚåõžāĀĆåČRimport
graphics.format.jpgèŁŽæåüåríjåĚëiijNæŮGäzúgraphics/__init__.pyåŠŊæŮGäzúgraphics/formats/__init__.py

çziĀd'gèČlāLEæŮüåĀŽèõl'__init__.pyçl'žçlĀåršåè;āĀĆä;EæYřæIJL'āžŽæČĚåEĚäyNårřèÇ;åÑĚåRñäz
äy;äylä;Nā■RiijŊ__init__.pyèÇ;åd'şçTlĀlèèGĤåLlāLæè;■åRælaaiU:

ħCRèfZæăüäyĂăylæŨĠğüz,çTlæLũârRřazēāzĚāzĔēĀŽèŁGimport
 pics.formatsælēāžčæŻŁimport graphics.formats.jpgāzēâRŁimport graphics.formats.pngăĂĆ
 __init__.pyçŽDăĖūāzŪăyyçTlçTlæsȚTaÑĖæNñârĖad'ŽăylæŨĠğüzăRŁăżűăĹrăyĂăyléĂz
 æTRéTRçŽĐćĹNăžRăŚYăiiJZăRŚçÖřiiJNă■şăj;ŁæşqæIJL__init__.pyæŨĠğüză■YăIJłiiJNp

12.3 10.3 ä;ŁçŦłçŻÿárzèùrâ;ĎăŦ■ārijāĒēāŇĒäÿ■ā■ŦēłāāİŰ

éŰóécŸ

ārĒāzčçăĀçzĎçzĠăĹŦăŇĒ,æČşçŦİmportēr■āŦēāzŌāŦēäÿĀäÿłāŇĒăŦ■æşşæİJL'çañçijŰçăĀēŁĠçŻĎă

èğçăĒşşæŰzæāĹ

ä;ŁçŦłāŇĒçŻĎçŻÿárzārijāĒērijŇă;ŁäÿĀäÿłēłāāİŰārijāĒēăŦŇăÿĀäÿłāŇĒçŻĎăŦēäÿĀäÿłēłāāİŰ
äÿ;äÿłä;Ňă■ŦrijŇăĀĠğèŌ;āİJlā;ăçŻĎăŰĠzūçşçzçşäÿŁæİJL mypackageāŇĒrijŇçzĎçzĠăēČäÿŇrijŻ

```
mypackage/  
  __init__.py  
A/  
  __init__.py  
  spam.py  
  grok.py  
B/  
  __init__.py  
  bar.py
```

āēČăđİJēłāāİŰmypackage.A.spamèēĀārijāĒēăŦŇçŻŌā;ŦäÿŇçŻĎăłāāİŰgrokīijŇăŌČăžŦērēāŇĒæŇñç

```
# mypackage/A/spam.py  
from . import grok
```

āēČăđİJēłāāİŰmypackage.A.spamèēĀārijāĒēäÿ■āŦŇçŻŌā;ŦäÿŇçŻĎăłāāİŰB.barīijŇăŌČăžŦērēā;ŁçŦł

```
# mypackage/A/spam.py  
from ..B import bar
```

äÿđ'äÿłimportēr■āŦēēČ;æşşāŇĒăŦŦēāūāsČăŇĒăŦ■rijŇēĀŇæŸŦă;ŁçŦłăžĒspam.pyçŻĎçŻÿárzèùrâ;Ďă

èŌlēŌž

āİJlāŇĒăĒērijŇæŰčăŦŦä;ŁçŦłçŻÿárzèùrâ;ĎăžşāŦŦä;ŁçŦłçZlārzèùrâ;ĎălēārijāĒēăĀČ
äÿ;äÿłä;Ňă■ŦrijŻ

```
# mypackage/A/spam.py  
from mypackage.A import grok # OK  
from . import grok # OK  
import grok # Error (not found)
```

ăČŦmypackage.AēŁžæüü;ŁçŦłçZlārzèùrâ;ĎăŦ■çŻĎäÿ■āĹ'ăžŇăđ'ĎăŸŦēŁžārēāūāsČăŇĒăŦ■çañçij
äÿ;äÿłä;Ňă■ŦrijŇăēČăđİJă;ăæŦžāŦŸăžĒāŇĒăŦ■rijŇă;ăārşăŁĒēāzæčĀæşşæŁ'ĀæİJL'æŰĠzūælēăŁŌæ■čæ;
ārŇæăūrijŇçañçijŰçăĀçŻĎăŦ■çğŦrijŻă;ŁçğžăĹăžčçăĀāŦŸă;ŰăžŦēž;ăĀČäÿ;äÿłä;Ňă■ŦrijŇăžşèŌÿæİJL'ă
āēČăđİJă;ŁçŦłçŻÿárzārijāĒērijŇēČăÿĀăĹĠēČ;ŏkīijŇçŻĎēĀŇă;ŁçŦłçZlārzèùrâ;ĎăŦ■ă;ĹăŦŦēČ;ăijŻăĠžēŰ

importer ■ aRēçŽĐ . aŠŃ . . çIJNètuæIēāŁæzŚçĹ,
 ä;EāōČæŃĠāōŽçŽōā;ŦāR■.äyžā;ŠāL■çŽōā;ŦiijŃ..BäyžçŽōā;Ŧ./BāĀČèŁŽçġ■ēr■æšŦāRléĀČçŦlāžŌimport
 äyŁäyŁä;Ńā■RiijŽ

```
from . import grok # OK
import .grok # ERROR
```

ār;çōāā;ŁçŦlçŽyāržārijāĒēçIJNètuæIēāČRæYrætRēġŁæŨĠāzūçšçzçšii;Ńä;EæYrāy■èČ;āŁrāōŽāzŁ'āŃĒ
 æIJĀāRŌiijŃçŽyāržārijāĒēāRléĀČçŦlāžŌāIJlāRŁéĀČçŽĐāŃĒäy■çŽĐælaāiŨāĀČārd'āĒūæYrāIJléaūā
 ä;ŃāēČiijŽ

```
% python3 mypackage/A/spam.py # Relative imports fail
```

āRēäyĀæŨzéİçii;ŃāēČæđIJä;äā;ŁçŦlPythonçŽĐ-méĀŁ'éāzæIēæL'ġēāŃāĒĹāL■çŽĐèĎŽæIJnii;ŃçŽyār
 ä;ŃāēČiijŽ

```
% python3 -m mypackage.A.spam # Relative imports work
```

æŽt'ād'ŽçŽĐāŃĒçŽĐçŽyāržārijāĒēçŽĐèČŃæŽrçšèèrE,èrūçIJN [PEP 328](#) .

12.4 10.4 āRēālaāiŨāŁĒāL'sæŁRād'ŽäyŁæŨĠāzū

éŨóécŸ

ä;āæČšārEäyĀäyŁælaāiŨāŁĒāL'sæŁRād'ŽäyŁæŨĠāzūāĀČä;EæYrā;äāy■æČšārEāŁĒēçççŽĐæŨĠāzūççç

èġčāEşæŨzæāŁ

çlŃāžRælaāiŨāRrāžèéĀŽèŁĠāRŸæŁRāŃĒæIēāŁĒāL'sæŁRād'ŽäyŁçŃŃçŃçŽĐæŨĠāzūāĀČèĀČèŽŚāy

```
# mymodule.py
class A:
    def spam(self):
        print('A.spam')

class B(A):
    def bar(self):
        print('B.bar')
```

āĀĠèō;ä;āæČšmymodule.pyāŁĒäyžäyđ'äyŁæŨĠāzūii;ŃæfRāyŁāōŽāzŁ'çŽĐäyĀäyŁçšzāĀČèçAāĀŽāŁrē
 èŁŽèŁŽäyŁçŽōā;ŦäyŃiijŃāŁŽāžžæäyŃæŨĠāzūii;Ž

```
mymodule/
__init__.py
a.py
b.py
```

āIJlā.pyæŨĠāzūäy■æRŠāĒēäžæyŃāžççāAii;Ž

```
# a.py
class A:
    def spam(self):
        print('A.spam')
```

āIJĭb.pyæŨĠāzūäy■æRŠāĖĕäzēäyŊāzčçăAĭijŽ

```
# b.py
from .a import A
class B(A):
    def bar(self):
        print('B.bar')
```

æIJĀāŖŎĭijŊāIJĭ__init__.py äy■ĭijŊāŕĖ2äyĭæŨĠāzūçšŸāŖĹāIJĭäyĀètŭĭijŽ

```
# __init__.py
from .a import A
from .b import B
```

āĖĈædIJæŊĹçĖĖĖZāzZæ■ēēld'ĭijŊæL'ĀāzġçŤšçŽĎāŊĖMyModuleārĖäĭIJäyžäyĀäyĭā■ŤäyĀçŽĎéĀz

```
>>> import mymodule
>>> a = mymodule.A()
>>> a.spam()
A.spam
>>> b = mymodule.B()
>>> b.bar()
B.bar
>>>
```

ēōlēōž

āIJĭēZāyĭçnāēĹĈäy■çŽĎäyžēēAēŨōēēŸæŸŕäyĀäyĭēōĭēōāēŨōēēŸĭijŊäy■çōāāĭāæŸŕāŖēāyŊæIJçŤĭā

```
from mymodule.a import A
from mymodule.b import B
...
```

ēĖZæāüēĈĭāüēäĭIJĭijŊäĭĖēĖZēōĭçŤĭāĹūāL'ġāŖŨæZĭād'ŽçŽĎèt'šæŊĖĭijŊçŤĭāĹūēēAçšēēAšäy■āŖŊ

```
from mymodule import A, B
```

ārzārŖŎēĀĖēĀŊēĭĀĭijŊēōĭ'mymoduleæĹŖäyžäyĀäyĭād'ġçŽĎæžŖæŨĠāzūæŸŕæIJĀäyŸēġAçŽĎāĀĈāĭĭ
ēĖZæāüāĀZçŽĎāĖšçŤōæŸŕāĹZāzžäyĀäyĭāŊĖçŽōāĭŤĭijŊäĭçŤĭ__init__.py
æŨĠāzūāĭēārĖæŕŖēĈĭāĹēçšŸāŖĹāIJĭäyĀètŭāĀĈ

āĭšäyĀäyĭāĭāĭŨēēĉāĹēāĹ'sĭijŊäĭāēIJāēēAçL'zāĹnæšĭæĎŖāžd'āŖĹāĭĭŤçŤĭçŽĎæŨĠāzūāŖ■āĀĈäyĭäy
from .a import A æĭēēŎūāŖŨāĀĈ

æŤŕ'äyĭçnāēĹĈēĈĭäĭçŤĭāŊĖçŽĎçŽyāržārĭjāĖēæĭēēAġāĖ■ārĖēāüāšĈæĭāāĭŨāŖ■çāñçijŨçăAġāĹŕæžŖäz

āĲĲāyžēŁŻāyĀçñāēŁĈçŽĎāzūāiĲyĲiĲŅāŕĒāzŅçz■āzūēŁšāŕiĲāĔēāĀĈāçĈāŽĲæL'Āçd'žĲiĲŅ__init__.pyæŪēçĀāĀŽāĴŕēŁŻāyĀçĈžĲiĲŅ__init__.pyæĲĴçžĒāĲčŽĎāŔŲāŅŪĲiĲŽ

```
# __init__.py
def A():
    from .a import A
    return A()

def B():
    from .b import B
    return B()
```

āĲĴēŁŻāyĲçĴĴæĲĲāy■ĲiĲŅçszĀāŠŅçszĬēçñæŽŁæ■cāyžāĲĲçññāyĀæñæēōŁēŪōæŪūāŁæĲĲ;æL'ĀēĲĲĀçŽĲāĲŅāçĲiĲŽ

```
>>> import mymodule
>>> a = mymodule.A()
>>> a.spam()
A.spam
>>>
```

āzūēŁšāŁæĲĲçŽĎāyžēçĀçĲiĲçĈzæŲŕçžgæL'ŁāŠŅçszāđŅæçĀæšēāŔŕēĈĲāiĲŽāy■æŪ■āĀĈāĲāāŔŕēĈĲāiĲŽ

```
if isinstance(x, mymodule.A): # Error
...

if isinstance(x, mymodule.a.A): # Ok
...
```

āzūēŁšāŁæĲĲçŽĎçĲĲšāōđāĲŅā■Ŕ, ēğĀæāĠāĠĒāžš multiprocessing/__init__.py çŽĎæžŔçāĀ.

12.5 10.5 āĴĲçĴĴāŚĲāŔ■çĴžēŪŕāŕiĲāĔēçŽōāĲĲāĴæĲççŽĎāžççāĀēŪōēçŲ

āĲāāŔŕēĈĲæĲĴĴāđ'ğēĠŔçŽĎāžççāĀĲiĲŅçŦsāy■āŔŅçŽĎāžžæĴēāĴæŦçāĲŕççzt'æŁd'āĀĈæŕŔāyĴēĈĴāĴæ

ēğĈāĒşæŪzæāĴ

āžŌæĲĲñēŦĴāyĴēōšĲiĲŅāĲēçĀāōŽāzĴĴāyĀāyĴēāūçžgPythonāŅĔĲiĲŅāĲĲāyžāyĀāyĴāđ'ğēŽĒāŔĴāĴĒāĲāĲçĲāĲāĲĲçžšāyĀāy■āŔŅçŽĎçŽōāĲĲēĠŅçžšāyĀçŽŲāŔŅçŽĎāŚĲāŔ■çĴžēŪŕĲiĲŅāĲæŲŕēçĀāĴāāŌžçŦĴāĴēārĴ

```
foo-package/
  spam/
    blah.py
```

(continues on next page)

(continued from previous page)

```
File "<stdin>", line 1, in <module>
AttributeError: 'module' object has no attribute '__file__'
>>> spam
<module 'spam' (namespace)>
>>>
```

æŽt'ād'ŽčŽĐāŇĚāŚ;āŘ■čl'žéŮt'āŁæAřāŘřäžæšēcIJŇ PEP 420.

12.6 10.6 éĜ■æŮřāŁæ;jaēlaaiŮ

éŮóécŸ

äjaæČšéĜ■æŮřāŁæ;jaŭščžŘāŁæ;jaŽĐælaaiŮrijŇāŽāāyžäjaāržāĚŮæžŘčāAēŁZēāŇāžEāŁōæŤžāĂĆ

èġčāEşæŮzæaŁ

äjaŁčŤlīmp.reload()æĬééĜ■æŮřāŁæ;jaĚĹāŁ'■āŁæ;jaŽĐælaaiŮāĂČäy;äyĹä;Ňā■ŘrijŽ

```
>>> import spam
>>> import imp
>>> imp.reload(spam)
<module 'spam' from './spam.py'>
>>>
```

èőĹéőž

éĜ■æŮřāŁæ;jaēlaaiŮāIJĹaijĀāŖŚāŚŇērČērŤèŁĜčĹŇäy■āyŷāyŷā;ŁæIJL'čŤĹāĂČä;EāIJĹčŤšāžġčŮřāćČā
reload()æŞēéŽĐ'āžEælaaiŮāžŤāsČā■ŮāĚyčŽĐāĚĚāōžrijŇāžŭéĂŽèŁĜéĜ■æŮřāŁæġèāŇāēlaaiŮčŽĐæžŘ
ärjačōāēČā■d'rijŇreload()æşæIJL'æŽt'æŮřāČŖāĂĹfrom module import
nameāĀĬèŁZæāŭäjaŁčŤlīimportēr■āŘēārijāĚēcŽĐāōŽāžŁ'āĂČäy;äyĹä;Ňā■ŘrijŽ

```
# spam.py
def bar():
    print('bar')

def grok():
    print('grok')
```

çŮřāIJĹāŘřāŁāžd'āžŚāijŘāijŽērĹijŽ

```
>>> import spam
>>> from spam import grok
>>> spam.bar()
```

(continues on next page)

(continued from previous page)

```
bar
>>> grok()
grok
>>>
```

äy■éÄÄGŽPythonä£ðæTžspam.pyçŽDæžRçäAñijNärEgrok()äG;æTæTžæLŘè£ŽæäüñijŽ

```
def grok():
    print('New grok')
```

çÖřaIJlāZđāLřāzd'āžŠāijRāijZèrlñijNéG■æŮřāLæ; ;ælaaiUñijNärIerTäyNè£ŽäyłaóðelNñijŽ

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

āIJlè£Žäyła;Nā■Räy■ñijNā;äçIJNāLřæIJL'2äyłçL'LæIJñçŽDgrok()äG;æTřècñāLæ; ;āÄĆéÄŽäyÿæIèèrt'
āŽāæ■d'ñijNāIJlçTšāžgçÖřāçCäy■āRřèÇ;éIJÄèçAéA£āĚ■éG■æŮřāLæ; ;ælaaiUāÄĆāIJlāzd'āžŠçÖřāçC

12.7 10.7 è£RèaÑçŽōā;TæLŮāŎÑçijl'æŮGäzú

éŮóécŸ

æCíæIJL'äyÄäyłaüßæLŘéT£äyžāNĚāRñād'ŽäyłæŮGäzúçŽDāžTçTlñijNāóČāüßè£IJäy■āE■æŸräyÄäyłç

èğçāEşæŮzæaŁ

āçCædIJā;äçŽDāžTçTlçlNāžRāüščzRæIJL'ād'ŽäyłæŮGäzúñijNā;āāRřāžèæLŁā;äçŽDāžTçTlçlNāžRæTç
äyçäyłä;Nā■RñijNā;āāRřāžèāČRè£ŽæäüāLŽāžççŽōā;TñijŽ

```
myapplication/
    spam.py
    bar.py
    grok.py
    __main__.py
```

āçCædIJ__main__.pyā■ŸāIJlñijNā;āāRřāžèçōÄā■TāIJřāIJléaüçžgççŽōā;Tè£RèaÑPythonèğçèGŁāŽlñijŽ


```
bash % python3 myapplication
```

èġċéĠŁăŻłăŕEăL'ġèqÑ__main__.pyăŰĠăzŭă;IJăyžăyžçłŊăžŔăĂĈ

ăĕĆăđIJă;ăăŕEă;ăċŽĐăžċăĂăL'ŞăŊĖăĹŔzipăŰĠăzŭijŊăŁŽġăĹĂăIJŕăŔŊăăŭăžŞăĂĈĉŦłijŊăyžă

```
bash % ls
spam.py bar.py grok.py __main__.py
bash % zip -r myapp.zip *.py
bash % python3 myapp.zip
... output from __main__.py ...
```

èőléőž

ăĹŽăžžăyĂăyłĉŽăŕŦăĹŰzipăŰĠăzŭăžŭăŭăžăĹă__main__.pyăŰĠăzŭăĹăŕEăyĂăyłăŽŦ'ăđ'ġĉŽĐPyth

ĉŦŝăžŎĉŽăŕŦăŦăŦŊzipăŰĠăzŭăŷŎăăĉăyăŷăŰĠăzŭăIJL'ăyĂĉĈăžăyăăŔŊijŊă;ăăŕŕăĈ;èŁŸăIJĂăĕĂăĉđă

```
#!/usr/bin/env python3 /usr/local/bin/myapp.zip
```

12.8 10.8 èŕžăŔŰă;ăăžŎăŊĖăyăĉŽĐăŦŕăăŎăŰĠăzŭ

éŰőéĉŸ

ă;ăĉŽĐăŊĖăyăăŊĖăŔŊăžĉăĂăIJĂăĕĂăŎžèŕžăŔŰĉŽĐăŦŕăăŎăŰĠăzŭăĂĈă;ăăIJĂăĕĂăŕ;ăăŕŕăĈ;ăIJŕĉŦ

èġĉăEşăŰžăăĹ

ăĂĠèőĹă;ăĉŽĐăŊĖăyăăĉŽĐăŰĠăzŭĉžĐĉžĠăŔăĕĈăyŊijŽ

```
mypackage/
__init__.py
somedata.dat
spam.py
```

ĉŎŕăIJłăĂĠèőĹspam.pyăŰĠăzŭăIJĂăĕĂăŕžăŔŰsomedata.dataăŰĠăzŭăyăăĉŽĐăĖăăŏžăĂĈă;ăăŕŕăžăĉŦłă

```
# spam.py
import pkgutil
data = pkgutil.get_data(__package__, 'somedata.dat')
```

ĉŦŝăăđ'ăžġĉŦşĉŽĐăŔŸăĠŔăŸŕăŊĖăŔŭăŕăĕŕăŰĠăzŭĉžĐăŎşăġŊăĖăăŏžăŽĐăăŰăĹĈăăŰĉŕăyşăĂĈ

èóìéőž

èĕAĕrżăŔŮæŦŕæ■ōæŮĜăzŭiijŊă;ăăŔŕĕĈ;ăijŽăĂĭăŔŖăžŎĉijŮăĖŽă;ĕĈŦĭăĖĖĈ;ŏĈŽĐĭ/
OăĽŖĕĈ;ĈŽĐăžĈĈăĀiijŊăĕCopen()ăĂĈă;ĖæŸŕĕĖŽĈĝ■æŮŭæŖŦăžŖæĭĬĽăŸĂăžŽĕŮŏĕĈŸăĂĈ

ĕĕŮăĖĽiijŊăŸĂăŸĭăŊĖăŕžĕĝĕĖĜĽăŽĭĈŽĐă;ŖăĽ■ăŭĕă;ĭĬĈŽŏă;ŦăĜăăžŎæŖæĭĬĽăŖŎĝăĽŭăĭĈăĂĈăŽăæŖ
ĈŋăăžŊiijŊăŊĖĕĂŽăŸŸăŏĽ'ĕĈĖă;ĭĬăŸž.zipăĽŮ.eggăŮĜăzŭiijŊĕĖŽăžŽăŮĜăzŭăžŭăŸ■ăĈŖăĭĬăŮĜăžŭ
pkgutil.get_data()ăĜ;æŦŕæŸŕăŸĂăŸĭĕŕżăŔŮæŦŕæ■ōæŮĜăzŭĈŽĐĕŋŸĈžĝăŭĕăĖŮiijŊăŸ■ĈŦĭĈŏăăŊĖæŸŕă
get_data()ĈŽĐĈŋăŸĂăŸĭăŖĈăŦŕæŸŕăŊĖăŖăŊĖăŖ■ĈŽĐă■ŮĈŋăŸŖăĂĈă;ăăŔŕăžĕĈŦ'æŖŎĕă;ĕĈŦĭăŊĖ

12.9 10.9 ăŖĖæŮĜăzŭăđ'žăĽăăĖĖăĽŖsys.path

ĕŮŏĕĈŸ

ă;ăæŮăæŖŦăŕijăĖĖă;ăĈŽĐPythonăžĈĈăĀăŽăăŸžăŏĈăĽ'ĂăĭĬĈŽĐĈŽŏă;ŦăŸ■ăĭĬsys.pathĕŊăĂĈă;ăæĈŖ

ĕĝĈăĖŖæŮŭæăĽ

æĭĬĽăŸđ'ĝĝ■ăŸŸĈŦĭĈŽĐăŮăijŖăŖĖæŮŖĈŽŏă;ŦăŮăžăĽăăĽŖsys.pathăĂĈĈŋăŸĂĈĝ■iijŊă;ăăŔŕăžĕă;ĕĈŦĭăŊĖ

```
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', ...]
>>>
```

ăĭĬĽĕĜăŏŽăžĽ'ăžŦĈŦĭĈĭŊăžŖăŸ■iijŊĕĖŽăăŮĈŽĐĈŖăĈăŖŸĕĜŖăŖăĭĬĽĭŊăžŖăŖăĽăŮŭĕŏĈ;ĕŏăĽŮ
ĈŋăăžŊĈĝ■æŮŭæŖŦăŸŕăĽŽăžăŸĂăŸĭ.pthăŮĜăzŭiijŊăŖĖĈŽŏă;ŦăĽŮăŸĭăĜăăĭĕiijŊăĈŖĕĖŽăăŮiijŽ

```
# myapplication.pth
/some/dir
/other/dir
```

ĕĖŽăŸĭ.pthăŮĜăzŭĕĭĬăĕĖAæŦĭăĭĬăŖŖăŸĭPythonĈŽĐsite-
packagesĈŽŏă;ŦiijŊĕĂŽăŸŸă;■ăžŖŮŮŕ/local/lib/python3.3/site-packages æĽŮĕĂĖ ~/local-
cal/lib/python3.3/sitepackagesăĂĈă;ŖĕĝĕĖĜĽăŽĭăŖŕăĽăŮŮiijŊ.pthăŮĜăzŭĕŊăĽŮăŸĭăĜăăĭĕĈŽĐă■ŸăĭĬ

èóìéőž

ăŖŦĕŦŮĕŦ'žăĽŽăĭŖăĽ'ĭæŮĜăzŭiijŊă;ăăŔŕĕĈ;ăijŽăĂĭăŔŖăžŎăĖŽăŸĂăŸĭăžĈĈăĀăĽ'ŊăĽĭĕŖĈĕĽĈsys.pat

```
import sys
sys.path.insert(0, '/some/dir')
sys.path.insert(0, '/other/dir')
```

èŽ;çĎűèŁŻèĈ;ăĂĬăűěă;ĬĴăĂĬĭĭĴŃă;ĒăĚŸřăĬĴăőđèűăŸ■ăđĂăŸžèĎĒăĭſĭĭĴŃăžŤăř;éĠŖéĂĚăĚ■ă;ĚçŤĴăĂ

```
import sys
from os.path import abspath, join, dirname
sys.path.insert(0, join(abspath(dirname(__file__)), 'src'))
```

èŁŻăŖĒsrcçŻőă;ŤăűzăĴăăĴŖpathéĠŃĭĭĴŃăŤŃăĒĴġèăŃăĒŤăĒăĒēă■ēēĴđ'çŽĎăžççăĂăĬĴăŖŃăŸĂăŸĴçŻőă;ŤăŤĂĈăĚĈăđĬă;ăăĒŤŃăĴăőĴ'èĈ packagesçŻőă;ŤăŤĂĈèŽ;çĎűçŤĴăžŌéĚ■ç;őpathçŽĎ.pthăŰĠăžűăĚĒăžăĒŤĴç;őăĬĴſite-packageséĠŃĭĭĴŃă;ĒăőĈéĚ■ç;őçŽĎèűřăĴĎăŖřăžăăŸŖçşçşçşăŸĴăžžă;Ťă;ăăŸŃăĬĴçŽĎçŻőă;ŤăŤĂĈăŽăă■đ'

12.10 10.10 éĂŽèŁĠă■ŰçņęăŸşăŖ■ăŕĭĵăĒēăĴăĴŰ

éŰőéćŸ

ăĵăăĈşăŕĭĵăĒēăŸĂăŸĴăĴăĴŰĭĭĴŃă;ĒăĚŸŖăĴăĴŰçŽĎăŖ■ă■ŰăĬĴă■ŰçņęăŸşéĠŃăĂĈă;ăăĈşăŕžă■ŰçņęăŸ

èğĉăĒşăĒŰžăăĴĴ

ăĵĚçŤĴimportlib.import_module()ăĠ;ăŤŖăĴăĒĴŤŃăĴăŖĭĵăĒēăŖ■ă■ŰăŸžă■ŰçņęăŸşçžăĠžçŽĎăŸĂăŸĴăĴ

```
>>> 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_moduleăŖĴăŸŖçőĂă■ŤăĬĴŖăĴġèăŃăŤŃăĬimportçŽŸăŖŃçŽĎă■ēēĴđ'ĭĭĴŃă;ĒăĚŸŖèĚŤăŽđçŤşăĴŖçăĚĈăđĬă;ăă■ĉăĬĴă;ĚçŤĴçŽĎăŤĒĭĭĴŃimport_module()ăžşăŖŖçŤĴăžŌçŽŸăŕžăŕĭĵăĒēăĂĈă;ĒăŸŖĭĭĴŃă;ăé

```
import importlib
# Same as 'from . import b'
b = importlib.import_module('.b', __package__)
```

èőĴèőž

ăĵĚçŤĴimport_module()ăĴŤŃăĴăŖĭĵăĒēăĴăĴŰçŽĎéŰőéćŸéĂŽăŸŸăĠžçŌŖăĬĴăžăăşŖçğ■ăŰžăĭŖçĭĴŰă

āIJāŨgçŽDāzčçāAīijNāIJL'æŨūā;āaijŽçIJNāLrçTlāžŌārijāĒēçŽDāEĒāzāG;æTŕ__import__()āĀĆāŕ;ġ
éĀŽāyŷæŽt'āōzæŸŞā;ġçTlāĀĆ

èĠlāōŽāzL'ārijāĒēēġĠŕçŽDēnŸçžgāōdā;NēgA10.11āŕRēLĆ

12.11 10.11 éĀŽèĒĠéŚl'ā■RēĒIJçlNāLāē;ġæġāġlŨ

éŨōécŸ

ā;āæČşèĠlāōŽāzL'PythonçŽDimportēŕ■āRēīijNā;ġā; ŨāōČèČ;āzŌēĒIJçlNāIJžāZlāyLēġéĀRæŸŌçŽDā

èġçāEşæŨzæāġL

éçŨāĒLèçAæRŔāGžæġçŽDæŸŕāōL'āĒlēŨōécŸāĀĆæIJnēLĆèōlēōžçŽDæĀġæČşāçĀçđIJæşāæIJL'āyĀ
āzşāŕsæŸŕēŕt'īijNāLŖāzñçŽDāyžèçAçŽōçŽDæŸŕæūsāĒēāĒLēæđŔPythonçŽDimportēŕ■āRēæIJžāLūāĀĆ
āçĀçđIJā;āçŔEēġçāžEæIJnēLĆāEĒēĀŕŌŖçŔEīijNā;āāŕsēČ;āđ'şāyžāĒūāzŨāzā;TçŽōçŽDēĀNēĠlāōŽāzL'i
æIJL'āžEēĒçāžŽīijNēōŕ'æLŖāzñçžġçç■āŔŖāL'■ēŕāĀĆ

æIJnēLĆæāyāŕČæŸŕēō;ēōāārijāĒēēŕ■āRēçŽDæL'ŕāsŤāLşèČ;āĀĆæIJL'ā;Ēāđ'Žçġ■æŨzæşŤāŔŕāzēāĀŽ
āy■ēĒĠāyžāžEæijŤçđ'žçŽDæŨzā;ġīijNāLŖāzñāijĀāġNāĒLæđDēĀāyNēġçēĒZāyſPythonāzççāAçzşæđDīijŽ

```
testcode/  
  spam.py  
  fib.py  
  grok/  
    __init__.py  
    blah.py
```

ēĒZāžZæŨĠāzūçŽDāEĒāōzāzūāy■ēĠ■ēçAīijNāy■ēĒĠæLŖāzñāIJlæŕŔāyſæŨĠāzūāy■æŤ;āĒēāžEāŕŖēĠ
ēĒZæāūā;āāŔŕāzēæŕNēŕŤāōČāzñāzūæşççIJNā;ŞāōČāzñēçnārijāĒēæŨūçŽDē;ŞāĠžāĀĆā;NāçCīijŽ

```
# spam.py  
print("I'm spam")  
  
def hello(name):  
    print('Hello %s' % name)  
  
# fib.py  
print("I'm fib")  
  
def fib(n):  
    if n < 2:  
        return 1  
    else:  
        return fib(n-1) + fib(n-2)  
  
# grok/__init__.py  
print("I'm grok.__init__")
```

(continues on next page)

(continued from previous page)

```
# grok/blah.py
print("I'm grok.blah")
```

æfZéGŃçŽDçŽDæYřăĖAeöyefZăžZæŮGăžŭă;IJăyžăġăăġŮećnèfIJġġŃeöfēŮăăĂĆ
ăžšèöyæIJĂçŏĂă■TçŽDæŮžăijRăřsæYřăřĖăŏCăžăăRŚăyCăĹrăyĂăyġwebæIJ■ăġăăŽġăyĹēĲăĂĆăIJġtestcode

```
bash % cd testcode
bash % python3 -m http.server 15000
Serving HTTP on 0.0.0.0 port 15000 ...
```

æIJ■ăġăăŽġēfRĖăNĖġăġăăRŎăĖ■ăRřăĹăyĂăyġă■TçNŃçŽDPythonĖğćēĢăăŽġăĂĆ
çăŏăĲă;ăăRřăžăă;ĲçŤĲ ĺllib eöfēŮăăĹrēfIJġġŃæŮGăžŭăĂĆăġNăĖĆrijŽ

```
>>> 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)

>>>
```

ăžŎēfZăyġæIJ■ăġăăŽġăĲăē;æžRăžççăĂæYřăŎăyNăġăæIJŃēĲçŽDăžçăĂăĂĆ
ăyžăĖæŽăžçæĹNăĲçŽDēĂžēĢ ĺlopen() æġăăŤŭēŽĖæžRæŮGăžŭġijŃ
æĲSăžŃēĂžēfĢēĢăŏŽăžĲimportĖ■ăRĖăġăăIJăăRŎăRřēĢăĲăĲăyŏăĲSăžăăĂžăĲăĂĆ

ăĲăē;ēfIJġġŃăġăăŮçŽDçŃăyĂçğ■ăŮžăşTæYřăĲZăžžăyĂăyġæYġăijRçŽDăĲăē;ăĢæTřăġăăŎăĲăĲ

```
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
```

ēfZăyġăĢæTřăijŽăyNĖ;æžRăžççăĂġijŃăžŭă;ĲçŤĲ ĺcompile()
ăřĖăŮġijŮĖřSăĲăyĂăyġăžççăĂăřžēşăy■ijŃ çDŭăRŎăIJăyĂăyġæŮřăĲZăžžçŽDăĲăăŮăřžēşçŽDă■ŮăĖy

```

>>> 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■čæĆä;äæL'ÄëğAīijŃârízăžŎçóĀā■ŦçŽĐæłaaİUëŁŻăyŁæŸřeaŃă;ŮéĂŽçŽĐăĂĆ
 äy■ēŁĞăŏĆăzŭæşæIJL'ăŦŃăĚăĹŕéĂŽăyŷçŽĐimportēŕ■ăŔăy■īijŃăēĆăđIJēēAæŦŕăŃAæŽt'énŸçžğçŽĐçz
 äyĂăyŁæŽt'éĚüçŽĐăĂŽæşŦæŸŕăŁŻăžžăyĂăyŁēĞăŏŽăžŁ'ăŕijăĚăăŽĹăĂĆçňňăyĂçğ■æŮzæşŦæŸŕăŁŻăž

```

# 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('.')[0] + '.py'

    def get_source(self, fullname):
        filename = self.get_filename(fullname)
        log.debug('loader: reading %r', filename)
        if filename in self._source_cache:
            log.debug('loader: cached %r', filename)
            return self._source_cache[filename]
        try:
            u = urlopen(filename)
            source = u.read().decode('utf-8')
            log.debug('loader: %r loaded', filename)
            self._source_cache[filename] = source

```

(continues on next page)

(continued from previous page)

```
        return source
    except (HTTPError, URLError) as e:
        log.debug('loader: %r failed. %s', filename, e)
        raise ImportError("Can't load %s" % filename)

    def is_package(self, fullname):
        return False

# Package loader for a URL
class UrlPackageLoader(UrlModuleLoader):
    def load_module(self, fullname):
        mod = super().load_module(fullname)
        mod.__path__ = [ self._baseurl ]
        mod.__package__ = fullname

    def get_filename(self, fullname):
        return self._baseurl + '/' + '__init__.py'

    def is_package(self, fullname):
        return True

# Utility functions for installing/uninstalling the loader
_installed_meta_cache = { }
def install_meta(address):
    if address not in _installed_meta_cache:
        finder = UrlMetaFinder(address)
        _installed_meta_cache[address] = finder
        sys.meta_path.append(finder)
        log.debug('%r installed on sys.meta_path', finder)

def remove_meta(address):
    if address in _installed_meta_cache:
        finder = _installed_meta_cache.pop(address)
        sys.meta_path.remove(finder)
        log.debug('%r removed from sys.meta_path', finder)
```

äyÑéÍcæYřäyÄäyłäzđ'āžŠaijŽerłiijŃæijTčđ'žāžEāęĆä;Tä;fcTłāL■éÍcçŽDāžččāAiiž

```
>>> # importing currently fails
>>> import fib
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
ImportError: No module named 'fib'
>>> # Load the importer and retry (it works)
>>> import urlimport
>>> urlimport.install_meta('http://localhost:15000')
>>> import fib
I'm fib
>>> import spam
```

(continues on next page)

(continued from previous page)

```
I'm spam
>>> import grok.blah
I'm grok.__init__
I'm grok.blah
>>> grok.blah.__file__
'http://localhost:15000/grok/blah.py'
>>>
```

[illegible]

```
# 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)

```
sys.path
```



```

        return fib(n-1) + fib(n-2)
>>>

```

èõlèõž

åIJlèrèçžEèõlèõžäzNåL■iijNæIJL'çCžèeAaijžèrCçŽDæYriijNPythonçŽDæIaaiUãAAaÑEåŠNårijaEëæIJ
 å■şä;ççžRèiNäyřarNçŽDPythonçIÑazRåSÝázşâ;LårSèC;çş;éÅŽåõCžñãĀC
 æLŠaIJlè£ŽéGNæÕlè■RäyÄazŽaAijçŽDåÕžèrççŽDæÚGæaçåŠNäzççs■iijNåNĒæNñ im-
 portlib module åŠN PEP 302. æÚGæaçåEĒåõžåIJlè£ŽéGNäy■aijŽècñéG■ad'■æRŘålriijNäy■è£GæLŠaIJlè£Ž

éçÚåĒLriijNæCædIJä;æCşålŽazžäyÄäyIæŮrçŽDæIaaiUåržèşaiijNä;ççTĪ imp.
 new_module() åG;æTřriijŽ

```

>>> import imp
>>> m = imp.new_module('spam')
>>> m
<module 'spam'>
>>> m.__name__
'spam'
>>>

```

æIaaiUåržèşæĀŽäyÿæIJL'äyÄazŽæIJşæIJZåşðæĀgriijNåNĒæNñ __file__
 iijLè£RèaÑæIaaiUåLæ;ç;èr■āRèçŽDæÚGazūāR■iijL' åŠN __package__ (åÑĒāR■)āĀC

åĒŮæñaiijNæIaaiUaijŽècñègçéGŁaZlçijŞa■YètuæIēāĀCæIaaiUçijŞa■YāRřazēāIJlā■ŮāĒy
 sys.modules äy■ècñæL;ålRřāĀC åŽäyÿæIJL'azEè£ŽäyIçijŞa■YæIJZålŮriijNēĀŽäyÿāRřazēāRçijŞa■YāŠ

```

>>> import sys
>>> import imp
>>> m = sys.modules.setdefault('spam', imp.new_module('spam'))
>>> m
<module 'spam'>
>>>

```

æCædIJçžŽåõŽæIaaiUåüşçžRā■YāIJlèCčázLårşaijŽçŽT æÕèèŮa;UåüşçžRècñålŽazžè£GçŽDæIaaiUř

```

>>> 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
>>>

```

çTşazŌålŽazžæIaaiUa;LçõĀa■TrijNā;LåõzæYşçijŮāEŽçõĀa■TāG;æTřæřTæçCññäyĀéČlāLEçŽD
 load_module() åG;æTřāĀC è£ŽäyIæŮzæaLçŽDäyÄäyIçijçCzæYřa;LéŽ;ad'DçRĒad'■æICæČĒāĒtæřT

äyžāẸāđ'ĐçŘĒäyÄäyġāÑĒijÑā;āēēAēĠ■āŪrāōđçŌrāŽōēĀŽimportēr■āRēçŽĐāžTāsCēĀžē;ŚiijLāēŦāēČā
 æL'gēāÑēČčāžŽæŪĠāzūiijÑēō;ç;ōēŭrā;Đç■L'rijL'āĀCēŁZāyġāđ'■āĪCæĀġārsæŸrāyžāžĀāžLæIJĀāē;çŽt'æŌ
 æL'fāsŦimportēr■āRēā;ŁçōĀā■ŦiijÑā;ĒæŸrāijŽæIJL'ā;Łāđ'ŽçġzāŁlæS■ā;IJāĀC
 æIJĀēñŸāsČāyŁiijÑārijaĒēæS■ā;IJēčñāyÄäyġā;■āžŌsys.meta_pathāLŪēāġāy■çŽĐāĀIJāĒČēŭrā;ĐāĀlæšēæ
 āēČāđIJā;āē;SāĠžāōČçŽĐāĀijrijÑāijŽçIJNāLŕāyÑēĪCēŁZæāūiijŽ

```

>>> from pprint import pprint
>>> pprint(sys.meta_path)
[<class '_frozen_importlib.BuiltinImporter'>,
<class '_frozen_importlib.FrozenImporter'>,
<class '_frozen_importlib.PathFinder'>]
>>>
  
```

ā;SæL'gēāÑāyÄäyġēr■āRēæŦāēČimport fib æŪūiijÑēġcéĠāŽġāijŽēA■āŌĒsys.mata_pathāy■çŽ
 ēŦççŦġāōČāžñçŽĐ find_module() æŪžæšŦāōŽā;■æ■ççāōçŽĐāġāġŪāŁāē;ġāŽġāĀC
 āŦŕāžēēĀŽēŁĠāōđēĪNāēĪçIJNçIJNrijŽ

```

>>> 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
>>>
  
```

æšġāĐŦçIJN find_module() æŪžæšŦæŸŕæĀŌæāūāIJlæŦŕāyÄäyġārijaĒēāršēcñēġēāŦSçŽĐāĀC
 ēŁZāyġāēŪžæšŦāy■çŽĐpathāŦCæŦŕçŽĐā;IJçŦġāŸŦāđ'ĐçŘĒāÑĒāĀC
 āđ'ŽāyġāÑĒēčñārijaĒēēiijÑārsæŸrāyÄäyġāŦŕāIJlāÑĒēçŽĐ _____path____
 āśđæĀġāy■æL';āŁŕçŽĐēŭrā;ĐāLŪēāġāĀC ēēAæL';āŁŕāÑĒēçŽĐā■ŦçžĐāžūāršēēAæčĀæšēēŁZāžŽēŭrā;ĐāĀ
 æŦāēČæšġāēĐŦŕāžāžŌ xml.etree āŠŦ xml.etree.ElementTree
 çŽĐēŭrā;ĐēĒ■ç;ōiijŽ

```

>>> import xml.etree.ElementTree
Looking for xml None
Looking for xml.etree ['/usr/local/lib/python3.3/xml']
Looking for xml.etree.ElementTree ['/usr/local/lib/python3.3/xml/
↳etree']
  
```

(continues on next page)

(continued from previous page)

```
Looking for warnings None
Looking for contextlib None
Looking for xml.etree.ElementPath ['/usr/local/lib/python3.3/xml/
↳ etree']
Looking for _elementtree None
Looking for copy None
Looking for org None
Looking for pyexpat None
Looking for ElementC14N None
>>>
```

ǎIǐ sys.meta_path äyŁæšæŁ;ǎZícŽDä;■;ǒǎ;ŁéG■ēAǐǐNǎřEǎǒČǎzŎēYšǎd't'çgǎǎŁréYšǎř;ǐǐN

```
>>> del sys.meta_path[0]
>>> sys.meta_path.append(Finder())
>>> import urllib.request
>>> import datetime
```

çÖrãIJlä; açIJNäy■ãLřäzzä; Tè; ŞãGžăžErijŃNãZăăyžãrjãjĖĖëèçnsys.meta_pathäy■çŽĐãĖŭăžŬăőđã; Şăđ'Đç
èŁŽæŬăăĀŽiijŃNã;ăăRłtæIJL'ãIJlãrjãjĖĖäy■ăŃŸãIJlăłãlŬçŽĐæŬăăĀŽăL'■èç; çIJŃãLřăőČècñèğęãRŚiijŽ

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

ā;āāzNāL'■āōL'ēcĒēfGāyĀāyļæ■TēŌūæIJčšēāļāīUčŽDæšēæL;āŽIijNēfZāyļæYř
 UrlMetaFinder ċszčŽDāĒšēTōāĀĆ äyĀāyļ UrlMetaFinder āōdā;NēcnāūzāŁāāŁř
 sys.meta_path čŽDæIJnār;iiijNā;IJāyžæIJĀāŘŌäyĀāyļæšēæL;āŽīæŪžæāŁāĀĆ
 āēCædIJēcnērūāēŚĆčŽDæļāāīUāŘ■äy■ēČ;āōZā;■riijNāřsaijŽēcnēfZāyļæšēæL;āŽīād'DčŘĒæŌŁāĀĆ
 ād'DčŘĒāNēČčŽDæŪūāĀŽēIJāēēAæšļæDRriijNāIJlpathāRČæTřäy■æŃGāōŽčŽDāĀijēIJāēēAēcnāčĀæšērij
 āēCædIJāy■æYřriijNērēā■RēāļāīUāfĒēāzā;ŚāsđāžŌāĒūāzŪāēšēæL;āŽīāžūēcnāf;čTēæŌŁāĀĆ

```

    ħřzāžŎăÑĚčŽĐăĔúāžŮăđ' ĐçŘĚăRřăIĬĲ                                UrlPackageLoader
çşzäy■ēcñæL'ğĂŁřăĂĆ      ēŁZăylçşzäy■aijŻărıjaĖēăÑĖăŘ■ıijÑĖĂÑăYřăŌzăŁăē;ı;ħřzāžŤçŽĐ
__init__.py                æŪĞăzũăĂĆ        āōCăżşaijŽēōç;ç;ôăłăăİŮçŽĐ        __path__
āsďăĀğriįÑĖfŁZăyĀæ■ēăŹŁéG■ēēAııjŅ āZăäyžăIĬăŁăē;ı;ăÑĚčŽĐă■ŘăłăăİŮăŮŮēŁZăylăĀıajıjŽēcńaijăçžZă
find_module()   ěrÇçŦíăĂĆ  aşzāžŌēűřăŹĐçŽĐărıjaĖēēŚĴă■ŘăYřēŁZăžZăĀıăČşçŽĐăyĀăylăĽĴăśŦııjŅ
æĻSăznėĈ;çşēēĀŞııjŅsys.path æYřăyĀăyİPythonăşşēĽĴ;ăłăăİŮçŽĐçZăă;ŦăĻŮēăłııjŅă;ŊăēĈııjŽ

```

```
>>> from pprint import pprint
>>> import sys
```

(continues on next page)

(continued from previous page)

```
>>> pprint(sys.path)
['',
 '/usr/local/lib/python3.3.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']
>>>
```

āĲĲ sys.path āy■čŽĎæŕRāyĀāyġāōđā;ŠēČ;āijŽēcñēĲāđ ŰčŽĎčzŠāōŽāĲŕāyĀāyġæšēæĲ;āŽĲāŕžēšāy
ā;āāŔŕāžēēĀŽēĲĠæšēčĲĲ sys.path_importer_cache
āŌžĲĲNāyNēfŽāžZæšēæĲ;āŽĲijŽ

```
>>> 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/python3.3.zip': None}
>>>
```

sys.path_importer_cache æŕŦ sys.path āijŽæŽŦāđ ġčČzĲijŦ
āŽāyžāōČāijŽāyžæĲĲĀĲĲēcñāĲāē;āžččāĲčŽĎčZōā;Ŧēōŕā;ŦāōČāžñčŽĎæšēæĲ;āŽĲāĲČ
ēĲŽāNĒæNŕāNĒčŽĎā■ŔčŽōā;ŦĲijNēfŽāžZēĀŽāyŕāĲĲ sys.path
āy■æŦŕāy■ā■ŦāĲĲŽĎāĲČ

ēēĲæĲġēāŦ import fib ĲijŦāijŽēāžāžŔæčĀæšē sys.path āy■čŽĎčZōā;ŦāĲČ
āŕžāžŌæŕRāyġčZōā;ŦĲijNāŔ■ġŕāĲĲfibāĲāĲijŽēcñāijāčžZčŽyāžŦčŽĎ sys.
path_importer_cache āy■čŽĎæšēæĲ;āŽĲāĲČ ēĲŽāyġāŔŕāžēēōŦā;āāĲŽāžžēĲġāūščŽĎæšēæĲ;āŽĲāžūā

```
>>> class Finder:
...     def find_loader(self, name):
...         print('Looking for', name)
...         return (None, [])
...
>>> import sys
>>> # Add a "debug" entry to the importer cache
>>> sys.path_importer_cache['debug'] = Finder()
>>> # Add a "debug" directory to sys.path
```

(continues on next page)

(continued from previous page)

```
>>> sys.path.insert(0, 'debug')
>>> import threading
Looking for threading
Looking for time
Looking for traceback
Looking for linecache
Looking for tokenize
Looking for token
>>>
```

aIjleŹeGñijNä;aaRräzeäyžaR■a■UaAIJdebugaAiaLZāzžayÄaylæŮrçŽDçijŠa■Yāođā;ŠažūarEāoČeo,
 sys.pathäyLçŽDçñnäYÄaylāĀ aIjleL'ÄæIJLæŌēäyNælēçŽDārjāĒēäy■ijNä;āaijŽçIJNāLŖā;āçŽDæšē.
 äy■ēfGrijNçTšazŌāoČēfTāžđ (None, [])ijNēCčazLādDçRĒēfŽçlNāijŽçžgčz■ādDçRĒäyNäyÄaylāođā;Šā.

```
sys.path_importer_cache ċŻĎ;ŁĉŤłēćñäYĂăyİā■YāĆıİJl sys.path_hooks  
äy■ĉZĐăĜ;æŦrăLŮeăİaeŌğăLũăĂC ēŦTērŦăyNéİcçZĐă;Ŋă■RtiJŊăőCăijŽăyÉēZđ'cijŞă■YăzűczŻ  
sys.path_hooks æůzăLăăYĂăylăŪřčZĐèúră;ĐăčĂăşăĞ;æŦŕ
```

```
>>> 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■çåÇä;äæL'ÄëgAüjNcheck_path() äG;æTřecñæfRäyI sys.path
äy■çZDäöda;ŞërÇçTİaÄÇ äy■æüüjNçTşazÖæŁZäGżazE ImportError äijČäyyüjN
âTëëÇ;äy■äijZârŞçTşazEüjNŁazĚäZĚârĚæçÄæŞë;ñçgżăŁrsys.path_hooksçZDäyNäyÄäyİäG;æTřüjL'äÄÇ
çŞëéASzäZæÄÖæäüsys.pathæYřæÄÖæäüëcñad'DçRĚçZDřijNä;äârşëÇ;ædDäzzäyÄäyİëGİäöZăZL'ëürä;

```
>>> def check_url(path):
...     if path.startswith('http://'):
...         return Finder()
...     else:
...         raise ImportError()
```

(continues on next page)

(continued from previous page)

```
>>> sys.path.append('http://localhost:15000')
>>> sys.path_hooks[0] = check_url
>>> import fib
Looking for fib # Finder output!
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ImportError: No module named 'fib'

>>> # Notice installation of Finder in sys.path_importer_cache
>>> sys.path_importer_cache['http://localhost:15000']
<__main__.Finder object at 0x10064c850>
>>>
```

[illegible]

ašžāžŌēũrā; ĎārijaĖĖčŽĎāNĖād' ĎčRĖčĭā; ōæIJL'čĆžād'■æIĆijNāzūāyTēuš
 find_loader() æŪzæšTēfTāZĎāAijæIJL'āĖšāĀĆ āfžāžŌčŏĀ■TælaāIŪijNfind_loader()
 èĤTāZĎāyĀāyĭāĖĖčžĎ(loader, None)iiijN āĖūāy■čŽĎload-
 eræYřāyĀāyĭčTĭāžŌārijaĖĖæĭāāIŪčŽĎāĤæ;ĭāZĭāŏđā;NāĀĆ

[illegible]

find_loader() æfYðeAeĈj;äd'ĐçŘEäyÄäyłaŚ;āŘ■çl'žeŮt'āNĚāĀĈ
äyÄäyłaŚ;āŘ■çl'žeŮt'āNĚäy■æIJL'äyÄäyłaŘLæŞTçZĐāNĚĈZōā;ŤāŘ■iijNā;EæYřäy■ā■YāIJl__init__.pyæŮ
èfZæuüçZĐèrIiijNfind_loader() āfĒÉæzèfTāZđäyÄäyłaĒĈçzĐ(None, path)iijN
pathæYřäyÄäyłçZōā;ŤāLŮèāIiijNçTśāōCælēæđDāzžāNĚĈZĐāōZāZL'æIJL'__init__.pyæŮGāžúçZĐ__path__
āržāžŌèfZçg■æĈĒāEŤiijNārijaĒēæIJžāLŮāijZçzgçz■āL'■ēāNāŌzæĈĀæşesys.pathäy■çZĐçZōā;ŤāĀĈ
æçCæđIJæL';āLřāžEāŚ;āŘ■çl'žeŮt'āNĚiijNæL'ĀæIJL'çZĐçzŞæđIJēuřā;ĐēcnaĽāāLřāyĀēŭælēæđDāzžæIJA
āĒşāžŌāŚ;āŘ■çl'žeŮt'āNĚĈZĐæZt'äd'ZāfæAæřēuāRĈCeĀĈ10.5ārŘēŁCāĀĈ

æL'ÄæIJL'çŽĐaŃĚĈ:āŃĚaŃnāžEäyÄäyŃaĚĚĈlěura;Đěo;ç;oiijŃaŃRäzěaIJl__path__āsđæÄğäy■çIJŃ.

```
>>> import xml.etree.ElementTree
>>> xml.__path__
['/usr/local/lib/python3.3/xml']
>>> xml.etree.__path__
['/usr/local/lib/python3.3/xml/etree']
>>>
```

```

    äzNâL■æRŘâĹĹriijN__path__čŽĎěŔç;ŏæYřéĂžěfĜ                                     find_loader()
    æŮzæšTěĽTâZĎaĀijæŎġâĹŭčŽĎâĂĆ äy■ěĽĜiiijN__path__æŎěäyNæiěäžšěćnsys.path_hooksäy■čŽĎâĜ;æT

```

āZāæ■d'tijNā;EāNĒçŽDā■RçzDāzūēcāLāē;āRŌrijNā;■āžŌ__path__āy■çŽDāōdā;ŠāijŽēcā
 handle_url() āG;æTřæčĀæšēāĀĆ ēfZāijŽārijēGt'æŮřçŽD UrlPathFinder
 āōdā;NēcāLāZāzžāzūāyTēcāLāāĒēāLř sys.path_importer_cache āy■āĀĆ

ēfYæIJL'āylēŽ;çCzārsæYř handle_url() āG;æTřāzēāRĽāōČēu\$āĒēČlā;fçTlçŽD
 _get_links() āG;æTřāzNēŮt'çŽDāzd'āzŠāĀĆ āēČādIJā;āçŽDæšēæL;āZlāōdçŌřēIJāēēAā;fçTlāLřāĒū
 æIJL'āRřēČ;ēfZāzŽāēlāāIŮāijZāIJāēšēæL;āZlāē\$■ā;IJāIJšēŮt'ēfZēāNāZt'ād'ŽçŽDārijāĒēāĀĆ
 āōČāRřāzēārijēGt' handle_url() āŠNāĒūāzŮæšēæL;āZlēČlāLĒēZūāĒēāyĀçg■ēĀŠā;Šā;łçŌřçLūæĀĀā
 āyžāzĒēgčēGLēfZçg■āRřēČ;æĀgrijNāōdçŌřāy■æIJL'āyĀāylēcāLāZāzžçŽDæšēæL;āZlçijŠā■YijLærRāyĀ
 āōČāRřāzēēAēāĒ■āLāZāzžēG■ād'■æšēæL;āZlçŽDēŮōēčYāĀĆ
 āRēād'ŮrijNāyNēlççŽDāzčçāAçL'GāōjāRřāzēçāōāfĀæšēæL;āZlāy■āijZāIJlāLlāgNāNŮē\$;æŌēēZEāRĽçŽD

```
# Check link cache
if self._links is None:
    self._links = [] # See discussion
    self._links = _get_links(self._baseurl)
```

æIJĀāRŌrijNāšēæL;āZlçŽD invalidate_caches()
 æŮzæsTæYřāyĀāyĽāūēāĒūæŮzæsTijNçTlāēāyĒçRĒāĒēČlçijŠā■YāĀĆ
 ēfZāyĽæŮzæsTāE■çTlāēLūērČçTl importlib.invalidate_caches()
 çŽDæŮūāĀZēcāēgēāRŠāĀĆ āēČādIJā;āæČšēōl'URLārijāĒēēĀĒēG■æŮřēržāRŮē\$;æŌēāLŮēāłçŽDērĽāRřā

āržærTāyNāyd'çg■æŮzæāLijJLāfōæTžsys.meta_pathæLŮā;fçTlāyĀāyĽēūrā;DēŠl'ā■RijL'āĀĆ
 ā;fçTlsys.meta_pathçŽDārijāĒēēĀĒāRřāzēāNĽçĒgēGĽāūšçŽDēIJāēēAēGłçTšād'DçRĒēāāIŮāĀĆ
 ā;NāēČrijNāōČāzNāRřāzēāzŌæTřæ■ōāžŠāy■ārijāĒēæLŮāzēāy■āRñāzŌāyĀēLñāēāāIŮ/āNĒād'DçRĒæŮzāi
 ēfZçg■ēGłçTšāRñāūāēDŘāŠçlĀārijāĒēēĀĒēIJāēēAēGĽāūšēfZēāNāĒēēČlçŽDāyĀāzZçōāçRĒāĀĆ
 āRēād'ŮrijNāšžāzŌēūrā;DçŽDēŠl'ā■RāRĽæYřēĀĆçTlāzŌāržsys.pathçŽDād'DçRĒāĀĆ
 ēĀZēfGēfZçg■æL'āšTāLāē;çŽDāēāIŮēūšæZōēĀZæŮzāijRāLāē;çŽDçL'zæĀgæYřāyĀæāūçŽDāĀĆ

āēČādIJāLřçŌrāIJlāyžæ■cā;āēfYæYřāy■æYřā;LæYŌçZ;ijNēČčāzLāRřāzēēĀZēfGācdāLāāyĀāzZæŮ

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

æIJĀāŔŌījNāzžēōōä;æŁśĆzæŮŭéŮt'çIJNçIJN PEP 302 äžěāŔŁim-
portlibçŽDæŮĜæąčāĀĆ

12.12 10.12 árijaĒěælaaiŮçŽDāŔNæŮŮäŁōæŤzælaaiŮ

éŮóécŸ

ä;ăæČšçzZæšŔäyŭāšā■ŸāIJĀāŭaiŮäy■çŽDāĜ;æŤŕæŭzāŁæčĒěēŕāŽĪāĀĆ
äy■ēŁĜījNāL■æŔŔæŸŕēŁZäyŭāŭaiŮāšçzŔēčnārijaĒěāzŭäyŤēčnā;ŁçŤĪēŁĜāĀĆ

èĝčāĒşæŮzæąŁ

ēŁŽēĜNēŮóécŸçŽDæIJnēt'ĪāŕšæŸŕā;ăæČšāIJĀāŭaiŮēčnāŁæ;æŮŮæL'ĝēāNæšŔäyŭāŁĪā;IJāĀĆ
āŔŕēČ;æŸŕā;ăæČšāIJāyĀäyŭāŭaiŮēčnāŁæ;æŮŮēĝēāŔSæšŔäyŭāZđērČāĜ;æŤŕæĪēĀŽçšēā;ăāĀĆ

ēŁZäyŭēŮóécŸāŔŕāzēā;ŁçŤĪ10.11āŕŔēŁČäy■āŔNæāŭçŽDārijaĒěēŠŦā■ŔæIJzāŁŮæĪēāōđčŌŕāĀĆäyNēĪō

```
# postimport.py
import importlib
import sys
from collections import defaultdict

_post_import_hooks = defaultdict(list)

class PostImportFinder:
    def __init__(self):
        self._skip = set()

    def find_module(self, fullname, path=None):
        if fullname in self._skip:
            return None
        self._skip.add(fullname)
        return PostImportLoader(self)

class PostImportLoader:
    def __init__(self, finder):
        self._finder = finder

    def load_module(self, fullname):
        importlib.import_module(fullname)
        module = sys.modules[fullname]
        for func in _post_import_hooks[fullname]:
            func(module)
```

(continues on next page)

(continued from previous page)

```
self._finder._skip.remove(fullname)
return module

def when_imported(fullname):
    def decorate(func):
        if fullname in sys.modules:
            func(sys.modules[fullname])
        else:
            _post_import_hooks[fullname].append(func)
        return func
    return decorate

sys.meta_path.insert(0, PostImportFinder())
```

ðŁŹæüüijŃä;äårśåŔřäzëä;ŁçŦĲ when_imported() ðĈĚěřåŹĺäžĚüijŃä;ŃåĈüijŹ

```
>>> from postimport import when_imported
>>> @when_imported('threading')
... def warn_threads(mod):
...     print('Threads? Are you crazy?')
...
>>>
>>> import threading
Threads? Are you crazy?
>>>
```

ä;IJäyžäyÄäyĲæŽŦ åóđéŽĚçŽĎä;Ńå■ŔüijŃä;äårŔřëĈ;æĈşåIJĺåüşå■ŸåIJĺçŽĎåŕŹäzĲ'äyĲéĲæüzåĲäèĈĚé

```
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)
```

èőĲèőž

æIJñèŁĈæĲæIJřä;ĲëtŰäžŎ10.11årŔëŁĈäy■èőşèŁřèŁĠçŽĎåŕijåĚěéŦŦ'å■ŔüijŃäzűçĲ■ä;IJäŁőæŦžåĲĈ
@when_imported ðĈĚěřåŹĺçŽĎä;IJçŦĲæŸŕæşĺåĚŦåIJĺåŕijåĚěæŰűèĈńæŁĲæt'žçŽĎåd'ĎçŔĚåŹĺåĠ;å

```

    errecEeerZlæcAæšesys.modulesæIææšçIJNæIaaiUæYræRçIJšçZDâušçzRècnâLæ; ;äzEäÄC
    æCædIJæYrçZDèriijNèræad'DçRÊâZlècnçNâ■sèrCçTlâÄCäy■çDüriijNâd'DçRÊâZlècnæûzâLââLr
    _post_import_hooks                â■UâEÿäy■çZDäyÄäyIâLÛæaIäy■âÖzâÄC
    _post_import_hooks çZDä;IJçTlârsæYræTüéZEæL'ÄæIJL'çZDäyžæfRäyIæIaaiUæšlâEÏNçZDâd'DçRÊ
    äyÄäyIæIaaiUâRfäzææšlâEÏNâd'ZäyIâd'DçRÊâZlâÄC

```

```

    èeAèol'æIaaiUârijaEëâRÖègæâRŠæûzâLæçZDâLlâ;IJiijNPostImporter
    çszècnèð;ç;öäyžsys.meta_pathçññäyÄäyIâEÇçt'ââÄC âoCäijZæ■TèÖûæL'ÄæIJL'æIaaiUârijaEëæš■ä;IJâÄC

```

```

    æIJnèLCäy■çZDPostImporter çZDä;IJçTlâzüäy■æYræLæ; ;æIaaiUriijNèÄNæYrèGlâyæârijaE
    âodéZËçZDârijaEëècnâgTæt'çzZä;■äzÖsys.meta_pathäy■çZDâEüäzÛæšæL'çZlâÄC
    PostImporter                çszäy■çZD                imp.import_module()
    âG;æTrècnéÄŠâ;ŠçZDèrCçTlâÄC äyžäzEæAæâE■éZûâEëæUäçZfâ;IçÖriijNPostImporter
    äfIæNâäzEäyÄäyIæL'ÄæIJL'ècnâLæ; ;èfGçZDæIaaiUéZEâRlâÄC
    æCædIJäyÄäyIæIaaiUâR■â■YâIJlârsäijZçZt'æÖèècnâf;çTæÖL'âÄC

```

```

    â;ŠäyÄäyIæIaaiUècn                imp.import_module()                âLæ; ;âRÖriijN
    æL'ÄæIJL'âIJl_post_import_hooksècnæšlâEÏNçZDâd'DçRÊâZlècnèrCçTlriijNâ;fçTlâÛrâLæ; ;æIaaiUâ;IJäyžæ

```

```

    æIJL'äyÄçCzéIJæèeAæšlæDRçZDæYræIJnæIJäy■éÄCçTlâzÖèCçäzZéÄZèfG    imp.
    reload() ècnæY;âijRâLæ; ;çZDæIaaiUâÄC äzšârsæYrèrt'ijNæCædIJâ;ââLæ; ;äyÄäyIâzNâL■âûšècnâLæ
    âRæad'ÛriijNèeAæYrâ;ääzÖsys.modulesäy■âLæéZd'æIaaiUçDûâRÖâE■éG■æÛrârijaEëriijNâd'DçRÊâZlâRlâ
    æZt'ad'ZâEšäzÖârijaEëâRÖéŠr'â■RâfæAæfèrûâRÇèÄC PEP 369.

```

12.13 10.13 áóL'ècĚçgAæIJL'çZDâNĚ

éUóécY

ä;äæCšèeAâóL'ècĚäyÄäyIçññäyL'æÛzâNĚriijNâ;EæYræšææIJL'æIČéZŘârEâóCâóL'ècĚâLrçšçzçšPython
æLÛèÄEriijNâ;ââRfèC;æCšèeAâóL'ècĚäyÄäyIä;ZèGlâûsä;fçTlçZDâNĚriijNèÄNäy■æYrçšçzçšäyLéIæL'Äæ

ègçâEšæÚzæaI

PythonæIJL'äyÄäyIçTlæLûâóL'ècĚçZôâ;TüijNéÄZäyççszäijijâÄI~/local/lib/python3.3/site-
packagesâÄIâÄC èeAâijzâLûâIJlèfZäyIçZôâ;Täy■âóL'ècĚâNĚriijNâRfâ;fçTlâóL'ècĚéÄL'éqzâÄIJ-userâÄIâÄC

```
python3 setup.py install --user
```

æLÛèÄĚ

```
pip install --user packagename
```

âIJlsys.pathäy■çTlæLûçZDâÄIIsite-packagesâÄIçZôâ;Tä;■äzÖçšçzçšçZDâÄIIsite-
packagesâÄIçZôâ;TäzNâL'■âÄC âZâæm'd'riijNâ;ââóL'ècĚâIJlèGñéIççZDâNĚârsæfTçšçzçšâûšâóL'ècĚçZDâN
riijLâr;çôâzûäy■æÄzæYrèfZæûriijNèeAâRÛâEšäzÖçññäyL'æÛzâNĚçôâçRÊâZlriijNærTæCdistributedæLÛp

èõléõž

éÁŽāyāNĖāijŽēcāōL'ēċĒāLŕçşçzşçŽĐsite-packagesçŽōā;Tāy■āŌžīijNēurā;ĐçşzāijijāĀIJ/usr/local/
packagesāĀīāĀĈ āy■ēċĠīijNēċŽæāūāAŽēIJĀēċAæIJL'çōaçRĒāSŸæĪĈēŽŖāžūāyTā;ċçTĪsudoāS;āzd'āĀĈ
ārşçōŪā;āæIJL'ēċŽæāūçŽĐæĪĈēŽŖāŌzæL'gēāNāS;āzd'īijNā;ċçTĪsudoāŌzāōL'ēċĒāyĀāyĪæŪŕçŽĐīijNāRŕēĈ
āōL'ēċĒāNĖāLŕçTĪæLūçŽōā;Tāy■éÁŽāyāYŕāyĀāyĪæIJL'æTĪçŽĐæŪzæāLīijNāōĈāĒĀēōyā;āāLŽāzžā
āRēād'ŪīijNā;āēċYāRŕāzēāLŽāzžāyĀāyĪēŽŽæNşçŌŕāċĈīijNēċŽāyĪæLŠāznāIJlāyNāyĀēLĈāijŽēōsāLŕā

12.14 10.14 āLŽāzžæŪŕçŽĐPythonçŌŕāċĈ

éŪōēċŸ

ā;āæĈşāLŽāzžāyĀāyĪæŪŕçŽĐPythonçŌŕāċĈīijNçTĪæĪēāōL'ēċĒāīāīŪāŠNāNĖāĀĈ
āy■ēċĠīijNā;āāy■æĈşāōL'ēċĒāyĀāyĪæŪŕçŽĐPythonāĒNēŽĒīijNāžşāy■æĈşārçşçzşçPythonçŌŕāċĈāžğçTş

ēğĈāĒşæŪzæāL

ā;āāRŕāzēā;ċçTĪ pyenv āS;āzd'āLŽāzžāyĀāyĪæŪŕçŽĐāĀIJēŽŽæNşāĀĪçŌŕāċĈāĀĈ
ēċŽāyĪāS;āzd'ēċnāōL'ēċĒāIJPythonēğċēĠLāŽĪāRŕNāyĀçŽōā;TīijNāLŪWindowsāyĪēĪċçŽĐScriptsçŽōā;Tāy

```
bash % pyenv Spam
bash %
```

āijāçžŽ pyenv āS;āzd'çŽĐāR■ā■ŪæYŕāŕĒēċĀēċnāLŽāzžçŽĐçŽōā;TāR■āĀĈā;ŞēċnāLŽāzžāRŌīijNS

```
bash % cd Spam
bash % ls
bin include lib pyenv.cfg
bash %
```

āIJĪbinçŽōā;Tāy■īijNā;āāijŽæL;āLŕāyĀāyĪāRŕāzēā;ċçTĪçŽĐPythonēğċēĠLāŽĪīijŽ

```
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']
>>>
```

èóíèőž

```
bash % pyvenv --system-site-packages Spam
bash %
```

12.15 10.15 aLÉaRŠaŇĚ

éŮőécŸ

èġčǎEşæŮzæąŁ

```
projectname/  
  README.txt  
  Doc/  
    documentation.txt
```

408

(continued from previous page)

```
projectname/  
    __init__.py  
    foo.py  
    bar.py  
    utils/  
        __init__.py  
        spam.py  
        grok.py  
examples/  
    helloworld.py  
...
```

ðeAeðl'ä;äçŽDāNĖāRřāzēāRŠāyČāGžāŌžījNēēŪāĒLä;äēAçijŪāEZāyĀäyĭ setup.
py īijNčšzāijjāyNéĪcēŁZæāūījŽ

```
# 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'],  
)
```

äyNāyĀæ■ēījNārsæYřāŁZāzžāyĀäyĭ MANIFEST.in æŪGāzūījNāŁŪāGžæL'ĀæIJL'āIJlä;äçŽDāNĖāy

```
# MANIFEST.in  
include *.txt  
recursive-include examples *  
recursive-include Doc *
```

çaðāĪI setup.py āŠN MANIFEST.in æŪGāzūæŤĭāIJlä;äçŽDāNĖçŽDæIJĀéāūçžgçZōā;Ťāy■āĀĆ
äyĀæŪēā;āāūšçzRāAŽāzEēŁZāžŽījNā;āārsāRřāzēāČRāyNéĪcēŁZæāūæL'gēāNāŠ;āzđ' æĪēāŁZāzžāyĀäyĭæž

```
% bash python3 setup.py sdist
```

āōČāijŽāŁZāzžāyĀäyĭæŪGāzūæŤāēČāĀIprojectname-1.0.zipāĪ æĹŪ
āĀIJprojectname-1.0.tar.gzāĪ, āĒŪā;Šā;ĪēŤŪāžŌā;äçŽDčšzçzšāžšāRřāĀČāēČæđIJāyĀāŁGæ■cāyīījN
ēŁZāyĭæŪGāzūārsāRřāzēāRŠéĀAçzŽāŁnāžžā;ŁçŤĪæĹŪēĀĒāyĪāijæēGš Python Package In-
dex.

ēōĪēōž

āržāžŌčžfPythonāžččāAīijNçijŪāEZāyĀäyĭæŽōēĀŽçŽD setup.py
æŪGāzūēĀŽāyīāĭŁçōĀā■ŤāĀĆ äyĀäyĭāRřēČ;çŽDēŪōēčYæYřā;āāŁĒēāzæL'NāŁĪāŁŪāGžæL'ĀæIJL'æđDæ
äyĀäyĭāyīyēgAēŤŽēřārsæYřāzēāzēāRřāŁŪāGžāyĀäyĭāNĖçŽDæIJĀéāūçžgçZōā;ŤījNāŁYēōřāzEāNĖāRřāN

ād'gēČlāLEPythončlNāžRāSŸēČ;çšēēAŞrijNæIJL'ā;Ĺad'ŽçññäyL'æŮzāNĚçðaçRĚāZlā;ŽéĀL'æNl'rijN
 æIJL'āžZæŸřāyžāEæŽĚāžčæāGāĜEāžŠäy■ŽĎdistutilsāĀCæslæĎRæCædIJā;äā;ĭeŮeĚZāžZāNĚrijN
 çTlāĹuāRřēČ;äy■ēČ;āōL'ēčĚā;āçŽĎē;řāzūrijNēZĎ'eĭdāzŮāžnāušçzRāžNāĚĹāōL'ēčĚēĚĜæL'ĀēIJĀēēAçŽĎ
 æ■čāZāāēCæ■d'rijNā;āæZt'āžTēřēæŮūāLzēōřā;RēūLçōĀā■TēūĹāē;çŽĎēAŞçRĚāĀC
 æIJĀāē;ēōĹ'ā;āçŽĎāžčçāAā;ĤçTlāæāGāĜEçŽĎPython 3āōL'ēčĚāĀC
 āēCædIJāĚūāžŮāNĚāžšēIJĀēēAçŽĎēřlrijNāRřāžēēĀŽēĚĜāyĀāyĹāRřēĀL'ēāžæĭēæTřæNĀāĀC
 āřžāžŌæūL'āRĹāĹLřCæL'l'āsTçŽĎāžčçāAæL'ŠāNĚäyŌāĹEāRŠāřsæZt'ād'■æĭČçCzāžEāĀC
 çññ15çñāāržāĚšāžŌCæL'l'āsTçŽĎēĚZæŮzēĭççšēēřEæIJL'äyĀāžZēřēçzEēōšēgčrijNçL'zāĹnæŸřāIJl15.2ārRē

(continued from previous page)

```
u = request.urlopen(url+'?' + querystring)
resp = u.read()
```

æĆæđIĲă;ăéIĴăèĀă;ǣćŦÍPOSTæŨzæsȚăIĴlêrűəsĆăyźă;Şăy■ăRŚéĂĂăşşèérăRCăȚriijNăRfrazēărEăr
urlopen()ăĞ;ăȚriijNărsăČŘèfZăăüııjŽ

```
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()
```

æĆæđIĲă;ăéIĲĂèĖAăIĲăĲŖŚăĠžĉŽĐèŕuăŝĆăy■æŖŖă;ŽăyĂăžŽèĠăôŽăzĲ'ĉŽĐHTTĲăđ't'ĲĲŃă;ŃăĖĆăđ
user-agent â■Ūăōŧ,ăŖŖăžăĖăĲăžăžăyĂăyĲăŃăĖăŖăă■ŪăōŧăĂĲĲŽĐă■ŪăĖyĲĲŃăžăăăĲăžăžăyĂăyĲRequestă
urlopen() ĲĲŃăĖĆăyŃĲĲŽ

```
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()
```

requests

```
import requests
```

(continues on next page)

(continued from previous page)

```
# Base URL being accessed
url = 'http://httpbin.org/post'

# Dictionary of query parameters (if any)
parms = {
    'name1' : 'value1',
    'name2' : 'value2'
}

# Extra headers
headers = {
    'User-agent' : 'none/ofyourbusiness',
    'Spam' : 'Eggs'
}

resp = requests.post(url, data=parms, headers=headers)

# Decoded text returned by the request
text = resp.text
```

āĖšāžŌrequestsāžŠiijNāyĀāyĭāĀijāĭ ŪāyĀæRRçŽĎçL'žæĀğārsæŸřāōČèČ;āžēād'Žçğ■æŪžāijRāzŌērūā
resp.text āyēcžŽæĹSāžñçŽĎæŸřāžēUnicodeèğçčāAçŽĎā\$■āžTæŪGæIJñāĀČā;EæŸřiijNāçCæđIJāŌžē
resp.content iijNārsāijŽāĭ ŪāĹrāŌšāgNçŽĎāžNēŁŽāĹūæTřæ■ōāĀČāRēāyĀæŪžēĹciijNāçCæđIJēōŁēŪ
resp.json iijNēČčāžĹārsāijŽāĭ ŪāĹrJSONæāijāijRçŽĎā\$■āžTāĖĖāōžāĀČ

āyNēĹcēŁŽāyĭčđ'žāĭNāĹĹ'çTĹ requests āžŠāRŚēĭūāyĀāyĭHEAD-
ērūāśČriijNāžūāžŌā\$■āžTāy■æRRāRŪāGžāyĀāžŽHTTPāđ't'æTřæ■ōçŽĎā■ŪæōřiijŽ

```
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']
```

āyNēĹcæŸřāyĀāyĭāĹĹ'çTĹrequestsēĀŽēŁGāšžæIJñēōđ'ērAçŽžā;TPypicžŽĎāĭNā■RiijŽ

```
import requests

resp = requests.get('http://pypi.python.org/pypi?action=login',
                    auth=('user', 'password'))
```

āyNēĹcæŸřāyĀāyĭāĹĹ'çTĹrequestsārĖHTTP cookiesāžŌāyĀāyĭērūāśČāijāēĀŠāĹrāRēāyĀāyĭçŽĎāĭNā■

```
import requests

# First request
```

(continues on next page)

(continued from previous page)

```
resp1 = requests.get(url)
...

# Second requests with cookies received on first requests
resp2 = requests.get(url, cookies=resp1.cookies)
```

æIJĀāŔŌä;EāzúéİdæIJÄy■éĜ■èeAçŽDäyÄäyġä;Nā■ŔæYřçŦlrequestsäyLäijääEĖĖőžijŽ

```
import requests
url = 'http://httpbin.org/post'
files = { 'file': ('data.csv', open('data.csv', 'rb')) }

r = requests.post(url, files=files)
```

ěóİěőž

ārzāžŎçIJšçŽDā;ŁçőĀā■ŦHTTPāóçæŁuçnrāzččāAüijNçŦlāEĖç;őçŽD urllib
æġāāİŮéĀŽäyYāršèüšād'šāžEāĀCä;EæYřijNāeCædIJä;äèeAāAŽçŽDäy■äzĖäzĖĀŔæYřçőĀā■ŦçŽDGETæL
requests ād'gæY;èžnæL'NçŽDæŮüāĀžāžEāĀC

ä;NāeCüijNāeCædIJä;āāEšāőZāİZæNĀä;ŁçŦlāāGāGĖçŽDçİNāžŔāžšèĀNäy■èĀCèZŠāČŔ
requests èŁZæāüçŽDçñnāyL'æŮžāžšüijNéCčāZŁāžšèöyāršāy■ā;Ůäy■ā;ŁçŦlāžŦāsCçŽD
http.client æġāāİŮæİēāōđçŎŕeĜlāüšçŽDāžččāAāĀCærŦæŮžèŕ'üijNāyNéİççŽDāžččāAāšŦçd'žāžEāeČā

```
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)
```

āŔNæāüāİŕüijNāeCædIJāŁĖēāzçijŮāEŽæŮL'āŔŁāžčçŔĖāĀAēōd'ērAāĀAcookiesāžēāŔŁāĖüāžŮäyĀāž
urllib āŕšæY;ä;ŮçL'žāŁnāŁnæL■āšNāŦŕāŮēāĀCærŦæŮžèŕ'üijNāyNéİçèŁŽäyŁçd'žä;NāōđçŎŕāIJPython

```
import urllib.request

auth = urllib.request.HTTPBasicAuthHandler()
auth.add_password('pypi', 'http://pypi.python.org', 'username',
    ↪ 'password')
opener = urllib.request.build_opener(auth)

r = urllib.request.Request('http://pypi.python.org/pypi?
    ↪ :action=login')
u = opener.open(r)
```

(continues on next page)

(continued from previous page)

```
resp = u.read()

# From here. You can access more pages using opener
...
```

requests

requests

```
>>> 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'}
>>>
```

requests

requests

13.2 11.2 TCP

UőéćŸ

requests

ğċÅEşæŮzæąĹ

requests

```
from socketserver import BaseRequestHandler, TCPServer

class EchoHandler(BaseRequestHandler):
    def handle(self):
        print('Got connection from', self.client_address)
```

(continues on next page)

(continued from previous page)

```
while True:

    msg = self.request.recv(8192)
    if not msg:
        break
    self.request.send(msg)

if __name__ == '__main__':
    serv = TCPServer(('', 20000), EchoHandler)
    serv.serve_forever()
```

ãĲĲēŁŻæōtäzççäÄäy■ĲĲNä;ääōŽāzŁ'āžEäyÄäyŁçŁ'zæōŁçŽĎāđ'ĐçŘEçsziĲNāōđçŎřāžEäyÄäyŁ
handle() æŰzæşŦĲĲNçŦĲĲēäyžāōçæŁüçnrēŁđæŎčæĲ■āŁāĀĀĆ request
āsđæĀğæŸřāōçæŁüçnrsocketĲĲNclient_address æĲĲāōçæŁüçnrāĲĲāĀĀĆ
äyžāžEäyŦNērŦēŁŽāyŁæĲ■āŁāāŽĲĲNēŁŘēāNāōČāzūāŁ'ŞāĲĲĀĲēāđ'ŰäyÄäyŁPythonēŁŽçĲNēŁđæŎčēŁŽāyŁæ

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

āĲĲāđ'ŽæŰūāĀŽĲĲNāŦřāžēāĲĲāōžæŸŞçŽĎāđ'ŽāzŁ'äyÄäyŁäy■āŦNçŽĎāđ'ĐçŘEāŽĲāĀĀĆäyNēĲæŸřāyĀ
StreamRequestHandler āşžçsžārEäyÄäyŁçsžæŰĞāzūāŎčāŦçæŦç;ōāĲĲāžŦāśČsocketäyŁçŽĎäĲNā■Ŧ

```
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()
```

ēōĲēōž

socketserver āŦřāžēēōĲ'æŁŚāžnāĲĲāōžæŸŞçŽĎāđ'ŽāzŁçōĀā■ŦçŽĎTCPæĲ■āŁāāŽĲāĀĀĆ
āĲĲæŸřĲĲNä;äēĲĲāēçAæşŁæĐŦçŽĎæŸřĲĲNēzŸēōđ'æČēāEŦäyNēŁŽçğ■æĲ■āŁāāŽĲæŸřā■ŦçžŁçĲNçŽĎĲĲNä;
āçČāđĲĲā;āæČşāđ'ĐçŘEāđ'ŽāyŁāōçæŁüçnrĲĲNāŦřāžēāĲĲāğNāNŰäyÄäyŁ
ForkingTCPServer æŁŰēĀĒæŸř ThreadingTCPServer āřžēşāāĀĀĆĲNāēČĲĲŽ

```
from socketserver import ThreadingTCPServer
```

```
if __name__ == '__main__':  
    serv = ThreadingTCPServer(('', 20000), EchoHandler)  
    serv.serve_forever()
```

ä;fçTíforkæLÚçžfçlNæI■āLāZlæIJL'äylæ;IJāIJléUőécYārsæYřāōČāznāijZāyžæfRāylāōcæLūčnrēfđæ
çTšāzŌāōcæLūčnrēfđæŌēæTřæYřæšæaIJL'ēŽŘāLūčŽDiiJNāZāæ■d'äyÄäylæAūæDRčŽDézŠāōcāRřāzēāRŇ

āēČādIJā;āæNĚāfČēfZāyléUőécYiiJNā;āāRřāzēāLZāzzāyÄāylécDāĚLāLĚēĚ■ād'gārRčŽDāuēā;IJçžfç
ä;āāĚLāLZāzzāyÄāylæŽōēĀŽçŽDēlđçžfçlNæI■āLāZlīijNčDūāRŌāIJlāyÄāylçžfçlNæšāy■ä;fçTí
serve_forever() æŰzæşTælēāRřāLlāōČāznāĀČ

```
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()
```

äyĀēLŇæIēēōšiiJNāyÄäyl TCPServer āIJlāōdā;NāNŰčŽDæUūāĀŽāijŽçzŠāōZāzūæfĀæt'zçZyāzTçŽ
socket āĀČ äy■ēfGiiJNāIJL'æUūāĀŽā;āæČšēĀŽēfGēō;ç;ōæšRāzŽēĀL'éqāāŌžēřČæTř'āžTāyNčŽD
socket' iiJNāRřāzēēō;ç;ōāRČæTř bind_and_activate=False āĀČāēČāyNiiJŽ

```
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()
```

äyLēlčçŽD socket éĀL'éqāæYřāyÄäylēlđāyāæŽōēA■çŽDēĚ■ç;ōēqzīijNāōČāĚAēōyæI■āLāZlēG■æ
çTšāzŌēēAēčnczRāyā;fçTíāLriijNāōČēcnæTç;ōāLřçszāRŸēGRāy■iiJNāRřāzēçZř'æŌēāIJl
TCPServer äyLēlčēō;ç;ōāĀČ āIJlāōdā;NāNŰæI■āLāZlčçŽDæUūāĀŽāŌžēō;ç;ōāōČçŽDāĀijiiJNāēČāyN

```
if __name__ == '__main__':  
    TCPServer.allow_reuse_address = True  
    serv = TCPServer(('', 20000), EchoHandler)  
    serv.serve_forever()
```

āIJlāyLēlčçd'žā;Nāy■iiJNāLŠāznāijTçd'žāzĚāyd'çg■āy■āRŇçŽDād'DçŘĚāZlāšžçszīijL
BaseRequestHandler āŠŇ StreamRequestHandler iiJL'āĀČ
StreamRequestHandler æŽř'āLāçAřæt'zçČzīijNēČ;éĀŽēfGēō;ç;ōāĚūāzŰçŽDçszāRŸēGRælēæTřæNā


```

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!')

```

æIJĀāRŌijNēĒYēIJĀēēAæslæDRçŽDæYrāulād'gēČlāLEPythonçŽDénYāsČç;ŚçzIJælaāIŪijLæfTæČl
 RPCç■LijL'ēČ;æYrāzçñNāIJĪ socketsserver āLšèČ;āzNāyLāĀČ
 āzšārsæYrēt'ijNçŽt'æŌēā;ĲçTĪ socket āzSælēāōđçŌræIJ■āLāāZlāzšāzūāy■æYrā;LéŽ;āĀČ
 āyNēlæYrāyĀāyā;ĲçTĪ socket çŽt'æŌēçijŪçlNāōđçŌrçŽDāyĀāyāæIJ■āLāāZlçōĀā■Tā;Nā■RijŽ

```

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 aLZaZZUDPaeIJaaLaZi

eUoeCY

ajaaCsaodcOrayAaylaZazOUDPaaReoocZDaeIJaaLaZiaIeayOaocaeLuncraAZaLaAaC

egcaEsaUzaal

euSTCPayAauiijNUDPaeIJaaLaZiaZsaRrazeAzeGajfcTi socketserver
azSaLLaOzaYScZDenaLZazzaAC ajNaCiiNayNeIcaeYrayAaylcoAaTcZDaeUueUt aeIJaaLaZiijZ

```
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()
```

euSazNaLaaayAauiijNajaaELaOZazLayAaylaodcOr handle()
clzaolaeUzaesTcZDcszrijNayzaocaeLuncraedaeOeaeIJaaLaAaC eZaylcszcZD
request asdaeAgaeYrayAaylaNEaRnaZEaeTreaoaeLeaSNazTasCsock-
etarzesaZDaECczDaaCclient_address aNEaRnaZEaocaeLuncraIJraIAaC

aLSaznaIeaeTnerTayNeZaylaeIJaaLaZiijNeeUaELeReaNaocCiiNcDuaROaeL'SajAaRead' UayAayH

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

eoieoz

ayAaylaEyaadNcZDUDPaeIJaaLaZiaOeaeTuulReLcZDaeTreaoaeLe(aulAeAf)aSNaocaeLuncraIJraIAaA
aoCeeAaczZaocaeLuncraZdaRSayAaylaTreaoaeLeaACarzaZOaeTreaoaeLeZDaijaeAAiijN
ajaaZTerajaLcTisocketcZD sendto() aSN recvfrom() aeUzaesTaAC
aricoaijaczscZD send() aSN recv() azsaRrazeLalraRnaeucZDaeTLaedIiijN
ajEaeYraLaelcZDayd'aylaeUzaesTarfzaOUDPefdaeOeeANeIAeZt aeZoeAaAC

çTšāžŌæšqæIJL'āžTāsĆçŽDēđđæŌëijNUPDæIJ■āLāāZÍçŽýárzāžŌTCPæIJ■āLāāZÍæIēēōšāōđçŌrètūæI
 äy■ēfGijNUPDāđ'f'çTšæYřāy■āRfēIāçŽDijLāZāyžēĀŽāfæšqæIJL'āžžčñNēfđæŌëijNæūLæAřāRfēČ;āy
 āZāæ■đ' éIJĀēēAçTšā;āēGlaūsæIēāEšāōZēřæĀŌæūāđ' DçREäyčād' sāūLæAřçŽDæČĚāEřāĀČēfZāyIāušçz
 äy■ēfGēĀŽāyāēIēēřt'ijNāēCæđIJāRfēIāæĀgāržāžŌā;āçlNāžRā;LéG■ēēAijNā;āēIJĀēēAāĀšāL'āžŌāžRāI
 UDPéĀŽāyēčñçTlāIJléCčāžZāržāžŌāRfēIāijāē;SēēAæšCāy■æYřā;LénYçŽDāIJžāRĪāĀČā;NāēČijNāIJā
 æŪāēIJĀēfTāZđæAčād'■āyčād'šçŽDæTřæ■ōāNĚijLçlNāžRāRfēIJĀčōĀā■TçŽDāf;çTēāōCāzūçžgçz■āRŠāI

UDPServer çszæYřā■TçžfçlNçŽDijNāžšārsæYřēřt'āyĀæñāāRfēČ;āyžāyĀāyIāōcæLūçñrēfđæŌēæIJ■
 āōđēŽĚā;fçTlāy■ijNēfZāyIæŪāēōžæYřāržāžŌUDPeYæYřTCPéČ;āy■æYřāžĀāzLād'gēŪōēcYāĀČ
 āēCæđIJā;āæČšēēAāzūāRŠæS■ā;IJijNāRfāžēāōđā;NāNŪāyĀāyI ForkingUDPServer
 æLŪ ThreadingUDPServer āřžēšāijŽ

```
from socketserver import ThreadingUDPServer

if __name__ == '__main__':
    serv = ThreadingUDPServer(('', 20000), TimeHandler)
    serv.serve_forever()
```

çŽt'æŌēā;fçTl socket æIēāōđçŌřāyĀāyIUDPæIJ■āLāāZÍāzšāy■ēŽ;ijNāyNēIcæYřāyĀāyIā;Nā■Rijž

```
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 éĀŽèĚGCIDRāIJřāIĀçTšæLŘāržāžTçŽDIPāIJřāIĀéŽE

éŪōēcY

ā;āæIJL'āyĀāyI CIDRç;ŠçzIJāIJřāIĀæřTāēČāĀIJ123.45.67.89/27āĀIijNā;āæČšāřEāĚūē;ñæ■cæLŘāōČā
 ijLæřTāēČijNāĀIJ123.45.67.64āĀI, āĀIJ123.45.67.65āĀI, āĀē, āĀIJ123.45.67.95āĀI)ijL'

ēgčāEšæŪzæāL

āRřāžēā;fçTl ipaddress æIāāIŪā;LāōžæYšçŽDāōđçŌrēfZæūçŽDēōāçōŪāĀČā;NāēČijž

```

>>> 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 äzşâĖAëöÿăĈRæTřčzĎäÿĂæăŭčŽĎčť cáijTăRŮăĀijīijNăĭNăeĆīijŽ

```

>>> 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')
>>>

```

ăRëăđ'ŮīijNăĭăeĤYăRŕăzëæL'ğëąNçĭŞçzIJæĹŔăŚŸæčĂæşëăzNçşzçŽĎæŞăĭIJīijŽ

```

>>> a = ipaddress.ip_address('123.45.67.69')

```

(continues on next page)

(continued from previous page)

```
>>> a in net
True
>>> b = ipaddress.ip_address('123.45.67.123')
>>> b in net
False
>>>
```

äyÄäyHPaIJraIAaŠNç;ŠçzIJaIJraIAeČ;éÄŽeŁGäyÄäyHPaeÖěaRčaeIeaeNGaōŽiijNä;NaeĆiijŽ

```
>>> inet = ipaddress.ip_interface('123.45.67.73/27')
>>> inet.network
IPv4Network('123.45.67.64/27')
>>> inet.ip
IPv4Address('123.45.67.73')
>>>
```

ěóIěōž

ipaddress æIaaiUæIJL'ā;Ład'ŽçszāRfäzēēāIçd'žIPaIJraIAāAAç;ŠçzIJaŠNæÖěaRčāĀĆ
ā;Šä;äeIJÄeAæŠ■ā;IJç;ŠçzIJaIJraIAiijLæfTæČeğčædŘāĀAæL'Šā■rāĀAeIÑeřAç■L'iijL'çŽDæUúāĀŽaijŽā

ēeAæslæDRçŽDæYřiiijNipaddress æIaaiUeūšāĒūāzŮäyÄāzŽaŠNç;ŠçzIJçŽyāĒšçŽDæIaaiUæfTæČ
socket āžŠāžd'ēZEā;ŁārŠāĀĆ æL'ÄāžēriijNä;äy■ēČ;ā;ŁçTÍ IPv4Address
çŽDāōdä;NæIeāzčæŽfäyÄäyIaIJraIAa■UçņeäyšiiijNä;äeēŮāĒŁa;ŮæY;aijRçŽDä;ŁçTÍ
str() èiñæ■cāōČāĀĆä;NaeĆiijŽ

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

æŽt'ad'ŽçŽyāĒšāĒĒāōžiiijNerūāRČeĀĆ An Introduction to the ipaddress Module

13.5 11.5 āLŽāžžäyÄäyIçōĀa■TçŽDRESTæÖěaRč

ěUőećY

ä;äaeČsä;ŁçTÍäyÄäyIçōĀa■TçŽDRESTæÖěaRčéÄŽeŁGç;ŠçzIJeŁIJç;NæÖğāŁūæLŮeōŁeŮōä;äçŽDāžT

èġċaEṣæŮzæaġL

ædDāzžāyĀāyġRESTċŌæāijçŽĎæŌċaRċæIJĀçōĀā■TçŽĎæŮzæṣTæYřāLZāzžāyĀāyġlāṣžāžŌWSGIæā
3333ijLçŽĎāġLārRçŽĎāžṢijNāyNéġæYřāyĀāyġlāġNā■RijŽ

```
# 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
```

äyžāEāġçTġēZāyġēČāžæŽġijNāġāāRġæIJĀæçAçijŮāEŽāy■āRŃçŽĎāđDçŘEāŽġijNārśāČRāyNéġcèE

```
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()

```

ěĖAætNèrTáyNèfZáylæI■āLāāZlíjNā;āāRfázēā;ŁçTlāyĀāyŁætRèġLāZlæLŮ urllib
 āŠNāōCāzd'āzŠāĀCā;NāēCījŽ

```

>>> 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)

```
<?xml version="1.0"?>
<time>
  <year>2012</year>
  <month>11</month>
  <day>24</day>
  <hour>14</hour>
  <minute>49</minute>
  <second>17</second>
</time>
>>>
```

èóìèőž

āIJlċijŪāEZRESTæŌēāRċæŪūiijNēĀŽāyēĈċ;æŸræIJ■āŁažŌæŽŏēĀŽĉŽDHTTTPēfūæśĈāĀĈā;EæŸrē
ēfŽāžŽæŤræ■ōāžēāRĎĉg■æāGāGEæāijāijRċijŪĉāAīijNærŤæĈXMLāĀAJSONæLŪCSVāĀĈ
ārċŏaĉlNāžRċIJNāyŁāŌžā;ŁĉŏĀā■ŤiijNā;EæŸræžēēfŽĉg■æŪžāijRæRĀ;ŽĉŽDAPIāržžāŌā;Łād'ŽāžŤĉŤlċ

ā;NāēĈiijNēŤfæIJšēfRēāNĉŽĎĉlNāžRāRfēĈ;āijŽā;ĤĉŤlāyĀāyĪREST
APIælēāŏĉĈŌĉŽŚæŌgæLŪēfŁæŪ■āĀĈ ād'gæŤræ■ōāžŤĉŤlĉlNāžRāRfæžēā;ĤĉŤlĪRESTælēādĎāžžāyĀāyĪæ'
RESTēfŸēĈ;ĉŤlælēæŌgāLŪĉāñāžūēŏ;ād'GærŤæĈæIJžāŽlāžžāĀāijāæĎšāŽlāĀāūēāŌĈæLŪĉAſæšāāĀĈ
æŽt'ēG■ēēAĉŽĎæŸiijNREST APIāūšĉzRēĉnād'gēGRāŏĉæLŪĉnŕċijŪĉlNĉŌſāĈæL'ĀæŤræNāiijNærŤæĈ-
Javascript, Android, iOS■lāĀĈ āŽāæ■d'iijNāLl'ĉŤlēfŽĉg■æŌēāRċāRfæžēēŏl'ā;āāijĀāRŚāGžæŽt'āŁāād'■æ

āyžāžEāŏĉĈŌſāyĀāyĪĉŏĀā■ŤĉŽDRESTæŌēāRċiijNā;āāRlēIJĀēŏl'ā;āĉŽĎĉlNāžRāžĉĉāAæžāēūšPython
WSGIēĉnāæāGāGEāžŚæŤræNāiijNāRŊæŪūāžšēĉnĉzĪād'gēĈlāLEĉññāyL'æŪžwebæaEæđūæŤræNāāĀĈ
āŽāæ■d'iijNāēĈādIJā;āĉŽĎāžĉĉāAēAſā;ĤēfŽāyĪæāGāGEiijNāIJlāRŌēlĉĉŽĎā;ĤĉŤlēfGĉlNāy■āršāijŽæŽt'āŁ

āIJlWSGIāy■iijNā;āāRfæžēāĈRāyNēlĉēfŽæūĉžēāŏŽĉŽĎæŪžāijRāžēāyĀāyĪāRfēĈĉŤlāržžēā;ĉāijRæl

```
import cgi

def wsgi_app(environ, start_response):
    pass
```

environ āśdæĀgæŸſāyĀāyĪā■ŪāĒyriijNāNēāRŋāžEāžŌwebæIJ■āŁažŽlāēĈA-
pache[āRĈēĀĈInternet RFC 3875]æRĀ;ŽĉŽDCGIæŌēāRċāy■ēŌūāRŪĉŽĎāAīijāĀĈ
ēēAārEēfŽāžŽāy■āRŊĉŽĎāAīijæRĀRŪāGžælēiijNā;āāRfæžēāĈRēfŽāžLēfŽæūāEZiijŽ

```
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)
```

æŁŚāžñāsŤĉd'žāžEāyĀāžŽāyēēgAĉŽĎāAīijāĀĈenviron['REQUEST_METHOD']
āžĉēālēſūæśĈśšādNāēĈGETāĀPOSTāĀHEADĉ■L'āĀĈ environ['PATH_INFO']
ēāĪĉd'žēĉnēſūæśĈĉĎDæžRĉŽĎēſā;ĎāĀĈ ēŕĈĉŤlċ cgi.FieldStorage()
āRfæžēāžŌēſūæśĈāy■æRĀRŪāšēēſĉāRĈæŤſāžūārEāŏĈāžnāŤ;āĒēāyĀāyĪĉśšā■ŪāĒyāržžēāy■āžēā;ĤāRŌ

start_response('200 OK', [('Content-type', 'text/plain')])

```
def wsgi_app(environ, start_response):
    pass
    start_response('200 OK', [('Content-type', 'text/plain')])
```

yield b'Hello World\n'

```
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
```

yield b'Goodbye!\n'

```
def wsgi_app(environ, start_response):
    pass
    start_response('200 OK', [('Content-type', 'text/plain')])
    yield b'Hello World\n'
    yield b'Goodbye!\n'
```

class WSGIApplication:

```
class WSGIApplication:
    def __init__(self):
        ...
    def __call__(self, environ, start_response):
        ...
```

environ['params']

print()

yield b'Goodbye!\n'

āZāyZāĜāĜĒārzāžŌāIJāLāāZlāŠNāāEāēdūāYřāy■čnNčŽDrijNāāāRřāzēāRĒā;āçŽDčlNāzRāTĭāĚāzāā
æLŠāznā;ĤçTlāyNéIcçŽDāzççāAætNērTætNērTætIJnēLČāzççāAijŽ

```
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()
```

āyLéIcāzççāAāLZāzžāžEāyĀāyĭçōĀā■TçŽDæIJāLāāZlīijNçDūāRŌā;āārśāRřāzēāIēætNērTāyNā;āçŽD
æIJĀāRŌrijNā;Šā;āāĜĒād'ĜēfZāyĀā■ēāL'āsTā;āçŽDčlNāzRčŽDæŪūāĀZijNā;āāRřāzēāfōāTzēfZāyĭāz

WSGIæIJnēznāYřāyĀāyĭāĭLārRčŽDæāĜāĜĒāĀČāZāē■d'āōČāzūāēšāēIJL'æRŘāĭZāyĀāžZénYčžgçŽD
ēfZāžZā;āēĜlāūsāōdçŌřētuāIēāzšāy■ēŽĭāĀCāy■ēfĜāçCādIJā;āāČšēēAæZt'ād'ŽçŽDæTřāēNāijNāRřāzēē
WebOb æLŪēĀĚ Paste

13.6 11.6 éĀŽēĚGXML-RPCāōdçŌřçōĀā■TçŽDēĚIJčlNērČçTĭ

éŬōécY

ā;āāČšāLĭāLřāyĀāyĭçōĀā■TçŽDæŪzāijRāŌzāLġēāNēfRēāNāIJlēfIJčlNāIJzāZlāyLéIcçŽDPythončl

èġçāEşæŪzæāĭ

āōdçŌřāyĀāyĭēfIJčlNāŪzæşTērČçTĭçŽDæIJĀçōĀā■TæŪzāijRāYřā;ĤçTĭXML-
RPCāĀCāyNéIcæLŠāznāijTçd'žāyĀāyNāyĀāyĭāōdçŌřāžEēTō-
āĀijā■YāČlāLšēČçŽDçōĀā■TæIJāLāāZlīijŽ

```
from xmlrpc.server import SimpleXMLRPCServer  
  
class KeyValueServer:  
    _rpc_methods_ = ['get', 'set', 'delete', 'exists', 'keys']  
    def __init__(self, address):  
        self._data = {}  
        self._serv = SimpleXMLRPCServer(address, allow_none=True)  
        for name in self._rpc_methods_:  
            self._serv.register_function(getattr(self, name))  
  
    def get(self, name):  
        return self._data[name]  
  
    def set(self, name, value):
```

(continues on next page)

(continued from previous page)

```
        self._data[name] = value

    def delete(self, name):
        del self._data[name]

    def exists(self, name):
        return name in self._data

    def keys(self):
        return list(self._data)

    def serve_forever(self):
        self._serv.serve_forever()

# Example
if __name__ == '__main__':
    kvserv = KeyValueServer(('', 15000))
    kvserv.serve_forever()
```

äyÑéÍæĹŚäzñäzŌäyÄäyĭäöçæĹuçñræIJzâŽĭäyĹéÍæĭëèöĚéŮöæIJ■āŁāāŽĭijŽ

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

èöĭëöž

XML-RPC āŔfäzëèŏĭæĹŚäzñä;ĹäöžæŸŞçŽĎæĎĎéĀäyÄäyĭçŏĀ■ŤçŽĎèĭIJĭĹNerČçŤĭæIJ■āŁāāĀĆäĭ
éĀŽèĚĠäöČçŽĎæŮžæşŤ register_function() æĭëæşĭāĒŃāĠæŤĭijŇçĎŭāŔŌäĭĚçŤĭæŮžæşŤ
serve_forever() āŔfāĹĭäöČāĀĆ āIJäyĹéÍæĹŚäzñärĒëĚžäzŽæ■ëĭd'æŤĭāIJäyĀëtŭāĒŽāĹrāyÄäyĭçşž

```
from xmlrpc.server import SimpleXMLRPCServer
def add(x, y):
    return x+y

serv = SimpleXMLRPCServer(('', 15000))
```

(continues on next page)

(continued from previous page)

```
serv.register_function(add)
serv.serve_forever()
```

XML-RPCæŽt' éIJšāGžæIēçŽDāG;æTřāRlèČ;éĀĆçTlāžŌēČlāLEæTřæ■ōçšzādNīijNærTāeCā■Ūçņēāyš.
ārzažŌāĒūāžŪçšzādNāršā; ŪēIJĀēēAāAžāžZēčlād' ŪçŽDāLšēr;āžEāĀĆ
ā;NāēČīijNāēČædIJā;āēČšéĀŽēĒG XML-RPC āijāēĀŠāyĀāyIāržēšāōđā;NīijNāōđēŽĒāyLāRlæIJL'āžŪçŽD

```
>>> 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}
>>>
```

çšzāijijçŽDīijNārzažŌāžNēĒZāLūæTřæ■ōçŽDād' ĎçRĒāžšēušā;āēČšēšāçŽDāy■ād' IāyĀæāūīijŽ

```
>>> s.set('foo', b'Hello World')
>>> s.get('foo')
<xmlrpc.client.Binary object at 0x10131d410>

>>> _ .data
b'Hello World'
>>>
```

āyĀēLñæIēēōšīijNā;āāy■āžTērēārĒ XML-RPC æIJ■āLāāžēāĒñāĒSAPIçŽDæŪzāijRæŽt' éIJšāGžæIēāĀĆ
ārzažŌēĒŽçg■æČĒāĒīijNēĀŽāyāLĒāyČāijRāžTçTlčlNāžRāijŽæYřāyĀāyIæŽt' āē;çŽDēĀL'æNl'āĀĆ

XML-RPCçŽDāyĀāyIçijžçČzæYřāōČçŽDæĀgēČ;āĀĆSimpleXMLRPCServer
çŽDāōđçŌræYřā■TçžĒčlNçŽDīijN æL'ĀāžēāōČāy■ēĀĆāRlāžŌād' gādNçlNāžRīijNār;çōqæLŠāzñāIJl1.2ār
āRēād' ŪīijNçTšāžŌ XML-RPC āRĒæL'ĀæIJL'æTřæ■ōēČ;āžRāLŪāNŪāyžXMLæāijāijRīijNæL'ĀāžēāōČāijŽ
ā;ĒæYřāōČāžšæIJL'āijYçČzīijNēĒŽçg■æŪzāijRçŽDçijŪçāAāRřāžēēčnczIād' gēČlāLEāĒūāžŪçijŪçlNēr■ēlĀ
éĀŽēĒGā;ĒçTlēĒŽçg■æŪzāijRīijNāĒūāžŪēr■ēlĀçŽDāōčæLūçnrçlNāžRēČ;ēČ;ēōĒēŪōā;āçŽDæIJ■āLāāĀĆ

ēŽ;çDŪXML-RPCæIJL'ā;Lād' ŽçijžçČzīijNā;ĒæYřāēČædIJā;āēIJĀēēAāĒnéĀšædDāžzāyĀāyIçōĀā■Tē
æIJL'æŪūāĀZīijNçōĀā■TçŽDæŪzæāLāršāūšçžRēūšād' šāžEāĀĆ

13.7 11.7 āIJāy■āRŅçŽDPythonēgčēGŁāZlāžNēŪt'āžd'āžŠ

éŪōēčY

ā;āāIJāy■āRŅçŽDæIJžāZlāyLēIcēĒRēāNçlĀād' ŽāyIPythonēgčēGŁāZlāōđā;NīijNāzūāyNæIJZēČ;ād' šā

æĈædĪJä;äçŽDèğćéĜĹāŽĭēfĤRèaŊāĪJāŖŊäYĀāŖræĪJžāŽĭäYĹēĭćijŊēĈčāZĹä;āāŖŖāzēä;ĤçŤĭāŖēād'Ŭç
 èèAæĈšä;ĤçŤĭUNIXāššāēŬæŌēā■ŬēĭēāĹZāzžäYĀäYĭēfđæŌēijŊāŖĭēĪJĀçōĀ■ŤçŽDārEāĪJŖāĪAæŤzāEZäY

èèAäČšä;čçTíWindowsåS;ǎŘ■çóæAŞæIěǎLZǎzzèfđæŎëijŇǎRlèIJǎǎČRäyNéIcèfZæäüä;čçTlǎyǎÄyǎ

äyÄäyléÄŽčŤlăĜĖăŁZæŸřijŇă;ăäy■ēēAă;£çŤl multiprocessing
 ælĕăôđçŎřăyÄäylărŷăđ'ŮčŽĎăĚňăĚsăIJ■ăŁăăĂĆ Client() ăŠŇ Listener()
 äy■čŽĎ authkey ăŔĆăŤřçŤlălĕēôđ'ērAăŔSĕtűē£đăŎččŽĎčŷŁčŋŕçŤlăLŭăĂĆ
 ăēĆăđIJăŕĚēSĕăy■ăŕŷăijŽăžgçŤŷăyÄäylăijCăyŷăăĂĆă■đ'ăđ' ŮřijŇĕŕĕălăălŮăIJăĚăĂĆăăŔŁçŤlălĕăžžçŋŇĕŤĚ
 ăŷŇăēĆĥijŇăyđ'ăylĕgçĕĜŁăŽlăžŇĕŮŤ'ăŔŕăĹălăŔŎăŕsăijĂăğŇăžžçŋŇĕĚđăŎĕăžŷăăIJăđ'ĐçŔĚă\$ŔăylĕŮŕĕčŸ

æĈĉdĪJä;äēIJÄēēAårzāzTāsĈēfđæŌēāAŽæZt'ād'ŽčŽDæŌgāLūiijNærTāēCēIJÄēēAæTŕæNĀēūĒæŪūā
 ä;äæIJÄāē;ä;ŁĉTlāRēād'ŪčŽDāzSæLŪēÄĒæYŕāIJlénYāsĈsocketäyLælēāōđĈŌrēfZāzŽĈL'zæÄgāĈĈ

éŮőécŸ

äjäæČšålJläyÄäy læúLæAřäijäečŠásČæČ sockets äÄmultiprocessing
connections æLÚ ZeroMQ čŽDšžčAázNäyŁaóđŎřäyÄäyłčŎÄ■TčŽDěfIJclNěfGčlNěřČčTlíijLRPC

èġčǎẸșæŮźæąŁ

āĖĀĞ;æŦřėrũæsĆăĂăĤĈæŦřăŠñēŦăŽđăĬjă;ŁçŦłpickleçijŨçaĀăŘŎiiŇăİJläy■ăŦŇçŽĐēğćéĠăŻ
äyŇÉİcăYřäyĀăvŁcőĀă■ŦčŽĐPRCăd'ĐcŘĖăŽİiiŇăŦřăžēēcŋăŦt'ăŦŦLăŦřäyĀăvŁăIJ■ăLăLăZİäy■ăŦōziiŽ

(continues on next page)

(continued from previous page)

```
func_name, args, kwargs = pickle.loads(connection.  
→recv())  
  
# Run the RPC and send a response  
try:  
    r = self._functions[func_name](*args,**kwargs)  
    connection.send(pickle.dumps(r))  
except Exception as e:  
    connection.send(pickle.dumps(e))  
except EOFError:  
    pass
```

ċeAä;ŁçTłeŁZäyŁad'DçŘEąŻłiijŃä;ăeIĀĊeAārEăoCăŁăăEċăŁrăyĂăyŁeŭŁeAŕăeIĴăŁăŻłäyăĂĂCă;ăe
 ä;EăĲŕă;ŁçTł multiprocessing ăžŞăĲŕăIĀĊoĂăTçŽDăĂCăyNélcăĲŕăyĂăyŁR-
 PCăIĴăăŁăăŻłă;ŃăăŘłiijŽ

```
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')
```

äyžāEāzŌäyÄäyſefIJçÍNāóœLúcñrēóſéUőæI■āŁaāZlījŃā;æeIJÄèeAāŁZāzzāyÄäyīārzāzŦcŽĐçŦlæī

```
import pickle

class RPCProxy:
    def __init__(self, connection):
        self._connection = connection
    def __getattr__(self, name):
```

(continues on next page)

(continued from previous page)

```
def do_rpc(*args, **kwargs):
    self._connection.send(pickle.dumps((name, args,
    ↪kwargs)))
    result = pickle.loads(self._connection.recv())
    if isinstance(result, Exception):
        raise result
    return result
return do_rpc
```

èeAä;fcTlèfZäyläzççRÊçşziiNä;äeIJÄeAärEäEüäNËècEäLräyÄäylæIJ■äLäqäZlçZDëfðæÖëäyLéIcïijN

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

èeAæşlæDRçZDæYřä;Läd'ZæüLæAřäsCïijLærTæC multiprocessing
ïijL'äüşçzRä;fcTïpickleäzRäLÜäNÜäzEäTřæ■öäÄC äeCädIJæYřefZæäüçZDërlïijNärz
pickle.dumps() äŞN pickle.loads() çZDërcçTlèeAäÖzæÖL'äÄC

ëöleöž

RPCHandler äŞN RPCProxy çZDäşzæIJnäÄleüræYřä;LærTë;CçöÄä■TçZDäÄC
äeCädIJäyÄäyläöcæLüçnræCşëeAërççTlâyÄäylèfIJçlNäG;æTřïijNærTæC foo(1, 2,
z=3) ,äzççRÊçşzäLZäzžäyÄäyläNËäRnäzEäG;æTřäR■äŞNäRCæTřçZDäEÇçzD ('foo',
(1, 2), {'z': 3}) äÄC èfZäyläEÇçZDëcñpickleäzRäLÜäNÜäRÖéÄZèfGç;ŞçzIJèfðæÖëäRŞçTşäC
èfZäyÄæ■eäIJl RPCProxy çZD __getattr__() æÜzæşTèfTäZdçZD do_rpc()
éÜ■äNËäy■äöNæLRäÄC æIJ■äLäqäZlæÖæTüäRÖéÄZèfGpickleäR■äzRäLÜäNÜäüLæAřïijNæşæL;äG;æ
æL'gëaŇçzşædIJ(æLÜäijCäyÿ)ëcñpickleäzRäLÜäNÜäRÖèfTäZðäRŞéÄAçzZäöcæLüçnräÄCæL'SäzñçZDäö
multiprocessing èfZëaNeÄZäfaäÄC äy■èfGïijNèfZçg■æÜzäijRäRřäzëeÄCçTlâzÖäEüäzÜäzžä;TæÜL
äZëäZëÄRlèIJÄeAärEëfðæÖëärzësäæ■cæLRäRlèÄCçZDZeroMQçZDsocketärzësä■şäRfäÄC

çTšäzÖäzTäsCéIJÄeAä;IèTÜpickleïijNëCçäzLäöL'äEléÜöecYärséIJÄeAèÄCèZŠäZE
ïijLäZäyžäyÄäylèAłæYÖçZDëzŠäöcäRřäzëäLZäzžçL'zäöZçZDæüLæAřïijNëC;äd'şëöI'äzzæDRäG;æTřéÄZ
äZäæ■d'ä;äæřyèfIJäy■eAäEÄeöyæIèèGtäy■äfažzæLÜäIJlèöd'erAçZDäöcæLüçnrçZDRPCäÄCçL'zäLnä
èfZçg■äRlèC;äIJlæEëClècñä;fcTlïijNä;■äZÖëYşçAñácZäRÖéicäzūäyTäy■eëAärzäd'ÜäZt'èIJšäÄC

ä;IJäyžpickleçZDæZëzçïijNä;äazşëöyäRřäzëèÄCèZŠä;fcTlJSONäÄAXMLæLÜäyÄäzZäEüäzÜçZDç


```

    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)

```

èõìèõž

hmac èõd'èrAçŽDäyÄäylyyègAä;£çTlâIJzæŽræYřâEĚéCíæúLæAřéĂŽăfæçşzçzşăŞÑè£ŽçlNéUř'éĂŽ
ä;NăeĆiijNăeCădIJä;ăcijŪăEŽçŽDçşzçzşæúL'ăRĹăLăRăyĂäyłéŽEç;đ'äy■ăd'Žäyłăd'ĐçŘĚăŽlăzNéUř'çŽDěĂ
ă;ăăRřăžăä;£çTlâIJñèĹCăŪzæăLăĬëçăŏăłlăRlăIJL'ècňăĚAèőyçŽDè£ŽçlNăzNéUř'æL'■č;ă;ijæ■đ'éĂŽăfæ
ăžNăŏđăyĹiijNăšžăžŌ hmac çŽDèõd'èrAècň multiprocessing
æłăălŪă;£çTlâĬăŏđçŌřă■Řè£ŽçlNçŽt'æŌëçŽDěĂŽăfăăĂĆ

è£YæIJL'äyĂçĆzéIJĀèeAăijžerČçŽDæYřè£đæŌèèõd'èrAăŠNăLăăřEæYřăyđ'čăAăžNăĂĆ
èõd'èrAæĹRăĹšăžNăŘŌçŽDěĂŽăfææúLæAřæYřăžăæYŌæŪĜă;čăijRăRŚéĂAçŽDřijNăžžă;ŤăžžăRlèeAæČ

hmacèõd'èrAçŏŪæşŤăšžăžŌăŞĹăyNăĜ;æŤřăeĆMD5ăŠNăSHA-
1řijNăĚşăžŌè£ŽäyłăIJĬETF RFC 2104äy■æIJL'èrëçzEăžNçz■ăĂĆ

13.10 11.10 aJlČ;ŠčzIæIJ■āŁäÿ■āŁăăĚSSL

ěŮóécŸ

ä;ăăČšăóđčŎřăŸĂăŸłăšžăžŎsocketsčŽĐč;ŠčzIæIJ■āŁäÿ■āŁăăĚSSL■āŁăăĚ

ěğčăĚşăŮzăăĹ

ssl æłăăĹŮěČ;ăŸžăžŮăśČsocketěđăŎěăŮzăăĹSSLčŽĐăŮřăŤăăĹăĚ ssl.
wrap_socket() āĜ;ăŮřăŮěăŮŮăŸĂăŸłăăšă■ŸăIJčŽĐsocketă;IJăŸžăŮČăŮřăžŮă;ĤčŮĹSSLăśČăĹăăŤăăĚă
ă;ŮăĚČŮijŮăŸŮéłăăŸřăŸĂăŸłčŮĂă■ŮčŽĐăžŮč■ŮăIJ■āŁăăĹŮijŮěČ;ăIJăIJ■āŁăăĹŮŮăŸžăăĹĂăIJĹăđăăĹ

```
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))
```

äyÑéÍcæĹŚäzñæijŦçd'žäyÄäyĹaóçæĹûçñrè£đæŌëæIJ■āŁaāZÍçŽĐäžd'äzŠäĭNā■ŘãĀCăóçæĹûçñrăijŽèr

```
>>> 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£Žçğ■çŽt'æŌëād'ĐçŘĒāžŦāsĆsocketæŰzâijRæIJĹ'äyĹéŰóécŸārsæŸřaóČäy■èČ;āĭĹăë;çŽĐèũ\$æăĠăČ
äĭNāëČiijNçzĹād'gëČĹăĹæIJ■āŁaāZĹäzççăAġijĹHTTPăĀĀXML-
RPCç■ĹġijĹ'ăôđéŽĒäyĹæŸřăšžăžŌ socketserver äžŞçŽĐăĀĆ
ăóçæĹûçñrăzççăAāIJläyÄäyĹèĭČénŸāsČäyĹăôđçŌřăĀĆæĹŚäzñéIJĀëçAāŘëād'ŰäyĀçğ■çĹ■āĭőäy■āŘNçŽĐ.
éçŰăĒĹġijNăřzăžŌæIJ■āŁaāZĹèĀNĒĹĀġijNăŘrăzççăĀŽè£ĠăČRăyNéÍcè£Žæăũă;£çŦĹäyÄäyĹmixincşzæĹë

```
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
```

äyžăžĒă;£çŦĹè£ŽäyĹmixincşziijNăġăăRřăzçăřĒăôČèũ\$ăĒŰăžŰæIJ■āŁaāZÍçşzæăũăŘĹăĀĆăĭNāëČiijNăy
RPCæIJ■āŁaāZĹăĭNā■ŘġijŽ

```

# 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)

```
kvserve.serve_forever()
```

ä;ŁçŦİēŁŻäyŁæIJ■āŁāāŹİæŮüijŇä;āāŔřäzēä;ŁçŦİæŹōéĀŹçŹĎ xmlrpc.client
æİāāİŮæİēēŁđæŌēāōĈāĀĈ āŔİēIJĀēēAāIJİURLäy■æŇĠāōŹ https: ā■şāŔřijŇä;ŇāēĈijŹ

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

ārzāžŌSSLāōēāŁūçŋŕæİēēōšäyĀäyŁæŕŦē;Ĉād■æİĈçŹĎŮōēēŸæŸŕæĈä;Ŧçāōēōđ'æIJ■āŁāāŹİēŕAäzēæ
äy■āzŷçŹĎæŸŕijŇæŹĈæŮūēŁŸæşāæIJL'äyĀäyŁæāĠāĠEæŮzæşŦæİēēġçāEşēŁŻäyŁæŮōēēŸŕijŇēIJĀēēAēĠ
äy■ēŁĠijŇäyŇēİēççzŹāĠzäyĀäyŁä;Ňā■ŔŕijŇçŦİæİēāzŹçŋŇäyĀäyŁāōL'āĒİçŹĎXML-
RPCēŁđæŌēæİēçāōēōđ'æIJ■āŁāāŹİēŕAäzēŕijŹ

```
from xmlrpc.client import SafeTransport, ServerProxy
import ssl

class VerifyCertSafeTransport(SafeTransport):
    def __init__(self, cafile, certfile=None, keyfile=None):
        SafeTransport.__init__(self)
        self._ssl_context = ssl.SSLContext(ssl.PROTOCOL_TLSv1)
        self._ssl_context.load_verify_locations(cafile)
        if certfile:
            self._ssl_context.load_cert_chain(certfile, keyfile)
        self._ssl_context.verify_mode = ssl.CERT_REQUIRED

    def make_connection(self, host):
        # Items in the passed dictionary are passed as keyword
        # arguments to the http.client.HTTPSConnection()
        ↪constructor.
        # The context argument allows an ssl.SSLContext instance to
        # be passed with information about the SSL configuration
        s = super().make_connection((host, {'context': self._ssl_
        ↪context}))

        return s

# Create the client proxy
```

(continues on next page)

(continued from previous page)

```
s = ServerProxy('https://localhost:15000',
                transport=VerifyCertSafeTransport('server_cert.pem',
                                                    'client_cert.pem',
                                                    'client_key.pem'),
                allow_none=True)
```

æIJ■āLāāZīlāEēfAāzēāRŠéĀAçzZāōcæLūçñfrijNāōcæLūçñfælēçāōēōd'āōCçZDāRLæşTæĀgāĀCēfZçg
āēCædIJæIJ■āLāāZīlāēCşēçAçāōēōd'āōcæLūçñfrijNāRfāzēāfEæIJ■āLāāZīlāRfāLāzççāAāfōæTzāēCāyNūjZ

```
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()
```

äyžāzEēōl'XML-RPCāōcæLūçñfāRŠéĀAēfAāzēijNāfōæTz ServerProxy
çZDāLlāgNāNŪāzççāAāēCāyNūjZ

```
# Create the client proxy
s = ServerProxy('https://localhost:15000',
                transport=VerifyCertSafeTransport('server_cert.pem',
                                                    'client_cert.pem',
                                                    'client_key.pem'),
                allow_none=True)
```

èōlēōž

ērTçlĀāŌžēfRēāNæIJñēLCçZDāzççāAēC;ætNērTā;āçZDçşzçzşēĒç;ōēC;āLZāŠNçRĒēgçSSLāĀC
ārfrēC;æIJĀād'gçZDæNŠæLYæYfæCā;TāyĀæ■ēæ■çZDēŌūāRŪāLlāgNēĒç;ōkeyāĀAēfAāzēāSŅāĒūāzŪ

æLSègçcéGLāyNāLrāzTēIJĀēçAāTēijNāerRāyĀāyISSLēfðæŌēçZLçñfāyĀēLñēC;āijZæIJLāyĀāyIçgAē
ēfZāyIēfAāzēāNĒāRnāzEāĒñēSēāzūāIJlērRāyĀānāēfðæŌēçZDæŪūāĀZēC;āijZāRŠéĀAçzZāfzæŪzāĀC
ārzāzŌāĒnāĒSæIJ■āLāāZīlāijNāōCāzñçZDērAāzēēĀZāyāYēYfēcñæIČāĀAēfAāzēæIJzædDærTāēCVerisignāĀA
āyžāzEçāōēōd'æIJ■āLāāZīlç■;āR■ijNāōcæLūçñfāZdāfIā■YāyĀāz;āNĒāRnāzEāfāzæŌLæIČæIJzædDçZD
āJNāēCijNwebætRēgĀZlāfIā■YāzEāyžēçAçZDēōd'ērAæIJzædDçZDērAāzēijNāzūā;ççTlāōCælēāyžæfRā
ārzæIJnārRēLCçd'zāJNēĀNēlĀijNāRlæYfāyžāzEætNērTijNæLSāznāRfāzēāLZāzžēGtç■;āR■çZDērAāzēij

::

```
bash % openssl req -new -x509 -days 365 -nodes -out server_cert.pem -keyout
server_key.pem
```


ěóľěőž

árzäžŎad' gěČlálĚčlNāžRāŚŸæİēēōsāIJlāy■āRÑēfZčlNāžNéŮt' aijäēĀŠæŮĠāzūæRRēfřčņēāē;āČRæšā
ä;ĚæŸřijNāIJL'æŮūāĀŽāōČæŸřædĎāžžāyĀäyĽāRřæL'řāsTčšžčžšščŽĎā;ĽæIJL'čŤlčŽĎāūēāĚūāĀČä;NāēČ
ä;āāRřäžæIJL'ād'ŽäyĽPythonēğčēĠLāŽlāōđä;NijNāřĚæŮĠāzūæRRēfřčņēāijäēĀŠčžŽāĚūāōČēğčēĠLāŽlāē

send_handle() āŠŇ recv_handle() āĠ;æŤřāRĽēČ;ād' ščŤlāžŎ
multiprocessingēfđæŎēāĀČ ä;ĚčŤlāōČāžñæİēäžčæŽĚčōāēĀščŽĎä;ĚčŤlīijĽāRCēĀČ11.7ēĽČřijĽřijN
ä;NāēČřijNā;āāRřäžēēŎ'æIJ■āĽāāŽlāŠŇāūēä;IJēĀĚāRĎēĠāžčā■ŤčNñčŽĎčlNāžRæİēāRřāĽlāĀČäyNēİčæŸ

```
# 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]))
```

ēfŘēāÑēfZäyĽæIJ■āĽāāŽlīijNāRĽēIJĀēēĀæL'gēāŇ python3 servermp.py /tmp/servconn
15000 řijNāyNēİčæŸřčŽyāžŤčŽĎāūēä;IJēĀĚäžččāĀřijŽ

```
# workermp.py

from multiprocessing.connection import Client
from multiprocessing.reduction import recv_handle
import os
```

(continues on next page)

(continued from previous page)

```
from socket import socket, AF_INET, SOCK_STREAM

def worker(server_address):
    serv = Client(server_address, authkey=b'peekaboo')
    serv.send(os.getpid())
    while True:
        fd = recv_handle(serv)
        print('WORKER: GOT FD', fd)
        with socket(AF_INET, SOCK_STREAM, fileno=fd) as client:
            while True:
                msg = client.recv(1024)
                if not msg:
                    break
                print('WORKER: RECV {!r}'.format(msg))
                client.send(msg)

if __name__ == '__main__':
    import sys
    if len(sys.argv) != 2:
        print('Usage: worker.py server_address', file=sys.stderr)
        raise SystemExit(1)

    worker(sys.argv[1])
```

python3 workerm.py
/tmp/servconn .
sendmsg()

```
# server.py
import socket

import struct

def send_fd(sock, fd):
    '''
    Send a single file descriptor.
    '''
    sock.sendmsg([b'x'],
                  [(socket.SOL_SOCKET, socket.SCM_RIGHTS, struct.
    pack('i', fd))])
    ack = sock.recv(2)
    assert ack == b'OK'

def server(work_address, port):
    # Wait for the worker to connect
    work_serv = socket.socket(socket.AF_UNIX, socket.SOCK_STREAM)
    work_serv.bind(work_address)
    work_serv.listen(1)
```

(continues on next page)

(continued from previous page)

```
worker, addr = work_serv.accept()

# Now run a TCP/IP server and send clients to worker
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, True)
s.bind(('', port))
s.listen(1)
while True:
    client, addr = s.accept()
    print('SERVER: Got connection from', addr)
    send_fd(worker, client.fileno())
    client.close()

if __name__ == '__main__':
    import sys
    if len(sys.argv) != 3:
        print('Usage: server.py server_address port', file=sys.
↳ stderr)
        raise SystemExit(1)

    server(sys.argv[1], int(sys.argv[2]))
```

äyÑéÍcæYřä;ŁçŤlâcŮæŎčā■ŮçŽĎăŭčä;IJèĀĚăóđçŎřijŽ

```
# 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.
calcsz('i')))

    cmsg_level, cmsg_type, cmsg_data = ancdata[0]
    assert cmsg_level == socket.SOL_SOCKET and cmsg_type == socket.
↳ SCM_RIGHTS
    sock.sendall(b'OK')

    return struct.unpack('i', cmsg_data)[0]

def worker(server_address):
    serv = socket.socket(socket.AF_UNIX, socket.SOCK_STREAM)
    serv.connect(server_address)
    while True:
        fd = recv_fd(serv)
        print('WORKER: GOT FD', fd)
```

(continues on next page)

(continued from previous page)

```
with socket.socket(socket.AF_INET, socket.SOCK_STREAM,
    ↪fileno=fd) as client:
    while True:
        msg = client.recv(1024)
        if not msg:
            break
        print('WORKER: RECV {!r}'.format(msg))
        client.send(msg)

if __name__ == '__main__':
    import sys
    if len(sys.argv) != 2:
        print('Usage: worker.py server_address', file=sys.stderr)
        raise SystemExit(1)

    worker(sys.argv[1])
```

Unix Network Programming by W. Richard Stevens (Prentice Hall, 1990). Windows multiprocessing.reduction

13.12 11.12

éúóécŸ

æLŪèĀĒæŸræCædIJä;ŁçTÍăóĈçŽDèrläijŽărzä;ăçŽDçlNăžRăžğçTšăžĀăžLă;śăŠăĀĈ

èğĉăEşæŪzæąŁ

receive äžNăžŭrijNçDŭăRŌècnă;ăăőŽăžLçŽDăŽdèrĈæŪzæşTæLŪăĠ;æTŕæİèăd'DçŘĒăĈă;IJăyžăyĀăylăRŕèĈ;çŽDètuăğNçCžijNăyĀăylăžNăžŭél'săLlçŽDăqEăđŭăRŕèĈ;ăijŽăžèăyĀăylăôđçŌŕăžEă

```
class EventHandler:
    def fileno(self):
        'Return the associated file descriptor'
        raise NotImplemented('must implement')

    def wants_to_receive(self):
        'Return True if receiving is allowed'
        return False

    def handle_receive(self):
```

(continues on next page)

(continued from previous page)

```
'Perform the receive operation'
pass

def wants_to_send(self):
    'Return True if sending is requested'
    return False

def handle_send(self):
    'Send outgoing data'
    pass
```

èŁŻäÿłçşçŻĐăőđăŃăĲăÿžæŔŠăžűècŋăŤăăĚçşzăijăÿŃéİcèŁŻăăũçŻĐăžŃăžűăłçŎŕăÿ■ĲăŻ

```
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()
```

ăžŃăžűăłçŎŕçŻĐăĚşéŤőéŬăŤăĚæŸŕ select() èŕČçŤĲĲăŇăőČăijŻăÿ■ăŮ■è;őèŕcăŮŬăžűăŔŔèŕçŋè
ăĲĲèŕČçŤĲ select() äžŃăŤăĲăĲăžŃăžűăłçŎŕăijŻèŕcéŮőăŤăăĲĲçŻĐăđ'ĐçŔĚăŻĲăĲăĚşăőŹăŞĲăÿĂăÿ
çĐűăŔŎăőČăŕĚçşşăđĲăŤŮăłăŕŔăĲăŻçşŻ select() äĂČçĐűăŔŎ select()
èŁŤăžđăĜĚăđ'ĜăŎăŕŮăŤŮăŔŖăĂăçŻĐăŕžèşăçşĐăŤŔçŻĐăŤŮăłăĂČ
çĐűăŔŎçŻŸăžŤçŻĐ handle_receive() æŤŮ handle_send()
æŮžæşŤŕècŋèğăŔŖăĂČ

çĲŮăĚŻăŤçŤĲŬăŇăŔçŻĐăŮăăĂžĲĲăŇăEventHandler
çŻĐăőđăŃăĲăžècŋăŤăžžăĂČăŤăăČĲĲăŇăŇéİcăŸŕăÿđ'ăÿłçőĂăŤçŻĐăşžăžŎUDPçĲşçşĲăĲăăŤăçŻĐăđ

```
import socket
import time

class UDPServer(EventHandler):
    def __init__(self, address):
        self.sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
        self.sock.bind(address)

    def fileno(self):
        return self.sock.fileno()

    def wants_to_receive(self):
        return True
```

(continues on next page)

(continued from previous page)

```
class UDPTIMEserver(UDPServer):
    def handle_receive(self):
        msg, addr = self.sock.recvfrom(1)
        self.sock.sendto(time.ctime().encode('ascii'), addr)

class UDPEchoServer(UDPServer):
    def handle_receive(self):
        msg, addr = self.sock.recvfrom(8192)
        self.sock.sendto(msg, addr)

if __name__ == '__main__':
    handlers = [ UDPTIMEserver(('', 14000)), UDPEchoServer(('',
↪15000)) ]
    event_loop(handlers)
```

ætNërTēfZæōtāzčċāAīijNërTçİÄāzŌāRēād'ŪāyĀāyPythonēğčēĠāZlēđæŌěāōČīijŽ

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

āōđčŌrāyĀāyITCPæIJ■āŁāāZīāijŽæŽt'āŁāād'■æİĆāyĀçČzīijNāZāāyžærRāyĀāyIāōčæŁūçnréČ;èçAāLī
āyNēİčæŸrāyĀāyITCPāžTç■TāōčæŁūçnrāçNā■RīijŽ

```
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)

```

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;Nā■ŘčŽDāĚšēTōčCzæYřazŎād'ĐčŘEāZlāy■āLŮeālācdāLāāŠNāLāēZd'āōcæLūçnrçŽDæŠ■ā;IJā
 áržærŘäyÄäylēfđæŎëijNäyÄäylæŮřčŽDād'ĐčŘEāZlēcnāLŽāzzāzūāLāāLřāLŮeālāy■āĀĆā;ŠēfđæŎëēcnāĚ
 āēĆādIJā;äēfŘēāNčlNāzŘāzŭērTçlĀçTlTelnetæLŮçszāijijāūēāĚūēfđæŎëijNāōČāijŽārEā;āāRSéĀAçŽDæt

ěóěőž

ăđđēŽĚäyŁæL'ĀæIJL'čŽĎāžNāzúél' sālŁæqEæđūāŌšçŘĚëùšäyŁéİcçŽĎä;Nā■ŘçŽyāuōæŪāāGāāĀĆăōō
ä;ĚæŸřāIJāIJĀæäyāŁČçŽĎēČíāŁĚīijNéČ;āijŽæIJL'äyĀäyĽē;őērčçŽĎä;ŁçŌřæĬēæčĀæšēæt' zālŁsocketīijNā

āžNāzúél' sālĬĬ/OçŽĎäyĀäyĽāŘřēČ;āē;ād'ĎæŸřāōČēČ;ād'ĎçŘĚēİđāyāđ'ğçŽĎāzūāŘŚēŁđæŌēīijNēĀN
āžšāřsæŸřēřt'īijNselect() ēřČçŤīīijLāĬŪāĒūāzŪç■L'æŤĬçŽĎīijL'ēČ;çŽŚāŘñād'ğēĞŘçŽĎsocketāžūāš■
āIJā;ŁçŌřāy■äyĀæñāđ'ĎçŘĚäyĀäyĽāžNāzūīijNāzūāy■ēIJĀēēAāĒūāzŪçŽĎāzūāŘŚæIJzālŪāĀĆ

āžNāzúél' sālĬĬ/OçŽĎçijžçČzæŸřæšqæIJL'çIJšæ■čçŽĎāŘNæ■ēæIJzālŪāĀĆ
āēČāđIJāžzā;ŤāžNāzūāđ'ĎçŘĚāŽīāŪzæšŤēŸzāāđāĬŪāL'gēāNāyĀäyĽēĀŪāŪūēōāçōŪīijNāōČāijŽēŸzāāđ
ērČçŤĬēČcāžZāžūāy■æŸřāžNāzúél' sālĬēčŌāēijçŽĎāžšāG;æŤřāžšāijŽæIJL'ēŪōēčŸīijNāŘNæāūēēAæŸřæš

ārzāžŌēŸzāāđāĬŪēĀŪāŪūēōāçōŪçŽĎēŪōēčŸāŘřāžēēĀŽēŁĠārĚāžNāzūāŘŚēĀĀäyĽāĒūāzŪā■ŤçNñç
äy■ēŁĠīijNāIJāžNāzūā;ŁçŌřāy■āijŤāĒēāđ'ŽçžŁçĬNāšNād'ŽēŁZçĬNæŸřæřŤē;ČæčŸæL'NçŽĎīijN
äyNēİcçŽĎä;Nā■RāēijŤçđ'žāžĚāēČā;Ťā;ŁçŤĬ concurrent.futures
āĬāīŪāĬēāōđçŌīijŽ

```
from concurrent.futures import ThreadPoolExecutor
import os

class ThreadPoolHandler(EventHandler):
    def __init__(self, nworkers):
        if os.name == 'posix':
            self.signal_done_sock, self.done_sock = socket.
↪socketpair()
        else:
            server = socket.socket(socket.AF_INET, socket.SOCK_
↪STREAM)
            server.bind(('127.0.0.1', 0))
            server.listen(1)
            self.signal_done_sock = socket.socket(socket.AF_INET,
                                                    socket.SOCK_
↪STREAM)
            self.signal_done_sock.connect(server.getsockname())
            self.done_sock, _ = server.accept()
            server.close()

            self.pending = []
            self.pool = ThreadPoolExecutor(nworkers)

    def fileno(self):
        return self.done_sock.fileno()

    # Callback that executes when the thread is done
    def _complete(self, callback, r):

        self.pending.append((callback, r.result()))
        self.signal_done_sock.send(b'x')

    # Run a function in a thread pool
```

(continues on next page)

(continued from previous page)

```
def run(self, func, args=(), kwargs={}, *, callback):
    r = self.pool.submit(func, *args, **kwargs)
    r.add_done_callback(lambda r: self._complete(callback, r))

def wants_to_receive(self):
    return True

# Run callback functions of completed work
def handle_receive(self):
    # Invoke all pending callback functions
    for callback, result in self.pending:
        callback(result)
        self.done_sock.recv(1)
    self.pending = []
```

ãĬlãzççãÄäy■ĬijNrun() æŰzæsŦëcñçŦlãĭëârEãũëä;IJæRŘäzd'çzZãZðerČãĜ;æŦræsãĭijNãd'ĐçŘEãõN
ãõðéZĚãũëä;IJëcñæRŘäzd'çzZ ThreadPoolexecutor ãõðä;NãĀĆ
äy■ëĚĜäyÄäyĭëZ;çČzæŦřã■RërČëõaçõŰçzŞædIJãŠNãzNãzũã;ĭçŎřĭijNäyZãZĖëğçãEşãõČĭijNæĬSãznãĬZãz
ã;ŞçZĚĭNæšããõNæĬRãũëä;IJãRŎĭijNãõČãĭjZæĬġëãNçsZãy■çZD _complete()
æŰzæsŦãĀĆ èĚZãylæŰzæsŦãĖ■æŞRãylsocketãylLãĖZãĖĚã■ŰëĬCãzNãĬ■ãĭjZëõšæNÇëĭũçZDãZðerČãĜ;æŦ
fileno() æŰzæsŦëĚŦãZðãRëãd'ŰçZDëCçãylsocketãĀĆ äZãæ■d'ĭijNëĚZãylã■ŰëĬCëcñãĖZãĖĚæŰŰĭijNã
çDũãRŎ handle_receive() æŰzæsŦëcñæĤæt'zãZũãyZæĬĀæĬĬLãzNãĬ■æRŘäzd'çZDãũëä;IJæĬġëãN
ãĭççZ;ëõšĭijNërt'ãZĖëĚãZĬãd'ZëĤdæĬSëĜlãũséČ;æZŦãZĖãĀĆ
äyNëĭcæŦřãyÄäyĭçõĀã■ŦçZDæĬLããZĬĭijNæĭjŦçd'zãZĖãçCã;Ŧã;ĭççZĚĭNæšãĭëãõðçŎřëĀŰæŰũçZDë

```
# 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)
```

ëĤRëãNëĚZãylæĬ■ãLããZĬĭijNçDũãRŎërŦçĭĀçŦlãĖũãõČPythonçĬNãZŦæĭëætŦNërŦãõČĭijZ

```

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;äazTèrèèĈ;âIJläy■âRŇçİŮâRčäy■éĜ■âd'■çŽDæL'gèaŇèŁZäyİçİŇâžRrijŇâžúâyTây■aijŽâ;śâŞ■âLrâĖ
 âũşçzRéYĖërzaõŇâžĖŁZäyĀârRèŁĆrijŇéĈcäzĹä;äazTèrëä;ŁçTİèŁZéĜŇçŽDäzççâAâRŮiijşâzşèöyây
 äy■èŁĜrijŇâéĈæđIJä;äçRĖèğçäžĖâşžæIJñâŌşçRĖrijŇâ;âârşèĈ;çRĖèğçèŁZäžZæaĖæđúæL'Āä;ŁçTİçŽDæây
 ä;IJâyžâržâŽđërĈâĜ;æTŕçijŮçİŇçŽDæŽæžçrijŇâžŇâžúél'śâLİçijŮçâAæIJL'æŮûâĀŽaijŽâ;ŁçTİâLrâ■RçİŇri

13.13 11.13 âRŚéĀAäyŌæŌëæTúâd'gâdNæTŕçzĎ

éŮóécŸ

ä;äèĖAèĀŽèŁĜç;ŚçzIJèŁđæŌëâRŚéĀAâŖŇæŌëâRŮèŁđçz■æTŕæ■óçŽDâd'gâdNæTŕçzĎrijŇâžúâr;éĜR

èğçâĖşæŮzæaĹ

äyŇéİççŽDâĜ;æTŕâLİ'çTİ memoryviews æİèâRŚéĀAâŖŇæŌëâRŮâd'gæTŕçzĎrijŽ

```

# 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:]

```

äyžâžĖætŇèrTçİŇâžRrijŇéçŮâĖĹâLZâžžäyĀäyİèĀŽèŁĜsocketèŁđæŌëçŽDæIJ■âŁaâZİâŖŇâóçæLüçnrç

```

>>> from socket import *
>>> s = socket(AF_INET, SOCK_STREAM)
>>> s.bind(('', 25000))
>>> s.listen(1)
>>> c,a = s.accept()
>>>

```

âIJİâóçæLüçnrrijLâRëad'ŮâyĀäyİèğçéĜLâZİäy■rijL'rijŽ

```
>>> from socket import *
>>> c = socket(AF_INET, SOCK_STREAM)
>>> c.connect(('localhost', 25000))
>>>
```

æIJñèŁĆçŻDçŻDæăĜæŸřă;ăĊĵéĂŽèŁĜèŁđæŌëăĵăëŁŞăŸĂăŸłëŮĖăđ'ġæŤřçŻDăĂĊèŁŻçġæĊĖăĖŁçŻD
array æłăăĭŮăĽŮ numpy æłăăĭŮăĽăĽŁăžžæŤřçŻDĭĵŻ

```
# 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.])
>>>
```

ëőłëőž

ăĬĴăŤřăġăŕĖĖŽĖăđNăĽĖăŸĊăĵŔĕŏăçŏŮăŠNăžşăăNĕŏăçŏŮĊĬNăžŔăŸġĭĵNĕĜĭăŮăĖŽĊĬNăžŔăĽăăŏđĊ
ăŸġĖĤĜĭĵNĕĤĂăŸřă;ăĊăŏăŏđăĊşĖĤŽăăŮăĂŽĭĵNăĵăăŔŕĖĊĵéĬĂĖĖĂăŕĖăĵăĊŽDăŤřăġĖĵăġăĊăĽŔăŌşăġNă
ăĵăăŔŕĖĊĵĖŸéĬĂĖĖĂăŕĖăŤřăġăĽĜăĽ'săĽŔăđ'ŽăŸłăĭŮĭĵNăŽăăŸžăđ'ġĖĊĬăĽăăŠNĊĴşĊĬĴçŽăăĖşĊŽDăĜĶ

ăŸĂĊġăĖŮăşŤăŸřă;ĤĶŤĬăşŔĊġăĬĴăĽăăžŔăĽŮăNŮăŤřăġăăĂăŤăĂăŔŕĖĊ;ăŕĖăăĖŮĵăġăĊăĽŔăŸĂă
ăŸġĖĤĜĭĵNĕĤŽăăŮăĬĂĊžĬăĵŻăĽăăžžăŤřăġĖĊŽDăŸĂăŸłăđ'ăĽŮăĂĊ
ăŕşĊŏŮăĵăăŔăĽŸŕĖŽŮĊĖŌĊŽDăĂŽĖĤŽăžŽĭĵNăĵăĊŽDăžĊĊăĂăĬĂĊžĬĖŸăŸŕăĵăĬĴ'ăđ'ġĖĜŔĊŽDăŕŔăđNă

æIJñèŁĆéĂŽèŁĜăĵĤĬăĖĖăŸĖġăŽĶăşŤĊđ'žăžĖăŸĂăžŽéŤăşŤăşăĬĬăĂĊ
æIJñĕŕĭăŸĬĭĵNăŸĂăŸłăĖĖăŸĖġăŽĶăŕşăŸŕăŸĂăŸłăŮăşăŸăĬĴăŤřçŻDçŻDĖĖĖĊŮăşĊăĂĊăŸăăžĖăžĖăŸŕĖ
ăĖĖăŸĖġăŸĖġăŽĶĖŸĖĊĵăžăŸăăŕNĊŽDăŮăăĵăĵŔĖĵăġăĊăĽŔăŸăăŕNĊşşăđNăĽĖăĖăĤĖŖăŤřăġăăĂĊ
ĖĤŽăŸłăŕşăŸŕăŸNĖĬĖĤŽăŸłĖŕăŕĖĊŽDçŻDĖĊŽDĭĵŻ

```
view = memoryview(arr).cast('B')
```

ăŏĊăŖĖăŕŮăŸĂăŸłăŤřçŻDăŕăžŮăŕĖăăĖŮĵăġăăŸžăŸĂăŸłăŮăġăăŕŮăŮăŮăĤĊçŻDăĖĖăŸĖġăŽĶăăĂă
ăŕŤăĖĊ socket.send() æĽŮ send.recv_into() äĂĊ
ăĬĴăĖĖĖĊŕĭĵNĕĤŽăžžăŮăşŤĖĊĵăđ'şĊŽŕăŖĖăşăĬĬăĬĤŽăŸłăĖĖăŸăNăžăşşăĂĊăĶNăĖĊĭĵNsock.
send() ċŽŕăŖĖăžŖĖăĖĖăŸăăŕşĊŤşăŤřăġĖĖăĂăŸăĖĬĂĖĖĂăđ'ăĽŮăĂĊ send.
recv_into() äĵĤĬĖĤŽăŸłăĖĖăŸăNăžăşşăĬăŸăžăŖĖăŕŮăşăĬĬăŽDĖĶşăĖĖĊĵşăĖşăNăžăĂĊ

ăĽŤăŸNĊŽDăŸĂăŸłĖŽĶĊĊăŕşăŸŕsocketăĜĶăŤŕăŕĖĊ;ăŕĬăşăĬĬăĬĤăĖĖăŖĖăăĂĊ
ĖĂŽăŸŸăĽĖăŏşĭĵNăĽşăžăăŮăŮăĵĤĬăĶăĽăđ'ŽăŸăăŕNĊŽD send() äŠN recv_into()

æIëäijæ; ŞæTt'äylæTřčzDãĀĆ äy■čTlæNĕāfČtījNæfRæñæŞ■ä;IJāRŌtījNĕğEāZ;äijZĕĀZĕfGāRŚĕĀAæLŪ
æŪřčZDĕğEāZ;āRNæāūāzŞæYřāEĒā■YĕĕEçZŪāsCāĀĆāZāæ■d'tījNĕfYæYřæşæIJL'āzzā;TčZDād'■āLŪæŞ
ĕfZĕGNæIJL'äylĕŪĕĕYārsæYřæŌĕāRŪĕĀĒāfĒĕāzāzNāĒLçşĕĕAŞæIJL'ād'ZāřSæTřæ■ĕĕAĕĕnāRŚĕĀ
āzĕä;ŁāōCĕČ;ĕcDāLEĕĒ■āyĀäylæTřčzDæLŪĕĀĒçāōāfIāōCĕČ;ārĒæŌĕāRŪčZDæTřæ■ōæT;āĒĕāyĀäylāūs
āĕCædIJæşāāŁdæşTčşĕĕAŞçZDĕrIījNāRŚĕĀAĕĀĒārsā;ŪāĒLārĒæTřæ■ōād'gārRāRŚĕĀAĕĕGæIĕījNçDūā

14 çññā■AžNçnáīijŽāzūāRŚçijŪčlŇ

ārZāžŌāzūāRŚçijŪčlŇ, PythonæIJL'ād'Žçğ■éTĒæIJŞæTřæNĀçZDæŪzæşT,
āNĒæNñād'ŽçžĒçlŇ, ĕřČçTlā■ŘĕfZçlŇ, āzĕāRĒāRĎçğ■āRĎæāūçZDāĒşāžŌçTşĕLĒāZlāG;æTřčZDæLĀāū
ĕfZāyĀçnāārĒāijŽçžZāGzāzūāRŚçijŪčlŇāRĎçğ■æŪzĕlççZDæLĀāūğ,
āNĒæNñĕĀŽçTlçZDād'ŽçžĒçlŇæLĀæIJřæzĕāRĒāzūĕāNĕōāçŌŪçZDāōđçŌræŪzæşT.

āČRçzŘĕlNäyřārNçZDçlŇNāžRāSŸæL'ĀçşĕĕAŞçZDĕČĕæū,
ād'gāōūæNĒāfČāzūāRŚçZDçlŇNāžRæIJL'æ;IJāIJlçZDā■śéZl'. āZāæ■d',
æIJñçnāçZDäyžĕĕAçZōæāGāzNäyĀæYřçžZāGzæZt'āŁāāRřāĕāĕŮāSŇæYşĕřČĕřTčZDāzççāĀ.

Contents:

14.1 12.1 āRřāŁlāyŌāAlJæ■ćçžĒçlŇ

éŪĕĕY

ä;āĕĕAäyžĕIJĀĕĕAāzūāRŚæL'gĕāNçZDāzççāĀāŁZāzž/ĕTĀæřAçžĒçlŇ

ĕğcāEşæŪzæāŁ

threading āžŞāRřæzĕāIJlā■TçNñçZDçžĒçlŇNäy■æL'gĕāNāzzā;TčZDāIJl
Python äy■āRřæzĕĕřČçTlçZDāržĕşāāĀĆā;āāRřæzĕāŁZāzžāyĀäyl Thread
āržĕşāāzūārĒā;āĕĕAæL'gĕāNçZDāržĕşāāzĕ target āRČæTřčZDā;ĕāijRæRŘä;ŽçžZĕĕāržĕşāāĀĆ
äyNĕlĕæYřāyĀäylçŌĀā■TčZDä;Nā■RīijZ

```
# 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()
```

ä;Šä;äälZäzäæ;äyÄäyŁçžŁłNärzèsqāRŌiijNèrēāržèsqāzūāy■äijŽčnNā■šaeL'gēāNiiJNēZd'ēIdä;äerČčTlā
 start() æŪzæšTiiJLā;Šä;äerČčTlā start() æŪzæšTæŪiijNāōČaijŽerČčTlā;äaijæéĀšèŁæİēçŽDāG;æT
 POSIX çžŁłNāLŪèĀĒäyÄäyŁ Windows çžŁłNiiJLiiJNèŁZāžŽçžŁłNārEçTšaeŠ■ä;IjçšçžšæİēāĒlāİČčōaçŁ

```

if t.is_alive():
    print('Still running')
else:
    print('Completed')
    
```

ä;äazšāRfäzēārEäyÄäyŁçžŁłNāLāāĒēāLrā;ŠāL■çžŁłNiiJNāzūç■L'ā;ĒāōČčZLæ■ciijŽ

```
t.join()
```

PythonēgčēĠLāŽlçŽt'ālRæL'ĀæIJL'çžŁłNéČ;çžŁLæ■cāL'■äz■äflāNĀēŁRēāNāĀČāržāžŌēIJĀēçAēTŁæ
 ä;NāēČiiJŽ

```

t = Thread(target=countdown, args=(10,), daemon=True)
t.start()
    
```

āRŌāRrçžŁłNæŪæšTç■L'ā;ĒiijNāy■ēŁGriiJNēŁZāžŽçžŁłNaijŽāIJlāyžçžŁłNçZLæ■cæŪūēĠlāLéTĀ
 éZd'āžEāēČäyLæL'Āçd'žçŽDäyŁ'äyŁæŠ■ä;IiijNāzūæšqæIJL'ād'lād'ŽāRfäzēāržçžŁłNāAžçŽDāžNæČĒāĀČ

```

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)
    
```

æēČædIjçžŁłNæL'gēāNäyĀāžŽāČRl/OēŁZæūçŽDēŸzāqdaēŠ■ä;IiijNéČčāZLēĀŽēŁGē;ōērčæİēçZLæ■
 ä;Nā■RāēČäyNiiJŽ

```

class IOTask:
    def terminate(self):
        self._running = False

    def run(self, sock):
        # sock is a socket
        sock.settimeout(5)          # Set timeout period
    
```

(continues on next page)

(continued from previous page)

```
while self._running:
    # Perform a blocking I/O operation w/ timeout
    try:
        data = sock.recv(8192)
        break
    except socket.timeout:
        continue
    # Continued processing
    ...
# Terminated
return
```

èóíëõž

çŤšăžŎăĖlăšĂegćéĠĖŤĀiijĠGILiijL'çŽĐăŎšăŽăiijŊPython
çŽĐçžĤċlNěcńéŽŔăĹŭăĹŕăŔŇăŷĂæŮŭăĹzăŔlăĖĀèőŷăŷĂăŷłçžĤċlNăL'ğèąŇèĤŽæăŭăŷĂăŷłæL'ğèąŇăĹăđŇ
çŽĐçžĤċlNăŽŦ' éĂĈçŦlăžŎăđ'ĐçŔĖl/OăŠŇăĖŮăžŮĖĤĤĂèçĂăžŭăŔŠăL'ğèąŇçŽĐéŸzăăđăš■ă;IJiijĹăŦĤăçŦ
æIJL'æŮŭă;ăăiijŽçIJŇăĹŕăŷŇè;žèĤŽçğ■ĂŽèĤĠçžğæL'Ĥ Thread
çšzæĹăôđçŎŕçŽĐçžĤċlNăiijŽ

```
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()
```

ăŕ;çőăĤŽæăŭăžšăŔŕăžèăŭčă;IJiijŇă;ĖèĤŽă;Ĥă;Ůă;ăçŽĐăžčçăĂă;ĹèŦŮăžŎ
threading äžŠiijŇăL'Ăăžèă;ăçŽĐèĤŽăžŽăžčçăĂăŔĹèĈ;ăĤĤçžĤċlNăŷĹăŷŇăŮĠăŷ■ă;ĤçŦlăĂĈăŷĹăŮĠăŷ
threading äžŠăŮăăĖšçŽĐiijŇèĤŽæăŭăŕšă;Ĥă;ŮèĤŽăžŽăžčçăĂăŔŕăžèèćŇŦlăIJlăĖŮăžŮçŽĐăŷĹăŷŇăŮĠ
multiprocessing æĹăăĹŮăIJlăŷĂăŷłă■ŦçŇŇçŽĐèĤŽċlNăŷ■ăL'ğèąŇă;ăçŽĐăžčçăĂăiijŽ

```
import multiprocessing
c = CountdownTask(5)
p = multiprocessing.Process(target=c.run)
p.start()
```

ăĖ■ăŋăéĠ■ŤšŷiijŇèĤŽæőŧăžčçăĂăžĖĂĈçŦlăžŎ

CountdownTask

çszæYřäzëçNñçñNäzŌåöðéZĚçZĎázüâRŠæL'Næøt̃ijLād'ŽçžŁçłNăĀĀad'ŽèŁZçłNç■Lç■L̃ijL'åöðçŌřçZĎæ

14.2 12.2 âLd'æŮ■çžŁçłNæYřäRĕaũşçžRăRřăĹ

éŮöécY

ä;ăăũşçžRăRřăĹläžEäyÄäyŁçžŁçłÑijNă;EæYřä;ăæÇşşşéAŞăöÇæYřäy■æYřçIJşçZĎaũşçžRăijĂăğNèŁ

èğçăEşşæŮzæaĹ

çžŁçłNçZĎäyÄäyĹăĚşéTōçL'zæĂğæYřæfRäyŁçžŁçłNéČ;æYřçNñçñNèŁRĕaŃäyTçŁüæĀăy■ăRřécĎæp
threading âžŞäy■çZĎ Event âřzèşăăĂĆ Event âřzèşăăNĚăRňäyÄäyĹăRřçTşçžŁçłNèöŁç;öçZĎăĹăRũæ
âřzèşăy■çZĎăĹăRũæăĠăŮèçnéöŁç;öäyžăĂĠăĂĆăÇăĎIJăIJL'çžŁçłNç■L'ă;ĚäyÄäyĹ
event âřzèşăijNĚăNèŁZäyĹ event âřzèşăçZĎăăĠăŮäyžăĂĠijNéČčázĹèŁZäyŁçžŁçłNăřEăijZèçñäyĂçZt' éYz
event âřzèşăçZĎăĹăRũæăĠăŮèöŁç;öäyžçIJş̃ijNăöČăřEăT'd' éĚŞæL'ĂæIJL'ç■L'ă;ĚèŁZäyĹ
event âřzèşăçZĎçžŁçłNăĂĆăÇăĎIJăyÄäyŁçžŁçłNç■L'ă;ĚäyÄäyĹăũşçžRèçnéöŁç;öäyžçIJşçZĎ
event âřzèşăijNéČčázĹăöČăřEăĹç;TĕèŁZäyĹăžNăžüijNçžğçž■æL'ğĕăNăĂĆ
äyNèŁçžZĎäzççăĂăşTçd'žăžEăçCă;Tă;ŁçTĹ Event æĹă■RĕřČçžŁçłNçZĎăRřăĹĹijZ

```
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')
```

ă;Şă;ăæL'ğĕăNèŁZăöğăžççăĂijNăĂIJcountdown is runningăĂĹ æĂzæYřæYçd'žăIJĹ
ăĂIJcountdown startingăĂĹ äžNăRŌæYçd'žăĂÇèŁZæYřçTşăžŌă;ŁçTĹ
event æĹă■RĕřČçžŁçłÑijNă;Ĺă;ŮäyžçžŁçłNèèAç■L'ăĹř countdown ()
ăĠă;TřèŁ;ŞăĠzăRřăĹăĹăæĂřăRŌijNăL'■èČ;çžğçž■æL'ğĕăNăĂĆ

èóíèőž

event áržèsáæIJĀāē;ā■Tæñāā;fçTlíijNārsæYřèrt'ijNā;āāLZāzzāyĀāył event
áržèsāijNēól'æ\$RāyłçžfçlNç■L'ā;ĒēfZāyłáržèsāijNāyĀæUēēfZāyłáržèsāēēñēō;ç;ōāyžçIJ\$iiijNā;āārsāžTèrē
clear() æŰzæşTælēēG■ç;ō event áržèsāijNā;EæYřā;LéZ;çāōāfIāōL'āĒlāIJræyĒçRĒ
event áržèsāāzūāržāōČēG■æŰřetNāĀijaĀCā;LāRřēČ;āijZāRSçTšēTZeĒGāžNāzūāĀAæ■zéTAæLŰēĀĒāĒūā
event áržèsāçŽDāzčçāAāijŽāIJłžfçlNāE■æñāç■L'ā;ĒēfZāył event
áržèsāāzNāL'■æL'gēāNiiijL'āĀCāçCādIJāyĀāyłçžfçlNéIJĀēçAāy■āAIIāIJrēG■ād'■ā;fçTł
event áržèsāijNā;āæIJĀāē;ā;fçTł Condition áržèsāæIēāzčæŽēāĀCāyNéIççŽDāzčçāAā;fçTł
Condition áržèsāāōđçŌřāžEāyĀāyłāSłæIJšāōŽæŰūāŽlíijNāerRā;ŠāōŽæŰūāŽlèūĒæŰūçŽDæŰūāĀŽiiijNā

```
import threading
import time

class PeriodicTimer:
    def __init__(self, interval):
        self._interval = interval
        self._flag = 0
        self._cv = threading.Condition()

    def start(self):
        t = threading.Thread(target=self.run)
        t.daemon = True

        t.start()

    def run(self):
        '''
        Run the timer and notify waiting threads after each interval
        '''
        while True:
            time.sleep(self._interval)
            with self._cv:
                self._flag ^= 1
                self._cv.notify_all()

    def wait_for_tick(self):
        '''
        Wait for the next tick of the timer
        '''
        with self._cv:
            last_flag = self._flag
            while last_flag == self._flag:
                self._cv.wait()

# Example use of the timer
ptimer = PeriodicTimer(5)
ptimer.start()

# Two threads that synchronize on the timer
```

(continues on next page)

(continued from previous page)

```
def countdown(nticks):
    while nticks > 0:
        ptimer.wait_for_tick()
        print('T-minus', nticks)
        nticks -= 1

def countup(last):
    n = 0
    while n < last:
        ptimer.wait_for_tick()
        print('Counting', n)
        n += 1

threading.Thread(target=countdown, args=(10,)).start()
threading.Thread(target=countup, args=(5,)).start()
```

eventárzèsaqŽDäyÄäyléG■èeAçL'žćĆzæYřa;ŠaǒČěcñèőç;ioäyžçIJšæUüaijŽaTd'éEŠæL'ÄæIJLç■L'ā;Ě
Condition árzèsaqælēæŽēāzčāĀČèĀČèŽSäyÄäyNèfZæotä;fcTlāfaāRüéGRāodčŎřçŽDäzčçāAiijŽ

```
# 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()
```

èfRëaŇäyLèçžŽDäzčçāAārEäijŽaRřaLläyÄäylçžfcłNæsāiijŇä;EæYřázúæšaqæIJL'äzĀäzLäžNæČĚaRS

```
>>> sema.release()
Working 0
>>> sema.release()
Working 1
>>>
```

çijŮaEŽæūL'āRLāLřād'gēGRçŽDçžfcłNéŮt'āRŇæ■ēēŮőécYçŽDäzčçāAäijŽeol'ā;āçŮZäy■æñšçTšāĀC

14.3 12.3 çŹŒćÍNéŮťéĂŽăĒą

éŮóécŸ

äĳăçŽĐćÍNăžŘăy■æIJL'ăd'ŽăyłçžŹŒćÍNġijŃăĳăéIJĂèĕAăIJlèŹăžŽçžŹŒćÍNăžNéŮťăôL'ăĚlăIJřăžd'æ■ćăĒą

èğĉăĒşæŮžæąĹ

ăžŮăyĂăyłçžŹŒćÍNăŘŝăŘĕăyĂăyłçžŹŒćÍNăŘŝéĂAæŤřæ■óæIJĂăôL'ăĚlçŽĐăŮžăijŘăŘřĕČĳăřŝăŸřăĳçŤl
queue äžŞăy■çŽĐĕŸşăĹŮăžĒăĂĆăĹŽăžžăyĂăyłĕćnăd'ŽăyłçžŹŒćÍNăĚŝăžnçŽĐ
Queue äřžĕŝăġijNĕŹăžŽçžŹŒćÍNéĂŽĕŹĜăĳçŤl put() äŞŃ get()
æŞ■ăĳIJăĚăŘŝéŸşăĹŮăy■æŮžăĹăăĹŮĕĂĚăĹăéŽd'ăĚČĉťăăăĂĆăĹNăĕĆġijŽ

```
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 äřžĕŝăăŮşçžŘăNĚăŘnăžĒăŹĒĕĕĂçŽĐĕŤġijŃăĹĂăžĕăĳăăŘřăžĕĕĂŽĕŹĜăôČăIJĹăd'ŽăyłçžŹŒćÍNé
ăĳŞăĳçŤlĕŸşăĹŮăŮġijŃă■ŘĕřČĉŤŝăžĝĕĂĚăŞŃăŮĹĕťžĕĂĚçŽĐăĒşĕŮ■éŮóécŸăŘřĕČĳăĳijŽăIJL'ăyĂăžŽĕ

```
from queue import Queue
from threading import Thread

# Object that signals shutdown
_sentinel = object()

# A thread that produces data
```

(continues on next page)

(continued from previous page)

```
def producer(out_q):
    while running:
        # Produce some data
        ...
        out_q.put(data)

    # Put the sentinel on the queue to indicate completion
    out_q.put(_sentinel)

# A thread that consumes data
def consumer(in_q):
    while True:
        # Get some data
        data = in_q.get()

        # Check for termination
        if data is _sentinel:
            in_q.put(_sentinel)
            break

        # Process the data
        ...
```

æIJnäḲNäy■æIJL'äyÄäyḲL'zæōŁçŽDāIJræŪzījŽæūLèt'zèĀĒāIJlērzaĹrèŁŽäyḲL'zæōŁāĀijazNāŔŌçnN
ārḲçōæYšāLŪæYræIJĀāyḡēgAçŽDçžfçlNéŪt'éĀŽāfææIJzāLŪījNā;EæYrāz■çDūāŔrāzèèGĹāūséĀŽèŁGāLZ
Condition āŔYéGRæĪēāNĒèçĒä;āçŽDæTŕæ■ōçzŠædDāĀCäyNè;zèŁŽäyḲāNā■ŔæijTçd'zāžEāçCä;TāLZ

```
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;ŁçTĪéYšāLŪæĪèèŁZèaŊçžfçlNéŪt'éĀŽāfææYrāyĀäyĹā■TāŔSāĀĀäy■çāōāōŽçŽDèŁGçlNāĀCéĀŽāy
task_done() āŠN join() īijŽ

```

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()

```

æĈæđĬäÿÄäÿłçžĚċĬNéĬJĀēēAāĬJläÿÄäÿläÄĬJæūĹet'zèÄĚâĬĬçžĚċĬNâd'DçŘEāōNçL'zâōŽçŽDæTṛæ■ō
 Event æṬ;āĹrāÿÄètūā;ĚçṬlījNēĚZæūāÄĬJçṬšäzgeÄĚâĬĬārsāRrāzēēÄŽēĚGēĚZäÿĬEventāržēsæĬēçŽSæṭ

```

from queue import Queue
from threading import Thread, Event

# A thread that produces data
def producer(out_q):
    while running:
        # Produce some data
        ...
        # Make an (data, event) pair and hand it to the consumer
        evt = Event()
        out_q.put((data, evt))
        ...
        # Wait for the consumer to process the item
        evt.wait()

# A thread that consumes data

```

(continues on next page)

(continued from previous page)

```
def consumer(in_q):
    while True:
        # Get some data
        data, evt = in_q.get()
        # Process the data
        ...
        # Indicate completion
        evt.set()
```

ěóľěőž

ášžāžŎčŏĀ■TéYšāLŮčijŮāEZād'ŽčžčlNčlNāžRāIJlād'ŽæTṛæČĚāEṭāyNæYṛāyĀäylæfTè;ČæYŎæZā
ä;£çTlčžčlNéYšāLŮæIJLäyĀäylèçAæšlæĎRçŽĎēŮŏécYæYṛijNāRŠéYšāLŮäy■æŭzāLāæTṛæ■ŏéqzæŮŭā

```
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 áržèsæēRŘä;ŽäyĀäžŽāIJlā;ŠāL■äyLäyNæŮĜā;LæIJLçTlčŽĎēŽĎāLāçL'zæĀgāĀČæfTæČāIJ
Queue áržèsæēŮŭæRŘä;ŽāRréĀLçŽĎsizeāRCæTṛælēčŽRāLŮāRřāžēæŭzāLāāLréYšāLŮäy■çŽĎāĚČčt'ā
āĀIJæŭLèt'zāĀlçŽĎēĀšāžēāfñijNéČčāžLā;£çTlāŽžāŏžād'gārRçŽĎēYšāLŮārsāRřāžēāIJléYšāLŮāŭšæzæç
get()āšNput()æŮžæšTéČ;æTṛæNĀēldeYžāqđæŮžāijRāšNēŏ;āŏžēŭĒæŮŭijNā;NāēČijŽ

```
import queue
q = queue.Queue()

try:
    data = q.get(block=False)
except queue.Empty:
    ...

try:
```

(continues on next page)

(continued from previous page)

```
q.put(item, block=False)
except queue.Full:
    ...

try:
    data = q.get(timeout=5.0)
except queue.Empty:
    ...
```

put() æÚæşŦåŠNäyÄäyİaŽžāōŽād' gārRçŽĐéYşāLŪäyÄëtüä;çŦİijNèfZæūā;ŞéYşāLŪāūsæzæŪūârş.

```
def producer(q):
    ...
    try:
        q.put(item, block=False)
    except queue.Full:
        log.warning('queued item %r discarded!', item)
```

q.get() æŦæāūçŽDæŞā;IæŪİijNèüEæŪüèGİaLİçzLæ■cāzēä;ŁæçĀæşēcžLæ■cæāGāfŪİijNā;āāžTèrēā;çŦİ
q.get() çŽĐârRéĀL'ârCæŦř timeout İijNāçCäyNİijŽ

```
_running = True

def consumer(q):
    while _running:
        try:
            item = q.get(timeout=5.0)
            # Process item
            ...
        except queue.Empty:
            pass
```

q.qsize() İijN q.full() İijN q.empty()
ç■L'āōđŦİæŪæşŦåRřazēēŌūârŪäyÄäyİéYşāLŪçŽDā;ŞāL'■ād' gārRāŠNçLúæĀAāĀĆā;EēçAæşİæDŦİijN
empty() āLd'æŪ■āGžèçŽäyİéYşāLŪäyžç' žİijNā;EāRŦæŪūârçād' ŪäyÄäyİçžçİNāRèÇ;āūsçzRāŦŞçfZæ

14.4 12.4 çzŽāĖŞéŦōéČİāLĚāŁæéŦĀ

éŬōécŸ

ä;äçIĀèçAāržād' ŽçžçİNçİNāžRāy■çŽDäyŦçŦNāNžāŁæéŦĀäzēēAŁāĖ■çndāžL'æİāžūāĀĆ

èġċàEşæŮzæąŁ

èċAąIJład'ŽçžŁċłŃċłŃăžŔăy■ăôŁ'ăĖłă;ŁċŤłăŔŕăŔŸăŕžèşąijŃă;ăĖIJăċċAă;ŁċŤł thread-
ing âžŞăy■ċŹĐ Lock âŕžèşąijŃăŕşăĈŔăyŃĕ;žĖŁăŸłă;Ńă■ŔĖŁăăŭijŽ

```
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 âŕžèşăăŞŃ with ċŕ■ăŔĕăİŮăŷĂĕŧăă;ŁċŤłăŔŕăžĕăĖĬĕŕAăžŞăŮĕăŁ'ġĕăŃijŃăŕşăŸŕăŕŔăŋăăŔłăĖ
with ċŕ■ăŔĕăŃĖăŔŋċŹĐăžċċăAăİŮăĂĈwith ċŕ■ăŔĕăijŽăIJĬĖŁăŸłăžċċăAăİŮăŁ'ġĕăŃăŁ'■ĖĠăŁĬĖŮăăŔŮĖŤ

ėóĬėōž

ċžŁċłŃĖŕĈăžĕăIJŋĕŧ'łăŷŁăŸŕăŷ■ċăôăôŽċŹĐŮijŃăŽăă■d'ijŃăIJład'ŽçžŁċłŃċłŃăžŔăy■ĖŤŽĕŕŕăIJŕă;ŁċŤł
ăIJłăŷĂăžŽăĂIJĖĂAċŹĐăĬ Python âžċċăAăŷ■ijŃăŸĬăijŔĖŮăăŔŮăŖăŃĖĠăŁăŤĬĖŤAăŸŕăĬŁăŷŷĖġAċŹĐăĂ

```
import threading

class SharedCounter:
    '''
    A counter object that can be shared by multiple threads.
    '''
    def __init__(self, initial_value = 0):
        self._value = initial_value
        self._value_lock = threading.Lock()

    def incr(self, delta=1):
        '''
```

(continues on next page)

(continued from previous page)

```

    Increment the counter with locking
    '''
    self._value_lock.acquire()
    self._value += delta
    self._value_lock.release()

def decr(self, delta=1):
    '''
    Decrement the counter with locking
    '''
    self._value_lock.acquire()
    self._value -= delta
    self._value_lock.release()

```

çZÿærTāzŌēfZçg■æYĭ;āijRèrCçTĭlçZDæŪzæsTĭijNwith èr■āRēæZt'āŁāaijYēZĒijNāzšæZt'äy■āózáYš
release() æŪzæsTæLŪēĀĒçlNāzRāIĭlēŌūā;ŪēTāzNāRŌāzgcTšāijCāyēfZāyd'çg■æCĒāEĭijLā;fçTĭ
with èr■āRēāRfāzēāfĭērAāIĭlēfZāyd'çg■æCĒāEĭāyNāz■ēC;æ■ççāōēGLæTĭ;ēTĀijL'āĀC
äyžāzEēAŁāĒ■āGzçŌræ■zēTāçZDæCĒāEĭijNā;fçTĭlēTāæIJzāŁūçZDçlNāzRāzTērēēō;āóZāyžæfRāyŁçzŁçl
āIĭI threading āzšāy■ēfYæRŖā;ZāzEāĒūāzŪçZDāRŖNæ■ēāŌšēr■ijNærTāēC RLock
āŠN Semaphore āržēšāāĀCā;EāYræāzæ■ōāzēā;ĀçzRēlNĭijNēfZāzZāŌšēr■æYrçTĭlāzŌāyĀāzZçL'zæŌŁçZ
RLock ĭijLāRfēĠ■āĒēēTĀijL'āRfāzēēcñāRŖNāyĀāyŁçzŁçlNād'ZānāēŌūāRŪĭijNāyžēēAçTĭlāēāōđçŌrāšzāz
SharedCounter çšĭijZ

```
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)
```

aIJayLeZzeZaylaNaRaYrijNasqaIJL'arzaerRayAaylaodaNaYcZDaRraRYarzesqaLaetArijNaRUe.
 decr æUzasTaAC eZcgaodcOraUzaijRcZDäyÄäytcL'zcCzaYrijNäUäeoZzeZaylcSzaIJL'ad'ZarSäylooda.

äfaaRüéGRärfzèsaaëYräyÄäyläzzçñNäIJläĖsäžñèóaaëTřãZlãšžçãÄäyŁçŽDãRÑæ■ěãŮšër■ãĀĆãęCæđIJeóaaë
ër■ãRëärĖëóaaëTřãZlãGRlīijŇçžĚćlNèćnáĚAèöyæL'gèaÑãĀĆwith
ër■ãRëæL'gèaŇçžŠæĪšãRŮīijŇèóaaëTřãZlãĹāīijŠãĀĆãęCæđIJeóaaëTřãZlãyžŮīijŇçžĚćlNãĖĖćnéYzãađīijŇçž

```
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)
```

ãęCæđIJä;äärzçžĚćlNãRÑæ■ěãŮšër■çŽDãžTãšĆçRĖëóžãŠNãóđçŮřæĎšãĖT'èüćīijNãRřäzëãRĆèĀĆæŠ

14.5 12.5 éYšæ■cæ■zéTĀçŽDãĹæéTĀæIJžãĹŮ

éUóécY

ä;äæ■cãIJläĖZäyÄäyläđ'ŽçžĚćlNćlNãžRīijNãĖŮäy■çžĚćlNéIJĀęęÄäyĀæñæèŮãRŮĹäđ'ŽäyĹéTĀīijNæc

èğcãĖşæ■zéTĀéUóécYçŽDäyĀçg■æŮzæaĹ

ãIJläđ'ŽçžĚćlNćlNãžRäy■īijNæ■zéTĀéUóécYäĹLäđ'gäyĀéĆlãĹEæYřçŤsäžŮçžĚćlNãRÑæŮüëŮüãRŮĹä
æŮüãĀŽãRŠçŤšëYzãađīijŇëĆcãžĹëĖŽäyŁçžĚćlNãřšãRřèĈ;éYzãađãĖŮäzŮçžĚćlNçŽDæL'gèaŇīijNãžŮèĀŇã
èğcãĖşæ■zéTĀéUóécYçŽDäyĀçg■æŮzæaĹæYräyžćlNãžRäy■çŽDærRäyÄäyĹéTĀãĹEéĖ■äyÄäylãTřäyĀçŽ
æYřéĹäyÿãóžæYšãóđçŮřçŽDīijŇçđ'žäĹNãęCäyŇīijŽ

```
import threading
from contextlib import contextmanager

# Thread-local state to stored information on locks already acquired
_local = threading.local()

@contextmanager
def acquire(*locks):
    # Sort locks by object identifier
    locks = sorted(locks, key=lambda x: id(x))

    # Make sure lock order of previously acquired locks is not
    ↪violated
    acquired = getattr(_local, 'acquired', [])
    if acquired and max(id(lock) for lock in acquired) >=
    ↪id(locks[0]):
        raise RuntimeError('Lock Order Violation')
```

(continues on next page)

(continued from previous page)

```
# Acquire all of the locks
acquired.extend(locks)
_local.acquired = acquired

try:
    for lock in locks:
        lock.acquire()
    yield
finally:
    # Release locks in reverse order of acquisition
    for lock in reversed(locks):
        lock.release()
    del acquired[-len(locks):]
```

acquire() *acquire all of the locks*

```
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()
```

acquire all of the locks

```
import threading
x_lock = threading.Lock()
y_lock = threading.Lock()

def thread_1():
```

(continues on next page)

(continued from previous page)

```
while True:
    with acquire(x_lock):
        with acquire(y_lock):
            print('Thread-1')

def thread_2():
    while True:
        with acquire(y_lock):
            with acquire(x_lock):
                print('Thread-2')

t1 = threading.Thread(target=thread_1)
t1.daemon = True
t1.start()

t2 = threading.Thread(target=thread_2)
t2.daemon = True
t2.start()
```

æĆædIJä;æefRëqÑefZäylçL'ŁæIJñçŽDäzčçäAijjÑäfĚäöŽäijZæIJL'äyÄäylçzŁçłNäRŚçTšät'ſæžČijÑä

```
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
>>>
```

äRŚçTšät'ſæžČçŽDäŎšäZäaIJläžŎijjÑæfRäylçzŁçłNéČ;èöřä;TçĬÄeĜläušäušçzRèŎüäRŮäŁřçŽDěTÄã
acquire() äĜ;æTřäijZæčÄæšëäzNäL'■äušçzRèŎüäRŮçŽDěTÄäLŮëäłijjÑ
çTšäzŎéTÄæYřæNL'çĚğä■ĜäžRæŎšäLŮëŎüäRŮçŽDrijjÑæL'ÄäžëäĜ;æTřäijZeöd'äyžäzNäL'■äüşëŎüäRŮç

èőlèőž

æ■zéTÄæYřæfRäyÄäyläd'ŽçzŁçłNçłNäžRéČ;äijŽélcäyt'çŽDäyÄäyléŮőécYñijŁäřsäČRäöČæYřæfRäyÄ
çzŁçłNäRĬč;äRŊæŮüäŁæĬäŊäyÄäyléTÄrijjÑefZæäüçłNäžRäřsäy■äijŽećnä■zéTÄéŮőécYæL'ÄäžRæL'řäÄ
æ■zéTÄçŽDæčÄætNäyŎæAčäd'■æYřäyÄäyläĜäžŎšææIJL'äijYéŽĚçŽDèğčäĚşæŮzæäŁçŽDæL'ſäT

è£RèaŃçŽDæŮûāĀŽaijZæfRéŽTäyĀæōtæŮúéŮt' éĜ■ç;ōèōæTřāŽlíijŃāIJlæšææIJL' āRŚçTšæ■zéTAcŽDæČ
èúĚæŮúíijŃè£ŽæŮúçlŃāžRaijŽéĀŽè£ĜéĜ■āRřèĜlèžnæAçāđ' ■āLřæ■čāyçŁúæĀAāĀĆ

éA£āĚ■æ■zéTAcYřāRēad' ŮäyĀçg■èĝčāEšæ■zéTAcŮóécYçŽDæŮžaijRíijŃāIJlè£ŽçlŃèŮūāRŮéTAcŽ
æ■zéTAcŁúæĀAāĀĆèfAæYŌārščTŽçzŽèfzèĀĚā;IJäyžçzČāžāžEāĀĆéA£āĚ■æ■zéTAcŽDäyžèeAæĀlæČšæ
æ■zéTAcŽDäyĀäyġā£ĚèeAæġāžūíijŃāžŌèĀŃéA£āĚ■çlŃāžRè£ŽāĚēæ■zéTAcŁúæĀAāĀĆ

äyŃéġcāžēäyĀäyġāĚšāžŌçž£çlŃæ■zéTAcŽDçzRāĚyēŮóécYrijŽāĀIJāŠšā■ēāōūārśéd' RéŮóécYāĀIrijŃā
éġcāL■æIJL' äyĀççŮéē■āŠŃäyĀāRġç■ūā■RāĀĆāIJlè£ŽéĜŃæfRäyġāŠšā■ēāōūāRřāžèçIJŃāĀŽæYřäyĀäyġçŃ
æĀlèĀČāĀAāRČēē■äyLçg■çŁúæĀAäy■çŽDäyĀäyġāĀĆéIJāēeAæšġāēDRçŽDæYrijŃæfRäyġāŠšā■ēāōūāRČ
éČčāžLāžŮāžnāžTäyġéČ;āRġlèČ;æŃ£çġĀäyĀāRġç■ūā■RāġRāIJlèČčāĎfrijŃçŽt' āLřéè£æ■žāĀĆæ■đ' æŮūāžŮā
äyŃéġcāYřäyĀäyġçōĀā■TçŽDä;£çTġæ■zéTAcA£āĚ■æIJžāLŮèĝčāEšāĀIJāŠšā■ēāōūārśéd' RéŮóécYāĀIçŽDæ

```
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()
```

æIJāRŌíijŃèeAçŁ' žāLŃæšġāēDRāLříijŃäyžāžEéA£āĚ■æ■zéTAcíijŃæL' ĀæIJLçŽDāLāéTAcæŠ■ā;IJā£Ěē
acquire() āĜ;æTřāĀĆāēČēđIJāžççāĀäy■çŽDæšRéČġāĤEçzTè£Ĝacquire
āĜ;æTřçŽt' æŌèçTšèrúeTAcíijŃéČčāžLæTř' äyġæ■zéTAcA£āĚ■æIJžāLŮāřšäy■ētūā;IJçTġāžEāĀĆ

14.6 12.6 ä£ġā■Yçž£çlŃçŽDçŁúæĀAā£æAř

éŮóécY

ä;äeIJāēeAā£ġā■Yæ■čāIJlè£RèaŃçž£çlŃçŽDçŁúæĀAíijŃè£ŽäyġçŁúæĀAāržāžŌāĚūāžŮçŽDçž£çlŃæY

èĝčāEšæŮžæaĤ

æIJL' æŮūāIJāđ' Žçž£çlŃçijŮçlŃäy■íijŃä;äeIJāēeAāRġā£ġā■Yā;ŠāL■è£RèaŃçž£çlŃçŽDçŁúæĀAāĀĆ
èeAē£ŽāžLāAŽíijŃāRřā;£çTġthread.local() āLŽāžžäyĀäyġæIJŃāIJřçž£çlŃā■YāĆġāržèšāĀĆ
āržè£ŽäyġāržèšaçŽDāšđæĀĝçŽDā£ġā■YāŠŃēržāRŮæŠ■ā;IJéČ;āRġaijŽāržæL' ĝeāŃçž£çlŃāRřèĝAíijŃèĀŃāĚ

äJäyžä;ŁçŤlæIJñäIJřāYāĆłŻDäyÄäyŁæIJL'èŭččŻDăōđéŽĚä;ŇāRiijŇ
 èĂĈeŽŚāIJl8.3ārRèŁĈăōŽāzL'èŁĠçŻD LazyConnection äyŁäyŇæŮĠçōāçŘĚāZłçśzāĂĈ
 äyŇéłĉæŁŚāzñāržāōĈēŁZēāŇäyÄāzŻārRçŻDăŁōæŤzä;Łā;ŮăōĈăRřāzēéĂĈçŤlāzŌăđ'ŽçžŁçłŇiijŽ

```
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
```

äzčĉăAäyriijŇeĠlăŭsēġĈāršāržāžŌ self.local äśđæĂġçŻDä;ŁçŤlāĂĈ
 āōĈĉēñāLlăġŇāŇŮäyžäyÄäyŁ threading.local() āōđä;ŇāĂĈ
 äĚŭāzŮæŮzæşŤæŞā;IJĉñāYāĆłäyž self.local.sock çŻDăēŮæŌēāŮăržēsāăĂĈ
 æIJL'äzĚēŁZāzŻārşāRřāzēāIJlăđ'ŽçžŁçłŇäyăōL'ăĚłçŻDä;ŁçŤl LazyConnection
 āōđä;ŇāzĚăĂĈă;ŇăēĈiijŽ

```
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()
```


āōČāzŇæL'ĀāzēēāŇā; ŪēĀŽčŽDāŌšāZāæYŕæfRāyłčžŁćłŇāijŽāŁZāzžāyĀāyłēGłāūsāyŠāsđčŽDāēŪæŌ
āZāæ■d'rijŇā; Šāy■āRŇčŽDčžŁćłŇæL'gēāŇāēŪæŌēā■ŪæŠ■ā; IJæŪūrijŇčŤśāžŌæŠ■ā; IJčŽDæYŕāy■āRŇčŽ

èóìèőž

āIJlād'gēČlāŁEćłŇāžRāy■āŁZāzžāŠŇæŠ■ā; IJčžŁćłŇčŁ'žāōŽčŁūæĀĀāzūāy■āijZæIJL'āžĀāzŁēŪŌēčYā
āy■ēŁGrijŇā; ŠāGžāžEēŪŌēčYčŽDæŪūāĀZrijŇēĀŽāyŷæYŕāZāyžæšRāyłŕžēšāēčŇād'ŽāyłčžŁćłŇā; ŁčŤlāŁ
æŕŤāēČāyĀāyłāēŪæŌēā■ŪæŁŪæŪGāzūāĀČā; āāy■ēČ; ēŌl'æL'ĀæIJL'čžŁćłŇāĒsāžŇāyĀāyłā■ŤčŇŇāŕžēšārijŇ
āZāyžād'ŽāyłčžŁćłŇāRŇæŪūēŕžāŠŇāĒŽčŽDæŪūāĀZāijZāžgčŤšæūūāzšāĀČ
æIJŇāIJčžŁćłŇā■YāČlēĀŽēŁGēŌl'ēŁZāžZēłDæžRāŕłēČ; āIJlēcŇā; ŁčŤlčŽDčžŁćłŇāy■āRŕēgĀælēēgčāEšēŁZ

æIJŇēŁČāy■rijŇā; ŁčŤl thread.local() āRŕāzēēŌl'
LazyConnection čśzæŤŕæŇĀāyĀāyłčžŁćłŇāyĀāyłēŁđæŌērijŇ
ēĀŇāy■æYŕāŕžāžŌæL'ĀæIJL'čŽDēŁŽčłŇēČ; āRlæIJL'āyĀāyłēŁđæŌēāĀČ

āĒūāŌščRĒæYŕrijŇæfRāył threading.local() āŌđā; ŇāyžæfRāyłčžŁćłŇčł'æŁd'člĀāyĀāyłā■Ťč
æL'ĀæIJL'æŽŌēĀŽāŌđā; ŇæŠ■ā; IJæŕŤāēČēŌūāRŪāĀĀāŁŌæŤžāŠŇāŁāēZd'āĀijāžĒāžĒæŠ■ā; IJēŁŽāyłā■Ūā
æfRāyłčžŁćłŇā; ŁčŤlāyĀāyłčŇŇčŇŇčŽDā■ŪāĒYāŕšāRŕāzēāŁlērĀæŤŕæ■ŌčŽDēŽŤčēzāžĒāĀČ

14.7 12.7 āŁZāzžāyĀāyłčžŁćłŇæšā

éŪŌēčY

ā; āāŁZāzžāyĀāyłāūēā; IJēĀĒčžŁćłŇæšārijŇčŤlāēāŠ■āžŤāŌčāŁūčŇŕēŕūæšČæŁŪæL'gēāŇāĒūāžŪčŽDā

ēgčāEšæŪzæāŁ

concurrent.futures āĠ; æŤŕāžŠæIJL'āyĀāył ThreadPoolExecutor
čśzāRŕāzēēčŇčŤlāēāŌŇæŁRēŁŽāyłāžžāŁāāĀČ āyŇēlčæYŕāyĀāyłčŌĀā■ŤčŽDŤCPæIJ■āŁāāŽlrijŇā; ŁčŤlāž

```
from socket import AF_INET, SOCK_STREAM, socket
from concurrent.futures import ThreadPoolExecutor

def echo_client(sock, client_addr):
    '''
    Handle a client connection
    '''
    print('Got connection from', client_addr)
    while True:
        msg = sock.recv(65536)
        if not msg:
            break
        sock.sendall(msg)
    print('Client closed connection')
    sock.close()

def echo_server(addr):
```

(continues on next page)

(continued from previous page)

```
pool = ThreadPoolExecutor(128)
sock = socket(AF_INET, SOCK_STREAM)
sock.bind(addr)
sock.listen(5)
while True:
    client_sock, client_addr = sock.accept()
    pool.submit(echo_client, client_sock, client_addr)

echo_server('', 15000))
```

æĈædIĵ;æĈsæL'NâLlâLZâzzâ;æĜlâũŝĈŽĎĉžĈlNæšâijŇ
éĂŽâyyâRfäzë;ĤĈTlâyĂäyIQueueæIëè;zæIġâôđĈŎřăĂĈâyNéIcæYřäyĂäyIĉl■ăġôäy■ăRŇă;EæYřæL'NâLlâô

```
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)
```

ThreadPoolExecutor çZyârfzâžŒL'NâLlâôđŒŒçŽDäyÄäylâejâd'DâIJlâžŒâôČä;£â;Ů
äzzâLâæRŘäžd'èÄĖäŽt'æŮzâ;£çŽDäzŒëcñerČçTlâĠ;æTřäy■èŒŮâRŮĚĚTâZđâĀijāĀČä;NâçĆijNä;ääRřèČ

```
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;Nâ■Räy■ĚĚTâZđçŽDhandleâržzèšâijŽäyôä;âad'DçRĚæL'ÄæIJL'çŽDĚYzâadäyŒâ■Rä;IJiijNçDúâRŒŒ
çL'zâLñçŽDijNâ.result() æ\$■ä;IJäijŽĚYzâadĚĚçlNçŽt'âLřâržâžTçŽDâĠ;æTřæL'ğëâNâŒNâLŘäžüĚĚ

èŒlèŒž

éÄŽäyyæİëèŒŒijNä;ääžTèrëéAĚâĚ■çijŮâĚŽçžĚçlNâTřëĠRâRřäzëæŮâéŽRâLŮâćĚĚTĚçŽDçlNâžRâĀČ

```
from threading import Thread
from socket import socket, AF_INET, SOCK_STREAM

def echo_client(sock, client_addr):
    '''
    Handle a client connection
    '''
    print('Got connection from', client_addr)
    while True:
        msg = sock.recv(65536)
        if not msg:
            break
        sock.sendall(msg)
    print('Client closed connection')
    sock.close()

def echo_server(addr, nworkers):
    # Run the server
    sock = socket(AF_INET, SOCK_STREAM)
    sock.bind(addr)
    sock.listen(5)
    while True:
```

(continues on next page)

(continued from previous page)

```
client_sock, client_addr = sock.accept()
t = Thread(target=echo_client, args=(client_sock, client_
→addr))
t.daemon = True
t.start()

echo_server(('', 15000))
```

ā;çōæfZāylāzšāRfāzēāuēā;IijN ā;EæYrāōCāy■ēČ;æŁtā;æIJL'āzžērTāZ;ēĀŽēfGāŁZāzžād'gēGRčž
ēĀŽēfGā;ŁčTīēcDāĒLāLīāgNāNŪčŽDčžŁčNāēsāiijNā;āāRfāzēēō;ç;ōāRŌNæUūēfRēāNčžŁčNčŽDāyLēZŌRā

ā;āāRfēČ;āijZāĒšāfČāŁZāzžād'gēGRčžŁčNāijZāIJL'āzĀāzLāRŌædIJāĀČ
čŌřāzčæŠ■ā;IJčžžčžšāRfāzēā;Lē;žæI;čŽDāŁZāzžāGāā■ČāylčžŁčNčŽDčžŁčNāēsāāĀČ
čTŽēGšijNāRŌNæUūāGāā■ČāylčžŁčNč■L'ā;Ēāuēā;IJāzūāy■āijZāfzāĒūāzŪāzččāĀāžgčTšæĀgēČ;ā;šāŠ■āĀ
ā;ŠčDūāzEijNāēCædIJæL'ĀæIJL'čžŁčNāRŌNæUūēcāTā'ēEšāzūčNā■šāIJCPUāyŁæL'gēāNijNēČčāřsāy■
ēĀŽāyŷijNā;āāzTērēāRīāIJī/Oād'DčRĒčŽyāĒšāzččāĀāy■ā;ŁčTīčžŁčNāēsāāĀČ

āŁZāzžād'gčŽDčžŁčNāēsāčŽDāyĀāyLāRfēČ;ēIJāēēĀāĒšæšŁčŽDēŪōēčYæYrāĒĒā■YčŽDā;ŁčTīāĀČ
ā;NāēČijNāēCædIJā;āāIJIOS XčžžčžšāyLēlčāŁZāzž2000āyŁčžŁčNijNčžžčžšæY;čd'žPythonēfZčNā;ŁčTīā
āy■ēfGijNēfZāyLēōāčŌŪēĀŽāyŷæYræIJL'ērāuōčŽDāĀČā;ŠāŁZāzžāyĀāyŁčžŁčNāēUūijNāēŠ■ā;IJčžžčžšāi
æT;ç;ōčžŁčNčŽDæL'gēāNāēāLijLēĀŽāyŷæYr8MBād'gārRijL'āĀČā;EæYrēfZāyLāĒĒā■YāRīæIJL'āyĀārR
āZāæ■d'ijNPythonēfZčNā;ŁčTīāLrčŽDčIJšāōdāĒĒā■YāĒūāōdā;LārR
ijLærTāēČijNāržāžŌ2000āyŁčžŁčNāēlēēōšijNārīā;ŁčTīāLrāzē70MBčŽDčIJšāōdāĒĒā■YijNēĀNāy■æYr
āēCædIJā;āæNēāfČēŽZæNšāĒĒā■Yād'gārRijNārřāzēā;ŁčTī
stack_size() āG;æTŕælēēZ■ā;ŌāōČāĀČā;NāēČijZ
threading.

```
import threading
threading.stack_size(65536)
```

āēCædIJā;āāLāāyLēfZāēIāēf■āRēāzūāĒ■āēfRēāNāL'ēlččŽDāŁZāzž2000āyŁčžŁčNērTēlNijN
ā;āāijZārŠčŌřPythonēfZčNārīā;ŁčTīāLrāzēā;gāēC210MBčŽDēŽZæNšāĒĒā■YijNēĀNčIJšāōdāĒĒā■Y
æšlāDRčžŁčNāēāLād'gārRāfĒēāzēGšārŠāyž32768■ŪēLČijNēĀŽāyŷæYrčžžčžšāĒĒā■Yēātd'gārRijL409

14.8 12.8 čōĀā■TčŽDāžūēāNčijŪčN

ēŪōēčY

ā;āæIJL'āyŁčNāžRēēĀæL'gēāNCPUārĒēZEādnāuēā;IijNā;āæČšēōl'āzŪāŁ'čTīād'ŽæāyCPUčŽDāijYā

ēgčāĒšæŪzæāŁ

concurrent.futures āžSæRRā;ZāžĒāyĀāyŁ ProcessPoolExecutor čšzijN
ārřēcNčTīāēāIJāyĀāyLā■TčNčžŽDPythonēgčēGLāZīāy■æL'gēāNēōāčŌŪārĒēZEādnāG;æTŕāĀČ
āy■ēfGijNēēĀā;ŁčTīāōČijNā;āēēŪāĒLēēĀæIJL'āyĀāzŽēōāčŌŪārĒēZEādnčžŽDāzžāŁāĀČ
æŁSāžnēĀŽēfGāyĀāyŁčŌĀ■TēĀNāōdēZēčŽDā;Nā■RālēēijTčd'žāōČāĀČāĀGāōZā;āæIJL'āyŁApache
webæIJ■āŁāZīāŪēāfŪčZōā;TčŽDgzipāŌNčijl'āNēijZ

```
logs/
  20120701.log.gz
  20120702.log.gz
  20120703.log.gz
  20120704.log.gz
  20120705.log.gz
  20120706.log.gz
  ...
```

ěĚŽäŸÄæ■ēāĀĜēōĳæfRäŸlæŮēāĤŮæŮĜäzŮāĒĚāōžčšžäijjäŸŇéĬcèĚŽæăŮijŽ

```
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 -
...
```

äŸŇéĬcæŸŕäŸÄäŸlèĎŽæĬŇŷijŇāĬĬèĚŽäzŽæŮēāĤŮæŮĜäzŮäŸ■æšēæLĳăĜžæL'ÄæĬJL'èōĚéŮōèĚĜrobot

```
# 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)

```

aL■éIcçŽDçlNāžRä;fçTlāžEéĀŽāyçŽDmap-reduceéĀāijælēçijŪāEŽāĀĆ āĜ;æTř
 find_robots() āIJlāyĀāyĽæŪĜāzūāR■éZEāRĽāyLāAŽmapæ\$■ā;IJiijNāzūārEçz\$ædIJæsĜæĀzāyžāyĀā
 āz\$ārsæYř find_all_robots() āĜ;æTřāy■çŽD all_robots éZEāRĽāĀĆ
 çŌrāIJiijNāAĜeō;ā;āæČçèeAāŁōæTžèfZāyĽçlNāzRèōl'āōČā;fçTlād'ŽæāyCPUāĀĆ
 ā;ŁçōĀā■TāĀTāĀTāRĽéIJĀèeAārEmap()æ\$■ā;IJæZŁæ■cāyžāyĀāyĽ
 concurrent.futures āž\$āy■çTšæĽRçŽDçšzāijijæ\$■ā;IJā■šāRřāĀĆ
 āyNéIcæYřāyĀāyĽçōĀā■TāŁōæTžçL'ŁæIJñiijŽ

```

# 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')

```

(continued from previous page)

```
for ipaddr in robots:
    print(ipaddr)
```

éÅžè£Gè£Žäyłä£óæŤzâŔŔiijNè£ŔèaŊè£ŽäyłèDŽæIJnăžgçŤšâŔŊæăüçŽDçz\$ædIJiijNă;EæŸfâIJlăŽŽ.
ăđđéŽĚçŽDæĂgèĈ;ăijŸăŊŮæŤŁædIJæăžæ■ăă;ăçŽDæIJžăŽlCPUæŤŕèGRçŽDăy■ăŔŊèĂŊăy■ăŔŊăĂĈ

èőlèőž

ProcessPoolExecutor çŽDăĚyăđŊçŤlæşŤæĈăyŊiijŽ

```
from concurrent.futures import ProcessPoolExecutor

with ProcessPoolExecutor() as pool:
    ...
    do work in parallel using pool
    ...
```

ăĚŮăŐşçŔEæŸŕiijNăyĂăył ProcessPoolExecutor
ălŽăžžNăyłçŊŋçŋŊçŽDPythonèğçéĠLăŽlriijŊ NæŸŕçşzçzşşăyŁéłçăŔŕçŤlCPUçŽDăyłæŤŕăĂĈă;ăăŔŕăžééĂŽ
ProcessPoolExecutor(N) ælëăŋŋæŤž âđ'ĐçŔEăŽlæŤŕéGRăĂĈè£Žäyłăđ'ĐçŔEæşăăijŽăyĂçŽŕ'è£Ŕèa
çĐŮăŔŐăđ'ĐçŔEæşăèçŋăĚşéŮ■ăĂĈăy■è£ĠriijŊçlŊăžŔăijŽăyĂçŽŕ'ç■L'ă;ĚçŽŕ'ălŔæL'ĂæIJL'æŔŔăžđ'çŽDă

èçŋăŔŔăžđ'ălŔæşăăy■çŽDăŮëă;IJăŋĚéqžèçŋăŐžăžL'ăyžăyĂăyłăĠ;æŤŕăĂĈæIJL'ăyđ'çğ■æŮžæşŤăŐžæ
ăçĈăđIJă;ăæĈşèŋŕ'ăyĂăyłăLŮëăŋăŐŋŕijăLŮăyĂăył map()
æş■ă;IJăžŮëăŊæLğëăŊçŽDèŕlriijŊăŔŕă;£çŤl 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)
```

ăŔëăđ'ŮriijNă;ăăŔŕăžëă;£çŤl pool.submit() ælëæL'ŊăŁłçŽDăŔŔăžđ'ă■ŤăyłăžžăŁăŕiijŽ

```
# Some function
def work(x):
    ...
    return result

with ProcessPoolExecutor() as pool:
    ...
    # Example of submitting work to the pool
```

(continues on next page)

(continued from previous page)

```
future_result = pool.submit(work, arg)

# Obtaining the result (blocks until done)
r = future_result.result()

...
```

Future
result()

```
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)
```

-
-
-
-
-

-

14.9 12.9 Python

éÜöécŸ

èġċaEşəÚzæaġL

ar;ċoqPythonaōNāĒlæTræNĀad'ŽçžŁłŃçijŪłŃŃijN ā;EæYrèġċéĠŁāZłċŽDĊer■ēlĀāōđċŌrēĊlāLEāIJl
āōđéZĒāyŁiijNēġċéĠŁāZłēcnāyĀāyġāĒlāsĀēġċéĠŁāZłēTĀāġlæLd'ċlĀŃijNāōĊċāōāġlāzzā;TæŪūāĀZēĊ;āR
GILæIJĀād'ġċŽDēŪōēċYārsæYrPythonçŽDād'ŽçžŁłŃçlŃāzRāzūāy■ēĊ;āLl'ċTlād'ZæāyCPUçŽDāijYāŁē
ŃijLærTæĊāyĀāyġā;ŁçTlāzĒāđ'ZāyŁçžŁłŃçŽDēōāċŪāfEēZEāđNçlŃāzRāRlāijZāIJlāyĀāyġā■TCPUāyŁēlēċ

āIJlēōlēōžæZōēĀZçŽDĠILāzNāL■ŃijNæIJLāyĀçĊzēēĀāijžērĊçŽDæYrGILāRlāijZā;sā■āLrēĊcāzZāy
āēĊādIJā;āçŽDçlŃāzRāđ'ġēĊlāLEāRlāijZæūLāRŁāLl/OŃijNærTæĊç;ŚçzIJāzd'āzŃijNēĊcāzLā;ŁçTlād'Žç
āZāyžāōĊāznād'ġēĊlāLEāŪūēŪr'ēĊ;āIJlç■LāġĒāĀĊāōđéZĒāyŁiijNā;āāōNāĒlāRfāzēæTlāfĊçŽDāLZāzā
ċŌrāzċæS■ā;IJçşçzçşēfRēāNēfZāzLād'ŽçžŁłŃçNæşæIJLāzzā;TāŌNāLZŃijNæşāāTēāRræNēāfĊçŽDāĀĊ

ēĀNāržāzŌā;ġlētŪCPUçŽDçlŃāzRŃijNā;āēIJĀēēĀāijDæyĒæžæL'ġēāNçŽDēōāċŪçŽDçL'zçĊzāĀĊ
āġNāēĊŃijNāijYāNŪāzTāsĊçŌŪæşTēēĀærTā;ŁçTlād'ŽçžŁłŃçNēfRēāNāfā;Ūād'ZāĀĊ
çşzāijijçŽDŃijNçTśāzŌPythonæYrèġċéĠŁæL'ġēāNçŽDŃijNāēĊādIJā;āārEēĊcāzZæĀġēĊ;çŞūēċLāzçċāĀçġzā
ēĀşāžēāzşāijZæRŘā■ĠçŽDāġLāfānāĀĊāēĊādIJā;āēēĀæS■ā;IJæTŕçzDŃijNēĊcāzLā;ŁçTlNumPyēfZæāūçŽD
æIJĀāRŌŃijNā;āēfYāRfāzēēĀĊēZŚāyNāĒūāzŪāRrēĀL'āōđċŌræŪzæāLŃijNærTæĊPyPyŃijNāōĊēĀZēfĠāy
ŃijLāy■ēfĠāIJlāEŽēfZæIJnāzççŽDæŪūāĀZāōĊēfYāy■ēĊ;æTræNĀPython 3ŃijLāĀĊ

ēfYæIJLāyĀçĊzēēĀæşlæDRçŽDæYŕŃijNçžŁłŃçNāy■æYfāyŞēŪlçTlālēāijYāNŪæĀġēĊ;çŽDāĀĊ
āyĀāyġCPUā;ġlētŪādNçlŃāzRāRrēĊ;āijZā;ŁçTlçžŁłŃçNēlēċōāĊRēāyĀāyġāZlā;ċçTlāLūçTŃēlēāĀāyĀāyġ
ēfZæŪūāĀZŃijNĠILāijZāzġçTşāyĀāzZēŪōēċYŃijNāZāāyžāēĊādIJāyĀāyġçžŁłŃçNēTfæIJşæNĀæIJL'GILçŽD
āzNāōđāyŁiijNāyĀāyġāEŽçŽDāy■āē;çŽDĊer■ēlĀæLl'āsTāijZārŃijēĠt'ēfZāyġēŪōēċYæZt'āLāāyēēĠŃijN
ar;ċoqāzçċāĀçŽDēōāċŪēĊlāLEāijZærTāzNāL■ēfRēāNçŽDæZt'āfānāzZāĀĊ

ērt'āzEēfZāzLād'ZŃijNçŌrāIJlæĊşērt'çŽDæYræLŚāznæIJLāyđ'çġ■ç■ŪçTælēēġċāEşGILçŽDçijçĊzā
ēēŪāĒLŃijNāēĊādIJā;āāōNāĒlāūēā;IJāzŌPythonçŌrāċĊāy■ŃijNā;āāRfāzēā;ŁçTl
multiprocessing ælāāIŪālēāLZāzzāyĀāyġēfZçlŃçNæşāŃijN
āzūāĊRā■RāRŃād'DçRĒāZlāyĀæāūçŽDā;ŁçTlāōĊāĀĊāġNāēĊŃijNāĀĠāēĊā;āæIJLāēĊāyNçŽDçžŁłŃçNāzçç

```
# 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)
    ...
```

āfōæTzāzçċāĀŃijNā;ŁçTlēfZçlŃçNæşāŃijZ

```
# 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()
```

èfZäyléÄZèfGä;fçTlâyÄäylæLÄaûgâl'çTléfZçlNæsäègçâEşazEGILçZDèUóécYāĀĆ
 â;ŞäyÄäylçZçlNæCşèeAæL'gèaŃCPUârEéZEădNăuëä;IJæŮürijŃäijZârEăzzâLăaŔSçzZèfZçlNæsäāĀĆ
 çDûâRŌèfZçlNæsäâijZâIJlâRèad'ŮäyÄäylèfZçlNäy■ârRâLlâyÄäylâ■TçNñçZDPythonègçéGŁăZlæIěăuëä;I
 â;ŞçZçlNç■L'â;ĒçzŞædIJçZDæŮûâĀZâijZèGŁæTçGILăĀĆăzûäyTrijŃçTşazŌèôaçŏŮăzzâLăaIJlâ■TçNñèg
 âIJlâyÄäylâd'ZæäyçşzçzşäyLéIçijŃä;ââijZârSçŌrèfZäylæLÄæIJrâRfäzèèol'ä;ăâ;Lăè;çZDâl'çTlâd'ZCPU
 âRèad'ŮäyÄäylègçâEşGILçZDç■ŮçTæYrâ;fçTlCæL'âsTçijŮçlNæLÄæIJrâĀĆ
 äyzeèAæĀIæCşæYrârEèôaçŏŮârEéZEădNăzzâLăe;ñçgçzçZCijŃeûşPythonçNñçŃNrijŃâIJlâuëä;IJçZDæŮû
 èfZârRfäzèèĀZèfGâIJlCăzççăÄäy■æŔSâĒèäyŃéIçèfZæăuçZDçL'zæôLăôRæIěăŏNæLRijZ

```
#include "Python.h"
...

PyObject *pyfunc(PyObject *self, PyObject *args) {
    ...
    Py_BEGIN_ALLOW_THREADS
    // Threaded C code
    ...
    Py_END_ALLOW_THREADS
    ...
}
```

âeĆădIJă;ăä;fçTlâEüâzŮăuëâEüèôfèŮôCèr■éĀrijŃârTâeĆârzâzŌCythonçZDctypesăzŞrijŃä;ăäy■éIJă
 â;ŃæĀçrijŃctypesâIJlêrCçTlCæŮûâijZèGŁăLéGŁæTçGILăĀĆ

èóIèöZ

èöyâd'ZçlNăzŔâSŸâIJlélçărzçZçlNæĀgèC;éŮóécYçZDæŮûâĀZrijŃél'ňäyLârşâijZæĀtç;†GILijŃăzĀă
 âĒŮăôðèfZæăuâ■Ŕad'lây■ăŌZèAşazşad'lad'l'çIJşăzEçCzâĀĆ
 â;IJâyZâyÄäylçIJşăôðçZDă;Ńâ■RrijŃâIJlâd'ZçZçlNçZDç;ŞçzIJçijŮçlNäy■çèdçgYçZD
 stalls âŔrèC;æYrâZăyZâEüâzŮăŌşâZăærTâeCâyÄäyIDNSæşèæLçăzûæŮürijŃèĀŃeûşGILæŕnæŮăâĒş
 æIJârŔŌă;ăçIJşçZDèIJăèeAăĒLăŌzæŔdæGĀ;ăçZDăzççăAæYrârEçIJşçZDèçŃGILă;şâŞ■ăLŕăĀĆ
 âŔŃæŮûèfYèeAæYŌçZçGILăd'gèĀlăLÉçC;ăzTèŕeâŔlăĒşæşlCPUçZDăd'DçŔEèĀŃäy■æYŕl/O.

âeĆădIJă;ăăGEăd'Gă;fçTlâyÄäylâd'DçŔEăZlæşârijŃæşlæDŔçZDæYŕèfZæăuâĀZæŮL'ârLăLŕæTŕæ■C
 èçŃæL'gèaŃçZDæŞ■â;IJéIJăèeAæTçâIJlâyÄäyléĀZèfGdèŕ■âŔăôŏZăZL'çZDPythonăĜ;æTŕây■rijŃäy■èC;

ázúäyTãGjæTřãRĆæTřãŠNěTãZđãAijãĚěązēęAãĚijãđżpicklēãĀĆ
 årNæãüiijNěęAæL'gëąNçZDăzzãŁaęGRãĚĚązēűşăđ'şăđ'găzëăijjēęęęćlăđ'ŮçZDěĂZăĚąăijĂęTĂăĂĆ
 årĚăđ'ŮăyĂăyĹęZjçCzæYřăjŞăüüăRLăj;ĤçTlçzĤçłNăŠNěĤçłNăşăçZDăŮüăĂZăijZēđl'ăjăăŁăđ't'çŮ
 âęĆăđIJăjăđęAãRÑæŮüăj;ĤçTlăyđ'ëĂĚiijNæIJĂăęjăIJłłNăžRăRřăŁăŮüiijNăŁZăzzăzzăj;TçzĤçłNăzNăL'■ă
 çDúăRŌçzĤçłNăj;ĤçTlăRÑæăüçZDěĤçłNăşăăĹëđĤZëąNăđCăznçZDěđăçđŮăřĚęZĚăđNăüëăjIJăĂĆ
 CæL'l'ăsTæIJăĚĠēęAçZDçL'zăjAæYřăđCăznăŠNPythonēğçĚĠăZlăYřăĤIæNăçNñçNçZDăĂĆ
 äzşârşæYřęřt'iijNăęĆăđIJăjăăĠĚăđ'ĠăřĚPythonăy■çZDăzzăŁăăĹĚĚĚ■ăĹrCăy■ăŌzæL'gëąNiiijN
 äjăĚIJăĚęAçăđăĤCăzççăAçZDăŞ■ăjIJëüŞPythonăĤIæNăçNñçNiiijN
 ěĤZăřşæĎRăŞçłĂăy■ĚęAăj;ĤçTlPythonæTřæ■đçzŞăđDăzčăRLăy■ĚęAĚrCçTlPythonçZDÇ
 APIăĂĆ årĚăđ'ŮăyĂăyĹăřşæYřăjăđęAçăđăĤCæL'l'ăsTæL'ĂăZçZDăüëăjIJæYřęűşăđ'şçZDiiijNăAijăŁŮăjăă
 äzşârşæYřęřt'CæL'l'ăsTæNĚĚt'şęŮăăZĚăđ'gëĠRçZDěđăçđŮăzzăŁăăijNëĂNăy■æYřăřşæTřăGăăyĹęđăçđŮăĂĆ
 ěĤZăžZēğçăĚşGILçZDăŮzæăŁăzŮăy■ĚçjĚĂCçTlăžŌăL'ĂăIJL'ĚŮĚĚYăĂĆ
 äjNăęĆiijNăşRăžZçşzăđNçZDăžTçTlçłNăžRăĚęĆăđIJĚĚñăĹĚęğçăyžăđ'ZăyĹęĤçłNăđ'DçĤĚçZDĚřlăžŮăy■Ě
 äzşăy■ĚçjăřĚăđCçZDĚĤlăĹĚăzççăAæTzæĹRĤĚĚĚĚĹăL'gëąNăĂĆ
 årzăžŌĚĤZăžZăžTçTlçłNăžRiiijNăjăăăřşęęAęĠăüşĚIJăşĤĚğçăĚşşæŮzæăŁăžĚ
 iijLăřTăęĆăđ'ZĚĤçłNĚĚĚŮăăĚşăžăĤĚĚ■YăNžiiijNăđ'ZĚğçăđRăZlĚĚĚĚăNăžŌăRÑăyĂăyĹęĤçłNç■L'iiijL
 æLŮĚĂĚiijNăjăĚĚYăRřăžĚĚĂĤĚZşăyNăĚŮăžŮçZDĚğçĚĠăZlăđđçŌriijNăřTăęCPyPyăĂĆ
 äZĚęğçæZt'ăđ'ZăĚşăžŌăIJlCæL'l'ăsTăy■ĚĠăTjGILiijNĚřŮăRĤĚĂĤ15.7ăŠN15.10ăřRĚĤCăĂĆ

14.10 12.10 áŌZăZL'ăyĂăyĹActorăzzăŁă

ĚŮĚĚY

ăjăăĤşăđZăZL'ĚüşactorăĹăăijRăy■çşăăijjăĂIJactorsăĂĹęŞĚL'şçZDăzzăŁă

ĚğçăĚşşæŮzæăŁ

actorăĹăăijRăYřăyĂçğ■æIJăăRđ'ëĂAçZDăžşæYřæIJăĤĚđăžŮăNăŠNăĹĚăyĤăijRĚĚăçđŮĚğç
 äžNăđăyĹiijNăđCăđ'l'çTşçZDçđŮă■TăĠğæYřăđCăęĆă■đ'ărŮăñĤĚŌçZDĚĠĚęAăŌşăZăăzNăyĂăĂĆ
 çđŮă■TăĹĚĚđŮiijNăyĂăyĹactorăřşæYřăyĂăyĹăžŮăRşæL'gëąNçZDăzzăŁăăijNăRĹăYřçđŮă■TçZDăL'gëąNăR
 âş■ăžTĚĤZăžZăŮĹăAřăŮüiijNăđCăRřĚĤjĚĤYăijZçzZăĚŮăžŮactorăRşĚĂAæZt'ĚĤZăyĂăĚĚZDăŮĹăAřă
 actorăžNĚŮt'çZDĚĂZăĚăęYřă■TăRşăŠNăijCă■ĚçZDăĂĆăZăă■đ'iijNăŮĹăAřăRşĚĂAĚĂĚăy■çşĚĚAşşæŮĹ
 äzşăy■ăijZăĚĚăŮăĹăRăyĂăyĹăŮĹăAřăŮşĚĤăđ'DçĤĚçZDăžđăžTăĹŮĚĂZçşĚăĂĆ

çzşăRLăj;ĤçTlăyĂăyĹçzĤçłNăŠNăyĂăyĹęYşăĹŮăRřăžĚăĹăđzæYşçZDăđZăZL'actoriijNăjNăęĆiijZ

```

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
    '''

```

```

    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()

```

```

    ẽŁZäyŁä;Nä■Räy■iijNä;ää;ŁçTŁactorăõđä;NçŽĐ
    æŰzæsŦāRŚéĀAæŰLæAřçzŽăõČăznăĀĆăĔŰæIJăĀLŰæŸriijNẽŁZäyŁæŰzæsŦaijŽārEæŰLæAřæŦ;ăĔëäyĀäy
    çĐŰăRŌârEăĔë;ňăzd'çzŽăd'ĐçŘEèçnăŌëăRŰæŰLæAřçŽĐäyĀäyŁăĔĔçŁçŽŁçĹNăĀĆ
    close() æŰzæsŦēĀŽẽŁĠăIJléŸšăĹŰäy■æŦ;ăĔëäyĀäyŁçL'zăõŁçŽĐăŠĹăĔġăĀijriijLActorExitriijL'æĹăĔĔşéŁ
    çŦĹăLŰăRřăžẽĔĀŽẽŁĠçzġæLŁActorăžŰăõŽăzL'ăõđçŌřëĠăŰăđ'ĐçŘEēĀžẽ;Śrun()æŰzæsŦæĹăăõŽăzL'æŰř
    ActorExităijĈăyŷçŽĐă;ŁçŦĹăřsæŸřçŦĹăLŰăĠăăõŽăzL'ăžççăĀăRřăžẽăIJléIJăĔēAçŽĐăŰăăĀŽăĹă■ŦēĈ
    riijLăijĈăyŷèçŋget()æŰzæsŦæŁZăĠăžăŰăijăăŠ■ăĠăăŌžriijL'ăĀĆ

```

```

    ăĔĈăđIJă;ăæŦ;ăő;ăřzăžŌăRŊă■ăăŠŊăijĈă■ăăŰLæAřăRŚéĀAçŽĐēēAăśĈriijŊ çşzac-
    torăřzèsăēŸăRřăžẽĔĀŽẽŁĠçŦşæĹRăŽĹăĹëçŏĂăŊŰăăõŽăzL'ăĀĆă;NăēĈriijŽ

```

```

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()

```

èóíèőž

actoræłaijRçŽĐē■ĖāŁŻārsāIJlāžŌāōČçŽĐčōĀā■ŦæĀğāĀĆ
āōđēŽĖāyŁiijNēŁŽēĠNāzĖāzĖāRlæIJL'āyĀāyłæāyāŁČæŞ■ā;IJ send()
çŦŽēĠsīijNāržāžŌāIJlāšžāžŌactorçşzçzşşāy■çŽĐāĀIJæŭŁæAŕâĀiçŽĐæşŽāNŪæçĆāŁŦāRŕāzēāŭšāđ'Žçğ■æŪ
ā;NāçĆīijNā;āāRŕāzēāzēāĖČçzĐā;ćāijRāijāéĀšæāĠç■;æŭŁæAŕīijNèł'actoræL'ğēāNāy■āRŦçŽĐæŞ■ā;IJiij

```
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)
```

ā;IJāyžāRēād'ŪāyĀāyłā;Nā■RīijNāyNēłççŽĐactorāĖAēōyāIJlāyĀāyłāŭēā;IJēĀĖāy■ēŁRēāNāzzæĐRçŽ
āžŭāyŦēĀŽēŁĠāyĀāyłçL'žæłŁçŽĐResultāržèšāēŁŦāŽđçzŞæđIJiijŽ

```
from threading import Event
class Result:
    def __init__(self):
        self._evt = Event()
        self._result = None

    def set_result(self, value):
        self._result = value

        self._evt.set()

    def result(self):
        self._evt.wait()
        return self._result

class Worker(Actor):
    def submit(self, func, *args, **kwargs):
        r = Result()
        self.send((func, args, kwargs, r))
        return r
```

(continues on next page)

(continued from previous page)

```
def run(self):
    while True:
        func, args, kwargs, r = self.recv()
        r.set_result(func(*args, **kwargs))

# Example use
worker = Worker()
worker.start()
r = worker.submit(pow, 2, 3)
print(r.result())
```

æIJĀāŔŌījNāĀIJāŔSéĀĀāĀĪāyĀāyĪāzzāLāæŭLæAŕçŽDæçCāſtāŔŕāzēēcnæL'ſāTāLŕād'ŽēſZçĪNçTŽ
äĴNāēČījNāyĀāyĪçšzactorāržēšaçŽD send() æŪzæſTāŔŕāzēēcnçijŪçĪNēōl'āōČēČĴāIJāyĀāyĪāēŪæŌēāŪ
æLŪēĀŽēſGæſŔāžZæŭLæAŕäyæŭt'äzŭījLæŕTāçCAMQPāĀAZMQçL'ījL'æĪēāŔSéĀĀāĀC

14.11 12.11 āōdçŌŕæŭLæAŕāŔSāyČ/ēōcéYĒæĪāđN

éŪōécŸ

äĴæIJL'āyĀāyĪāšzāžŌçžſçĪNēĀŽāſaçŽDçĪNāžŔījNæČšēōl'āōČāznāōdçŌŕāŔSāyČ/ēōcéYĒæĪāāijŔçŽ

ēğçāEşæŪzæāL

ēçĀāōdçŌŕāŔSāyČ/ēōcéYĒçŽDæŭLæAŕēĀŽāſæĪāāijŔījN
äĴāēĀŽāyŷēçĀāijTāEçāyĀāyĪāTçNñçŽDāĀIJāžd'æcæIJzāĀĪæLŪāĀIJçĴSāĒşāĀĪāŕžēšāāIJāyžæL'ĀæIJL'a
āžşāŕſæYŕēſ'ījNāyçŽt'æŌēārEæŭLæAŕāzŌāyĀāyĪāzzāLāāŔSéĀĀāLŕāŔçāyĀāyĪījNēĀNæYŕārEāĒŭāŔSé
çDŭāŔŌçTšāžd'æcæIJzārEāōČāŔSéĀĀçžZāyĀāyĪæLŪāđ'ŽāyĪēcnāĒşēĀTāzzāLāāĀCāyNēĪcæYŕāyĀāyĪēĪd

```
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]
```

äyÄäyläzd' æ■caeIJzärsæYräyÄäylæZöéÄZärzéšaiijNët' šet'čçzt'æŁd'äyÄäylæt'zèuČčZDèöcéYĚèÄĚéZ
æfRäyläzd' æ■caeIJzéÄŽèŁGäyÄäyläR■çgräöZä;■iijNget_exchange()
éÄŽèŁGçzZäöZäyÄäyläR■çgrèŁTäZdçZyāzTčZD Exchange äöđä;NāĀĆ
äyNéİcaeYräyÄäylçöĀā■Tä;Nā■RiijNæijTčd'žāzEāēČä;Tä;ŁçTlāyÄäyläzd' æ■caeIJziijŽ

```
# 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)
```

är;çöāārzāzÖèŁZäyléUöécYæIJL'ā;Łād'ŽčZDāRŸçg■iijNäy■èŁGäyGāRYäy■çēzāĒūāöUāĀĆ
æūŁæAřaijŽècnāRŠéĀAçzZäyÄäyläzd' æ■caeIJziijNçDūāRÖāzd' æ■caeIJzaijZārEāöČāznāRŠéĀAçzZècnçzŠā

èóİèőž

éÄŽèŁGéYšāŁUāRŠéĀAæūŁæAřçZDāzzāŁæŁŮçžŁčİNçZDæİāaijRā;ŁāözæYšècnāöđçÖřāzūāyTāzš
äy■èŁGiijNä;ŁçTlāRŠāyČ/èöcéYĚæİāaijRçZDāē;ād'DæZt'āŁāæYŌæY;āĀĆ

éēŪāĒĒiijNä;ŁçTlāyÄäyläzd' æ■caeIJzāRřāzèçöĀāNŪād'gēČİāŁEæūŁ'āRŁāŁřçžŁčİNéÄŽāŁaçZDāūēā;I
æŪāēIJĀāŌzāEŽéÄŽèŁGād'ŽèŁZçİNæİāāİŪæİēæŠ■ā;IJād'ŽäylçžŁçİNiiijNä;āāRİēIJĀēēAā;ŁçTlēŁZäyläzd' æ
æšRçg■çİNāžēäyŁiijNēŁZäylārseūšæŪēāŁŪæİāāİŮçZDāūēā;IJāŌšçRĒçszaiijjāĀĆ
āödéZĒäyŁiijNāöČāRřāzèē;zæİçZDègçēĀçİNāžRäy■ād'ZäyläzzāŁāāĀĆ


```
class DisplayMessages:
    def __init__(self):
        self.count = 0
    def send(self, msg):
        self.count += 1
        print('msg[{}]: {}'.format(self.count, msg))

exc = get_exchange('name')
d = DisplayMessages()
exc.attach(d)
```

āĖšāžŌāzd' æ■cæIJžčŽDāyĀāyġaRrēČ;éŪóécŸæŸrāfzāžŌēócéŸĖēĀĖčŽDæ■čçāōčzŠāōŽāŠNēgčçzŠāĀ
 āyžāžĖæ■čçāōčŽDčōāçRĖēġDæžRīijNæfRāyĀāyġczŠāōŽčŽDēócéŸĖēĀĖāĤĖēqazēIJĀczĤLēēĀēgčçzŠāĀĆ
 āIJāžčcāĀāy■ēĀžāyŷāijŽæŸrāČRāyNéĪcēĤZæāūčŽDāġāijRīijŽ

```
exc = get_exchange('name')
exc.attach(some_task)
try:
    ...
finally:
    exc.detach(some_task)
```

```
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)
```

489

```

@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

```

æIJĀāŔŌēſŸāŽŦērēæſĭæĐŔçŽĐæŸŕāĒſäžŌäzd' æ■æIJžçŽĐæĀĭæČſæIJL'āĭLād'Žçğ■çŽĐæL'ĭāsŦāōđ
 āĭNāēČĭijNāzd' æ■æIJžāŔŕāzēāōđçŌŕāyĀæŦŦ'āyĭæūLæAŕéĀŽéAſéZEāŔLæLŪæŔŔāĭŽāzd' æ■æIJžāŔ■çğ
 āzd' æ■æIJžēſŸāŔŕāzēēčnæL'ĭāsŦāŔāLēāyČāijŔēōaçōŪçĭNāžŔāy■ĭijLæŕŦāēČĭijNāŕEæūLæAŕēŭŕçŦſāŔŕā

14.12 12.12 äĭĒçŦĭçŦſæĹŔāŽĭāzčæŽĒçžĒçĭN

éŬōécŸ

äĭāæČſäĭĒçŦĭçŦſæĹŔāŽĭĭijLā■ŔçĭNĭijL'æŽĒäzčçſžçžſçžĒçĭNæĭēāōđçŌŕāžūāŔſāĀČēſŽāyĭæIJL'æŪūāſ

èğčĀEſæŪzæāĹ

ēēAäĭĒçŦĭçŦſæĹŔāŽĭāōđçŌŕēĠāŭſçŽĐāžūāŔſĭijNāĭāēēŪāĒĹēēAāŕžçŦſæĹŔāŽĭāĠĭæŦŕāſN
 yield ēŕ■āŔēæIJL'æŭſāLzçŔEēğčāĀČ yield ēŕ■āŔēāijŽēōĭ'āyĀāyĭçŦſæĹŔāŽĭæNČēŭāōČçŽĐæL'gēāNĭ
 āŕEçŦſæĹŔāŽĭāĭſāAŽæſŔçğ■āĀIJžāzāŁāāĀĭāžūāĭĒçŦĭāzžāŁāā■ŔāĭIJāĹĠæ■æĭēæŽĒæ■čāōČāzñçŽĐæL'g

yield

```
# 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
```

```
from collections import deque

class TaskScheduler:
    def __init__(self):
        self._task_queue = deque()

    def new_task(self, task):
        '''
        Admit a newly started task to the scheduler
        '''
        self._task_queue.append(task)

    def run(self):
        '''
        Run until there are no more tasks
        '''
        while self._task_queue:
            task = self._task_queue.popleft()
            try:
                # Run until the next yield statement
                next(task)
                self._task_queue.append(task)
            except StopIteration:
                # Generator is no longer executing
                pass

# Example use
sched = TaskScheduler()
sched.new_task(countdown(10))
sched.new_task(countdown(5))
```

(continues on next page)

(continued from previous page)

```
sched.new_task(countup(15))
sched.run()
```

TaskScheduler çşzâIJläyÄäyİa;İçÖräy■è£RëaÑçTşæLRâZİéZEâRLâÄTâÄTæfRäyİeÇ;è£RëaÑâLİç
è£RëaÑè£Zäyİa;Nâ■RİijNè;ŞaGzæCäyNİijZ

```
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
...
```

âlRæ■d'äyza■cİijNæLSäznâôdeZEäyLâûşçzRâôdçÖräžEäyÄäyİaÄIJæŞ■ä;IJçşzçzşâÄİçZDæIJÄârRæäy
çTşæLRâZİlâG;æTřarsæYřèôd'äyžİijNèÄNyieldèr■âRëæYřäzzâLæNÇèİçZDä£aâRûâÄC
èřČäžæâZİlâ;İçÖræčAæşëäzzâLââLÛealçZt'âlRæşæIJL'äzzâLæçAæL'gëaNäyžæ■câÄC

âôdeZEäyLİijNä;ääRrèČ;æČşèeAä;£çTİçTşæLRâZİæİeâôdçÖřçôÄa■TçZDäzûâRSäÄC
éČcäzLİijNâIJlâôdçÖřactoræLÛç;ŞçzIJæIJ■âLââZİçZDæUûâÄZä;ääRräžčä;£çTİçTşæLRâZİæİeæZ£äzççZ£ç

äyNéİççZDäžčçAæijTçd'žäžEä;£çTİçTşæLRâZİæİeâôdçÖräyÄäyİay■ä;İetÛçž£çÍNçZDactorİijZ

```
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)

ðŏŇăĖĹăĭjDăĜĈēĴăŏĵăzĉĉăĂēĲĂĊēĂăĴt' æŭsăĖēĉŽDă■ēăzăĭĭjNăj;ĖăĲŕăĖŝēTŏĉĆzăĲĹăžŎăTŭēZĖă
 æĲćnèt' ĹăyĹĭĭjNètĈăzēăŽĹăĲăĲĹēĲĂĊēĂăĴŖŖŖĂăĴŽDăŭĹăĂŕăŭăĭjŽăyĂĉŽt' ēĴŖēăŇĉĹăăĂĈ
 ĊŏăăTŕĉTŝăĹŖăŽĹăĭjŽĉzĴēĴăŭsăŖŖŖĂăĴŭĹăĂŕăžŭăĲĹăyĂăyĹēĂŖă;ŖăĴĉŖăy■ĉzŖăĲăăĂĈ
 äyŇēĹăăĲŕăyĂăyĹăŽt'ăĴăénŸĉzĝĉŽDăj;Ňă■ŖĭĭjNăĭjTĉd' žăzĖăj;ĴĉTĹĉŖăĹŖăŽĹăĲăăŏĉĉŖăyĂăyĹăžŭă

```
from collections import deque
from select import select
```

```

# 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)

```

```

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)

```

```

# Wrapper around a socket object for use with yield
class Socket(object):
    def __init__(self, sock):
        self._sock = sock
    def recv(self, maxbytes):
        return ReadSocket(self._sock, maxbytes)
    def send(self, data):
        return WriteSocket(self._sock, data)
    def accept(self):
        return AcceptSocket(self._sock)
    def __getattr__(self, name):
        return getattr(self._sock, name)

if __name__ == '__main__':
    from socket import socket, AF_INET, SOCK_STREAM
    import time

    # Example of a function involving generators. This should
    # be called using line = yield from readline(sock)
    def readline(sock):
        chars = []
        while True:
            c = yield sock.recv(1)
            if not c:
                break
            chars.append(c)
            if c == b'\n':
                break
        return b''.join(chars)

    # Echo server using generators
    class EchoServer:
        def __init__(self, addr, sched):
            self.sched = sched
            sched.new(self.server_loop(addr))

        def server_loop(self, addr):
            s = Socket(socket(AF_INET, SOCK_STREAM))

            s.bind(addr)
            s.listen(5)
            while True:
                c, a = yield s.accept()
                print('Got connection from ', a)
                self.sched.new(self.client_handler(Socket(c)))

        def client_handler(self, client):
            while True:
                line = yield from readline(client)

```

(continues on next page)

(continued from previous page)

```
if not line:
    break
line = b'GOT:' + line
while line:
    nsent = yield client.send(line)
    line = line[nsent:]
client.close()
print('Client closed')

sched = Scheduler()
EchoServer(('', 16000), sched)
sched.run()
```

èĚæōtāzčĥAæIJL'çĆzād'■æĪCāĀĆāy■èĚGriiĴNāōČāōđçŎřāžEāyĀāyĴāŕRādNçŽDæS■ā;IJçşçzçşāĀĆ
æIJL'āyĀāyĴāŕşçzĴçŽDāzzāŁæēYſāĴŪriiĴNāzūāyTēĚYæIJL'āŽāĴ/OāijSçIJăçŽDāzzāŁaç■L'ā;ĒāNzāşşāĀĆ
èĚYæIJL'ā;Ĵād'ŽērČāžēāŽĴet'şet'cāIJĴāŕşçzĴēYſāĴŪāŠNĴ/Oç■L'ā;ĒāNzāşşāzNēŪt'çğzāŁĴāzzāŁāāĀĆ

èőléőž

āIJĴēđDāzzāşżāžŎçTſæĴŔāŽĴçŽDāzūāŔŚæāEæđūæŪriiĴNēĀŽāyāijŽā;ĴçTĴæŽt'āyāyèğAçŽDyielđā;ćā

```
def some_generator():
    ...
    result = yield data
    ...
```

ā;ĴçTĴēĚŽçğ■ā;ćāijŔçŽDyielđēr■āŔēçŽDāĴ;æTŕēĀŽāyāyèćñçğŕāyžāĀIJā■ŔćĴNāĀĴāĀĆ
ēĀŽèĚĴērČāžēāŽĴiijŴyielđēr■āŔēāIJĴāyĀāyĴā;ĴçŎřāy■èćnād'ĐçŔēiijNāēĆāyNriiĴŽ

```
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
```

èĚŽēGNçŽDēĀzē;ŚćĴ■ā;ōæIJL'çĆzād'■æĪCāĀĆāy■èĚGriiĴNēćnāijăçžŽ
send() çŽDāĀijāōŽāzĴ'āžEāIJĴyielđēr■āŔēēEſæĴēæŪūçŽDēĚTāŽđāĀijāĀĆ
āŽāæ■đ'riiĴNāēĆæđIJāyĀāyĴyielđāĴEād'ĴāIJĴāŕzāzNāL■yielđ-
æTŕæ■ōçŽDāŽđāžTāy■ēĚTāŽđçzşæđIJæŪriiĴNāijŽāIJĴāyNāyĀæñā send()
æş■ā;IJēĚTāŽđāĀĆ āēĆæđIJāyĀāyĴçTſæĴŔāŽĴāĴ;æTŕāĴŽāijĀāğNēĚŔēāNriiĴNāŔŚēĀĀyĀāyĴNoneāĀijāij

ēŽđ'āžEāŔŚēĀĀāĀijād'ŪriiĴNēĚYāŔŕāžēāIJāyĀāyĴçTſæĴŔāŽĴāyĴēĴcāL'ğēāNāyĀāyĴ
close() æŪzæşTāĀĆ āōČāijŽārijeĴt'āIJĴæL'ğēāŴyielđēr■āŔēæŪūæĴŽāĴžāyĀāyĴ
GeneratorExit āijCāyāyriiĴNāzŎēĀNçzĴLæ■cāL'ğēāNāĀĆ

æCædIJæfZäyÄæ■èèö;èðaiijNäyÄäyIçTšæLRäZläräzææ■TèÖüèfZäyIaijCäyÿázüæL'gèaÑæyËçRææS■ä;I.
 årNæäüèfYäräzæä;fçTlçTšæLRäZlçZD throw() æÜzæſTäIJyield-
 èr■ärææL'gèaÑæÜüçTšæLRäyÄäyIäzzæDRçZDæL'gèaÑæNGäzd'äÄC
 äyÄäyIäzzäLæèrCäzæZläräL'çTlæöCæIèaIJlèfRæaÑçZDçTšæLRäZlây■ad'DçRææTZeëräÄC

æIJÄärÖäyÄäyIä;Nä■Räy■ä;fçTlçZD yield from èr■äræèèççTlæIèäöðçÖrä■RçlNriijNäräzæèèçnäE
 æIJnèr'läyLäræYræEæÖgälüæIÇæÄRæYÖçZDäijæ;ççzZæÜçZDäG;æTträÄC
 äy■äCRæZöèÄZçZDçTšæLRäZlriijNäyÄäyIä;fçTl yield from
 èçnèrCçTlçZDäG;æTträRäzæèfTäZdäyÄäyIä;IJäyZ yield from
 èr■äræççZædIJçZDäÄijäÄC äEšäZÖ yield from çZDæZt'ad'ZäfaæAraRäzæäIJl PEP
 380 äy■æL;älRäÄC

æIJÄärÖriijNæCædIJä;fçTlçTšæLRäZlçijÜçlNriijNèeAæRRæEšä;äçZDæYræöCèfYæYræIJL'ä;Läd'Zç
 çL'zälNæYriijNä;äa;Üäy■älRäzzä;TçZççlNäräzææRRä;ZçZDäè;äd'DäÄCä;NæCriijNæCædIJä;æL'gèaÑ
 äöCäijZäræEäT'räyIäzzäLæaÑCèüçšèeAšæS■ä;IJäöNæLRäÄCäyZäZègçæEšæfZäyIèÜöèçYriijN
 ä;äaRlèC;éÄL'æNl'ärææS■ä;IJägTæt';çzZäRæad'ÜäyÄäyIäRäzæçNñçNèfRæaÑçZDçZçlNæLÜèfZçlNäÄ
 äRæad'ÜäyÄäyIèZRälüæYræd'gèCläLEPythonäZšäZüäy■èç;ä;Læ;çZDäEijäöZäšZäZÖçTšæLRäZlçZDçZçl
 æCædIJä;äeÄL'æNl'èfZäyIæÜzæaLriijNä;äaijZäRšçÖrä;äeIJäèeAèGläüæTzæEzä;Läd'ZæaGäGæZšäG;æ
 ä;IJäyZæIJnèLÇæRRälçZDä■RçlNäSñçZyÄEšæLÄæIJçZDäyÄäyIäšççäÄèCÑæZriijNäräzææšççIJN
 PEP 342 äSñ äÄIJä■RçlNäSñäZüärSçZDäyÄèÜäIJL'èüçç;çlNäÄI

PEP 3156 årNæäüæIJL'äyÄäyIäEšäZÖä;fçTlæ■RçlNçZDäijCæ■I/OæIäadNäÄC
 çL'zälNçZDriijNä;äay■äræC;èGläüæÖZäöðçÖräyÄäyIäZTäSÇçZDä■RçlNèrCäzæZläräÄC
 äy■èfGriijNäEšäZÖä■RçlNçZDäÄIæCšæYræ;Läd'ZæTäæaÑäZšçZDäšççäÄijN äNæNñ
 gevent, greenlet, Stackless Python äzæäRäLäEüäZÜçšZäijijäüèçlNäÄC

14.13 12.13 ad'ZäyIçZçlNéYšälÜè;öèrc

éÜöèçY

ä;äæIJL'äyÄäyIçZçlNéYšälÜèZEäRlriijNæCšäyZälRæIèçZDäEÇçt'æ;öèrcäöCäznriijN
 årseüšä;äayZäyÄäyIäöçæLüçnèrèüæSÇäÖzè;öèrcäyÄäyIç;SçZIJèfðæÖèçZEäRlçZDæÜZäijRäyÄæäüäÄC

ègçæEšæÜzæaL

ärZäZÖè;öèrcéÜöèçYçZDäyÄäyIäyYègAègçæEšæÜzæaLäy■æIJL'äyIä;LärSæIJL'äZççšèeAšçZDæLÄäü
 æIJnèr'läyLèöšäEüæÄIæCšärsæYriijZärZäZÖærRäyIä;äæCšèeAè;öèrcçZDèYšälÜriijNä;äälZäZäyÄärZèfðæ
 çDüäRÖä;äaIJlÄEüäy■äyÄäyIäèÜæÖèä■ÜäyLélçijÜäEZäZççäAæIèæäGèrEä■YäIJlçZDæTträ■öriijN
 äræad'ÜäyÄäyIäèÜæÖèä■ÜèçnäijäçZ select() æLÜçšZäijijçZDäyÄäyIè;öèrcæTträ■öälRè;çZDäG;æTträ

```
import queue
import socket
import os

class PollableQueue(queue.Queue):
    def __init__(self):
        super().__init__()
```

(continues on next page)

(continued from previous page)

```
# Create a pair of connected sockets
if os.name == 'posix':
    self._putsocket, self._getsocket = socket.socketpair()
else:
    # Compatibility on non-POSIX systems
    server = socket.socket(socket.AF_INET, socket.SOCK_
→STREAM)
    server.bind(('127.0.0.1', 0))
    server.listen(1)
    self._putsocket = socket.socket(socket.AF_INET, socket.
→SOCK_STREAM)
    self._putsocket.connect(server.getsockname())
    self._getsocket, _ = server.accept()
    server.close()

def fileno(self):
    return self._getsocket.fileno()

def put(self, item):
    super().put(item)
    self._putsocket.send(b'x')

def get(self):
    self._getsocket.recv(1)
    return super().get()
```

âĬĬĬZäyĭäzççäÄäy■ĭĭjNäyÄäyĭäŨrçŽD Queue äöđäĭŇçszäđNēcñäöŽäzL'ĭĭjNäzTāsCæYřäyÄäyĭäcñēf
âĬĬĬUnixæĬJžāZĭäyĭçŽD socketpair() äĜĭæTřèČĭèĭzæĬĭçŽDäĬZäzžēfZæäüçŽDäēŨæŖēä■ŨäĀC
âĬĬĬWindowsäyĭLēĬĭĭjNäĭäāfĒēāzäĭçTĭçszäĭĭjĭäzççäÄæĬæĭäæNšäöCäĀC
çDŭäŖŖŖäöŽäzL'æŽŖēÄŽçŽD get() äŖN put() æŨzæşTäĬĬĬZäzŽäēŨæŖēä■ŨäyĭLēĬæĭæL'gèāNĭ/OæŞ
put() æŨzæşTäĬĬĬäŖĬæTřæŖŖæTĭäĒēēYşäĬŨäŖŖŖäĭjZäĒZäyÄäyĭä■Tä■ŨēĬCäĬŖæŞŖäyĭäēŨæŖēä■Ũäy■ä
ēĀN get() æŨzæşTäĬĬĬäzŖēYşäĬŨäy■çğžēZd'äyÄäyĭäĒČt'äæŨäĭjZäzŖäŖäđ'ŨäyÄäyĭäēŨæŖēä■Ũäy■ä
fileno() æŨzæşTäĭçTĭäyÄäyĭäĜĭæTřæŖTäēC select()
æĭēēŖēfZäyĭēYşäĬŨäŖŖäzēēcñēŖēŖcäĀC äöČäzĒäzĒäŖĭæYřæŽt'ēĬJšäzĒäzTäsČēcñ
get() äĜĭæTřäĭçTĭäĬŖçŽDsocketçŽDäŨĜäzŭæŖŖēfŖçñçēĀNäŭšäĀC
äyNēĬæYřäyÄäyĭäĭNä■ŖĭĭjNäöŽäzL'äzĒäyÄäyĭäyžäĬŖæĬēçŽDäĒČt'äçZŞæŖĜäđ'ZäyĭēYşäĬŨçŽDäŭ

```
import select
import threading

def consumer(queues):
    '''
    Consumer that reads data on multiple queues simultaneously
    '''
    while True:
        can_read, _, _ = select.select(queues, [], [])
        for r in can_read:
```

(continues on next page)

(continued from previous page)

```
        item = r.get()
        print('Got:', item)

q1 = PollableQueue()
q2 = PollableQueue()
q3 = PollableQueue()
t = threading.Thread(target=consumer, args=(q1,q2,q3,))
t.daemon = True
t.start()

# Feed data to the queues
q1.put(1)
q2.put(10)
q3.put('hello')
q2.put(15)
...
```

æÇæðIä;æŕŦçIÄëŔëqNãöÇrijNä;äaijZãRŠçÖŕëŹäyŁæúŁët zèÄĖaijZæŎëãRŮãŁŕæL'ÄæIJL'çŽĐëćn

ëöleöž

ărzäžŎë;öëŕcéİdçsæŮĞäzũărzèşaijNærŦæÇéYşãLŮëĂŽäyŷëÇ;æYŕærŦë;ČæçYæL'ŇçŽĐëŮőëćYãÄĆ
ä;ŇäëÇrijNäëÇæðIä;äay■ä;ŁçŦlāyŁëİççŽĐäëŮæŎëã■ŮæŁÄæIJŕijN
ä;ääŦŕäyÄçŽĐëÄL'æNŦ'ärşæYŕçijŮãEžZäzčçäAæİëä;ŁçŎŕéA■ãŎEëŹäžŽéYşãLŮäzũä;ŁçŦlāyÄäyŁäöŽæŮüä

```
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)
```

ëŹæüüäÄŽäĖüäöđäy■ãŔŁçŔĖrijNëŹYäijŽaijŦäĖëäĖüäžŮçŽĐæĂğëČ;éŮőëćYãÄĆ
ä;ŇäëÇrijNäëÇæðIä;ŮŕçŽĐæŦŕæ■öëćnãŁääĖëãŁŕäyÄäyŁëYşãLŮäy■rijNëĞşărŞëëAëŁš10æŕŋçğŞæL■ëČ;è
æÇæðIä;ääžNãL'■çŽĐë;öëŕcéŹYëëAãŎžë;öëŕcãĖüäžŮărzèşaijNærŦæÇç;ŠçzIJäëŮæŎëã■ŮëČçëŹYäijZæL
ä;ŇäëÇrijNäëÇæðIä;äæČşãŔNæŮüë;öëŕcãëŮæŎëã■ŮäŠNëYşãLŮijNä;ääŔŕëČ;ëëAăČŕäyŇëİçëŹæüüä;Łç

```
import select

def event_loop(sockets, queues):
    while True:
        # polling with a timeout
        can_read, _, _ = select.select(sockets, [], [], 0.01)
        for r in can_read:
```

(continues on next page)

(continued from previous page)

```

    handle_read(r)
for q in queues:
    if not q.empty():
        item = q.get()
        print('Got:', item)

```

æfZäy!æŪzæqLéĀŽæfGārEēYšāLŪāSŅāēŪæŌēā■Ūc■L'āRŅārZā;ĒæIēēgčāEšzāzEāđ'gēCīāLEcZĐEŪō
 äyÄäy!ā■TcNŋcZĐ select() ērČcTīāRrēcŋāRŅāŪūcTīāIēē;ōērcāAC
 ā;ŁcTīlēŭEāŪūāLŪāEūāzŪāšzāzŌāŪūēŪr'čZĐāIJzāLŪāIēāL'gēāNāSīāIJšāĀgācĀāšēāzūāšqāIJL'āfĒēē
 cTŽēGšijNāēCādIJāTŗā■ōēcŋāLāāĒēāLŗāyÄäy!ēYšāLŪiijNāūLēr'zēĀĒāGāāzŌāRŗāzēāōđāŪūcZĐēcŋēĀ
 āŗ;çōāijZāIJL'äyĀcČzcZāzTāsČcZĐI/Oā■šēĀŪiijNā;ŁcTīāōČēĀŽāyyajZēŌūā;ŪāZr'āē;čZĐāS■āzTāŪ

14.14 12.14 aJÍUnixçszçšäyŁéíçáŘřáŁíáóŁæŁd'eÈZçÍN

éŮőécŸ

ä:äæČšcijŮãĚžÄyÄäyľä;ĲäyžäyÄäyľäĲĲUnixæĹŮčšzUnixčšzčžšäyĹéĺčèĹřèaŇčŽĐaŮĹæĹd'èĹŽčÍNèĹ

èġčǎẸșæŮźæąŁ

áĹZázžäYÄäyĹæ■čçaočŽDáoĹæŁd'èfŽćÍNéIJĀēēAäyÄäyĹčš;çaočŽDčšzčžšèrČćTíāžRÁĹŮāžēāRĹáržāž
 äyNéÍćčŽDžžččāAāsTčd'žžEæĀŌæūāōŽāzĹ'äyÄäyĹáoĹæŁd'èfŽćÍNijNāRfāžēāRfāĹlāRŌā;ĹáožæYŠčŽĹ

```
#!/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)

```

(continued from previous page)

```
raise SystemExit(1)

if sys.argv[1] == 'start':
    try:
        daemonize(PIDFILE,
                   stdout='/tmp/daemon.log',
                   stderr='/tmp/dameon.log')
    except RuntimeError as e:
        print(e, file=sys.stderr)
        raise SystemExit(1)

main()

elif sys.argv[1] == 'stop':
    if os.path.exists(PIDFILE):
        with open(PIDFILE) as f:
            os.kill(int(f.read()), signal.SIGTERM)
    else:
        print('Not running', file=sys.stderr)
        raise SystemExit(1)

else:
    print('Unknown command {!r}'.format(sys.argv[1]), file=sys.
→stderr)
    raise SystemExit(1)
```

èeAaRřaLlèfZäyIaóLæLd'èfZçlNijNçTlæLúeIJÄeAä;fçTlæCäyNçZDäS;äzd'iijZ

```
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
...
```

áoLæLd'èfZçlNäRřææáoNäÉlälJlãRÖäRřæRřæNijNäZæ■d'èfZäyIaS;äzd'aijZçnNä■sèfTäZdaÁĆ
äy■èfGijNä;ääRřææäCRäyLéÍcéCæäüæšçIJNäyÖáoCçZyãEšçZDpidæÜGäzüäŠNæUëäfÜäÁĆèeAäAIJæ

```
bash % daemon.py stop
bash %
```

èóIèőž

æIJnèLĆáoŽázL'ázEäyÄäyIaG;æTř daemonize() iijNäIJlçlNäZRaRřaLlæUüecnèrCçTlæ;fä;ÜçlNäZR
daemonize() äG;æTřaRlæÖëäRÜäEšéTöa■ÜäRCæTřijNèfZæäüçZDërläRréAL'äRCæTřaIJlècnä;fçTlæÜ
áoCäijZäijZäLüçTlæLüäCRäyNéÍcéfZæäüä;fçTláoCřijZ


```
daemonize('daemon.pid',
          stdin='/dev/null',
          stdout='/tmp/daemon.log',
          stderr='/tmp/daemon.log')
```

èĀNāy■æYřăĈRāyNéİcèŁZæăăRŃçŁăy■æyĔçŽĎërĈçŦīijŽ

```
# Illegal. Must use keyword arguments
daemonize('daemon.pid',
          '/dev/null', '/tmp/daemon.log', '/tmp/daemon.log')
```

ăĹZăžžăyĀăyĹăŌĹăŁd'èŁZĉĹNçŽĎă■éēĹd'çIJNāyĹăŌžăy■æYřăĹăYŖsæĠĈijNă;EæYřăĹd'gă;ŞæĀĹăĈ
 éēŪăĒĹijNāyĀăyĹăŌĹăŁd'èŁZĉĹNăĒĒăžĎēAăžŌĹŪēŁZĉĹNāy■èĎŖçēzăĀĈ ēŁZæYřçŦŖs
 os.fork() æŞ■ăIJăĹăăŌNăĹRçŽĎīijNăžŭçŋNă■şèçŋĹŪēŁZĉĹNçŽĹă■căĀĈ

ăIJĹă■RēŁZĉĹNăRŸăĹRă■d'ăĎĹăŖŌīijNërĈçŦī os.setsid()
 ăĹZăžžăyĒăyĹăĒĹăŪřçŽĎēŁZĉĹNăijŽērīijNăžŭēŌç;ŏă■RēŁZĉĹNāyžēēŪēēĒăĀĈ
 ăŌĈăijŽēŌç;ŏēŁZăyĹă■RēŁZĉĹNāyžăŪřçŽĎēŁZĉĹNçŽĎçŽĎēēŪēēĒijNăžŭçŋăŏăĹăy■ăijŽăĒă■IJĹăŌğăĹŪç
 ăēĈăĎIJēŁZăžŽăRŃăyĹăŌžăĎ'Ĺē■ŦăžžīijNăžăăyžăŏĈēIJăēēAăřĒăŏĹăŁd'èŁZĉĹNăRŃçŽĹĉŋŕăĹēççēzăijĀăžŭ
 ērĈçŦī os.chdir() ăŖŖ os.umask(0) æŦžăRŸăžĒă;ŞăĹă■ăŭēă;IJçŽŏă;ŦăžŭēĠç;ŏæŪĠăžŭăĹĈēŽŖæ
 ăĹŏæŦžçŽŏă;ŦēĀžăyŸăYřăyĹăē;ăyžăĎŖīijNăžăăyžēēŁZæăăRăžēă;Ĺă;ŪăŌĈăy■ăĒă■ăŭēă;IJăIJĹēēŋăRăĹăĹă

ăŖăĎ'ŪăyĀăyĹërĈçŦī os.fork() ăIJĹēŁZēĠNăZŦ'ăĹăçēĎçĠYçĈăăĀĈ
 èŁZăyĀă■ēă;Ĺă;ŪăŌĹăŁd'èŁZĉĹNăĎ'ŏăŌžăžĒēŌăăRŪăŪřçŽĎăŌğăĹŪçŽĹĉŋŕçŽĎēĈ;ăĹZăžŭăyŦēŏĹ'ăŏĈă
 īijĹăIJĹĹēĹăyĹīijNërēĎămonăŦ;ăijĈăžĒăŏĈçŽĎăijŽērĹēēŪēēĒă;Ōă;■īijNăžăă■d'ăĒăžşæşăēIJĹăĹĈēŽŖ
 ăr;çŏăq;ăăRăžēăĹ;çŦēēŁZăyĀă■ēīijNă;EæYřăIJăăē;ăy■ēēAēŁZăžĹăĀžăĀĈ

ăyĀăŪăŏĹăŁd'èŁZĉĹNēēŋă■ççăŏçŽĎăĹēççēzīijNăŏĈăijŽēĠ■æŪŕăĹăğNăŖŪăăĠăĠĒ/OăŦăĒăŖăĠăĠă
 èŁZăyĀăĈăĹăĹăēIJĹçĈēŽē;æĠĈăĀĈēŭşăăĠăĠĒ/OăŦăĈçŽăăĒşçŽĎăŪĠăžŭăŕŕžēşăçŽĎăijŦçŦīăIJĹēğçēĠă
 īijĹsys.stdout, sys.__stdout__ç■ĹīijĹăĀĈ ăžĒăžĒçŏăĀ■ŦçŽĎăĒşēŪ■
 sys.stdout ăžŭēĠ■æŪŕăŖăĠăŏĈăYřăqNāy■ăŽçŽĎīijN
 ăžăăyžăşăăĹăşŦçşēēAşăŏĈăYřăŖăēĒĹēĹēĈ;æYřçŦīçŽĎăYř sys.stdout ăĀĈ
 èŁZēĠNīijNăĹŖăžŋăĹŖăijĀăžĒăyĹă■ŦçNŋçŽĎăŪĠăžŭăŕŕžēşăīijNăžŭērĈçŦī os.
 dup2() īijN çŦīăŏĈăĹēăžçăŽēēŋ sys.stdout ă;ĹçŦīçŽĎăŪĠăžŭăŖŖēĹŕçŋēăĀĈ
 èŁZăăŭīijNsys.stdout ă;ĹçŦīçŽĎăŌşăğNăŪĠăžŭăijŽēēŋăĒşēŪ■ăžŭçŦŖăŪřçŽĎăĹăēŁŖă■căĀĈ
 èŁYēēAăijžērĈçŽĎăYřăžă;ŦçŦīăžŌăŪĠăžŭçijŪçăĀăĹŪăŪĠăIJăĎ'ĎçŖēçŽĎăăĠăĠĒ/OăŦăĒăYăijŽă

ăŌĹăŁd'èŁZĉĹNçŽĎăyĀăyĹēĀžăyŸăŏĎēŭŦăYřăIJăyĀăyĹăŪĠăžŭăy■ăĒăĒēēŁZĉĹNĎīijNăRăžēēēŋăĹă
 daemonize() ăĠ;æŦřçŽĎăIJăăŖŌēĈăĹăĒăĒăžĒăyĹăŪĠăžŭīijNă;EæYřăIJĹĹNăžŖçŽĹă■căŪŭăĹă
 atexit.register() ăĠ;æŦŖăşĹăĒăŖăyĀăyĹăĠ;æŦŖăIJĹPythoneğçēĠăžĹçŽĹă■căŪŭăĹăğăŋăĀĈ
 ăyĀăyĹăŕŕžăžŌSIGTERMçŽĎăĹăăŖăĎ'ĎçŖēăžĹçŽĎăŏžăžĹăŖăŖăăŭēIJăēēAēēŋăijYēZĔçŽĎăĒşēŪ■ăĀĈ
 ăĹăăŖăĎ'ĎçŖēăžĹçŏăĀ■ŦçŽĎăĹăăžăžē SystemExit() ăijĈăyŸăĀĈ
 ăĹŪēŏyēŁZăyĀă■ēçIJNāyĹăŌžăşăăĒēēAīijNă;EæYřăşăēIJĹăŏĈīijN
 çŽĹă■căĹăăŖăăijŽă;Ĺă;Ūăy■ăĹăğăŋ atexit.register()
 ăşĹăĒçŽĎăyĔçŖēăş■ă;IJçŽĎăŪŭăĀžăŕşăĹăăŌ'ăžĒēğçēĠăžĹăĀĈ
 ăyĀăyĹăĹăăŌ'èŁZĉĹNçŽĎă;Nă■RăžççăăăŖăžēăIJĹĹNăžŖăIJăăŖŌçŽĎ stop
 ăŖ;ăžĎ'çŽĎăş■ăIJăy■çIJNăĹăĀĈ

æZŦ'ăĎ'ŽăĒşăžŌçijŪăĒăŏĹăŁd'èŁZĉĹNçŽĎăĹăăŖăŖăžēăşēçIJNăĀĹUNIX
 çŖŕăçĈēŋYçžğçijŪçĹNăĀŊ, çŋăžŖçĹĹ by W. Richard
 Stevens and Stephen A. Rago (Addison-Wesley, 2005)ăĀĈ

är;çøãðČæŸřăĚşæşlăyŌCér■ēlĀçijŪćlŊiijŊă;EæŸřæL'ĀæIJL'çŽĐăĚăőžéČ;éĀČçŤlăžŌPythoniijŊăŽăăŷžæL'ĀæIJL'éIJĀèçAçŽĐPOSIXăĜ;æŤřéČ;ăŔřăžěăIJlăăĜăĜĚăžŞăy■æL'ĵăĹŕăĀĆ

15 çňňă■AăŷL'çnáiiijŽèĐŽæIJňçijŪćlŊăŷŌçşżçzşçóaçŘĚ

èőŷăđ'Žăžžă;£çŤlPythonă;IJăŷžăŷĂăŷłshellèĐŽæIJňçŽĐæŽăžçiiijŊçŤlăĬăőđçŌŕăŷŷçŤlçşżçzşăžžăĽă

Contents:

15.1 13.1 éĀŽè£ĜéĜ■ăőŽăŘŚ/çóăéAŞ/æŪĜăžúæŌěăŔŪèĴŞăĚě

éŪőéčŸ

ă;ăăŷŊæIJŽă;ăçŽĐèĐŽæIJňæŌěăŔŪăžžă;ŤçŤlăĽűèőđ'ăŷžæIJĀçóĀă■ŤçŽĐèĴŞăĚěæŪžăijŔăĀĆăŊĚæéĜ■ăőŽăŘŚæŪĜăžúăĹŕèřèèĐŽæIJňiijŊăĽŪăIJlăŚ;ăžđ'èăŊăŷ■ăiijăéĀŞăŷĂăŷłæŪĜăžúăŘ■æĽŪæŪĜăžúăŘ■

èĝčăĚşæŪžæăĴ

PythonăĚĚç;őçŽĐ fileinput æĴăăĽűèőŕ'è£ŽăŷłăŔŸăĴŪçóĀă■ŤăĀĆăçĀđIJă;ăæIJL'ăŷĂăŷłăŷŊéĬéç

```
#!/usr/bin/env python3
import fileinput

with fileinput.input() as f_input:
    for line in f_input:
        print(line, end='')
```

éĆčăžĽă;ăăŕşéČ;ăžžăĽ■ēĬææŔŔăĹŕçŽĐæL'ĀæIJL'æŪžăijŔăĬăăŷžæ■đ'èĐŽæIJňæŔŔă;ŽèĴŞăĚěăĀĆăĀfilein.pyăžžăŕĚăĚűăŔŸăŷžăŔŕæL'ĝèăŊæŪĜăžúiiijŊ éĆčăžĽă;ăăŔřăžěăČŔăŷŊéĬéç£ŽăăŷłèŕČçŤlăőČiijŊ

```
$ 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.
```

èőĬèőž

fileinput.input()ăĽŽăžžăžűè£ŤăŽđăŷĂăŷł FileInput çşżçŽĐăőđăĴŊăĀĆèŕèăőđăĴŊéŽđ'ăžĚæŊěæIJL'ăŷĂăžŽæIJL'çŤlçŽĐăŷőăĽl'æŪžæşŤăđ'ŪiijŊăőČè£ŸăŔŕèçŋă;ŞăĀŽăŷĂăŷłăŷĽăăŽăă■đ'iijŊăŤŕ'ăŔĽèŭăĬèiijŊăçĀđIJăĽŚăžŋèçĀăĚŽăŷĂăŷłæL'Şă■ŕăđ'ŽăŷłæŪĜăžűèĴŞăĜžçŽĐèĐŽæIJň

```
>>> import fileinput
>>> with fileinput.input('/etc/passwd') as f:
>>>     for line in f:
...         print(f.filename(), f.lineno(), line, end='')
```

(continues on next page)

(continued from previous page)

```

/etc/passwd 1 ##
/etc/passwd 2 # User Database
/etc/passwd 3 #

<other output omitted>

```

éÅŽefGärEäöČä;IJäyžäyÄäyläyLäyNæŮĞçöaçŘEäZlā;ŁçŤlīijNāRfāžēçāōāŁlāōČäy■aE■ä;ŁçŤlæŮūæŮ
èĀNäyTæŁSāznāIJlāžNāRŌēŁŸæijŤčd'žāžEFileInputčŽDäyÄāžZæIJŁçŤlçŽDäyōāŁlæŮžæŤlæīēēŌū

15.2 13.2 çŁæ■ćłÍŃăŹŔăŹŮçŁŹăĞŹéŤŹèŕŕăĖăæŦŕ

éŮóécŸ

ä;äæČšăŔŚăăĞăĜĖĕŤŽĕŕŕæĽŠă■ŕăyĂæĬæūĹæĀŕăžūēĤŤăŽďæšŔăyĹēĬďéŽūčĹūæĂĀçăĀæĭčžĹæ■čĹ

èġčǎẸșæŮźæąŁ

ä;äæIJL'äyÄäy!çlNäžRâČRäyNé!çèfZæäüçzLæ■çijNæLZâGžäyÄäy! SystemExit
äijÇäyÿijNä;ççTléTžèrræüLæAřä;IJäyžâRČæTřâĂČä;NäèÇijŽ

```
raise SystemExit('It failed!')
```

ăŃăĭjŽăřĚæŭŁæAřăĬJ sys . st derr äy■æL'Să■riijŃcĐŭăRŔŃŃlŃăžRăzēcŁŭæĂAçăA1éĂĂăGžăĂĆ

ěóíěž

æIñèŁĆèŽ;čĎůǻ;Łç§■ǎřRiiŋNǎ;EǣYřǻŌČěČ;ěğcǻEǵsǻIJǻǻEZēDŽæIJnæUűčŽĎäyǺǻlǻyǻyǻyǻğǻéUőécYǻ
ǻžšǻřsǻYřérťiiŋNǻ;Sǻ;ǻǻčšèeAčZLǻ■ǻčšRǻyłčłNǻžRǻUűiiiŋNǻ;ǻǻRřėČ;ǻijŽǻČRǻyNéİčēfZǻǻǻǻEŽiiŋŽ

```
import sys
sys.stderr.write('It failed!\n')
raise SystemExit(1)
```

```

    æĈædIJä;ăçZt'æŌëârEæúLæAřä;IJäyžăŔĈæȚăriăjăçzŽ
    SystemExit()
iijNéĈcăžLă;ăăŔăřăžëçIJAçȚăăĚüăžŪæ■ēld'iijN
    æřăĈăĈim-
portèr■ăŔëăĹŪăřEĚȚZërăăúLæAřăĚZăĚë sys.stderr

```

15.3 13.3 èğçæđŘăŚĵăzd'èąŃéĂL'éąż

éŮóécỲ

ä;äçŽĎçÍŇăŹŔăĉĆă;ŤĕĈ;ăđ'šĕğĉăđŔăŜ;ăzd'ĕăŇĕĂĹ'ĕăziiĲă;■ăžŮsys.argvăy■iiĲ

èġčǎẸșæŮźæąŁ

argparse ælɑɑlʊɑRrɛcncʰlæiɛəgcædRɑSjɑzd'ɛɑNɛʌl'ɛɑzɑʌCɑyNɛlɑyʌyɫɔʌɑTɑjNɑRɑɪjTɕd'z

```
# 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)
```

ěřečlŇāžŘāōŽāzL'āžEäyÄäyIaeĆäyNä;ŁćTlćŽDāŚ;āzd'eaNěgċædŘāŽliiž

```
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:
  -h, --help            show this help message and exit
  -p pattern, --pattern pattern
                        text to search for
  -v, --verbose          print more information
  -o OUTFILE, --out OUTFILE
                        file to write results to
  --speed {slow,fast}   search speed (slow or fast)
```

(continues on next page)

(continued from previous page)

```
-h, --help            show this help message and exit
-p pattern, --pat pattern
                        text pattern to search for
-v                    verbose mode
-o OUTFILE            output file
--speed {slow,fast}   search speed
```

äyÑéÍççŽĎĎČlálĚæijŤčď'žāžĚçlŇāžŘäy■çŽĎæŤřæ■óéČlálĚāĎČäzŤčžĚğČārſprint()èr■āŘēçŽĎæLſ

```
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
```

āržāžŎéĀL'éāžāĀijçŽĎĎžŽäyĀæ■ēād'ĎçŘĚçŤſçlŇāžŘælēāĚſāōŽiijŇçŤlā;āèĠtāūsçŽĎĎĀžèçſælēæŽĎ
print() āĠ;æŤřāĎČ

èóíèőž

argparse ælāāIŮæŸřæāĠāĠĚāžſäy■æIJĀād'ğçŽĎælāāIŮāžŇäyĀiijŇæŇæIJL'ād'ğéĠŖçŽĎĎĚ■ç;óé
æIJñèLČāŖlæŸřæijŤčď'žāžĚāĚūäy■æIJĀāſžçāĎŽĎäyĀāžŽçL'žæĀġiijŇäyōāLſ'ā;āāĚēēŮlāĎČ

äyžāžĚğçæďŘāſ;āzd'ēāŇéĀL'éāžiiŇNā;āēçŮāĚĬlèçAāĬZāžžäyĀäyſ
ArgumentParser āōđäçŇiijŇ āžūā;ççŤl add_argument()

```
parser.add_argument(dest='filenames', metavar='filename', nargs='*')
```

```
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('-p', '--pat', metavar='pattern', required=True,
                    dest='patterns', action='append',
                    help='text pattern to search for')
```

```
parser.add_argument('--speed', dest='speed', action='store',
                    choices={'slow', 'fast'}, default='slow',
                    help='search speed')
```

509

15.4 13.4 è£ŘèàÑæÙúàijzàGžárEçăAèŁŞăĚěæRŘçd'ž

éŮóécŸ

ä;ääEŽāZēÄy!èĎŽæIJñijNè£ŘèàÑæÙúéIJĀèēAäyĀäy!ārEçăAāĀĆæ■d'èĎŽæIJñæŸřāzd'āžŠāijRçŽĎñij
èĀÑæŸřéIJĀèēAāijzàGžāyĀäy!ārEçăAèŁŞăĚěæRŘçd'žñijNèđŁ'çŤ!æLûèG!āûsèŁŞăĚěāĀĆ

èğcāEşæŮzæaĹ

è£ŽæŮúāĀŽPythonçŽĎ getpass æ!āā!Ůæ■çæŸřä;āæL'ĀéIJĀèēAçŽĎāĀĆä;āāRřāzèèđŁ'ä;āāŁèŁzæ!Ł
āžŮāyŤāy■āijŽāIJŁŤ!æLûçzŁčnrāZđæŸŁārEçăAāĀĆāyNé!çæŸřāEūā;ŞăžççāAñijŽ

```
import getpass

user = getpass.getuser()
passwd = getpass.getpass()

if svc_login(user, passwd):      # You must write svc_login()
    print('Yay!')
else:
    print('Boo!')
```

āIJ!æ■d'āžççāAäy■ñijNsvc_login() æŸřä;āèēAāđđçŮřçŽĎāđ'ĎçRĚārEçăAçŽĎāG;æŤñijNāEūā;Şç

èó!èőž

æş!æĎRāIJ!āL'■é!çāžççāAäy■ getpass.getuser()
äy■āijŽāijzàGžçŤ!æLûāR■çŽĎèŁŞăĚěæRŘçd'žāĀĆ āđČāijŽæāžæ■đèřčŤ!æLûçŽĎshel-
lçŮřāçCæLŮèĀĚāijŽä;Īæ■đæIJñāIJçşzçzşçŽĎārEçăAāžŞñijLæŤřæÑĀ pwd
æ!āā!ŮçŽĎāžşāRñijLæ!ēā;ŁçŤ!ā;ŞāL'■çŤ!æLûçŽĎçŽzā;ŤāR■ñijN

āēČæđIJā;āæČşæŸŁçd'žçŽĎāijzàGžçŤ!æLûāR■èŁŞăĚěæRŘçd'žñijNā;ŁçŤ!āĚĚç;řçŽĎ
input āG;æŤñijŽ

```
user = input('Enter your username: ')
```

è£ŸæIJL'äyĀçČzā;ŁéG■èēAñijNæIJL'āžŽçşzçzşāRřèČ;äy■æŤřæÑĀ getpass()
æŮzæşŤéŽŘèŮRèŁŞăĚěārEçăAāĀĆ è£Žçğ■æČĚāĚāyNñijNPythonāijŽæRŘāL'■è■æāŚ!ä;āè£ŽāžŽéŮóécŸñij

15.5 13.5 èŮāRŮçzŁčnrçŽĎāđ'ğārR

éŮóécŸ

ä;āéIJĀèēAçşééAŞā;ŞāL'■çzŁčnrçŽĎāđ'ğārRāzèä;Łæ■ççāōçŽĎæāijāijRāNŮèŁŞăGžāĀĆ

[illegible]

```
try:
    out_bytes = subprocess.check_output(['cmd', 'arg1', 'arg2'],
    ↪ timeout=5)
except subprocess.TimeoutExpired as e:
    ...
```

```
out_bytes = subprocess.check_output('grep python | wc > out',  
    ↪ shell=True)
```

èóíèőž

```
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)
```

512


```
# To interpret as text, decode
out = stdout.decode('utf-8')
err = stderr.decode('utf-8')
```

```
subprocess ælɑɑlUɑrʒɑžŌä; ietŰTTYçŽDɑd' ŰéClɑS; äzd' äy■āRLéĀĆçTlāĀĆ
ä; NāēĆiijNä; ääy■ēČ; ä; fçTlāōČæIēēGlaLlāNŰäyÄäyłçTlæLūē; ŠāĒēārEçāAçŽDäzzāŁajijLærTāēCäyÄäyłs
ēfZæŰūāĀŽiijNä; äēIJĀēēAä; fçTlāLřçññäyL æŰžælɑɑlUāžEiijNærTāēCāšžāžŌēSŰāŘ■çŽD
expect āōūæŰRçŽDāūēāĒūiijLpexpectæLŰçšžaiijçŽDiiijL
```

15.7 13.7 ād'■āLŰæLŰēĀĒçgžāLlæŰĠäzūāŠNçŽōā;T

éŰōécŸ

```
ä; äæČšēēAād'■āLŰæLŰçgžāLlæŰĠäzūāŠNçŽōā;TiiijNä; EæŸrāRLäy■æČšēēČçTlshellāS; äzd' āĀĆ
```

ēğcāEşæŰžæqL

```
shutil ælɑɑlUæIJL'ā; Lād' Žä; fæ■ūçŽDāG; æTřāRřäžēād'■āLŰæŰĠäzūāŠNçŽōā;TāĀĆä; fçTlētūæIēē
```

```
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)
```

```
ēfZāžZāG; æTřçŽDāRCæTřēČ; æŸrā■Űçñēäyšā; cāijRçŽDæŰĠäzūæLŰçŽōā;TāŘ■āĀĆ
āžTāsĆēr■āžL ælææNšāžEçšžaiijçŽDUnixāS; äzd' iijNāēCäyLēlççŽDæšléGlēClāLēāĀĆ
```

```
ézŸēōd' æČĒāEłäyNiiijNāržāžŌçñēāRūēŠ; æŌēēĀNāūsēēfZāžZāS; äzd' ād' DçŘEçŽDæŸrāōČæNĠāŘSçŽ
ä; NāēĆiijNāēCādIJæžRæŰĠäzūæŸrāyÄäyłçñēāRūēŠ; æŌēiijNēČčāžLçŽōæāGæŰĠäzūārEäijŽæŸrçñēāRūē
āēCādIJä; āāRlæČšād'■āLŰçñēāRūēŠ; æŌēēIJñēžñiijNēČčāžLēIJĀēēAæNĠāōŽāĒšēTōā■ŰāRCæTř
follow_symlinks ,æČäyNiiijŽ
```

```
āēCādIJä; äæČšāfiçTžēćnād'■āLŰçŽōā;Täy■çŽDçñēāRūēŠ; æŌēiijNāČRēfZæāūāĀŽiijŽ
```

```
shutil.copytree(src, dst, symlinks=True)
```

```
def ignore_pyc_files(dirname, filenames):
    return [name in filenames if name.endswith('.pyc')]

shutil.copytree(src, dst, ignore=ignore_pyc_files)
```

```
shutil.copytree(src, dst, ignore=shutil.ignore_patterns('*~', '*.pyc',  
↳ ''))
```

ä;fçTl shutil ad'■áLúæŮGäzúáSÑçZóä;TäzşäſŞçöÄ■TäzEçCzâRgāĂC
 äy■ēfGriiŃārſzāžŌæŮGäzúāĖĈæTŗæ■ōāfæAŗiiŃcopy2() ēfZæāũçZĎāG;æTŗāRlēĈ;ār;ēGlāũsæIJĀad'gē
 òðſēŮðæŮūēŮŗāĀAāLZāzæŮūēŮŗāSŃæiĈēZŖēfZāzZāşzæIJnāfæAŗaijZēcñāfIçTZriiŃ
 ä;EæYŗārſzāžŌæL'ĀæIJL'ēĀĖāĀACLsāĀAēTĎæzŖforkāSŃāĖŮūāzŮæZŗæũsāsĈæñaçZĎæŮGäzúāĖĈcāfæA
 èſZāyſēYā;Ůā;ſetŮāzŌāzTāsĈæŞ■ä;IJçşzçzşçşzādŃāSŃçTlæLúæL'ĀæNēæIJL'çZĎēðſēŮðæiĈēZŖāĂC
 ä;āēĀZāyſy■āijZāŌzā;fçTl shutil.copytree() āG;æTŗælēæL'gēāŃçşzçşşād'Gäz;āĂC
 ä;Şād'DçRĖæŮGäzúāR■çZĎæŮūāĀZiiŃNæIJĀāē;ä;fçTl os.path
 äy■çZĎāG;æTŗælēçāōāfIæIJĀad'gçZĎāRfçgçzæd'■æĀgriiŃLçL'zāLŃæYŗāRŃæŮūēēAēĂĈçTlāžŌUnixāSŃW
 ä;NāçĈiiŃZ

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

```
try:
    shutil.copytree(src, dst)
except shutil.Error as e:
```

```
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)
```

æIJñèŁĆæijTčd'žćŽĐèŁžăžŽăĜ;æTřéČ;æYřæIJăÿÿèĜAçŽĐăĂĆăÿ■èŁĜiijNshut il
èŁYæIJL'æŽř'ăd'ŽćŽĐăŠŇăd'■ăĹúæTřæ■ôćŽÿăĚšćŽĐăŠ■ăIJăĂĆ
ăôĆćŽĐăŮĜăeăă;ĹăAijă;ŮăÿĂćIJŇiijNăRĆèĂĆ Python documentation

ēfZāzZāG;æTřēfYæIJL'ā;Łāđ'ZāĚüāzŮēĀL'ēāzīijNčTlāžŌæŮēāfŮæL'Sā■řāĀēčĎāčĀāĀAæŮGāzŮā
āŔCēĀČ shutilæŮGāæč

15.9 13.9 éĀŽēŁGæŮGāzŮāR■æšēæL'ġæŮGāzŮ

éŮōēčY

ä;äēIJĀēēAāĚZāyĀäylæŮL'āŔLāLŔæŮGāzŮæšēæL'ġæS■ä;IJčZĎēĎŽæIJñijNāēŤāēČāŕzæŮēāfŮā;Šæā
ä;äy■æČšāIJPythonēĎŽæIJñäy■ērČčTlshellīijNāēLŮēĀĚä;āēēAāōđčŌŕāyĀāžZshelläy■ēČ;āAŽčZĎāŁšēČ

ēğčāEšæŮzæāŁ

æšēæL'ġæŮGāzŮīijNāŔŕā;ŁčTl os.walk() āG;æTřīijNāijāyĀäylēāŮčžğčZōā;ŤāR■čZāōČāĀČ
āyNēlæYŕāyĀäylā;Nā■RīijNæšēæL'ġčL'zāōŽčZĎæŮGāzŮāR■āzŮč■ŤāžŤæL'ĀæIJL'čņēāŔLæIāāzŮčZĎæŮ

```
#!/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])
```

āfIā■YēĎŽæIJñäyžæŮGāzŮfindfile.pyīijNčĎŮāŔŌāIJlāS;āzd'ēāNäy■æL'gēāNāōČāĀČ
æŇGāōŽāLlāğNæšēæL'ġčZōā;ŤāžēāŔLāR■ā■Ůā;IJāyžā;■č;ōāŔCæTřīijNāēČāyNīijZ

ēōlēōž

os.walk() æŮzæšŤäyžæŁSāzñēA■āŌēčZōā;ŤæāSīijN
æŕŔæñæēZāĚēāyĀäylčZōā;TīijNāōČāijZēfŤāZđāyĀäylāyL'āĚČčzĎīijNāNĚāŔñčZyāržāžŌæšēæL'ġčZōā;Ť
āžēāŔLēČčāylčZōā;ŤāyNēlččZĎæŮGāzŮāR■āLŮēāfāĀČ

āržāžŌæŕŔāylāĚČčzĎīijNāŔlēIJĀæčĀæŤNāyĀäyNčZōæāGæŮGāzŮāR■æYŕāŔēāIJlæŮGāzŮāLŮēāfāy■
os.path.join() āŔLāzŮēŮŕā;ĎāĀČ äyžāžēēAēāĚ■āēGæĀłčZĎēŮŕā;ĎāR■æŕŤāēČ ./
./foo//bar īijNā;ŁčTlāžēāŔēāđ'Ůāyđ'āylāG;æTřælēāfōæ■ččzSāđIJāĀČ čññāyĀäylæYŕ
os.path.abspath() āōČæŌēāŔŮāyĀäylēŮŕā;ĎīijNāŔŕēČ;æYŕčZyāržēŮŕā;ĎīijNæIJĀāŔŌēfŤāZđčzIā
čññāžNāylæYŕos.path.normpath() īijNčTlælēēfŤāZđæ■čāyŷēŮŕā;ĎīijNāŔŕāžēēğčāEšāŔNæŮIJæIEā

ār;čōāēfZāylēĎŽæIJñčZyāržāžŌUNIXāzšāŔŕāylēlččZĎā;Łāđ'ZæšēæL'ġælēēōšēēAçōĀā■Ťā;Łāđ'Zīij
āzŮāyTīijNēfYēČ;ā;Łē;zæĲčZĎāLāāĚēāĚüāzŮčZĎāŁšēČ;āĀČ
æŁSāzñāĒ■æijŤčđ'zāyĀäylā;Nā■RīijNāyNēlččZĎāG;æTřæL'Sā■ŕæL'ĀæIJL'æIJĀēfSēčñāfōæŤžēfGčZĎæŮ

```
#!/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]))
```

āIĴā■d'āĜ;æTŗçŽDā\$žçāĀāzNāyLūijNā;ŁçTĴos,os.path,globç■ŁçšzāijjæĴāāIŪijNā;āāršèĈ;āōđçŌřæŽŲ
āRřāRĈèĀĈ5.11ārRèŁĈāŠŅ5.13ārRèŁĈç■ŁçŽyāĒşçñăèŁĈăĀĈ

15.10 13.10 èrzāRŪéĒ■ç;őæŪĜäzŪ

éŪőéćŸ

æĀŌæăüèrzāRŪæŽőéĀŽ.iniaēāijāijRçŽĎéĒ■ç;őæŪĜäzŪrijš

èĝčāĒşçæŪzæāĴ

configparser æĴāāIŪèĈ;èćñçTĴæĴèèrzāRŪéĒ■ç;őæŪĜäzŪāĀĈä;ŅāçĈijNāAĜèđç;ä;ăæIJL'æçĈäyŅç

```
; config.ini
; Sample configuration file

[installation]
library=%(prefix)s/lib
include=%(prefix)s/include
bin=%(prefix)s/bin
prefix=/usr/local

# Setting related to debug configuration
[debug]
```

(continues on next page)

(continued from previous page)

```
log_errors=true
show_warnings=False

[server]
port: 8080
nworkers: 32
pid-file=/tmp/spam.pid
root=/www/root
signature:
=====
Brought to you by the Python Cookbook
=====
```

äyÑéÍcæYřäyÄäylerzâRŮŮâŠŇæRRâRŮŮâĚŮäy■âÄijçŽĎä;Nâ■ŘijŽ

```
>>> 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
\=====
>>>
```

âĖČæđIæIJL'éIJÄĖAřijNâ;æĖYĖČ;ăĖŏăŤzéĚ■ç;ŏăžŮä;ĖçŤÍ
æŮžæşŤăŖĖâĚŮâĖŽăŽďăĹŕæŮĜăžŮäy■ăĂĆă;NăĖĆijŽ

cfg.write()

```
>>> cfg.set('server', 'port', '9000')
>>> cfg.set('debug', 'log_errors', 'False')
>>> import sys
>>> cfg.write(sys.stdout)
```

```
[installation]
library = %(prefix)s/lib
include = %(prefix)s/include
bin = %(prefix)s/bin
```

(continues on next page)

(continued from previous page)

```
prefix = /usr/local

[debug]
log_errors = False
show_warnings = False

[server]
port = 9000
nworkers = 32
pid-file = /tmp/spam.pid
root = /www/root
signature =
=====
Brought to you by the Python Cookbook
=====
>>>
```

ðóíèõž

éĚ■;øæŮĜäzŭä;IJäyžäyÄçġ■āRřèræÄġāŁŁäē;çŽĎæäijäijRiijNéIdäyÿéÄĆçŤlāzŎā■ŸāĆlćlNāžRäy■;ç
āIJlærRäyĥéĚ■;øæŮĜäzŭäy■iijNéĚ■;øæŤræ■ōäijŽècñāĹēçzDiiJLæfŤæĆäŁNā■Räy■çŽĎāÄIInstallationā
āÄIdebugāÄI āŠN āÄIserverāÄIiijL'āÄĆ æfRäyĥāĹēçzDāIJlāĚŭäy■æNĠāōŽārzažŤçŽĎāRĎäyĥāRŸéĠRāÄ

ārzažŎāRřāōđçŎrāRñæäŭāŁšèÇ;çŽĎéĚ■;øæŮĜäzŭāŠNPythonæžRæŮĜäzŭæŸræIJL'ā;Łād'ġçŽĎäy■
éēŮāĚLiijNéĚ■;øæŮĜäzŭçŽĎēr■æšŤèçAæŽr'èĠçŤsäzŽiijNäyNéĹççŽĎētNāÄijēr■āRēæŸrç■L'æŤĹçŽĎiij

```
prefix=/usr/local
prefix: /usr/local
```

éĚ■;øæŮĜäzŭäy■çŽĎāR■ā■ŮæŸrāy■āNžāĹēād'ġārRāēZçŽĎāÄĆäŁNāēÇiijŽ

```
>>> cfg.get('installation','PREFIX')
'/usr/local'
>>> cfg.get('installation','prefix')
'/usr/local'
>>>
```

āIJĹēġçæđRāÄiijçŽĎæŮŭāÄŽiijNgetboolean() æŮžæšŤæššæL'äzzä;ŤārřæāNçŽĎāÄijaÄĆäŁNāēÇ

```
log_errors = true
log_errors = TRUE
log_errors = Yes
log_errors = 1
```

æĹŮēöÿéĚ■;øæŮĜäzŭāŠNPythonäžççāAæIJÄād'ġçŽĎäy■āRñāIJlāzŎiijNāōCāzŭäy■æŸrāzŎäyĥēÄñ
æŮĜäzŭæŸrāōL'èçĚäyÄäyĥæŤr'ā;ŠècñèrzaRŮçŽĎāÄĆāēÇæđIJççrāĹrāžEāRŸéĠRæZĤæ■çiijNāōCāōđéŽĚā
ä;ŁāēÇiijNāIJlāyNéĹçēZäyĥéĚ■;øäy■iijNprefix āRŸéĠRāIJlā;ĤçŤĹāōÇçŽĎāRŸéĠRāzNāL'■æĹŮäzNāf

```
[installation]
library=%(prefix)s/lib
include=%(prefix)s/include
bin=%(prefix)s/bin
prefix=/usr/local
```

ConfigParser æIJL'äylåóæYŞècñá;ègEçZĐçL'záĀgæYřáóČèČ;äyĀæñæřzâRŮâd'ŽäyléĚ■ç;óæŮ
ä;NâeČiijNâAĞèö;äyĀäylçTlæLûâČRäyNéIcéfZæâũædDéĀäazEäzŮäznçŽĐéĚ■ç;óæŮGäzŭijŽ

```
; ~/.config.ini
[installation]
prefix=/Users/beazley/test

[debug]
log_errors=False
```

èřzâRŮèfŽäylæŮGäzŭijNâóČârseČ;èu\$ázNâL'■çŽĐéĚ■ç;óâRLázüètũæIěãĀČæČiijŽ

```
>>> # 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
>>>
```

äzTçzEègČârşäyN prefix âRŸéGRæYřæĀŌæâũèeEçZŮâĚüazŮçZyâĚşâRŸéGRçZĐiijNærTæČ
library çZĐèö;åóŽâĀijăĀĆ äžgçTşèfZçg■çzŞædIJçZĐâŌşâZæYřâRŸéGRçZĐæTzâEžEĞGâRŮçZĐæ
ä;ââRřäzëâČRäyNéIcéfZæâũâAŽerTélNiiijŽ

```
>>> cfg.get('installation', 'library')
'/Users/beazley/test/lib'
>>> cfg.set('installation', 'prefix', '/tmp/dir')
>>> cfg.get('installation', 'library')
'/tmp/dir/lib'
>>>
```

æIJĀâRŌèfYæIJL'â;LéG■èeAäyĀçČzèeAæşláĐRçZĐæYřPythonázüäy■èČ;æTřæNĀ.iniaŮGäzŭaIJlâ
çåöâfIä;âũşçzRâRĆéYĚäžEçconfigparseræŮGæaçäy■çZĐèf■æşTèřæČĚäzëâRLæTřæNĀçL'záĀgăĀĆ

15.11 13.11 çŻćŃĀ■TèĎŽæIJñáćđāŁăæŮěăŁŮāŁšèĈ;

éŮóécŸ

ä;ääŸNæIJŽāIJlèĎŽæIJñāŠŇćÍŇāžRäŸ■ārĒērŁæŮ■ăŁæAŕāĒŽāĒĒæŮěăŁŮæŮĜāžŰāĀĆ

èğĉăĒşæŮžæąŁ

æŁŠā■ŕæŮěăŁŮæIJĀćŃĀ■TæŮžāijRæŸŕä;ŁćŦÍ logging æĹāāIŮāĀĆă;ŇāĒĆŕijŽ

```
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()
```

äŸŁéĬăžŦäŸĹæŮěăŁŮērĈćŦŕijŁcritical(), error(), warning(), info(),
debug()ŕijŁăžééŽ■ăžRæŮžāijRēāĈđ'žäŸ■āŖŇćŽĎäŸéĒĜ■ćžġāŁŇāĀĆ
basicConfig() ćŽĎ level āŖĆæŦŕæŸŕäŸĀäŸĹēŁĜæzd'ăŽĹāĀĆ
æŁĀæIJŁćžġāŁŇā;ŎăžŎæ■đ'ćžġāŁŇćŽĎæŮěăŁŮăŮĹæAŕēĈ;āijŽēćŇāŁ;ćŦĒæŎĹāĀĆ
æŕŖäŸĹloggingæŠ■ă;IJćŽĎāŖĆæŦŕæŸŕäŸĀäŸĹăŮĹæAŕā■ŮĉŇæŸŕijŇāŖŎéĬăĒēŮşäŸĀäŸĹæĹŮăđ'ŽäŸĹāŖĆ
æđĎēĀăæIJĀćŽŁćŽĎæŮěăŁŮăŮĹæAŕćŽĎæŮŰāĀŽæĹŠāžŇā;ŁćŦĹăžĒ%æŠ■ă;IJĉŇæĹēæāijāijRāŇŮăŮĹæA

ēŁŖēāŇēŁŽäŸĹćÍŇāžRāŖŎŕijŇāIJĹæŮĜāžŰ app.log äŸ■ćŽĎāĒēăŃăžŦērēæŸŕäŸŇéĬēŁŽæăŮŕijŽ

```
CRITICAL:root:Host www.python.org unknown
ERROR:root:Could not find 'spam'
```

ăĒĆăđIJă;ăæĈşæŦžāŖŸē;ŠăĜžć■ŁćžġŕijŇă;ăāŖŕäžēăŁŃæŦž basicConfig()
ērĈćŦĹäŸ■ćŽĎāŖĆæŦŕāĀĆă;ŇāĒĆŕijŽ

```
logging.basicConfig(
    filename='app.log',
    level=logging.WARNING,
    format='%(levelname)s:%(asctime)s:%(message)s')
```

æIJĀăŔŎë;ŞăĠzăŔŸæĹŔăĉCăyŊiijŽ

```
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
```

äyĹéĬçŽĐæŮëăŮéĚ■ç;őéČ;æŸŕçăŋçijŮçăAăĹŕçĹŊăzŔăy■çŽĐăĂĉăĉCăđIJă;ăăČşă;ĚçŤĹéĚ■ç;őăŮČ
ăŔŕăzëăČŔăyŊéĬçēŹæăüăĚăŤž basicConfig() ëŕČçŤĹiijŽ

```
import logging
import logging.config

def main():
    # Configure the logging system
    logging.config.fileConfig('logconfig.ini')
    ...
```

ăĹŹăzžăyĂăyĹăyŊéĬçēŹæăüçŽĐæŮĠăzŭiijŊăŔ■ă■ŮăŔă logconfig.ini iijŽ

```
[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
```

ăĉCăđIJă;ăăČşăĚăŤžéĚ■ç;őiijŊăŔŕăzëçŽŕ æŎëçijŮë;ŞăŮĠăzŭlogconfig.iniă■şăŔŕăĂĈ

èõléõž

ărjçõårzäžÕ logging ælaaiUèĀŃaušæIJL'â;Lād'ŽæZt'énYçžğçŽDěĚ■ç;õéĀL'ëazīijŃ
äy■ēfĜēfŽēĜŃçŽDæŪzæāLārzäžÕçõĀ■TçŽDçĪNāzRāŠŃēDŽæIJŃaušçzRēūšād' šāžEāĀC
ārĪæČšāIJlērČçTĪæŪēāfŪæŞ■ä;IJāL'■āĒLæL'gèaŃäyŃbasicConfig()āĜ;æTŗæŪzæşTīijŃä;ăçŽDçĪNāzRāršē
ăçCædIJä;ăæČşēAä;ăçŽDæŪēāfŪæūLæAŗāEŽāLŗæăĜăĜEēTŽēŗrāy■īijŃēĀŃäy■æYŗæŪēāfŪæŪĜăz
basicConfig() æŪūäy■āijæŪĜăzūāR■ārCæTŗā■şāRŗāĀCă;ŃăçCīijŽ

```
logging.basicConfig(level=logging.INFO)
```

basicConfig() āIJĪĪNāzRāy■ārĪēC;ècŃæL'gèaŃäyĀæŃāāĀCăçCædIJä;ăçĪ■ārŌæČşæTŗāRŸæŪēā
āršēIJăēçAāĒLēŌūāRŪ root logger iijŃçDūāRŌçŽt' æŌēāfōæTŗāōCăĀCă;ŃăçCīijŽ

```
logging.getLogger().level = logging.DEBUG
```

éIJĀēçAāijžērČçŽDæYŗæIJŃēLČārĪæYŗæijTçd'žāžE logging
ælaaiUçŽDäyĀāžZāşzæIJŃçTĪæşTāĀC āōCārŗāzēāAŽæZt'ād'ŽæZt'énYçžğçŽDāōZāLūāĀC
āĒşāžŌæŪēāfŪāōZāLūāŃŪäyĀäyĪā;Lāē;çŽDçTŗæZŗæYŗ Logging Cookbook

15.12 13.12 çžZāĜ;æTŗāžŞăcdāLăæŪēā£ŪāŁşèČ;

éŪōécŸ

ä;ăæČşçZæŞRāyĪāĜ;æTŗāžŞăcdāLăæŪēāfŪāŁşèČ;īijŃä;EæYŗāRĪLāy■ēČ;ā;şāŞ■āLŗēCčāžZäy■ä;£çTĪ

èğcāEşæŪzæāĪ

ārzäžŌæČşēAæL'gèaŃæŪēāfŪæŞ■ä;IJçŽDāĜ;æTŗāžŞēĀŃaušīijŃä;ăāžTēŗēāLZăžzäyĀäyĪayŞăşdcŽD
logger āržēşāīijŃāzūāyTāČRāyŃéĪcēfŽæăūāLĪăĜŃāŃŪēĒ■ç;õīijŽ

```
# 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;£çTĪēfZäyĪēĒ■ç;õīijŃēzYēōd' æCĒāEĭtäyŃäy■āijZæL'Şă■ŗæŪēāfŪāĀCă;ŃăçCīijŽ

```
>>> import somelib
>>> somelib.func()
>>>
```

äy■ëfGüijNäeCædIJéË■ç;ðè£GæUëå£UçşzçzşüijNéCçázLæUëå£UæüLæAřæL'Sa■řāřsaijĀāğNçTşæTŁi

```
>>> import logging
>>> logging.basicConfig()
>>> somelib.func()
CRITICAL:somelib:A Critical Error!
>>>
```

ëólēōž

é£ŽāyŷæIëèðšüijNä;äy■āžTèřēāIJlāG;æTřāžŠāzččāAäy■ēĠāũséË■ç;ðæUëå£UçşzçzşüijNæLŮèĀĚæY
èřČçTl get_logger(__name__) āLŽāžzāyĀäyġāŠNèřČçTlæġāġIŮāRŇāR■çŽDlog-
geræġāġIŮāĀC çTšāžŌæġāġIŮéC;æYřāTřāyĀçŽDüijNāZāæ■d'āLŽāžzçŽDloggerāžšāřEæYřāTřāyĀçŽDāĀC
log.addHandler(logging.NullHandler()) æS■ā;IJāřEäyĀäyġçl'žād'DçŘEāZlçzŠāōZāLřāL
äyĀäyġçl'žād'DçŘEāZlçzYèōd'äijŽā£;çTēèřČçTlæL'ĀæIJLçŽDæUëå£UæüLæAřāĀC
āZāæ■d'üijNäeCædIJä;£çTlèřēāG;æTřāžŠçŽDæŮūāŽè£YæšqæIJL'éË■ç;ðæUëå£UüijNéCçázLāřEäy■äijŽæL
è£YæIJL'äyĀçCzāřsæYřāřzāžŌāRĎäyġāG;æTřāžŠçŽDæUëå£UéË■ç;ðāRřāžææYřçŽyāžŠçNñçñNçŽDüij
ä;NāeCüijNāřzāžŌāeCāyNçŽDāžččāAüijŽ

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

āIJlè£ŽéGŇüijNæāžæUëå£UëcñéË■ç;ðæLŘāžĚāžĚè;ŠāGžERRORæLŮæZt'énYçžġāLñçŽDæüLæAřāĀ
äy■è£GüijN somelib çŽDæUëå£UçžġāLñècñā■TçNñéË■ç;ðæLŘāRřāžčè;ŠāGždebugçžġāLñçŽDæüLæAřā
āČŘè£ZæāũæZt'æTžā■TçNñæġāġIŮçŽDæUëå£UéË■ç;ðāřzāžŌèřČèřTæIëèðšæYřā;LæŮžā;£çŽDüijN
āZāäyžā;āæŮāeIJĀāŌžæZt'æTžāžzā;TçŽDāĒlāsĀæUëå£UéË■ç;ðāĀTāĀTāRlēIJĀeĀāfōæTžā;āæČšèeĀæZ

Logging HOWTO èřçzEāžNçz■āžEāeCä;TéË■ç;ðæUëå£UæġāġIŮāŠNāĚüāžŮæIJLçTlæLĀāũgüijNāRřā

15.13 13.13 āóđçŌřäyĀäyġlēōāæŮūāŽl

éŮōécY

ä;āæČšèōřā;TçĲNāžRæL'gēāNād'ŽäyġāžzāLqæL'ĀèLšèt'žçŽDæŮüéŮt'

èġċaEşæŪzæaĹ

time æĹaâĹŪăŃĖăŔnăĹLăd'ŽăĢĵæŦŕæĹæL'ġeăŃeŭşæŪŭéŪt' æĹĹL'ăĖşçŽĎăĢĵæŦŕăĂĈ
ăŕĵçŏăăĈCæ■d'ĲĵŃăĂŽăŷŷæĹŚăžŋăĲĴăĹĹæ■d'ăşžçăĂăžŃăŷĹæđĎăĂăŷĂăŷĹæŽt' éŋŸçžġçŽĎæŐăŔcæĹæ

```
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()
```

èĴŽăŷĹçşzăŏŽăžĹ'ăžEăŷĂăŷĹăŔfăžèèçŋĴĹăĹŭæăžæ■ŏéĹĂèĈAăŔfăĹăĂăăĹĹæ■căŞŃéĢ■çĴŏçŽĎèŏă
ăŏĈăĲĴăĹĹĹ elapsed âşđæĂġăŷ■èŏŕăĴæŦt'ăŷĹæŭĹăĂŪæŪŭéŪt'ăĂĈ
ăŷŃéĹcæŸfăŷĂăŷĹăĴŃă■ŔæĹæăĲĴçđ'žæĂŐæăŭăĴçĴĹăŏĈĲĴ

```
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)

```

èõlèõž

æIJñèŁĆæRŘä; ŽäžEäyÄäyİçõÄâ■TèĀŇăódçTİçŽDçşzæİěăódçŎræŰüéŰt' èõră; TăzèăRĹèĀŰæŰüéõaç
 âŖŇæŰüäzşæŸrărzä; ŁçTİwithèr■ăRēäzèăRĹäyLäyŇæŰĠçõaçRĖăŽİă■RèõõçŽDäyÄäyİä; Łăë; çŽDæijTçd' ž

ăIJlèõqæŰüäy■èçAèĀĈèŽSäyÄäyİäžTăşĆçŽDæŰüéŰt' âĠ; æTŕeŰõécŸăĀĈäyĀèĹŇæİèèrt' iijŇ
 ä; ŁçTİtime.time() æĹŰtime.clock() èõaçõŰçŽDæŰüéŰt' çş; äžèăZăæŞ■ă; IJçşzçzşçŽDäy■ăŖŇăij
 èĀŇă; ŁçTİtime.perf_counter() âĠ; æTŕăRŕăžèçqõăİä; ŁçTİçşzçzşäyĹéİćæIJĀçş; çqõçŽDèõqæŰüăŽ

äyĹèĤŕăžççăĀäy■çTşTimer çşzèõră; TçŽDæŰüéŰt' æŸŕéŞşèăĹæŰüéŰt' iijŇăžŰăŇĖăŖŇăžĖæĹĀæIJĹă
 âçĈăđIJă; äăŖĹæĈşèõaçõŰèŕèèçŽçİŇæĹĀèĹşèt' zçŽDĈCPUæŰüéŰt' iijŇăžTèŕèă; ŁçTİtime.
 process_time() æİěäzçæŽİijŽ

```

t = Timer(time.process_time)
with t:
    countdown(1000000)
print(t.elapsed)

```

time.perf_counter() âŞŇ time.process_time()
 éĈ; äijŽèĤTăZđăŖRæTŕă; çăijRçŽDçğŞæTŕæŰüéŰt' äĀĈăódéŽĖçŽDæŰüéŰt' âĀijæşqæIJĹăžzä; TăĎŖăzĹ' iij

æŽt'ăđ' ŽăĖŞăžŎèõqæŰüăŞŇæĀğèĈ;ăĹĖæđŖçŽDă; Ňă■RèŕŰăŖĈèĀĈ14.13ăŕŖèĹĈăĀĈ

15.14 13.14 éŽŖăĹŰăĖĖă■ŸăŞŇCPUçŽDă; ŁçTİéĠŖ

éŰõécŸ

ä;ăæĈşărzăIJĹUnixçşzçzşäyĹéİćèĤRèăŇçŽDçĹŇăžŖèõ; ç; õăĖĖă■ŸæĹŰCPUçŽDă; ŁçTİéŽŖăĹŰăĀĈ

èğçăĖşæŰzæăĹ

resourceăĹăăİŰèĈ;ăŖŇæŰüæĹğèăŇèĤŽăyđ' äyİăžzăĹăăĀĈă; ŇăëĈiijŇèçĖéŽŖăĹŰCPUæŰüéŰt' iij

```

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

```

çİNâzRèfRèaÑæUüijÑSIGXCPU æfqaRûaIJæUüéU't'èfGæIJæUüècñçTşæLRrijÑçDûaRÕæL'gèaÑæY
 èeAéZŖaLûaEĖĖ■Yä;£çTlrijNèøçç;ôaRfä;£çTlçZDæÄzaEĖĖ■YäAija■şaRrijNæCâyNrijZ

```

import resource

def limit_memory(maxsize):
    soft, hard = resource.getrlimit(resource.RLIMIT_AS)
    resource.setrlimit(resource.RLIMIT_AS, (maxsize, hard))

```

âCRèfZæuèøçç;ôäzEaEĖĖ■YéZŖaLûaRÕrijÑçİNâzRèfRèaÑaLŖæşæIJL'ad'Zä;ZaEĖĖ■YæUüaijZæL
 MemoryError aijCâyäĖĖ

èøléøž

âIJæIJnèLCä;Nâ■Räy■rijNsetrlimit() âG;æTŖècñçTlæIèèøçç;ôçL'zâoZètDæzRäyLéIççZDè;réZŖ
 è;réZŖaLûaYŖäyÄäyIaAijrijNâ;ŞeüEèfGèfZäyIaAijçZDæUüaAZæŞ■ä;IJçşççzşéAZäyÿaijZaRSéAAäyÄäy
 çañéZŖaLûaYŖçTlæIèæNĖgâoZè;réZŖaLûèÇ;èøçç;ôçZçZDæIJÄad'gâAijaĖĖÄZäyÿæIèèørijNèfZäyIçTşçşç
 âr;çôaçañéZŖaLûaRfäzæTzârRäyÄçCzrijNâ;EæYŖæIJÄâè;äy■èeAä;£çTlçTlæLûèfZçİNâOzæôæTzâĖĖ

setrlimit() âG;æTŖèfYèÇ;ècñçTlæIèèøçç;ôa■RèfZçİNæTŖèGRãAAæL'ŞaijÄæUĖGäzüæTŖäzèaRLç
 æZt'ad'ZèŖæÇĖèrûaRCèĖĖ resource æIaaiUçZDæUĖGæaçĖĖĖ

éIJÄèeAæşlæDRçZDæYŖæIJnèLCæĖĖäôzâRlèÇ;éĖĖÇTlâžOUnixçşççzşrijNâzûäyTäy■æIèfAæL'ÄæIJL
 æŖTäeCæLSäznâIJlætNèŖTçZDæUüaAZrijNâoÇèÇ;âIJLinuxäyLéIcæ■câyÿèfRèaÑrijNâ;EæYŖaIJIOS
 XäyLâ■t'äy■èÇ;äĖĖ

15.15 13.15 aRraLäyÄäyIWEBætRègŁaZÍ

éUóécY

ä;äæČšéĀŽèŁGèDŽæIJñāRraLäyÄäyIætRègŁaZÍlāzūæL'SāijĀæŃGāōŽčŽDURLč;Séat

èğcāEşæÚzæaŁ

webbrowser ælaaiUèČ;ècncŤlæIēāRraLäyÄäyIætRègŁaZÍlāzūāyŤāyŌāzşāRraUāāEşāĀCā;ŃāçC

```
>>> import webbrowser
>>> webbrowser.open('http://www.python.org')
True
>>>
```

āōČāijZā;ŁčŤlészYèōd'ætRègŁaZÍæL'SāijĀæŃGāōŽč;SéatāĀCāçCædIJä;æŁYæČşārč;SéatæL'SāijĀæŃGāōŽč

```
>>> # 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
>>>
```

èŁZæāuārsāRražæL'SāijĀäyĀäyIæŮřčŽDætRègŁaZÍŁŮāRçæŁŮèĀĒæāGç■iijŃāRlèçAætRègŁaZÍæŤ

æçCædIJä;äæČşæŃGāōŽætRègŁaZÍŁçşādnŤijŃāRražæ;ŁčŤlæ webbrowser.get()
āG;æŤræIēæŃGāōŽæşRäyŁçL'zāōŽætRègŁaZÍāĀCā;ŃāçCŤijŽ

```
>>> c = webbrowser.get('firefox')
>>> c.open('http://www.python.org')
True
>>> c.open_new_tab('http://docs.python.org')
True
>>>
```

ārzažŌæŤræŃAçŽDætRègŁaZÍāRçğraŁŮèālaRraşçéYĒ'PythonæŮGæaç <<http://docs.python.org/3/library/webbrowser.html>>'_

èóléōž

āIJlèDŽæIJñāy■æL'SāijĀætRègŁaZÍæIJL'æŮūāĀZāijZā;ŁæIJLčŤlāĀCā;ŃāçCŤijŃæşRäyIèDŽæIJñæL'
ä;äæČşāfñéĀşæL'SāijĀäyĀäyIætRègŁaZÍæIēçāōāŁIāōČāušçzRæ■čāyYèŁRēāŃāžEāĀC
æŁŮèĀĒæYræşRäyŁçlŃāžRāžèHTMLč;SéatæāijāijRè;ŞāGžæŤræ■ōiijŃā;äæČşæL'SāijĀætRègŁaZÍæşççIJN
äy■çōæYrāyŁēlčāŞŁçg■æČĒEŤiijŃā;ŁčŤlæ webbrowser ælaaiUèČ;æYrāyĀäyIçōĀā■ŤāōđçŤlčŽDèğcāEşæ

16 çññå■AåŽŽçñäiijŽæŧNèrŧãĀAèŕCèrŧaŠÑaijCâyŷ

èŕŧèŧNèŕŷæŷŕåĹæçŠçŽDiiijNä;EæŷŕèŕCèŕŧiijšåršæšæéCçäzĹæIJL'èüçäzEăĂCäzNăođæŷŕiijNăIJIPytl

Contents:

16.1 14.1 æŧNèrŧstdoutèĹŠăĜž

éUóécŷ

ä;ăçŽDĹNăžŔäy■æIJL'äyĹæŰzæŷŧaijŽèĹŠăĜžăĹŕæăĜăĜEèĹŠăĜžăy■iijĹsys.stdoutiijL'ăĂCäzšåršæŷŕèŕ
ä;ăæCšăEžăyĹæŧNèrŧæĹèèŕAæŷŰăoČiijNçzŽăoŽăyĀăyĹèĹŠăĜžèĹNçŽyăžŧçŽDèĹŠăĜžèČ;æ■čăyŷæŷĹçd'žă

èĝčăEşæŰzæăĹ

ä;ăçŧĹ unittest.mock æĹăăĹŰăy■çŽD patch() äĜ;æŧŕiijN
ä;ăçŧĹèŧăĹèĹđăyŷçôĂă■ŧiijNăŕŕăžèăyžă■ŧăyĹæŧNèrŧæĹăæNš sys.stdout
çDăăŔŌăŽđæzŽiijN äžüăyŧăy■ăžĝçŧšăd'ĝéĜŔçŽDăyt'æŰăăŔŷéĜŔæĹŰăIJăŧNèrŧçŧĹăNçŽt'æŌăæŽt'éĹ

ä;IJăyžăyĀăyĹă;Nă■ŕiijNæĹŠăžñăIJ mymodule æĹăăĹŰăy■ăoŽăzĹ'ăçCăyNăyĀăyĹăĜ;æŧŕiijŽ

```
# mymodule.py
```

```
def urlprint(protocol, host, domain):  
    url = '{}://{}.{}'.format(protocol, host, domain)  
    print(url)
```

ézŸèod'æČĚăEŧăyNăEĚç;ôçŽD print äĜ;æŧŕăijŽăŕEèĹŠăĜžăŕŠéĂăăĹŕ sys.
stdout äĂC äyžăžEæŧNèrŧèĹŠăĜžçIJšçŽDăIJĹéCçéĜNiiijNă;ăăŕŕăžèă;ăçŧĹăyĀăyĹæŽĚèžnăŕžèšăæĹèăĹăæN
ä;ăçŧĹ unittest.mock æĹăăĹŰçŽD patch() æŰzæŷŧăŕŕăžèăĹæŰză;ăçŽDăIJăŧNèrŧèŕŔèăNçŽDăyĹ
ăžüăyŧă;ŷæŧNèrŧăoNăĹŔæŰăăĂŽèĜăĹĹèŕŧăŽđăoČăžñçŽDăŌšæIJL'çĹăăĂăĂCăyNéĹăæŷŕăŕž
mymodule æĹăăĹŰçŽDæŧNèrŧăžçčăĀiijŽ

```
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)
```

èóìèõž

```
urlprint() àĜ;æTṛæŌěàRŮäyL'äyłàRCæTṛijNætNèrTæŮzæṣTajjĀāgNaijZāĒLèõç;ōærRäyĀäyłā
expected_url àRŸéGRècñèõç;ōæLŔāNĒāRñæIJṣæIJZçŽDèç;ṢāGžçŽDā■ŮçñëäyṣāĀĆ
```

```
unittest.mock.patch() àĜ;æTṛècñçTlā;IJäyĀäyłäyLäyNæŮĜçōaçRĒāZlriijNā;ŁçTl
StringIO      àrżèṣæİēäzçæZŁ      sys.stdout      fake_out
àRŸéGRæŸrāIJlèrèēŁZçlNäy■ècñāLZāzzçŽDēİæNṣàrżèṣāāĀĆ      àIJlwith-
èr■āRēäy■ä;ŁçTlāōČāRfāzēæL'gèāNāRĎçg■æçĀæṣēāĀČā;Ṣwithèr■āRēççṢæİṣæŮüriijNpatch
äijZārĒæL'ĀæIJL'äyIJèēŁæĀčād'■āLṛætNèrTajjĀāgNāL'■çŽDçLūæĀĀāĀĆ
æIJL'äyĀçČzēIJĀēēĀæṣlāĎRçŽDæŸræṢRāzZārZPythonçŽDCæL'l'āsTārreČ;äijZāŁ;çTēæŌL
sys.stdout      çŽDēĒç;ōēĀNçZt'æŌěāĒZāĒēāLṛæāGāĒĒç;ṢāGžäy■āĀĆ
éZṚāžŌçrGāZĒriijNæIJñèŁČäy■äijZæūL'āRŁāLṛēŁZæŮzéİççŽDèõṣègçriijNāōČēĀČçTlāžŌçžrPythonāzççāĀ
āçČādIJā;äçIJṣçŽDēIJĀēēĀāIJlCæL'l'āsTäy■æ■TèŌūI/OriijNā;āāRfāzēāĒLæL'ṢäijĀäyĀäyłäy'æŮūæŮGāzŮ
æZt'ād'ZāĒṣāžŌæ■TèŌūāzēā■Ůçñëäyṣā;čäijRæ■TèŌūI/OāŠN      StringIO
àrżèṣæṛuāRCéŸĒ5.6ārRēŁČāĀĆ
```

16.2 14.2 àIJlā■TāĒĈætNèrTäy■çžZārżèṣæL'ṢèāēäyA

éŮóécŸ

```
ä;āāĒZçŽDā■TāĒĈætNèrTäy■éIJĀēēĀçzZæNĠāōŽçŽDārżèṣæL'ṢèāēäyĀriijN
çTlāİēæŮ■ēİĀāōČāznāIJlætNèrTäy■çŽDæIJṣæIJZēāNäyziijLærTāçCriijNæŮ■ēİĀècñèrČçTlāŮūçŽDāRCæT
```

èğčāĒṣæŮzæāŁ

```
unittest.mock.patch()      àĜ;æTṛāRṛècñçTlāİèèğčāĒṣèŁZäyłēŮóécŸāĀĆ
patch() èŁŸāRṛècñçTlā;IJäyĀäyłèçĒēēāZlāĀäyLäyNæŮĜçōaçRĒāZlāLŮā■TçNñā;ŁçTlriijNār;çōāāzŮ
ä;NāēCriijNäyNéİçæŸrāyĀäyłārĒāōČā;ṢāĀZèçĒēēāZlā;ŁçTlçŽDä;Nā■RriijZ
```

```
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)
```

```
āōČēŁŸāRfāzēècñā;ṢāĀZäyĀäyłäyLäyNæŮĜçōaçRĒāZlriijZ
```

```
with patch('example.func') as mock_func:
    example.func(x)          # Uses patched example.func
    mock_func.assert_called_with(x)
```

æIJĀāŘŔĭjNä;æĕŸāŔřäzĕæL'NāĽĽçŽDä;ĤçŦĽăŔĈæL'ŞĕăĕăŸAĭjŽ

```
p = patch('example.func')
mock_func = p.start()
example.func(x)
mock_func.assert_called_with(x)
p.stop()
```

ăĕĆăđIJăŔŕĕĈ;çŽDĕŕĭĭjNä;ăĕĈ;ăđ'şăŔăăĽăĕĈĖĕĕŕăŽĽăŞNăŸĽăŸNăŨĜĉŏăçŔĖăŽĽăĭĕçzŽăđ'ŽăŸĽăŕzĕş

```
@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:
        ...
```

èőĽĕőŽ

patch() æŎĕăŔŨăŸĂăŸĽăŭşă■ŸăĽĽăŕzĕşăçŽDăĖĽĕŭŕă;ĎăŔ■ĭjNăŕĖăĖŭăŽĤæ■ăŸŸăŸĂăŸĽăŨŕçŽDăĽăŔăŖăĭĕçŽDăĀĭĵăĭjŽăĽĽĭĕĈĖĕĕŕăŽĽăĜ;æŦŕăĽŨăŸĽăŸNăŨĜĉŏăçŔĖăŽĽăŭŔăŔŔĖĖĜĽăĽăĀĈăđ'■ăŽđăĭĕăĕŸĖĕđ'æĈĖăĖŦăŸNĭjNăL'ĂæĽĽăĀĭĵăĭjŽĕĉn MagicMockăđă;NăŽĤăzĉăĂĈă;NăĕĈĭjŽ

```
>>> x = 42
>>> with patch('__main__.x'):
...     print(x)
...
<MagicMock name='x' id='4314230032'>
>>> x
42
>>>
```

ăŸ■ĕŤĜĭjNä;ăăŔŕăzĕĕĂŽĕŤĜçzŽ patch() æŔŔă;ŽçŋăžNăŸĽăŔĈæŦŕăĭĕăŕĖăĀĭĵăŽĤæ■ăĽŔăzză;Ŧ

```
>>> x
42
>>> with patch('__main__.x', 'patched_value'):
...     print(x)
...
patched_value
```

(continues on next page)

(continued from previous page)

```
patched_value
>>> x
42
>>>
```

ěćńçŤlæİëä;IJäyžæŽŁæ■ćăĀijçŽĐ MagicMock āōđä;NèĈ;ād'şæłæNşāRrērĈçŤlāržèşqāŠŇāōđä;NăĂ
äzŮäzñèōřă;ŤāržèşqçŽĐä;ŁçŤlăŁqæAřázŭăĚĀeōÿă;ăæL'ğèqŇæŮ■élĀæčĀæşēijŇă;ŇăeĈijŽ

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

äyĂēĹŋæİēēōřijŇēŁZăžZæŞ■ă;IJăijŽăIJăyĂăyĹă■ŤăĚĈæŤNērŤăy■ăōŇæĹŖăĂĈă;ŇăeĈijŇăĀĞèō;ă;

```
# example.py
from urllib.request import urlopen
import csv

def dowprices():
    u = urlopen('http://finance.yahoo.com/d/quotes.csv?s=@^DJI&f=sll
↪')
```

(continues on next page)

(continued from previous page)

```
lines = (line.decode('utf-8') for line in u)
rows = (row for row in csv.reader(lines) if len(row) == 2)
prices = { name:float(price) for name, price in rows }
return prices
```

æ■čāÿÿæİēēōsīijÑēŁŻāÿİāĞ;æŦřāijŽā;ŁçŦİ urlopen() äzŌWe-
bāÿŁēİcēŌūāŦŪæŦřæ■ōāzūēğçædŘāōČāĀČ āİĴā■ŦāĒČætŦNērŦāÿ■īijÑā;āāŦřāzēçzŻāōČāÿĀāÿİēčĎāĒŁāōŽ

```
import unittest
from unittest.mock import patch
import io
import example

sample_data = io.BytesIO(b'''\r\n
"IBM",91.1\r\n
"AA",13.25\r\n
"MSFT",27.72\r\n
\r\n
''')

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()
```

æİĴñā;Ŧāÿ■īijÑā;■āzŌ example æİāāİŪāÿ■çŽĎ urlopen()
āĞ;æŦřēčñāÿĀāÿİæİāæŦŦāřzēsāæŽēāzçīijŦ ēřēāřzēsāāijŽēŦŦāŽđāÿĀāÿİāŦĒāŦñætŦNērŦæŦřæ■ōçŽĎ
ByteIO().

ēŁŸæİĴĴāÿĀçČzīijŦāİĴæŁŦēāēāÿĀæŪūæŁŦāzñā;ŁçŦİāzē example.
urlopen æİēāzçæŽē urllib.request.urlopen āĀČ
ā;Ŧā;āāŁŻāzžēāēāÿĀçŽĎæŪūāĀŽīijŦā;āāŦĒēāzā;ŁçŦİāōČāznāİĴætŦNērŦāzççāĀāÿ■çŽĎāŦ■çğřāĀČ
çŦŦsāzŌætŦNērŦāzççāĀā;ŁçŦİāzē from urllib.request import urlopen ,ēČčāzĴ
dowprices() āĞ;æŦř āÿ■ā;ŁçŦİçŽĎ urlopen() āĞ;æŦřāōđēŽēāÿŁāřsä;■āzŌ
example æİāāİŪāzēāĀČ

æİĴñēŁČāōđēŽēāÿŁāŦİæŸŦāřz unittest.mock æİāāİŪçŽĎāÿĀæñætŦĒāŦİē;Ďæ■čāĀČ
æŽŦāđŽæŽŦēŦŸçžğçŽĎçŁzæĀğīijŦēŦūāŦČēĀČ āōŸæŪžæŪĞæç

16.3 14.3 `assertRaises()` and `assertRaisesRegex()`

Example

Example: `assertRaises()` and `assertRaisesRegex()` can be used to test that a function raises a specific exception.

Example

Example: `assertRaises()` and `assertRaisesRegex()` can be used to test that a function raises a specific exception.

```
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')
```

Example: `assertRaises()` and `assertRaisesRegex()` can be used to test that a function raises a specific exception.

```
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')
```

Example

Example: `assertRaises()` and `assertRaisesRegex()` can be used to test that a function raises a specific exception.

```
class TestConversion(unittest.TestCase):
    def test_bad_int(self):
        try:
            r = parse_int('N/A')
        except ValueError as e:
            self.assertEqual(type(e), ValueError)
```

```
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() æŮzæſTäijŽad’DčŘEæL’ÄæIJL’çzEèŁĆijŃŃaZäæ■d’ä;äzTèrëä;ŁçTlĹăŮCăĂĆ

assertRaises() çŽDäyÄäyŁçijçCzæYřăŮčĽŃäy■ăzEäijCăyÿăEüă;ŞçŽDăÄijæYřăd’ŽărSăĂĆ

äyžăEæſNërTäijCăyÿăÄijŃŃŃăRřăzëä;ŁçTl assertRaisesRegex() æŮzæſTŃijŃ

ăŮčĂŖăŖăŖŃæŮŮæſNërTäijCăyÿçŽDă■YăIJăzëăŖŁéĂŽèŁGă■căŁZăijŖăŃzëĚ■äijCăyÿçŽDă■ŮçŋäyşëăŁçd

```
class TestConversion(unittest.TestCase):
    def test_bad_int(self):
        self.assertRaisesRegex(ValueError, 'invalid literal .*',
                                parse_int, 'N/A')
```

assertRaises() assertRaisesRegex()

```
class TestConversion(unittest.TestCase):
    def test_bad_int(self):
        with self.assertRaisesRegex(ValueError, 'invalid literal .*
↳ '):
            r = parse_int('N/A')
```

ä;Eä;äčŽDætNërTæuĹ'äRĹäĹräd'ŽäyĹæĹ'gëaNæ■ēēld'čŽDæŮüäÄŽēfŽčg■æŮzæšTāršä;ĹæIJĹčŤlāzE;

16.4 14.4 ǻĖætĖNĕrTĕĹŠǻĠžĉTĭǻĖUĕǻĤUĕőřǻĤǻĹrǻŮĠǻžŮǻŷ■

éŮőécŸ

ä;äÿÑæIJŽăŔĖ■TăĚĈætŊërTçŽĐè;ŠăĠžăĖŽăĹrăĹræšŔăÿtæŨĠăžăüä;■ăŐzĭjŊĕĂŊăy■æŸræĹŠă■ŕă

èğčǎẸșæŮźæǻŁ

èŁRèàÑā■TāĖĈætNèrTävÄäylävÿèğAæŁĂæIJråršæYråIJlætNèrTæŮĞazüāžTéĈlāŁāāĖäyNéicèŁZæø,

```
import unittest

class MyTest(unittest.TestCase):
    pass
```

(continues on next page)

```
if __name__ == '__main__':
    unittest.main()
```

```
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)
```


16.5 14.5 åÆçTëæLÚæIJşæIJZætNërTåd'sèt'ě

éUőécŸ

äjäæČşåIJlā■TāĚČætNërTäy■āfçTëæLÚæăGèőŕæşŘăžZætNërTäijŽæŃL'çĚgécĎæIJşèĚŘèāŃād'sèt'ěā

èğčāEşæŮzæąŁ

unittest ælāaiŮæIJL'èčĚéčŕāŽlāŔŕçTlælēæŮğāLūāržæŃĠăōŽætNërTæŮzæşTçŽĎād'ĎçŘĚijŃäçN

```
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()
```

ăęČæđIJäjäåIJlMacăyŁèĚŘèāŃèĚZæőžăžččāAijŃä;ăäijŽă;ŮāLŕăęČăyŃèçŞăĠžiiJŽ

```
bash % python3 testsample.py -v
test_0 (__main__.Tests) ... ok
test_1 (__main__.Tests) ... skipped 'skipped test'
test_2 (__main__.Tests) ... skipped 'Not supported on Unix'
test_3 (__main__.Tests) ... ok
test_4 (__main__.Tests) ... expected failure

-----
↳---
```

(continues on next page)

(continued from previous page)

```
Ran 5 tests in 0.002s
```

```
OK (skipped=2, expected failures=1)
```

èóìèőž

```
skip()      èċĚēřǎŽíĚČ;èċńċŤlǎİēǎǣ;çŤǣ$Řǎyłǎ;ǎy■ǎĈşèĤŘǎǎŇċŽĎǎŤŇǎŤǎĈ
skipIf() ǎŤŇ skipUnless() ǎřžǎžŎǎ;ǎǎŤǎĈşǎİĤǎ$ŘǎyłċL'žǎőŽǎžşǎŤǎĤŤŤPythonċL'ĤǎİŇǎĤŮǎĤŮ
ǎ;ĤċŤĤ @expected ċŽĎǎđ'sèt'èċĚēēřǎŽíĤēǎǎĤēōřĚĈǎžŽċǎőǎőŽǎijŽǎđ'sèt'ċŽĎǎŤŇǎŤŤijŇǎžŮǎyŤǎřžēĤ
ǎǣ;çŤǣŮǎşŤċŽĎċĚēēřǎŽíĤēŤŤǎŤǎžēċńċŤlǎİēċĚēēřǎŤŤǎyłǎŤŇǎŤŤşžijŇǎŤŤǎĈċijŽ
```

```
@unittest.skipUnless(platform.system() == 'Darwin', 'Mac specific_
↳tests')
class DarwinTests(unittest.TestCase):
    pass
```

16.6 14.6 ǎđ'ĎċŘĚǎđ'ŽǎyłǎijĈǎyŷ

éŮőéćŸ

ǎ;ǎǎİĤL'ǎyǎǎyłǎžċċǎǎċL'ĤǎēōŤǎŤŤēĈ;ǎijŽǎĤŽǎĤǎđ'Žǎyłǎy■ǎŤŇċŽĎǎijĈǎyŷijŇǎǎŮǎǎL'■ēĈ;ǎy■

èġċǎĤşǎŮǎǎǎĤ

ǎċĈǎđİǎ;ǎǎŤǎžēĈŤĤǎ■ŤǎyłǎžċċǎǎǎİŮǎđ'ĎċŘĚǎy■ǎŤŇċŽĎǎijĈǎyŷijŇǎŤǎžēǎŤĤǎőĈǎžŇǎŤ;ǎĤēǎyǎǎ

```
try:
    client_obj.get_url(url)
except (URLError, ValueError, SocketTimeout):
    client_obj.remove_url(url)
```

ǎİĤēĤŽǎyłǎ;Ňǎ■ǎy■ijŇǎĤĈċēŮǎy■ǎžžǎ;ŤǎyǎǎyłǎijĈǎyŷǎŤŤĈŤşǎŮŮēĈ;ǎijŽǎĤġēǎŇ
remove_url() ǎŮǎşŤǎĈ ǎċĈǎđİǎ;ǎĈşǎřžǎĤŮǎy■ǎşŘǎyłǎijĈǎyŷēĤŽǎŇǎy■ǎŤŇċŽĎǎđ'ĎċŘĚijŇǎŤ
except ċŤ■ǎŤēǎy■ijŽ

```
try:
    client_obj.get_url(url)
except (URLError, ValueError):
    client_obj.remove_url(url)
except SocketTimeout:
    client_obj.handle_url_timeout(url)
```

ǎĤĤǎđ'ŽċŽĎǎijĈǎyŷǎijŽǎİĤL'ǎŤĈċžġǎĤşşžijŇǎŤǎžǎžŎēĤŽċġ■ǎĈĤǎĤijŇǎ;ǎǎŤŤēĈ;ǎ;ĤċŤĤǎőĈǎžŇċŽĎǎ

```
try:
    f = open(filename)
except (FileNotFoundError, PermissionError):
    pass
```

āŕŕāzēēcēŃĜ■āĒŽāyžījŽ

```
try:
    f = open(filename)
except OSError:
    pass
```

OSError æŸŕ FileNotFoundError āŠŃ PermissionError
āijĈāyŷĉŽĎāšžčšāĀĈ

èóĹèőž

ārĉōāđ'ĎĉŔĒāđ'ŽāyĹāijĈāyŷæIJñēžnāžūæšāžĀāžĹĉĹ'žæōĹĉŽĎījŃāy■èĚĜā;āāŕŕāzēā;ĚĉŦĪ
as āĖšéŦōā■ŮāĹēēŮōāĹŮēcāĹĹŽāĜžāijĈāyŷĉŽĎāijŦĉŦīijŽ

```
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)
```

èĚŽāyĹāĹŃā■Ŕāy■īijŃ e āŔŸéĜŔæŃĜāŔŠāyĀāyĹēcāĹĹŽāĜžĉŽĎ OSError
āijĈāyŷāōđāĹŃāĀĈ èĚŽāyĹāIJĹā;āæĈšæŽŦ'èĚŽāyĀæ■ēāĹĒāđŔēĚŽāyĹāijĈāyŷĉŽĎæŮūāĀŽāijŽāĹĹæIJĹ'ĉŦīij
āŔŃæŮūēĚŸēĉAæšĹæĎŔĉŽĎæŮūāĀŽ except ěŕ■āŔēæŸŕéāžāžŔæĉĀæšĉŽĎījŃĉñāyĀāyĹāŃzéĒ■ĉ
ā;āāŕŕāzēā;ĹāōžæŸšĉŽĎāđĎēĀāđ'ŽāyĹ except āŔŃæŮūāŃzéĒ■ĉŽĎæĈĒā;ĉīijŃæŕŦæĉĈīijŽ

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

FileNotFoundError OSError æŽŕäýÄëĽñijŇăĎČăŔŕăŇzéĚ FileNotFoundErroŕ äijČăýýijŇăžŎăŸŕăŕŝăŸŕçññăýĂăýĽăŇzéĚ ŽĎăĂČăĽĽăŕČăŕŤčŽĎăŮăĂŽijŇăęČăđĽăĵăăŕŕăŝŔăýĽčĽăžăžăijČăýýăĵăăŔŕăŕăĚăŽăĚŔăŝęĽŇăŕăăijČăýýčŽĎ __mro__ăŝđăĂġăĽăĤŇăĂŝăŕŔăġĽăĂČăŕŤăĈijŽ

```
>>> FileNotFoundError.__mro__
(<class 'FileNotFoundError'>, <class 'OSError'>, <class 'Exception'>
→,
 <class 'BaseException'>, <class 'object'>)
>>>
```

äýĽăĽăĽăŮăăĽăýăăžăĵăŤăýĂăýĽčŽŕăĽŕ BaseException čŽĎčŝŕăĈĵăĈĵăĈŇăĽăžŎ
except ŕăăŔăăĂČ

16.7 14.7 æŇŤăŮăæĽĂæĽĽăĵăĈăýý

éŮăéčŸ

æĂŎăăŮăŇŤăŮăăžčăĂăýăčŽĎăĽĂæĽĽăĵăĈăýýijŝ

èġčăĚŝăŮăăăĽ

ăĈŝăĸăĂăŇŤăŮăăĽĂæĽĽăĵăĈăĵăĈăýýijŇăŔŕăŕăĚăĈŕăŎăăŇŤăŮă Exception
ăŝăŔŕijŽ

```
try:
    ...
except Exception as e:
    ...
    log('Reason:', e) # Important!
```

ăĚŽăýĽăŕăĸăijŽăŇŤăŮăăŽđăžĚ SystemExităĂĂ KeyboardInterrupt
ăŝŇ GeneratorExităžŇăđŮčŽĎăĽĂæĽĽăĵăĈăýýăĂČ
ăęČăđĽăĵăăĸăŸăĈŝăŇŤăŮăăĚăýĽăýĽăijČăýýijŇăŕăĚ ExceptionăŤăžăĽŔ
BaseExceptionăŝăŔŕăĂČ

èŮăăăž

ăŇŤăŮăăĽĂæĽĽăĵăĈăýýăĂžăýýăŸŕčŤŝăžŎčĽŇăžŔăŝŸăĽăŝŔăžŽăđăăĽăĈăŝăĽăĵăăžăŮăýăĈĵăĈŮăăęČăđĽăĵăăăýăŸŕăĽčŕăĤčĈăžăžăžăŸăŸŕčĵăăĚăýăŸŝăŕčăŕŤăžčăăĂčŽĎăýĂăýĽčŮăăŤăŮă
ăĈăžăăęČăđĽăĵăăĸăŝăŝăĈăđĽăĵăăĸăĽăŇŮăŇŤăŮăăĽĂæĽĽăĵăĈăýýijŇăĈăžăĽăĽăŝŔăýĽăĽăŮăijĽăăęČăđĽăĵăăŝăăĽăĚăăăăĂžijŇăĽăŮăăĂžăăĵăĽăŝăĽăĽăŮăŕăĸăŝŸăýăĈăăĬăăđŕăĤăđă

```
def parse_int(s):
    try:
        ...
    except:
        ...
```

(continues on next page)

(continued from previous page)

```
n = int(v)
except Exception:
    print("Couldn't parse")
```

èŕŦçĭĂèĚŔèąŇèĚŽăŷłăĜĭ;æŦŕĭĭjŇçzŞæđĬJăĕĆăŷŇĭĭjŽ

```
>>> parse_int('n/a')
Couldn't parse
>>> parse_int('42')
Couldn't parse
>>>
```

èĚŽæŮŭăĂŽăĭ;ăăŕşăĭjŽæŇăăđ't' æĈşĭĭjŽăĂĬJèĚŽăŖŇăŽđăžŇăŦĬĭĭjŞăĂĬ
ăĂĜăĕĆăĭ;ăăĈŔăŷŇéĬèĚŽæăŭéĜ■ăĚŽèĚŽăŷłăĜĭ;æŦŕĭĭjŽ

```
def parse_int(s):
    try:
        n = int(v)
    except Exception as e:
        print("Couldn't parse")
        print('Reason:', e)
```

èĚŽæŮŭăĂŽăĭ;ăèĈĭ;èŬăŕŮăĕĆăŷŇéĭ;ŞăĜŷĭĭjŇæŇĜæŸŬăžĒæĬJĬ'ăŷĭçĭjŮçĭŇéŦŽèŕŕĭĭjŽ

```
>>> parse_int('42')
Couldn't parse
Reason: global name 'v' is not defined
>>>
```

ăĭĬæŸŬăŸĭĭjŇăĭ;ăăžŦèŕèărĭ;ăŖŕèĈĭ;ăŖĒăĭjĈăŷŷăđ'ĐçŖĒăŽĭăŵŽăžĬ'çŽĐçşĭ;ăĜĒăŷĂăžŽăĂĈ
ăŷ■èĚĜĭĭjŇèĕĂæŸŕăĭ;ăăĬĒéăžæ■ŦèŬăŕĬ'ĂæĬJĬ'ăĭjĈăŷŷĭĭjŇçăŵăĬĬæĬ'Şă■Ŗæ■ĈçăŵçŽĐèŕĬæŮ■ăĬăæĂŕæĬŮă

16.8 14.8 áĬŽăžžèĜĬăŵŽăžĬ'ăĭjĈăŷŷ

éŬŵéćŸ

ăĬĬăĭ;ăăđĐăžžçŽĐăžŦçŦĬçĬŇăžŖăŷ■ĭĭjŇăĭ;ăæĈşăŕĒăžŦăŖŖăĭjĈăŷŷăŇĒèĕĒæĬŖèĜĬăŵŽăžĬ'çŽĐăĭjĈăŷŷă

èĝĉăĒşşæŮžæăĬ

ăĬŽăžžæŮŕçŽĐăĭjĈăŷŷăĭ;ĬçŵĂă■ŦăĂŦăĂŦăŵŽăžĬ'æŮŕçŽĐçşŷĭĭjŇèŵĬ'ăŵĈçžĝæĬ'ĤèĜĬ
Exception ĭĭjĬæĬŮèĂĒæŸŕăžžăĭ;ŦăŷĂăŷłăŭşă■ŸăĬĬçŽĐăĭjĈăŷŷçşăđŇĭĭjĬ'ăĂĈ
ăĭŇăĕĈĭĭjŇăĕĈăđĬJăĭ;ăçĭjŮăĒŽçĭ;ŞçzĬJçŽăĒşçŽĐçĬŇăžŖĭĭjŇăĭ;ăăŖŕèĈĭ;ăĭjŽăŵŽăžĬ'ăŷĂăžŽçşžăĭĭjĭăĕĆăŷŇçĭ

```
class NetworkError(Exception):
    pass
```

(continues on next page)

```
class HostnameError(NetworkError):
    pass

class TimeoutError(NetworkError):
    pass

class ProtocolError(NetworkError):
    pass
```

čDůŘŮčTlæLũarsãRřazěaČŘéĀŽäÿëĆčæuü; ěčTlěŁZăZjajĆäÿÿăžEiijNă; NăeĆiijŽ

```
try:
    msg = s.recv()
except TimeoutError as e:
    ...
except ProtocolError as e:
    ...
```

èóíèőž

ĕĠłăŏŽăzĹ'ăijĈăyŷĉșăžTĕrĕăĂzăŸřĉžġăĹ'ĤĕĠłăĚĖĉ;ŏĉŽĎ	Exception
ĉșăiijN æĹŪĕĂĚăŸřĉžġăĹ'ĤĕĠĤĕĈăžZăIJnĕžnăřsăŸřăžŎ	Exception
ĉžġăĹ'ĤĕĂNăĤĕĉŽĎĉșăĂĈ ăř;ĉŏăăĹ'ĂăIJĹ'ĉșăăRŇăŪăăžșĉžġăĹ'ĤĕĠ	BaseException
ijNă;Ėă;ăăŷăăžTĕrĕă;ĤĉTĤĕŹăŷłășžĉșăĤĕăŏŽăzĹ'ăŪřĉŽĎăijĈăyŷăĂĈ	BaseException
ăŸřăŷĉșĉžĉșĖĂăĠăijĈăyŷĕĂNăĤĤĉŽĎijNăřTăĖĈ	KeyboardInterrupt
æĹŪ	
SystemExit äžĕăRĹăĖŪăžŪĕĈăžZăijŽĉžĹăžTĉTĤăRŖŖĂăĤăăăăRŭĕĂNăĂăĠăžĉŽĎăijĈăyŷăĂĈ	
ăžĂăăđ'ijNăăTĕŎĖĖŹăžZăijĈăyŷăIJnĕžnăřsăăžĂăžĹăĎRăžĹăĂĈ	
ĕŹăăŭĉŽĎĕřijijNăĂĠăăĈă;ăĉžġăĹ'Ĥ	BaseException
ăRĕĖĈ;ăijŽăřijĕĠ'ă;ăĉŽĎĕĠăŏŽăzĹ'ăij	

aIjIcIŃažRäy■aijTāĒēēGĽaōŽāzL'aijCāyāaRřāzēā;ǣā;Ūā;āčŽDāžččāAæŽr'āĒūāRřerzæĀgġijŃēČ;æyĒæ
 ēfYæIJL'āyĀčg■ēō;ēōāæYřāřĒēGĽaōŽāzL'aijCāyāēĀŽēfĠčzġāL'ǣzčDāRĽēūālēāĀCāIjIād'■āIČāžTčTĽIŃ
 ā;ǣcTĽIāšzčszālēāLēcžDāRĎc■aijCāyācszāzšæYřā;LāēIJL'cTĽIčŽDāĀCāōČāRřāzēēōl'cTĽāLūā■TēŌūāyĀā

```
try:
    s.send(msg)
except ProtocolError:
    ...
```

ä:äeſYëČ:æ■TèŬuæŽt'ad'gëŇČăŽt'čŽDajĭČăvŷiiĭŇărsăČRăvNéİcèſŽăũiiĭŽ

```
try:
    s.send(msg)
except NetworkError:
    ...
```

æĈædIjä;äæĈsǎōžǎzL'çŽdæŨřaijCǎyyéG■ǎEžǎžE __init__() æŨzæſTijjN
çǎōǎflǎ;ǎä;ççTlǎL'ǎæIJL'ǎRĈǎTřerĈçTl Exception.__init__() iijNä;NǎeCiiž

```
class CustomError(Exception):
    def __init__(self, message, status):
        super().__init__(message, status)
        self.message = message
        self.status = status
```

çIJNäyLåŒzæIJL'çCzæĜæĀhijNäy■èĜExceptionçŽDézYëód'èaŒäyžæYřæŒëâRŪæL'ĀæIJL'äijæéĀŠç
 .args åśdæĀğäy■. åĴLåd'ŽâĒüāzŪāĜ;æTřāžŠāŠŒéĀlĀLEPythonāžŠézYëód'æL'ĀæIJL'äijCāyŸéĀ;āĒĒéāzæ
 .args åśdæĀğhijN'āZāæ■d'æĀĀdIJä;āāĒ;çTěāžEēĒZāyĀæ■ēhijNä;āāijŽāRŠçŒřæIJL'āžZæŪūāĀZā;āāōŽāz
 äyžāžEæijTçd'ž .args çŽDä;ĒçTřhijNēĀĀçŽSāyNäyNēlĀēĒZāyĴā;ĒçTřāEĒç;ōçŽD Run-
 timeError' äijCāyŸçŽDāžd'āžŠāijŽēřhijN æşĴæĀRçIJNraiseēē■āRēäy■ā;ĒçTřçŽDāRĀçæTřäyĴæTřæYřæĀŒæā

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

āĒşāžŒāĴZāžzèĜĴāōŽāzL'äijCāyŸçŽDæŽt'ād'ŽāĒæAřhijNēřūāRĀçèĀĀ'PythonāōYæŪzæŪĜæāç
 <<https://docs.python.org/3/tutorial/errors.html>>'_

16.9 14.9 æ■TèŒŪāijCāyŸāRŒæLZāĜzāRēād'ŪçŽDāijCāyŸ

éŪŒéçY

ä;āæĀşæ■TèŒŪāyĀäyĴāijCāyŸāRŒæLZāĜzāRēād'ŪāyĀäyĴāy■āRŒçŽDāijCāyŸhijNāRŒæŪŪēĒYāĴŪāIJ

èĜçāEşæŪzæāĴ

äyžāžEēŞçæŒēāijCāyŸhijNä;ĒçTř raise from ēē■āRēæĴēāzçæŽĒçōĀā■TçŽD raise
 ēē■āRēāĀĀ āōĀhijŽēōĴ'ä;āāRŒæŪūāĒĴçZāyĴ'āyĴāijCāyŸçŽDāĒæAřāĀĀçā;NāēĀhijŽ

```
>>> def example():
...     try:
...         int('N/A')
...     except ValueError as e:
...         raise RuntimeError('A parsing error occurred') from e
↪e
```

(continues on next page)

(continued from previous page)

```
...
>>> example()
Traceback (most recent call last):
  File "<stdin>", line 3, in example
ValueError: invalid literal for int() with base 10: 'N/A'
```

äyŁéÍćŻĎâijĈäyÿæŸřäyŇéÍćŻĎâijĈäyÿäžğçŤšçŻĎçŽt' æŌěăŌšăŽăijŽ

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "<stdin>", line 5, in example
RuntimeError: A parsing error occurred
>>>
```

ăIJlăZđæžřäy■ăRřäzěçIJŇăLřijŇäyđ' äyłâijĈäyÿéČ;ěćŋæ■ŤěŌuăĂĆ
ěęAæĈšæ■ŤěŌuěŁZăăüçŻĎâijĈäyÿrijŇă;ăăRřäzěă;ŁçŤlăyĂäyłçŏĂă■ŤçŻĎ except
ěŋ■ăRěăĂĆ äy■ěŁGrijŇă;ăěŁŸăRřäzěěĂŽěŁGăšççIJŇăijĈäyÿăržesăçŻĎ __cause__
ăśđăĂğăĬěüšëyłâijĈäyÿéŞ;ăĂĆăĹŇăęĈrijŽ

```
try:
    example()
except RuntimeError as e:
    print("It didn't work:", e)

    if e.__cause__:
        print('Cause:', e.__cause__)
```

ă;ŠăIJĬ except âİŮäy■ăRĹăIJĬăRěăđ' ŮçŻĎâijĈäyÿěćŋæŁZăĜžăŮüăijŽăřijęĜt' äyĂäyłéŽŘěŮŔçŻĎă

```
>>> 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'
```

ăIJlăđ' ĎçŘĚäyŁěřâijĈäyÿçŻĎăŮüăĂŽrijŇăRěăđ' ŮäyĂäyłâijĈäyÿăŔŚçŤšăžĚrijŽ

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "<stdin>", line 5, in example2
NameError: global name 'err' is not defined
>>>
```

ěŁZăyłăĹŇă■Răy■rijŇă;ăăŔŇăŮüěŌuăĹŮăžĚäyđ' äyłâijĈäyÿçŻĎăŁăăĂřijŇă;ĚăŸărřăijĈäyÿçŻĎğğç

ðēŹæŮŭāĀŹīījŅNameError āijĈāyŷēćnā;IJāyŷĉĹŅāzŔæIJĀçzĹāijĈāyŷēćnæŁŹāĠŹīījŅēĀŅāy■æŸřä;■āžŮ
 æĈædIJīījŅā;āæĈşāŋ;çŦæŮŬāijĈāyŷēŹ;īījŅāŔřä;ŋçŦĪ 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
>>>
    
```

èõlèõž

āIJlèõžèõāžzĉĉāAæŮīījŅāIJlāŔēād'ŮāyĀāyĪ except āžĉĉāAāĪŮāy■ā;ŋçŦĪ raise
 èř■āŔēçŹDæŮŭāĀŹā;āēçAçŁ'zāĹnārŔāŋĈāzEāĀĆ ād'ġād'ŹæŦřæĈĒāEřāyŅīījŅēŋŹçġ■
 raise èř■āŔēēĈ;āžŦērēēćnæŦzæĹŔ raise from èř■āŔēāĀĆāžşāřsæŸřēřt'ā;āāžŦērēā;ŋçŦĪāyŅēlĉēŋŹçġ

```

try:
...
except SomeException as e:
    raise DifferentException() from e
    
```

ðēŹæāŭāĀŹçŹDāŮşāŹæŸřä;āāžŦērēæŸçd'žçŹDārĒāŮşāŹæŹ;æŮēēřŭælēāĀĆ
 āžşāřsæŸřēřt'īījŅDifferentException æŸřçŹt'æŮēāžŮ SomeException
 ēā■çŦşēĀŅælēāĀĆ ðēŹçġ■āĒşçşzāŔřāzēāžŮāŹdæžřçzşædIJāy■çIJŅāĠŹælēāĀĆ

æĈædIJā;āāĈŔāyŅēlĉēŹæāŭāĒŹāžĉĉāAīījŅā;āāž■çDŭāijŹā;ŮāĹŔāyĀāyĪēŹ;æŮēāijĈāyŷīījŅ
 āy■ēŋĠēŹāyĪāžŭæşæIJŁ'āĹŁāyĒæŹŕçŹDēřt'æŸŮēŋŹāyĪāijĈāyŷēŹ;āĹŔāžŦæŸřāĒēĈĪāijĈāyŷēŋŸæŸřæş

```

try:
...
except SomeException:
    raise DifferentException()
    
```

ā;Şā;āā;ŋçŦĪ raise from èř■āŔēçŹDērīījŅāŔřā;ŁāyĒæēŹçŹDēāĹæŸŮæŁŹāĠŹçŹDæŸřçŋnāžŅāyĪā
 æIJĀāŔŮāyĀāyĪā;Ņā■Ŕāy■ēŹŔēŮŔāijĈāyŷēŹ;āŋæAřāĀĆ
 āř;çŋæēŹŔēŮŔāijĈāyŷēŹ;āŋæAřāy■āĹ'āžŮāŹdæžřīījŅāŔŔæŮŭāŮĈāžşāyćād'sāžĒāĹŁād'ŹæIJŁ'çŦĪçŹDērĈ
 āy■ēŋĠāyĠāžŅçŹĒāžşç■ŮīījŅæIJŁ'æŮŭāĀŹāŔlāŋĹçŦŹēĀĆā;ŞçŹDāŋæAřāžşæŸřāĹŁāIJŁ'çŦĪçŹDāĀĆ

16.10 14.10 éĜæŮŕæŁŻăĜžèćŋæ■TèŎũçŽĎaijCăyŷ

éŮóécŸ

äĵăăIJăyĂăyĭ except äĭŮăy■æ■TèŎũăžEăyĂăyĭaijCăyŷiiijŃçŎŕăIJăçšéĜæŮŕæŁŻăĜžăőČăĂĆ

èġčăEşæŮzæąŁ

çőĂă■TçŽĎăĭçŁĭăyĂăyĭă■TçŃŋçŽĎ rasie èŕ■ăŔěă■şăŔŕiijŃăĭŃăęĆriijŽ

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

èóĭèőž

èŁŽăyĭéŮóécŸéĂŽăyŷăŸŕăĭŞăĭăéIJăęĂăIJăæ■TèŎũăijCăyŷăŔŎæŁġèăŃăşŔăyĭăş■ăĭIJiijŁăŕŤăęĆèăăyĂăyĭăĭŁăyŷèġĂçŽĎçŤĭăşŤăŸŕăIJăæ■TèŎũăŁĂăIJĬăijCăyŷçŽĎăđŦĎçŔĖăŽĭăy■iijŽ

```
try:
...
except Exception as e:
    # Process exception information in some way
...

    # Propagate the exception
    raise
```

16.11 14.11 èĭŞăĜžè■ęăŚŁăŋæAŕ

éŮóécŸ

äĵăăyŃăIJŽèĜĭăũççŽĎçĭŃăžŔèČĭçŤşăĬŔè■ęăŚŁăŋæAŕiijŁăŕŤăęĆăžşăijČçŁžăĂġăĬŮăĭçŁĭéŮóécŸ

èġċàEşæŮzæąŁ

èċAèŁŞăĠzäyÄäyłè■ēāŚŁæŮŁæAřijŇăŔřăĴçŤĪ
ăĠ;æŤřăĀĈăĴŇăēĈijŽ

warning.warn()

```
import warnings

def func(x, y, logfile=None, debug=False):
    if logfile is not None:
        warnings.warn('logfile argument deprecated',
↳DeprecationWarning)
    ...
```

warn() çŽĎăŔĈæŤřæŸřăyÄäyłè■ēāŚŁæŮŁæAřăŇăyÄäyłè■ēāŚŁçşziiŇè■ēāŚŁçşzæIJL'ăēĈăyŇăĠă
DeprecationWarning, SyntaxWarning, RuntimeWarning, ResourceWarning, æŁŮ FutureWarn-
ing.

ărzè■ēāŚŁçŽĎăĎ'ĎçŔĒăŔŮăEşăžŌăĴăăēĈăĴ;ŤèŤŔèăŇēġċéĠăŽĴăzēăŔĴăyÄăžZăĒŮăžŮéĒ■ċ;ŏăĀĈ
ăĴŇăēĈijŇăēĈăĎIJăĴăăĴçŤĪ-W all éĀŁ'éăžăŌzèŤŔèăŇPythonijŇăĴăăijŽăĴŮăĴŔăēĈăyŇçŽĎèŁŞăĠziiŇŽ

```
bash % python3 -W all example.py
example.py:5: DeprecationWarning: logfile argument is deprecated
  warnings.warn('logfile argument is deprecated',
↳DeprecationWarning)
```

éĀŽăyŷæĴēōşrijŇè■ēāŚŁăijŽèŁŞăĠzăĴŔăăĠăĠĒēŤŽerrăyĴăĀĈăēĈăĎIJăĴăăĈşèōşè■ēāŚŁèĴŇăēĈăyžă
-W error éĀŁ'éăžziiŇŽ

```
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 %
```

èőłēōž

ăIJăĴăçzt'æŁĎ'èĴřăžŮijŇăŔŔĈĎ'žçŤĴæĴŮăşŔăžZăŤăæAřijŇă;EăŸřăŔĴăy■éIJăēċAăŔĒăĒŮăyĴă■Ġăy
ăĴŇăēĈijŇăĀĠēōĴăĴăăĠĒăĎ'ĠăŤŏăŤzæşŔăyĴăĠ;æŤřăžŞăĴŮăăEăĎŮçŽĎăĴşēĈĴijŇăĴăăŔřăžăăĒĴăyžă;ă
ăĴăēŤŸăŔřăžèè■ēāŚŁçŤĴæĴŮăyÄăžZăŔzăžçċăAæIJL'éŮŏéċŸçŽĎăĴçŤĴæŮăijŔăĀĈ

ăĴIJăyžăŔĒăĎ'ŮăyÄäyłăĒĒēċŏăĠ;æŤřăžŞçŽĎè■ēāŚŁăĴçŤĴăĴŇă■ŔijŇăyŇéĴăijŤĈĎ'žăžEăyÄäyłăşăæ

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

ézYëød' æČĚāEĵäyNġijNāzūäy■æYřæL' ĀæIJL'è■ēāSŁæūLæAřéČ;äijŽāGžčŎřāĀĆ-W
 éĀL'ēāzèČ;æŎğāLūè■ēāSŁæūLæAřčŽDè;ŠāGžāĀĆ -W all
 äijŽè;ŠāGžæL' ĀæIJL'è■ēāSŁæūLæAřġijN-W ignore āĤ;çTēæŎL'æL' ĀæIJL'è■ēāSŁġijN-W
 error āřEè■ēāSŁè;ñæ■ēāL'ŘāijČāyŷāĀĆ āŘēād' ŪäyĀčg■ēĀL'æN'ġijNā;āèĤYāRřāzēā;ĤčTĪ
 warnings.simplefilter() āĤ;æTřæŎğāLūè;ŠāGžāĀĆ always
 āRČæTřāijŽèŏl'æL' ĀæIJL'è■ēāSŁæūLæAřāGžčŎġijN` ignore
 āĤ;çTēēřČæL' ĀæIJL'čŽDè■ēāSŁġijNerror āřEè■ēāSŁè;ñæ■ēāL'ŘāijČāyŷāĀĆ

ārzāžŎčŏĀā■TčŽDčTšæL'Řè■ēāSŁæūLæAřčŽDæČĚāEĵēĤZāžZāũšçZŘēūšād' šāžEāĀĆ
 warnings āĤāāĪŪāržēĤGāzđ' āŠNè■ēāSŁæūLæAřād' DčŘEæRŘä;ZāžEād' gēGRčŽDæŽt'énYčžgčŽDēĤ■ç;
 æŽt'ād' ŽāĤæAřēřūāRČēĀĆ PythonāŪĜæqč

16.12 14.12 ěřČērTāšžæIJñčŽDčĪNāžRāt'Ī'æžČéTŽērĪ

éŬóécY

ä;äçŽDčĪNāžRāt'Ī'æžČāRŎèrēæĀŎæūāŎžērČērTāŏČġijš

èğčāEşæŪzæqĪ

æēČæđĪä;äçŽDčĪNāžRāZāyžæšŘāyĤāijČāyŷēĀNāt'Ī'æžČġijNēĤŘēāN
 python3 -i someprogram.py āRřæL'ğēāNçŏĀā■TčŽDērČērTāĀĆ
 -i éĀL'ēāzāRřēŏl'čĪNāžRčZšæĪšāRŎæL'ŠāijĀäyĀäyĤāzđ' āžŠāijRshellāĀĆ
 çDūāRŎā;āāřsēČ;æšēçIJNçŎřāčČġijNā;NāēČġijNāAĜēŏ;ä;āæIJL'äyNēĪççŽDāžčçāAġijŽ

```
# sample.py

def func(n):
    return n + 10

func('Hello')
```

ēĤŘēāN python3 -i sample.py äijŽæIJL'čšzäijijæČāyNçŽDè;ŠāGžġijŽ

```
bash % python3 -i sample.py
Traceback (most recent call last):
  File "sample.py", line 6, in <module>
    func('Hello')
  File "sample.py", line 4, in func
    return n + 10
```

(continues on next page)

(continued from previous page)

```
TypeError: Can't convert 'int' object to str implicitly
>>> func(10)
20
>>>
```

æĈædIJă;ăçIJNăy■ăLřăyŁéÍcèŁZăăuĉŽDřijNăRřăzěăIJłÍNăžRăt'ĲăžĈăŘŎæL'ŠăijĂPythonĉŽDěřĈěřĲ

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

æĈædIJă;ăçŽDăžĉăAæL'ĂăIJłŽDĉŎřăĈăŁéŽŁèŎŭăRŮăžd'ăžŠshellijŁăřTăĉĈăIJăŠRăyŁăIJ■ăŁăŁă
éĂŽăyyăRřăzěă■TěŎŭăijĈăyyăRŎěĠăŭăæL'Šă■řeŭšëyŁăŁăqăAřăĂĈăŁNăĉĈijŽ

```
import traceback
import sys

try:
    func(arg)
except:
    print('**** AN ERROR OCCURRED ****')
    traceback.print_exc(file=sys.stderr)
```

èĖAæYřă;ăçŽDĉłNăžRăšăæIJL'ăt'ĲăžĈijNěĂNăRŁăYřăžġĉTšăžEăyĂăžZă;ăçIJNăy■ăĠĈĉŽDĉzŠăđŁ
ă;ăăIJăĎŠăĒt'ëŭĉĉŽĎăIJăŮzăRŠăĒěăyĂăyN print() ěr■ăRěăžšăYřăyŁăy■éTŽĉŽĎéĂL'æNĲăĂĈ
ăy■ěŁĠijNěĖAæYřă;ăæL'ŠĉŎŮěŁZăăŭăĂŽijNăIJL'ăyĂăžZăřRăŁĂăŭăRřăzěăyŏăŁĲăăăĂĈ
éĖŮăĒŁijN traceback.print_stack() åĠăTřăijŽă;ăĉłNăžRěŁRěăNăŁřěĈăyŁĉĈzĉŽĎăŮŭăĂŽăŁŽ

```
>>> def sample(n):
...     if n > 0:
...         sample(n-1)
...     else:
...         traceback.print_stack(file=sys.stderr)
...
>>> sample(5)
File "<stdin>", line 1, in <module>
File "<stdin>", line 3, in sample
File "<stdin>", line 3, in sample
File "<stdin>", line 3, in sample
```

(continues on next page)

(continued from previous page)

```
File "<stdin>", line 3, in sample
File "<stdin>", line 3, in sample
File "<stdin>", line 5, in sample
>>>
```

```
    aRɛɑd' ŨrijNä;æðfYɑRfrazəɑCRäyNéIcèfZæuä;fçTÍ          pdb.set_trace()
aIJläzzä;TāIJræŨzæL'NāLÍçŽDāRfāLérCèrTāZÍrijŽ
```

```
import pdb

def func(arg):
    ...
    pdb.set_trace()
    ...
```

```
    ä;ŞçlNāzRærTè;Cād'gèĀNä;äæÇşèrCèrTæŌgāLūætAçlNāzèāRĹāG;æTřāRCæTřçŽDæŨuāĀŽèfZäyĹā
ä;NāçCrijNäyĀæŨçèrCèrTāZÍrijĀāgNèfRèāNrijNä;āārşèÇ;ād'şä;fçTÍ
print          æIèègCætNāRYéGRāĀijæLŨæTşāGzæşRäyĹāŞ;äzd'aerTæçC          w
æIèèŨāRŨè;fçyĹāfæAfrāĀC
```

èõlèõž

```
    äy■èeAārEèrCèrTāijDçŽDèfGāžŌād'■æICāNŨāĀCäyĀāžZçōĀā■TçŽDèTŽèrrāRĹéIJĀèeAègCārşçlNā
aōdèZĒçŽDèTŽèrrāyĀèLñæYřāāEæāLçŽDæIJĀāRŌäyĀèāNāĀC
ä;āāIJĹāijĀāRŞçŽDæŨuāĀŽrijNāzşāRfrazèāIJĹā;æéIJĀèeAèrCèrTçŽDāIJræŨzæRŞāĒèäyĀäyN
print() āG;æTřāIèèfLæŨ■āfæAfrījLāRĹéIJĀèeAæIJĀāRŌāRŞāyCçŽDæŨuāĀŽāLæéZd'èfZāžZæL'Şā■
```

```
    èrCèrTāZÍçŽDäyĀäyĹāyègAçTĹæşTæYřègCætNæşRäyĹāuşçzRāt'l'æžCçŽDāG;æTřäy■çŽDāRYéGRāĀ
çşèeAşæĀŌæuāIJĹāG;æTřāt'l'æžCāRŌèfZāĒèèrCèrTāZÍāYřäyĀäyĹā;LæIJLçTĹçŽDæLĀèC;āĀC
```

```
    ä;Şä;äæÇşègçāL'ŨäyĀäyĹéIdäyŷād'■æICçŽDçlNāzRrijNāzTāşCçŽDæŌgāLūéĀžè;Şä;ääy■æYřā;LäyĒ
æRŞāĒè pdb.set_trace() èfZæuäçŽDèr■āRèārşā;LæIJLçTĹāžEāĀC
```

```
    aōdèZĒäyLijNçlNāzRāijŽäyĀçZt'èfRèāNāLřççrāLř          set_trace()
èr■āRèä;■ç;ōrijNçDūāRŌçñNél'nèfZāĒèèrCèrTāZÍāĀC çDūāRŌā;āārşāRfrazèāĀZæZt'ād'ŽçŽDāžNāžEāĀC
```

```
    æCædIJā;āä;fçTĹIDEæIèāĀŽPythonāijĀāRŞrijNéĀŽäyŷIDEéC;āijZæRŘä;ZèGĹāuşçŽDèrCèrTāZÍāIèā
æZt'ād'ŽèfZæŨzéIççŽDāfæAfrāRfrazèāRCèĀCä;ä;fçTĹçŽDIDEæL'NāEñāĀC
```

16.13 14.13 çžŽä;ăçŽDçlNāzRāĀZæĀgèC;ætNèrT

éŨóécY

```
    ä;äæÇşætNèrTä;ăçŽDçlNāzRèfRèāNæL'ĀèLsèt'zçŽDæŨéŨt'ázūāĀZæĀgèC;ætNèrTāĀC
```

èġċăEşæŮzæąŁ

ăĕĆădĬJă;ăăRĭăĖŸřċôĂă■TċŽĐăĈşăĭNĕřTăŸNă;ăċŽĐċĬNăžRăTĭ'ă;ŞĕŁşĕt'żċŽĐăŮŭĕŮt'ĭĭjŃ
éĂžăŸăă;ĤċŦĬUnixăŮŭĕŮt'ăĠ;ăTĭřăřşăăNăžĖĭĭjNăřTăĕĈĭjŽ

```
bash % time python3 someprogram.py
real 0m13.937s
user 0m12.162s
sys 0m0.098s
bash %
```

ăĕĆădĬJă;ăĕĤŸĖĬJăĕĕAăŸĂăŸĭċĬNăžRăRĐăŸĭċzĖĕŁĈċŽĐĕřĕċzĖăĖĖăŞĬĭjNăRăřăžăă;ĤċŦĬ
cProfileăĭăăĭŮĭĭjŽ

```
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 %
```

ăŸ■ĕĤĠĖĂžăŸăăĈĖăĖĤăŸřăžNăžŮĕĤŽăŸd'ăŸĭăđAċŋřăžNĕŮt'ăĂĈăřTăĕĈă;ăăŸşċzRċşĕĕAşăžċċăAĕĖĬ
ăržăžŮĕĤŽăžZăĠ;ăTĭřċŽĐăĂġĕĈ;ăĭNĕřTĭĭjNăRăřăžăă;ĤċŦĬăŸĂăŸĭċôĂă■TċŽĐĕĈĖĕřăŽĭĭjŽ

```
# timethis.py

import time
from functools import wraps

def timethis(func):
    @wraps(func)
    def wrapper(*args, **kwargs):
```

(continues on next page)

(continued from previous page)

```
start = time.perf_counter()
r = func(*args, **kwargs)
end = time.perf_counter()
print('{}.{} : {}'.format(func.__module__, func.__name__,
↪end - start))
return r
return wrapper
```

èeAä;ŁçŦłefZäyłecĚéērăZłijŇăŔłéIJĀēeAārEāĔūæŦłç;őăIJlă;ăēeAēfZēqŇæĀgēĈ;ætŇerŦçŽĎăĜ;æŦ

```
>>> @timethis
... def countdown(n):
...     while n > 0:
...         n -= 1
...
>>> countdown(10000000)
__main__.countdown : 0.803001880645752
>>>
```

èeAætŇerŦæŖăyłäzččăAăIŮefŖēqŇæŮŭéŮŦ'ijŇă;ăăŖfäzēăőŽăzL'ăyĀăyłäyŁăyŇæŮĜçőaçŖĒăŽłijŇ

```
from contextlib import contextmanager

@contextmanager
def timeblock(label):
    start = time.perf_counter()
    try:
        yield
    finally:
        end = time.perf_counter()
        print('{} : {}'.format(label, end - start))
```

ăyŇéłcæŸŕă;ŁçŦłefZäyłäyŁăyŇæŮĜçőaçŖĒăŽłçŽĎă;Ňă■ŖijŽ

```
>>> with timeblock('counting'):
...     n = 10000000
...     while n > 0:
...         n -= 1
...
counting : 1.5551159381866455
>>>
```

ărzăžŌætŇerŦă;ŁărŖçŽĎăzččăAçL'ĜăōŧefŖēqŇæĀgēĈ;ijŇă;ŁçŦł timeit
ăłqăłŮăijŽă;ŁăŮză;ŁijŇă;ŇăēĈijŽ

```
>>> 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` äijZæL'gëaŊçññäyÄäyIäRĆæTřäy■ēr■āRě100äyGæñāzŭeōaçoŭUēŁRëaŊæUŭéŮt'āĀĆ
çññāzŊäyIäRĆæTřäYřēŁRëaŊæTŊērTāzŊāL'■ēĚ■ç;ōçŎřācČāĀĆæCædIJä;āæČšæTzāRŸā;łçŎřæL'gëaŊæñ
āRřāzēāCRäyŊēIcēŁZæāŭeō;ç;ō number āRĆæTřčZDāĀijijZ

```
>>> timeit('math.sqrt(2)', 'import math', number=1000000)
1.434852126003534
>>> timeit('sqrt(2)', 'from math import sqrt', number=1000000)
1.0270336690009572
>>>
```

ěōlěōž

ā;ŠæL'gëaŊæĀgèČ;æTŊērTčZDæUŭāĀZijŊēIJĀèeAæšIæDRčZDæYřä;æeŎŭāRŮčZDčzŠædIJéČ;æYřē
time.perf_counter() āĜ;æTřäijZāIJlčzZāōZāzšāRřäyŁeŎŭāRŮæIJĀénYčš;āžēčZDēōaæUŭāĀijāĀĆ
äy■ēŁĜijŊāōČāz■čDŭēŁYæYřāšzāžŎæUŭéŠšæUŭéŮt'ijŊā;Łād'ZāZāčt'āaijZā;šā\$■āLřāōČčZDčš;çāōāžē
āēČædIJä;āārāzēŎæL'gëaŊæUŭéŮt'æZt'æDšāĒt'ēūčijŊä;ŁçTl time.process_time()
æIēāzčæZŁāōČāĀĆä;ŊāēČijZ

```
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
```

æIJĀāRŎijŊāēČædIJä;āæČšēŁZëaŊæZt'æŭsāĒēčZDæĀgèČ;āŁēædRřijŊēČčāzŁä;æēIJĀèeAēřēčzĒēYŮ
time āĀAtimeit āŠŊāĒŭāzŮčZyāĒšæIqāIŮčZDæŮĜæāčāĀĆ
ēŁZæāŭā;āāRřāzēčRĒēgčāŠŊāzšāRřčZyāĒščZDāŭōāijČāzēāRŁäyĀāzZāĒŭāzŮēZŭēYšāĀĆ
ēŁYāRřāzēāRĆēĀĆ13.13ārRēŁČäy■čZyāĒščZDäyÄäyIäL'ZāzžēōaæUŭāZlčščZDä;Ŋā■RāĀĆ

16.14 14.14 āŁæéĀšçIŊāžRēŁRëaŊ

éŮōécŸ

ā;āčZDčIŊāžRēŁRëaŊād'ŁæĒčijŊä;āæČšāIJläy■ä;ŁçTlād'■æiCæŁĀæIJræfTāēČCæL'řāsTāŁŮJITčijŮ

èġċàEşæŮzæąŁ

ăĖşăžŎċÍŇăžŔăijŸăŇŮċŽĐċñăyĂăyĹăĠĖăĹŹăŸŕăĂĪăy■èĕĂăijŸăŇŮăĂĪijŇċñăžŇăyĹăĠĖăĹŹăŸŕăăĕĊăđĪă;ăċŽĐċÍŇăžŔĕĤŔĕăŇċijŞăĖĊĲijŇĕĕŮăĖĹă;ăă;Ůă;ĤċŦĲ14.13ăŕŔĕĹĊċŽĐăĹĂăĪŕăĖĹăŕăăŏĊĕĤŹă

éĂŽăyŷăĪĕĕŏă;ăăijŽăŔŞċŎŕă;ăă;ŮċÍŇăžŔăĪĪăŕŞăŦŕăĠăăyĤċ■ċĊăĪŕăŮzĕĹsĕŕ'žăžĖăđ'ġĕĠŔăŮĭĕăŕŦăĕĊăĖĖă■ŸċŽĐăŦŕă■ŏăđ'ĐċŔĖăĹĤċŎŕăĂĊăyĂăŮĕă;ăăŏŽă;■ăĹŕĕĤŽăžŹċĊĲijŇă;ăăŕśăŔŕăžăă;ĤċŦĲăyŇ

ă;ĤċŦĲăĠăĠă;ăŦŕ

ăĹĹăđ'ŽċÍŇăžŔăŞŸăĹŽăijĂăġŇăijŽă;ĤċŦĲPythonĕŕ■ĖĹĂăĖŽăyĂăžŹċŏĂă■ŦĕĐŽăĪŇăĂĊăĲijŮăĖŽĕĐŽăĪŇċŽĐăŮăĂŽijŇĕĂŽăyŷăžăăĊŕăžĖăĖŽăŕŇăŮăċŷŞăđĐċŽĐăžċċăĂĲijŇăŕŦăĕĊĲijŽ

```
# somescript.py

import sys
import csv

with open(sys.argv[1]) as f:
    for row in csv.reader(f):

        # Some kind of processing
        pass
```

ăĹĹăŕŞăĪĪăăžŹċşĕĕĂŞĲijŇăĊŔĕĤŹăăăăŏŽăžĹăĪĪăĖĹăśĂĕŇĊăŹŕ'ċŽĐăžċċăĂĕĤŔĕăŇĕĲăĪĕĕĕĂăŕŦăŏŽăĕĤŹċġ■ĖĂăžĕăăŏăĲĊăŸŕċŦśăžŎăśĂĕĊĹăŔŸĕĠŔăŖŇăĖĹăśĂăŔŸĕĠŔċŽĐăŏđċŎŕăŮzăĲŕĲijĹă;ĤċŦĲăśĂĕĊĹăăŽăă■đ'ĲijŇăĕĊăđĪă;ăăĊşĕŏŦ'ċÍŇăžŔĕĤŔĕăŇăŹŕ'ăŕŇăžŹĲijŇăŔĹĖĪăĕĕĂăŕĖĕĐŽăĪŇĕŕ■ăŔĕăŦĹăĖĖăĠă;ăŦŕ

```
# 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])
```

éĂşăžĕċŽĐăăŏăĲĊăŔŮăĖşăžŎăŏđĕŽĖĕĤŔĕăŇċŽĐċÍŇăžŔĲijŇăy■ĕĤĠăăžăă■ŏċŷŔĕĹĖŇĲijŇă;ĤċŦĲăĠăĠă;ăŦŕ30%ċŽĐăĂġĕĊă;ăŔŔă■ĠăŸŕăĹăyŷĕġĂċŽĐăĂĊ

ăŕ;ăŔŕĕĊă;ăŎŹăŎĹăśđăĂġĕŏĤĕŮŏ

ăŕŔăyĂăŇăă;ĤċŦĲĊĲ(.)ăŞ■ăĲĲĲĕĪĕĕŏĤĕŮŏăśđăĂġċŽĐăŮăăĂŽăijŽăyĕăĪĕĕĹăđ'ŮċŽĐăĲăĖŦăĂăăăŏĊăĲijŽĕġĕăŔŞċĹăăŏŽċŽĐăŮzăşŦĲijŇăŕŦăĕĊ____getattribute____()ăŖŇ____getattr____()ĲijŇĕĤŽăžŹăŮzăşŦăĲijŽĕĤŹăăŇă■ŮăĖŸăŞ■ăĲăŞ■ăĲăĂĊ

éĂŽăyŷă;ăăŔŕăžăă;ĤċŦĲfrom module import nameĕĤŹăăăċŽĐăŕĲijăĖĖă;ăăĲijŔĲijŇăžăăŔĹă;ĤċŦĲċŷăăŏŽċŽĐăŮzăşŦăĂĊăĂĠĕŏĲă;ăăĪĪăăĊăyŇċŽĐăžċċăĂċĹăĠăŏŦĲijŽ

```
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)
```

åIJåŁŚāznæIJžāZlāyŁéİcætNerTçZĐæUúāĀZiijNefZāyİçİNāžRèŁsèt'zāžEāđ'ğæeĆ40çğŠāĀĆçÖřāIJā
compute_roots() āĠ;æTřæĆāyNiiž

```
from math import sqrt

def compute_roots(nums):

    result = []
    result_append = result.append
    for n in nums:
        result_append(sqrt(n))
    return result
```

āfōæTžāRŎçZĐçL'ŁæIJñefRèaÑæUúéUť'ād'ğæeĆæYř29çğŠāĀĆāTřāyĀāy■āRÑāzNād'ĐāřsæYřæúŁé.
çTÍ sqrt() āžcæZŁāžE math.sqrt() āĀĆ The result.
append() æŮzæşTècnètNçzZāyĀāyİāsĀéČlāRÝéĠR result_append
iijNçDūāRŎāIJāEĒēČlā;İçÖřāy■ā;İçTlāōČāĀĆ

āy■ēfĠiijNefZāžZæTžāRÝāRŁæIJL'āIJlād'gēĠRēĠāđ'■āžçāĀāy■æL■æIJL'æĐŘāžL'iijNæřTāeČā;İç
āZāæ■đ'iijNefZāžZāijYāNŮāžşāRŁæYřāIJāşRāžZçL'žāōZāIJřæŮzæL'■āžTèrècēcnā;İçTlāĀĆ

çŘEğçāsĀéČlāRÝéĠR

āžNāL'■æRRèfĠiijNāsĀéČlāRÝéĠRāijZæřTāÉlāsĀāRÝéĠRèfRèaÑæĀşāžæāfnāĀĆ
åržāžŎéçŞçZĀeōfēUōçZĐāR■çġriijNēĀŽefĠāřEefZāžZāR■çġrāRÝæLŘāsĀéČlāRÝéĠRāRřāžēāŁæēĀşçİN
ā;NāeĆiijNçIJNāyNāzNāL'■åržāžŎ compute_roots() āĠ;æTřefZèaÑāfōæTžāRŎçZĐçL'ŁæIJñiiž

```
import math

def compute_roots(nums):
    sqrt = math.sqrt
    result = []
    result_append = result.append
    for n in nums:
        result_append(sqrt(n))
    return result
```

āIJİēfZāyİçL'ŁæIJñāy■iijNşqrt āžŎ match æİāİUècēnæNŁāĠžāžūæTç;āĒēāžEāyĀāyİāsĀéČlāRÝéĠR

æĊædIJä;æĤRèaÑèĤZäyläzĉĉäAĭijÑäd' ġæĊĈèLset' z25ĉġSĭijLärzäzŌäzNäl'■29ĉġSäRLæYřäyÄäyĭæTzèĤZ
 èĤZäyléĉĭäd' ŪĉZĎäLæĤŌšäŌšäZæYřäZäyZärzäzŌäšĤéĤĭäRŸéĠR sqrt
 ĉZĎæšĕæL' ĭĉĉAāĤnäžŌāĤĭāsĤāRŸéĠR sqrt

ärzäzŌĉśzäy■ĉZĎāsđæĤġèĉĤéŪŏäzšāRÑæāüéĤĊTĭäzŌèĤZäylāŌšĊRĕāĤĊ
 éĤZāyŸæĭĕĉŏšĭijÑæšĕæL' ĭæšRäyĭāĤĭjærTāĉĊ self.name
 äĭjZærTèĉĤéŪŏäyÄäyĭāsĤéĤĭäRŸéĠRĕĉAæĤĉäyÄäzZāĤĊ āĭĭāĤĕĤĭäĭĭĉŌřäy■ĭijÑāRřäzĕārĤæšRäyĭēĭJĀē

```
# 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)
```

éAĤāĤ■äy■āĤĕĕAĉZĎæL'ĕśā

äzzä;TæŪūāĤZā;Šā;äā;ĤĊTĭéĉĭäd' ŪĉZĎäd' ĎĊRĕāsĊĭijLærTāĉĈĕĉĕĕrāZĭāĤĤāsđæĤġèĉĤéŪŏāĤĤæR
 ærTāĉĈĊĭJÑäyÑæĈäyNĉZĎèĤZäylĉśzĭijZ

```
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
```

ĉŌřāĭĭĕĤZèaÑäyÄäyĭĉŏĤā■TætNērTĭijZ

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

ārřäzĉĭJÑāĤĭijÑĕĉĤéŪŏāsđæĤġyĉZŸærTāsđæĤġxēĤÑēĭĤāĕĊĉZĎäy■æ■cäyĤĊĊzĉĊzĭijÑäd' ġæĊĈæĤ
 æĊædIJä;āāĭĭāĤĎRæĤġĉĊZĎērĭijÑĕĈĉāzĤārśēĭJĀēĉAĉĠ■ŪřāŏæġĤäyNārřäzŌyĉZĎāsđæĤġèĉĤéŪŏāz

æĊædIJæſqæIJL'æĤĖĖeAġijNærſä;ĤçTĭçōĀā■TāsđæĀğāRğāĀĆ
æĊædIJäzĖäzĖæYřāZāāyZāĖūāzŪçijŪçĭNēr■ēĭĀĖIJĀĖeAā;ĤçTĭgetter/setterāĠ;æTřārſāŌzāĤōæTřāzčçāAĖč

ä;ĤçTĭāĖĖç;ōçZĎāōzāZĭ

āĖĖç;ōçZĎæTřæ■ōçszādNærTāeCā■ŪçņēäyſāĀāāĖĊçzĎāĀāĤĭŪēāĭāĀāēZEāRĤĭāŠNā■ŪāĖYēČ;æYř
æĊædIJā;āæĊſēĠāūsāōđçŌřæŪřçZĎæTřæ■ōçzſæđĎġĭĤærTāeČēſ;æŌēāĤĭŪēāĭāĀāāzſēāqāēāſç■ĤġĭĤġġġ
ēĊčāzĤēæĀæĊſāIJĭæĀğēČ;āyĤē;ĭāĤřāĖĖç;ōçZĎēĀſāzēāĠāāzŌāy■ārřēČ;ġġġNāZāæ■đġġġNēĤYæYřāzŪāzŪ

ēAĤāĖ■āĤZāzžāy■āĤĖĖeAçZĎæTřæ■ōçzſæđĎāĤĭŪāđ'■āĤĭ

æIJL'æŪūāĀZçĭNāzRāſYæĊſæY;æſEāyNġġġNæđĎēĀāyĀāzZāzūæſqæIJL'æĤĖĖeAçZĎæTřæ■ōçzſæđ

```
values = [x for x in sequence]
squares = [x*x for x in values]
```

āzſēōyēĤZēĠNçZĎæĊſæſTæYřēēŪāĖĤārĖāyĀāzZāĀġæTūēZEāĤřāyĀāyĤāĤĭŪēāĭāy■ġġġNçĎūāRŌā;ĤçT
āy■ēĤĠġġġNçņāyĀāyĤāĤĭŪēāĭāōNāĖĤæſqæIJL'æĤĖĖeAġijNærRāzēçōĀā■TçZĎāĊRāyNēĭcēĤZæāūāĖZġġZ

```
squares = [x*x for x in sequence]
```

āyŌæ■đçZyāĖſġġġNēĤYēeAæſĭæĎRāyNēĊčāzZārZPythonçZĎāĖſāznæTřæ■ōæIJzāĤĭŪēĤĠāzŌāĤRæĤġ
æIJL'āzZāzžāzūæſqæIJL'ā;Ĥāē;çZĎçRĖēğçēĤĭŪāĤāzžPythonçZĎāĖĖā■YæĤāđNġġġNæzēçTĭ
copy.deepcopy() āzNçſçZĎāĠ;æTřāĀĆēĀZāyāĤĭēĤZāzZāzčçāAāy■æYřārRāzēāŌzæŌĤāđ'■āĤĭæſ

ēōlēōž

āĤĭāġYāNŪāzNāĤ'■ġġġNæIJL'æĤĖĖeAāĖĤçāTçĭ'ūāyNā;ĤçTĭçZĎçōŪæſTāĀĆ
ēĤĭæNĭ'āyĀāyĤāđ'■āĤčāēāyž O(n log n) çZĎçōŪæſTēeAærTā;āāŌzērĊæTĭ'āyĀāyĤāđ'■āĤčāēāyž
O(n**2) çZĎçōŪæſTæĤĭĀāyēāĖēçZĎæĀğēČ;ærRā■ĠēēĀāđ'ğāĤĭŪāđ'ZāĀĆ

æĊædIJā;āēĠĤĭŪā;āēĤYæYřā;ŪēĤZēāNāġYāNŪġġġNēĊčāzĤēŕūāzŌæTř'ā;ſēĀĊēZſāĀĆ
āĤIJāyžāyĀēĤNāĠĖĤĤġġNāy■ēeAārzcĭNāzRçZĎæRāyĀāyĤēĤĭĤĖēČ;āŌzāġYāNŪ,āZāyžēĤZāzZāĤōæTř
ā;āāzTēŕēāyſæſĭāzŌāġYāNŪāzğçTſæĀğēČ;çſūēēĤçZĎāIJæŪzġġġNærTāeČāĖĖēČĭā;ĤçŌřāĀĆ

ā;āēĤYēeAæſĭæĎRā;ōārRāġYāNŪçZĎçzſæđIJāĀĆā;NāeĊēĀĊēZſāyNēĭcāĤZāzžāyĀāyĤā■ŪāĖYçZĎā

```
a = {
    'name' : 'AAPL',
    'shares' : 100,
    'price' : 534.22
}

b = dict(name='AAPL', shares=100, price=534.22)
```

ārŌēĭcāyĀçğ■āĖZæſTæZĭ'çōĀæt'ĀāyĀāzZġġĤĭLā;āāy■ēIJĀĖeAāĤĭāĖſēTŌā■ŪāyĤē;ſāĖēāġTārŪġġĤĭL'ā
āy■ēĤĠġġġNāeĊædIJā;āārĖēĤZāyđ'āyĤāzčçāAçĤĭĠæōĤēĤZēāNæĀğēČ;æĤNērTārZærTæŪŪġġġNāġZārſçŌřā;Ĥç
dict() çZĎæŪzāġRāġZæĖčāzēZāĀ■āĀĆçIJNāĤĖēĤZāyġġġNā;āæYřāy■æYřæIJL'āĖſāĤĭæĤĤæĤĭĀæIJL'ā;
dict() çZĎāzčçāAēČ;æZæ■cāĤRçņņāyĀçğ■āĀĆāy■āđ'ſġġġNēĀĤæYŌçZĎçĭNāzRāſYārRāġZāĖſæſĭāzŪ

æĊædIJā;āçZĎāġYāNŪēeAæſĊærTē;ĊēNġġġNæIJNēĤĊçZĎēĤZāzZçōĀā■TæĤĭæĤřæzæēſāy■āZĖġġ
ā;NāeĊġġġNPyPyāūēçĭNæYřPythonēğçēĠāZĭçZĎāRēāđ'ŪāyĀçğ■āōđçŌřġġġNāōĊāġZāĤĖæđRā;āçZĎçĭNāz

[illegible]

559


```

    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
```

æĈædIJäYÄÄLGæ■čäyÿijNä;äärſäRfrazëäLäë;äzûä;ŁçTléGÑeÍcäŹZäZL'çŽDCäG;æTřazEäÄCä;NäçČ

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

ëöleöž

æIJnärRèLCæIJL'ä;Läd'ŽäÄijä;ÜæLSäznèrëçzEèöleöžçŽDäIJræŮzäÄC
 éçŮäĚLæYřärzäžÖCäŠNPythonäzčçäÄäYÄètūæL'SäNĚçŽDëÜöécYřijNäçČædIJä;äaIJlä;ŁçTÍ
 ctypes ælëèöŁëÜöçijŮerSäRÖçŽDCäzčçäÄijN éCčäzLéIJÄæAçäöäŁëŁZäyŁäĚsaznážSæT;äIJl
 sample.py ælaäIŮäRŇNäYÄäyŁäIJræŮzäÄC äyÄçg■äRfèC;æYřärEçTšæLRçŽD .
 so æŮGäzûæT;ç;öäIJlëçÄä;ŁçTÍläöČçŽDPythonäzčçäÄäRŇNäYÄäyŁçŽöä;TäyNäÄC
 æLSäznäIJl recipeäÄTsample.py äy■ä;ŁçTÍ __file__
 äRÝéGRælēæšçIJNäöČëcäŇäŮL'èçĚçŽDä;■ç;öijN çDüäRÖædDëÄäyÄäyŁæNĞäRŠäRŇNäYÄäyŁçŽöä;Täy■
 libsample.so æŮGäzûçŽDëürä;DäÄC

æĈædIJCäG;æTřazŠëcäŇäŮL'èçĚäLřäĚüäzŮäIJræŮzijNéCčäzLä;äärſëçÄäŁöæTřçZyāžTçŽDëürä;DäÄ
 æĈædIJCäG;æTřazŠäIJlä;æIJzäŽlāyŁëcäŇäŮL'èçĚäyžäyÄäyŁæäĞäĚäžŠäžEřijN
 éCčäzLäRfrazëä;ŁçTÍ ctypes.util.find_library() äG;æTřælēæšçæL'çijŽ

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

äyÄæŮëä;äçšëçÄšäžECäG;æTřazŠçŽDä;■ç;öijNéCčäzLärsäRfrazëäČRäyNéÍcèŁZæäüä;ŁçTÍ
 ctypes.cdll.LoadLibrary() ælēäLäë;äöČijN äĚüäy■ _path
 æYřæäĞäĚäžŠçŽDäĚlëürä;DijŽ

```
_mod = ctypes.cdll.LoadLibrary(_path)
```

āĠjæTřāžŠěcñāŁăēj;āŘŌrijNājăēIJĀēēAçijŪāEŽāGăäyĭēř■āRēāĬēāRŘāRŪčŁ'zāōŽčŽDčņēāRūāzūāēNč
ārśāČRāyNēĬēēZāyĭāzččāAçŁ'ĠăōťāyĀăăüijŽ

```
# 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
```

āIJĭēēZăōťāzččāĀăy■ijN. argtypes āśđāĀġæYřāyĀăyĭāēČčzDrijNāNēāRñāzEæšRăyĭāĠjæTřčŽDč
ēĀN .restype ārśæYřčŽyāžTčŽDēēTāZđčśzādNāĀČ ctypes
āōŽāzŁ'āžEāđ'ġēĠRčŽDčśzādNāržēsajijĹārTāēCc_double, c_int, c_short, c_floatç■Ĺ'ijĹ'ijN
āžčēāĭāžEāržāžTčŽDcæTřæ■ōčśzādNāĀČāēČăđIJă;ăæČšēōĹ'PythonēČ;ăđ'šāijăēĀŠæ■čçāōčŽDāRCæTřčśz
ēČčāzĹēēZāžŽčśzādNč■;āR■čŽDčzŠāōŽæYřā;ĹēĠ■ēēAçŽDăyĀăēāĀČāēČăđIJă;ăæšqæIJĹēēZāzĹāAŽiij
ēēYāRřēČ;āijZārijēĠræTř'ăyĭēġčēĠāZĭēēZčĭNāNčĀŌĹāĀČ
ā;ĲçTĭctypesæIJĹ'ăyĀăyĭēžzčČēčČzčŽDāIJræŪzæYřāŌšçTšçŽDcāžččāĀă;ĲçTĭčŽDæIJřēř■āRřēČ;ēũ\$Pytho
divide() āĠjæTřæYřāyĀăyĭā;Ĺăē;čŽDă;Nā■RijNāōČēAZēēĠăyĀăyĭāRCæTřēŽđ'āžēāRēăyĀăyĭāRCæTř
ār;çōāēēZæYřāyĀăyĭā;ĹăyĭēġAçŽDcæĹăēIJřijNā;EæYřāIJĹPythonăy■ā■t'ăy■çšēēAçæĀŌăăăyĀēZřčŽ
ăĲNāēČijNă;ăăy■ēČ;āČRāyNēĬēēēZăăüčōĀă■TčŽDăAŽiijŽ

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

ārśčŌŪēēZăyĭēČjæ■čçāōčŽDăüēă;IJijNāōČāijŽēēĬāR■PythonāržāžŌæTř'æTřčŽDăy■āRræZř'æTřzāŌšā
āržāžŌăŪĹ'āRĹāĹræNĠēŚĹčŽDāRCæTřijNājăēĀŽăyĭēIJĀēēĀēĹăđDăžăyĀăyĭčŽyāžTčŽDctypesāržēsā

```
>>> x = ctypes.c_int()
>>> divide(10, 3, x)
3
>>> x.value
1
>>>
```

āIJĭēēZēĠNrijNăyĀăyĭ ctypes.c_int āōđă;NēcñāŁZăžzāzūā;IJăyžăyĀăyĭāēNĠēŚĹēcñāijăēēZăŌzā
ēũ\$æZŏēĀŽPythonæTř'ă;çăy■āRŊčŽDæYřijNăyĀăyĭ c_int
āržēsqæYřāRřāžēēcñāŌăēTřčŽDăĀČ .value āśđāĀġāRřēcñçTĭāēēēŌăāRŪăĹŪăZř'æTřēēZăyĭāĀijāĀČ

āržāžŌēČčāžZăy■āČRPythonçŽDcēřČçTĭrijNēĀŽăyĭāRřāžēāEžZăyĀăyĭārçŽDāNēēēĀĠjæTřāĀČ
ēēZēĠNrijNāēĹsāžnēōĹ' divide() āĠjæTřēĀžēēĠăēČčzDăĭēēēTāZđăyđ'ăyĭçzšşăđIJijŽ

```
# 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() ãĢjæTřãRŁæYřäYÄäylæŮřčŽDæŇŠæLYãĀĆCäzččãAæIJšæIJZæŌëãRŮãŁřäYÄäylæŇĢëŠŁãŠ
 äjEæYřijŇãIJPythonäy■ijŇæŁSäznãŁĚëäzëĀĆëŽSëĚZäylæŮöëcYřijŽæTřčzDæYřãTëijšãŌĆæYřäYÄäylæŁ
 èĚYæYř array æŁãĀŮäy■ŽDäYÄäylæTřčzDijšèĚYæYřäYÄäyl numpy
 æTřčzDijšèĚYæYřëřt'æL'ÄæIJL'ëČjæYřijšãŌđëŽĚäYŁijŇNäYÄäylPythonãĀIJæTřčzDãĀĬæIJL'ãđ'Žġg■ãjčã

DoubleArrayType æijTčđ'žäZæÄŌæãũãđ'DčŘĚëĚŽġg■æČĚãĚtãĀĆ
 ãIJĚëĚZäylčszäy■ãŌŽäZŁ'äZĚäYÄäylæ■TäylæŮzæšT from_param() äĀĆ
 èĚZäylæŮzæšTčŽDëġSëL'sæYřæŌëãRŮäYÄäylæ■TäylæRĆæTřčDũãRŌãřĚãĚũãŘSäyŇëjñæ■cäyžäYÄäylæŘĬ
 ijŁæIJñãŁNäy■æYřäYÄäyl ctypes.c_double čŽDæŇĢëŠŁijL'ãĀĆ
 ãIJĬ from_param() äy■ijŇNäjããŘřäzëãĀŽäzzä;Tj;äæČšãĀŽčŽDäžŇãĀĆ
 ãRĆæTřčŽDčszãđŇãŘ■ëcñæRŘãRŮãĢZæĬëäzũëcñčTĬäžŌãĬĚãŘSãŁřäYÄäylæZt'ãĚũãjščŽDæŮzæšTäy■ãŌ
 äjŇãëČijŇNäëČãđIJäYÄäylæŁŮëãĬëcñäijäëĀŠëĚĢæĬëijŇëČčäZŁ typename äřsæYř list
 ijŇčDũãRŌ from_list æŮzæšTëcñëřČčTĬãĀĆ

ãřzäžŌãŁŮëãĬãŠŇãĚČčzDijŇfrom_list æŮzæšTãřĚãĚũëjñæ■cäyžäYÄäyl ctypes
 čŽDæTřčzDãřzëšãĀĆ èĚZäylčIJŇäYŁãŌzæIJLčČzãëĢæĀřijŇNäYŇĬëãŁSäznãjŁčTĬäYÄäylæZđ'äZŠãijŘãŁŇ
 ctypes æTřčzDijŽ

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

ãřzäžŌæTřčzDãřzëšãijŇfrom_array() æRŘãRŮãžTãšĆčŽDãĚĚã■YæŇĢëŠŁãžũãřĚãĚũëjñæ■cäyž
 ctypes æŇĢëŠŁãřzëšãĀĆäjŇãëČijŽ

```
>>> import array
>>> a = array.array('d', [1, 2, 3])
>>> a
array('d', [1.0, 2.0, 3.0])
>>> ptr_ = a.buffer_info()
>>> ptr
```

(continues on next page)

(continued from previous page)

```
4298687200
>>> ctypes.cast(ptr, ctypes.POINTER(ctypes.c_double))
<__main__.LP_c_double object at 0x10069cd40>
>>>
```

from_ndarray() æijTçd'žāžEāržāžŌ numpy æTřčzDčŽDèĭnæ■cæŠ■ä;IJāĀĆ
éĀŽèĚĜāōŽāzL DoubleArrayType çszāžūāIJĭ avg() çszādNç■ĭāR■äy■ä;ĚçTĭlāōČĭijN
éČčāzĹēfŽāyĭāG;æTřārsèČ;æŌēāRŪād'Žāyĭāy■āR■NçŽDçszæTřčzDèĭŠāĒēāžEĭijŽ

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

æIJñèĹĆæIJĀāRŌäyĀēČĭāĹēāRŠä;æijTçd'žāžEæĀŌæāūād'DčŘEäyĀäyĭçōĀā■TçŽDČçzŠæđDāĀĆ
āržāžŌçzŠæđDä;ŠĭijNä;āāRĭēIJĀēçAāČRäyNēĭčēfZæāūçōĀā■TçŽDāōŽāzLäyĀäyĭçszĭijNāNĒēāRñçZyāžTç

```
class Point(ctypes.Structure):
    _fields_ = [('x', ctypes.c_double),
                ('y', ctypes.c_double)]
```

äyĀæŪççszèçñāōŽāzLāRŌĭijNä;āārsāRfāžēāIJĭçszādNç■ĭāR■äy■æĹŪèĀĒæYřēIJĀēçAāōđä;NāNŪçzŠ

```
>>> p1 = sample.Point(1, 2)
>>> p2 = sample.Point(4, 5)
>>> p1.x
1.0
>>> p1.y
2.0
>>> sample.distance(p1, p2)
4.242640687119285
>>>
```

æIJĀāRŌäyĀāžZārRçŽDæRŘçd'žĭijŽāçCæđIJä;æČšāIJĭPythonäy■èøĚéŪōäyĀāžZārRçŽDČāG;æTřĭijN
ctypes æYřäyĀäyĭāĹæIJĭçTĭçŽDāG;æTřāžŠāĀĆ āřçōāāçCæ■d'ĭijNāçCæđIJä;æČšèçAāŌžèøĚéŪōäyĀā
Swig (15.9èĹĆāijŽèōšāĹr) æĹŪ CythonĭijĹ15.10èĹĆĭijĹāĀĆ

āržāžŌād'gādNāžŠçŽDèøĚéŪōæIJĭäyĭāyžèèAēŪōēçYĭijNçTšāžŌçtypesāžūāy■æYřāōNāĒĭēĜĭāĹāNŪř
éČčāzĹä;āārsāfĒēāžēĹsèt'žād'gēĜRæŪūēŪr'æĭēçijŪāĒZæĹĀæIJĭçŽDçszādNç■ĭāR■ĭijNārsāČRä;Nā■Räy
āçCæđIJāG;æTřāžŠād'šād'■æĭČĭijNä;æēfYā;ŪāŌžçijŪāĒZāĹĹād'ŽārRçŽDāNĒēçĒāG;æTřāŠNæTřæNĀçsz
āRēād'ŪĭijNēZd'ēīđä;āāūççzRāōNāĒĭçš;éĀŽāžEæĹĀæIJĭāžTāsČçŽDČæŌēāRççzEēĹĆĭijNāNĒæNñāEēĀ■
éĀŽāyāyĀäyĭāĹĹārRçŽDāžççāAçijžēŽūāĀAèøĚéŪōēūĹçTĭNæĹŪāĒūāzŪçszāijjēTŽēřfārsèČ;ēōĹPythonçĹ

äJäyž ctypes çŽDäyÄäylæŽŁäzčijNä;æŁYāRfäzëeÄČëŽŚäyŃCFFIäÄČCFFIæRŘä;ŽäžEā;Łād'Žç
ä;EæYřä;ŁçTÍCër■æşTāzūæTřæŃAæŽt'ād'ŽénYçžğçŽDCäzčçāAçşzādŃāÄČ
āŁřāEŽēŁŽæIJñäžēäyžæ■čijŃCFFIēŁYæYřäyÄäylçŽyāřzè;ČæŮřçŽDāũēčĹŃijŃ
ä;EæYřāŃČçŽDætAèqŃäžçæ■čāIJlāfñéAşäyŁā■GāÄČ çTŽeGşèŁYæIJLāIJlèŃlèŃzāIJlPythonāřEælēçŽDçL

17.2 15.2 çŃÄ■TçŽDCæL'řāsTæĹāĹU

èŮŃéčY

ä;āæČşäy■ä;ĹéĹāāEūāzŮāũēāEūijŃçŽt'æŎēä;ŁçTÍPythonçŽDæL'řāsTAPIælēçijŮāEŽäyÄäžŽçŃÄ■Tç

èğčāEşæŮzæāĹ

āržāžŎçŃÄ■TçŽDCäzčçāAřijŃædDāžžäyÄäylèĠāŃŽāzL'æL'řāsTæĹāĹUæYřā;ŁāŃzæYŞçŽDāÄČ
ä;IJäyžçññäyÄæ■čijNä;æŁIJāèçAçāŃāŁĹä;äçŽDCäzčçāAæIJL'äyÄäylæ■ççāŃçŽDād't'æŮGāzūāÄČä;ŃāçČij

```
/* 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);
```

éÄŽäyŷæĹèèŃijŃēŁŽäyĹād't'æŮGāzūēçAāržāžTäyÄäylāũşçzŘèçŃā■TçŃñçijŮērŚēŁĞçŽDāžŞāÄČ
æIJL'āžEēŁŽäžŽijŃäyŃéĹæŁŚāžñæijTçd'žäyŃçijŮāEŽæL'řāsTāĠ;æTřçŽDäyÄäylçŃÄ■Tā;Ńā■ŘijŽ

```
#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);
}

```

ěĕAçzŠăoŽefŽăyŁæL'ŕăŝTăĹăăiŮijŇăČŘăyŇéĹĕfŽăăăăLŽăzžăyĂăyŁ setup.py
 æŮĞăžŭijŽ

```
# 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']
                    )
      ]
)
```

python3 buildlib.py build_ext --inplace

```
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 %
```

sample.so

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

Python3


```

/* Indicate we're done working with the buffer */
PyBuffer_Release(&view);
return Py_BuildValue("d", result);
}

```

äyÑéíçæĹŚäzñæijTçd'žäyÑèĹZäyĹæĹl'āsTāĜ;æTṛæYṛæĆä;Tāũëä;IJçŽDiiJŽ

```

>>> 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
>>>

```

èóíèőž

ārEäyÄäyĹæTṛçzDāržèšäijäçzŽCāĜ;æTṛāRṛèĈ;æYṛäyÄäyĹæĹl'āsTāĜ;æTṛāAZçŽDæIJÄäyÿèġAçŽDäz
 āĹĹād'ŽPythonāžTçTīĈlNāžRīijNāžŌāZ;āĈRād'DçRĒāĹrçġSā■ēēōaçōŪīijNēĈ;æYṛāšzāžŌēñYæĀġèĈ;çŽD
 éĀŽèĹĜçijŪāĒZèĈ;æŌēāRŪāžūæS■ä;IJæTṛçzDçŽDäzççāAīijNä;āāRṛäzèçijŪāĒZāĹĹæ;çŽDāĒijāōžèĹZāžŽ
 èĀNäy■æYṛāRṛèĈ;āĒijāōžä;æĜĹāūsçŽDäzççāAāĀĆ

äzççāAçŽDāĒšéTōçĈzāIJläžŌ PyBuffer_GetBuffer() āĜ;æTṛāĀĆ
 çzŽāōŽäyÄäyĹäzæDRçŽDPythonāržèšäijNāōĈāijZèrTçlĀāŌžèŌūāRŪāžTāsCāĒĒā■YāfæAṛīijNāōĈçōĀā
 1. äijäçzŽ PyBuffer_GetBuffer() çŽDçĹ'žæōĹæāĜāĹŪçzŽāĜzāžĒæĹĹĀéIJçŽDāĒĒā■YçijSāĒšçsžāç
 äĹNāēĈīijNPyBUF_ANY_CONTIGUOUS èāĈd'žæYṛäyÄäyĹèĹdçz■çŽDāĒĒā■YāNžāššāĀĆ

āržāžŌæTṛçzDāĀā■ŪèĹCā■ŪçņäyšāŠNāĒŪāžŪçsžāijjāržèšæĀNēĹĀīijNäyÄäyĹ

Py_buffer çzŞædĐä;ŞāNĒāRñāzĒæL'ĀæIJL'āzTāsCāĒĒā■ŸçŽDāfæAřāĀC
 āōČāNĒāRñāyĀāyġæNĠāRŚāĒĒā■ŸāIJřāĪāĀĀād' ġārRāĀĀāĒČçř' āād' ġārRāĀĀæāijāijRāŠNāĒūāzŪçzĒĒ

```
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;
```

æIJñĒLCāy■īijNāĒLSāznāRġāĒŞæşġæŌēāRŪāyĀāyġāRñçşġāžæřōçCzæTřæTřçzĐä;IJāyžāRCæTřāĀC
 èĒAæCĀæŞēāĒČçř' āæŸřāRĒæŸřāyĀāyġāRñçşġāžæřōçCzæTřīijNāRġĒIJāĒĒĒĒĒĒ
 format āśđæĀġæŸřāy■æŸřā■ŪçņæyşāĪdāĀĪ. èĒŽāyġāzşæŸř struct
 æġāāĪŪçTġĒĒĒçijŪçāĀžNĒĒZāĪŪæTřæ■ōçŽDāĀC éĀŽāyŸæġēēōşīijNformat
 āRřāzēæŸřāzžā;TāĒījāōž struct æġāāĪŪçŽDāēāijāijRāNŪā■ŪçņæyşīijN
 āzūāyTāēCāđIJæTřçzĐāNĒāRñāzĒCçzŞædĐçŽDĒřġāōČāRřāzēāNĒāRñāđ' ŽāyġāĀijāĀC
 āyĀæŪæĒLSāznāūşçzRçāōāōŽāžĒāzTāsCçŽDçijŞā■ŸāNžāĒæAřīijNéCčāRġĒIJāēĒAçōĀā■TçŽDāřĒāōČāij
 āōđēŽĒāyġĪijNāĒLSāznāy■āĒĒāNĒāĒCæŸřāĒŌæāūçŽDæTřçzĐçşzādNāĒŪēĀĒāōCæŸřēčnāzĀāzĪāzŞāĪZ
 èĒŽāzşæŸřāyžāzĀāzĪēĒŽāyġāĠ;æTřēČ;āĒījāōž array æġāāĪŪāzşēČ;āĒījāōž numpy
 æġāāĪŪāy■çŽDæTřçzĐāžĒāĀC

āIJġēĒTāZđæIJāçzĪçzŞæđIJāžNāL'■īijNāzTāsCçŽDçijŞāĒşāNžēġĒāZġāĒĒēāzā;ĒçTĪ
 PyBuffer_Release() éĠĒæTġ;æŌĪāĀC āzNāĒĪāzēēĒAēĒŽāyĀæ■ēæŸřāyžāžĒēČ;æ■ççāōçŽDçōāçRĒ

āRñāūīijNāēIJñĒLCāzşāzĒāzĒāRġæŸřāēijTçđ' žāžĒæŌēāRŪæTřçzĐçŽDāyĀāyġāRçŽDāžççāAçĪĪĠāō
 āēCāđIJā;āçIJşçŽDēĒAād' ĐçĒĒæTřçzĐīijNā;āāRřēČ;āijŽçčřāĪřād' Žçzt' æTřæ■ōāĀĀād' ġæTřæ■ōāĀĀāy■ā
 éCčāzĪārsāġ;ŪāŌzā■æŽt' ēnŸçžġçŽDāyIJēēfāžĒāĀCā;āēIJāēĒAāRCēĀČāōŸæŪzæŪĠæāçæġēēŌūāRŪæZt'
 āēCāđIJā;āēIJāēĒAçijŪāĒŽæŪĪāRĪāĪĪæTřçzĐād' ĐçĒĒçŽDād' ŽāyġāĪĪāśTīijNéCčāzĪēĀŽēĒĠCytho

17.4 15.4 āĪĪCæĪĪ'āsTġæġāāĪŪāy■æŞ■āĪJéŽRāġcæNĠéŞĪ

éŪōēčŸ

āġāæIJL'āyĀāyġāĪĪ'āsTġæġāāĪŪēIJāēĒAād' ĐçĒĒCçzŞædĐä;Şāy■çŽDæNĠéŞĪĪijN
 āġĒæŸřā;āāRĪāy■æČşæŽt' ēIJşçzŞædĐä;Şāy■āzžā;TāĒĒēČĪçzĒĒĪCçzŽPythonāĀC

ēġčāĒşæŪzæāĪ

ēŽRāġççzŞædĐä;ŞāRřāzēāĪĪāōzæŸşçŽDēĀŽēĒĠāRġāōČāznāNĒēčĒāĪĪēČūāZĪāřzēşāy■æġēād' ĐçĒĒ
 ēĀČēŽŞāĒSāznā;Nā■RāzççāĀāy■çŽDāyNāĪŪCāzççāAçĪĪĠāēōīijŽ

```
typedef struct Point {
    double x,y;
} Point;

extern double distance(Point *p1, Point *p2);
```

äyÑéÍæYřäyÄäyİä;İçTİèCũăZŁăÑĚèčĚPointçzŞæđDä;ŞăŠŇ distance()
 åĜ;æTřçŽDæL'řāsTäzččăĂăđă;ŇijŽ

```
/* 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);
}

```

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

ĕĈũāZŁāŠŃCæŃĜēŠŁçšzaiijjāĀĈāIJlāĒĒēĈlīijŃāōĈāznēŌūāRŪāyĀāylēĀZçTlāēŃĜēŠŁāŠŃāyĀāylāR
 PyCapsule_New() āĜ;æTřřāŁāōzæYŠçŽĎēcŋāŁZāzžāĀĈ
 āRēād'ŪīijŃāyĀāylāRfēĀLčŽĎæđRæđĎāĜ;æTřēĈ;ēcŋçzŠāōZāŁrēĈũāZŁāyŁīijŃçTlāēāIJlēĈũāZŁāřzēsāē
 ěAæRŘāRŪēĈũāZŁāy■čŽĎæŃĜēŠŁīijŃāRřā;ĕçTlī PyCapsule_GetPointer()
 āĜ;æTřřāzūāŃĜāōZāR■çğřāĀĈ āēĈæđIJæRŘā;ŁçŽĎāR■çğřāŠŃēĈũāZŁāy■āŃzēĒ■æLŪāĒŪāzŪēTŻēřrāĜž
 æIJŋēŁĈāy■īijŃāyĀāřzāūēāĒūāĜ;æTřřāĀTāĀT PyPoint_FromPoint()
 āŠŃ PyPoint_AsPoint() ěcŋçTlāēāŁZāzžāŠŃāzŌēĈũāZŁāřzēsāy■æRŘāRŪ-
 Pointāōđā;ŃāĀĈ āIJlāzžā;TæL'l'āsTāĜ;æTřřāy■īijŃæŁSāznāijŽā;ĕçTlēŁZāzžāĜ;æTřēĀŃāy■æYřçŽt'æŌēā;
 ēŁZçğ■ēōŁēōāq;ĕā;ŪāŁSāznāRřāzēā;ŁāōzæYŠçŽĎāžTāřzārĒāēlēārZPointāžTāyŃçŽĎāŃĒēĕĒēçŽĎæŽt'æTž
 ā;ŃāēĈīijŃāēĈæđIJā;āĒĒsāōZā;ĕçTlāRēād'ŪāyĀāylēĈũāZŁāžĒīijŃēĈčāzŁāRlēIJāēēAæŽt'æTžēŁZāyđ'āylā
 āřzāzŌēĈũāZŁāřzēsāyĀāylēŽ;çĈzāIJlāžŌāđĈāIJ;āZđæTūāŠŃāĒēĒā■YçōāçŘĒāĀĈ
 PyPoint_FromPoint() āĜ;æTřřāŌēāRŪāyĀāyl must_free
 āRĈæTřīijŃ çTlāēāēŃĜāōZā;ŠēĈũāZŁēcŋēTĀāfAæŪūāžTāsĈPoint *
 çzŠæđĎā;ŠæYřāRēāžTēřēcŋāZđæTūāĀĈ āIJlāšŘāzŽCāzčçāĀy■īijŃā;ŠāsđēŪōēçYēĀŽāyā;ŁēŽ;ēcŋāđ'Ī
 çlŃāžRāŠYāRřāzēā;ĕçTlī extra āRĈæTřřālēāŌğāŁūīijŃēĀŃāy■æYřā■TæŪzēlççŽĎāĒsāōZāđĈāIJ;āZđæT
 ēēAæšlāĎRçŽĎæYřāŠŃçŌřāIJLēĈũāZŁāēIJLāĒēšçŽĎæđRæđĎāZlēĈ;ā;ĕçTlī
 PyCapsule_SetDestructor() āĜ;æTřřālēāŽt'æTžāĀĈ
 āřzāzŌēūLāRŁāŁrçzŠæđĎā;ŠçŽĎCāzčçāĒēĀŃēlĀīijŃā;ĕçTlēĈũāZŁāYřāyĀāylāfTē;ĈāRŁçŘĒēçŽĎ
 ā;ŃāēĈīijŃæIJLāŪūāĀZā;āāzūāy■āĒšāĕĈæŽt'ēIJççzŠæđĎā;ŠçŽĎāĒēĒēĈlāfāæĀřāLŪēĀĒāřāĒēūē;ŋāēĈā
 ēĀZēŁGā;ĕçTlēĈũāZŁīijŃā;āāRřāzēāIJlāōĈāyŁēlāēT;āyĀāylē;zēGRçzçççŽĎāŃĒēĕĒēĀZlīijŃçĎūāRŌārĒāō

17.5 15.5 äzÓæL'ŕásTælaaIÜäy■áoŽázL'áŠNárijáGžCçŽĐAPI

éUóécY

ä;äæIJL'äyÄäyI CæL'ŕásTælaaIÜäy■NâIJlâEĚčČláoŽázL'ázEā;Lād'ŽæIJL'çTlçŽĐāG;æTrijNä;äæČšārEā
APIā;ŽāĚüāzÜāIJŕæŰzā;ŁçTlāĀČ ä;äæČšāIJlâĚüāzÜæL'ŕásTælaaIÜäy■ā;ŁçTlēŁŽázZāG;æTrijNä;EæŰŕāy
āzūāyTēĀŽēŁGCçijŰērSāŽl/éS;æŌēāŽlāĚāAŽçIJNäyLāŌzçL'zāLnáđ'■æIČrijLæLŰēĀĚäy■āRŕēČ;āAŽāL

èğčāEşæŰzæaL

æIJnēLČäyžèeAéUóécYæŰŕæČā;Tād'ĐçŘE15.4ārRēLČäy■æRŘāLŕçŽĐPointāržèšāāĀČāzTçzEāZđāy.

```
/* 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);
}
```

çŌŕāIJlçŽĐēUóécYæŰŕæĀŌæāuārE PyPoint_AsPoint()
āŠN Point_FromPoint() āG;æTŕā;IJäyžAPIārījāGžījN
ēŁŽæāuāĚüāzÜæL'ŕásTælaaIÜēČ;ā;ŁçTlāzūēS;æŌēāŌČāznījNārTāēČāēČæđIJā;äæIJL'āĚüāzÜæL'ŕásTāzš
èeAèğčāEşēŁZäyŁēUóécYījNēēŰāĚLēeAäyž sample æL'ŕásTāEŽäyŁēŰŕçŽĐād't æŰGāzūāR■āRn
pysample.h iijNāeČäyNījŽ

```
/* 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;

#ifdef PYSAMPLE_MODULE
/* Method table in external module */
```

(continues on next page)

(continued from previous page)

```
static _PointAPIMethods *_point_api = 0;

/* Import the API table from sample */
static int import_sample(void) {
    _point_api = (_PointAPIMethods *) PyCapsule_Import("sample._point_
↪api", 0);
    return (_point_api != NULL) ? 1 : 0;
}

/* Macros to implement the programming interface */
#define PyPoint_AsPoint(obj) (_point_api->aspoint)(obj)
#define PyPoint_FromPoint(obj) (_point_api->frompoint)(obj)
#endif

#ifdef __cplusplus
}
#endif
```

èĚŽéĜŇæIJĀéĜ■ēçAçŽĎéČlāLEæŸřāĜ;æŦřæŇĜéŠĹèāĹ _PointAPIMethods .
āōČaijZāIJlārijāĜzæĹāāIŮæŮüēcñāĹiāġNāŇŮiijŇçDūāRŌārijāĒēæĹāāIŮæŮüēcñæšēæL;āĹřāĀĆ
āĤōæŦzāŌšāġŇçŽĎæLŦāsŦæĹāāIŮæĹēāāñāĒĒēāĹæāijāzūārĒāōČāČRāyŇéĹcēĤZæāūārijāĜzīijŽ

```
/* 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;
}

```

æIJĀŘŮijŇäyŇéÍæŸräyÄäyĽŰřčŽDæLŦŕāsŦæÍqâĪŮäĽŇā■ŘiijŇčŦĽæĽæĽæĽ;āzŭä;ĤčŦĽæŹăžZAPIā

```

/* 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}
};

```

(continued from previous page)

```
static struct PyModuleDef ptexamplemodule = {
    PyModuleDef_HEAD_INIT,
    "ptexample",           /* name of module */
    "A module that imports an API", /* Doc string (may be NULL) */
    -1,                    /* Size of per-interpreter state or -1 */
    PtExampleMethods       /* Method table */
};

/* Module initialization function */
PyMODINIT_FUNC
PyInit_ptexample(void) {
    PyObject *m;

    m = PyModule_Create(&ptexamplemodule);
    if (m == NULL)
        return NULL;

    /* Import sample, loading its API functions */
    if (!import_sample()) {
        return NULL;
    }

    return m;
}
```

çijŮerSèfZäyIæŮræIaaiŮæŮüijNä;äçTŽèGšäy■éIJÀèçAaŌzèÄČèŽŚæÄŌæäüarEaĜ;æTřřžŠæLŮäžčç
ä;NäçČüijNä;ääRřřžæäČRäyNéIcéfZæäüaLZäžžäyÄäyIçõÄa■TçŽD setup.py æŮĜäžüijŽ

```
# setup.py
from distutils.core import setup, Extension

setup(name='ptexample',
      ext_modules=[
          Extension('ptexample',
                  ['ptexample.c'],
                  include_dirs = [], # May need pysample.h
→directory
          )
      ]
)
```

æçCædIJäyÄaLĜæ■çäyüijNä;ääijŽaRŚçŌřä;äçŽDæŮræL'äśTäĜ;æTřèČ;äŠNäóŽäžL'äIJäEüäžŮæIaai
APIäĜ;æTřäyÄèüèfRèaNçŽDä;Läë;äÄČ

```
>>> import sample
>>> p1 = sample.Point(2,3)
>>> p1
<capsule object "Point *" at 0x1004ea330>
>>> import ptexample
```

(continues on next page)

æIJñēŁĆąšżăŹŌăyĂăyŭŁ'■æŔŔăŕŝæŸŕijŃēĆŭăZŁăŕžēŝăēČ;ēŌŭăŔŬăzză;Ŧă;ăăČŝēēAçŽĐăŕžēŝăçŽĐăēŹăăŭçŽĐŕŕijŃăŏŹăZŁ'ăŭăŭăŭăŹăăŃăăŸăăŸăŭăŤă;ăŦŕăŃŖēŖŤçŽĐçzŝăđĐă;ŝŕijŃăŤăăžăăyĂăyŭăŃăăŭăŃăăē sample. _point_ api.

āĖūāzŪæĺāāĬĖĈ;āđ'šāĬĬāřījāĖĖæŪūēŌūāRŪāĬĤrēfZāyĺāśđæĀğāzūæRŘāRŪāzŤāsĈĈZĎæŃĜēŚĬāĈĈ
 āzŃāōđāyĹījŃPythonæRŘā;ZāZĖPyCapsule_Import()
 āūēāĖūāĜj;æŤřījŃāyžāZĖāōŃāĬRæĹ'ĀæĬĬĈZĎæēēĹđ'āĈĈ
 ā;āāRĹēĬĬAæRŘā;ZāsđæĀğĈZĎāR■ā■Ūā■šāRřījĬĹæŤāēĈsample._point_apīīīĬĹījŃĈĎūāRŌāzŪāřśāījZāyĀ

[illegible]

æIJĀāRŌrijNēƷYæIJL'äyÄäyƭeĜ■ēeAçŽDāŌšāZāēol'ā;āāŌzā;ƷçTlēƷZäyƭeĀæIJræIēēŠçæŌēæIaaiUā
 æçCædIJā;āäy■æÇšā;ƷçTlæIJnæIJžçŽDæĀæIJrijNēCčā;āāršāfĒēāzā;ƷçTlāĒšāznāžSçŽDénYçžgçL'zæĀgā
 āçNæCrijNārEäyÄäyƭeŽōēĀŽçŽDAPiāĜ;æTṛæTçāĒēäyÄäyƭeĒšāznāžSāžūçāōāfIæL'ĀæIJL'æL'r'āsTæIaaiU
 èƷçg■æŪzæšTçqāōāōdāRfēaNūijNā;EæYrāōČçŽyārççZAçRŌrijNçL'zāL'næYrāIJād'gādnçšçzçšy■āĀC
 æIJnèLČæijTçd'zāžEāçCā;TēĀŽēƷGPythonçŽDæŽōēĀŽārijāĒēæIJzāLūāšNāzĒāzĒāGāyƭeCūāZLerČçTlæ
 ārçzāŌæIaaiUçŽDcijUērSrijNā;āāRlēIJĀēeAāōZāzL'ād't'æŪGāzūrijNēĀNāy■æIJĀēeAēĀCēZSāĜ;æTṛāžSçŽ

æZt'ad'ZăĖšăžŎăLr'çTlC APIăiěădĎĎĂăăL'răsTălăăIŮçŽĎăăăăAřăRřăzěăăRĆěăĎĎ
PythonçŽĎăŮĖăăăă

17.6 15.6 äžŒCèr■élĂäÿ■èrČçŤÍPythonäžččăA

ä;äæČšâIJİCäy■āōL'āĖİčŽĐæL'gèaŊæšŘäyİPythonèČčŤlázûēŤāŽđçzšæđIJçzŽCăĂĆ
ä;ŊæČcīİŊä;äæČšâIJİCèr■ēİÄäy■ä;ŁçŤlæšŘäyİPythonāĠ;æŤrā;IJäyžäyÄäyİāŽđèrČăĂĆ

aIJCèr■èlĀäy■èrCçTĭPythonéIdāyÿçöĀā■TĭijNāy■èfGēōç,èōāāLřāyĀāzZārRçĭ■eUĭāĀC
 äyNéIcçZĬCäzçcāAāSĬērL'ä;äæĀŌæäüāōL'āĒlçZĬDèrCçTĭijZ

```

#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
}

```

æAä;ŁçŦlèfZäyŁaĞ;æŦrijŊä;æeIJÄðeAeŌuāRŪäijæĀŠèŁĞæIèçŽDæšŘäyŁaũšā■ŸāIJlPythonèřČçŦlçŽ
 æIJLā;Łād'Žçg■æŮzæšŦāRřazèèŌ'ä;æeŁZæāũāAŽijŊ æŦāeČārEäyĀäyŁaRřerČçŦlāřzèšāijäçžŽäyĀäyŁæL
 äyŊéIcæŸřäyĀäyŁçŌĀā■Ŧä;Ŋā■ŘçŦlæIèæŌ' ečřazŌäyĀäyŁaŦŊāEèçŽDPythonèğçéĞŁāŽlāy■èřČçŦlāy

```

#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;
}

```

èeAædĐāžzä;Ŋā■ŘāžččāAijŊä;æeIJÄðeAçijŮerŠCāzũārEāŌČeŠ;æŌěāLřPythonèğçéĞŁāŽlāĀĆ
 äyŊéIcçŽDMakefileāRřāžæŦŽä;æĀŌæāũāAŽijLāy■eŁĞāIJlā;æIJzāŽlāyŁéIcēIJÄðeAäyĀāžŽēĒ■ç;ŋijL

```

all::
    cc -g embed.c -I/usr/local/include/python3.3m \
        -L/usr/local/lib/python3.3/config-3.3m -lpython3.3m

```

çijŮerŚázűeŁRèaŃaijŽāžgçŤšçszäijijäyŃeİcçŽĐèŁŚāGžiiž

```
0.00 0.00
0.10 0.01
0.20 0.04
0.30 0.09
0.40 0.16
...
```

äyŃeİcæŸräyÄäylçİāŁöäy■aŖŃçŽĐäŁŃa■ŖiiŃŃaŤçd'žāžEäyÄäylæLŤāŤāGĳæŤriiŃŃ
āōČæŌēaŖŮäyÄäylāŖŕēŤČçŤlāržèſaāŚŃāĒūāzŮāŖČæŤriiŃŃāzūāŖEāōČāznāijāēĀŠçzŽ
call_func() æİēāAŽæŤŃerŤiiž

```
/* 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);
}
```

äĳçŤİēŁŽäyĳæLŤāŤāGĳæŤriiŃŃāĳæēAāČŖäyŃeİcèŁŽæāūæŤŃerŤāōČiiž

```
>>> import sample
>>> def add(x, y):
...     return x+y
...
>>> sample.call_func(add, 3, 4)
7.0
>>>
```

èōİēōž

āēČædİJā;āāİJÍCèŕ■ēİÄäy■ērČçŤİPythoniiŃŃeēAèōŕä;ŖæİJĀēG■èēAçŽĐæŸŕCèŕ■ēİÄäijŽæŸräyžā;ŚāĀ
āžſāŕſæŸŕēŕŤiiŃŃCèŕ■ēİĀēŤſēŤčædĐēĀāāŖČæŤŕāĀĀēŤČçŤİPythonāGĳæŤŕāĀĀæčĀæſēāijČāyŷāĀĀæčĀæ

äĳİJäyžçŃŃäyĀæ■ēiiŃŃā;āāŁĒēāžāĒŁæİJL'äyÄäylēāŁcd'žā;āāŖEēēAērČçŤİçŽĐPythonāŖŕēŤČçŤlāržèſaā
ēŁŽāŖŕāžæŸräyÄäylāGĳæŤŕāĀĀçſzāĀĀæŮžæſŤāĀĀāĒĒçĳōæŮžæſŤæLŮāĒūāzŮāžzæĐŖāōđçŌŕāžE
__call__() æŚ■āĳİJçŽĐäyİJēēŁāĀČ äyžāžEçāōāŁæŸŕāŖŕēŤČçŤİçŽĐiiŃŃāŖŕāžæāČŖäyŃeİcçŽĐäžççāĀē
PyCallable_Check() āAŽæčĀæſēiiž

```
double call_func(PyObject *func, double x, double y) {
    ...
    /* Verify that func is a proper callable */
    if (!PyCallable_Check(func)) {
```

(continues on next page)

(continued from previous page)

```
fprintf(stderr, "call_func: expected a callable\n");
goto fail;
}
...
```

ãĲĲCäzççãAëĜÑäd' ĎçŘĚéT'Zèřřä; äéĲĲÄëAæäijäd' ŮčŽDärRäſČãĀĆäyÄèĽñæĲèèõřřijÑä; ääy■ēČ; äzĒä
éT'ZèřřäZT'èřä; ĲçTĲCäzççãAæŮzäijRæĲèècnäd' ĎçŘĚãĀĆĀĲĲēſŽéĜÑřijÑæĽSäzñæĽŠçõŮärĚäřzéT'ZèřřçŽD
abort() çŽDèT'Zèřřäd' ĎçŘĚäZĲãĀĆ äõČäijŽçzŠæĲšæŌĽæT' äyĲĲĲNäzRijÑäĲĲĲĲšäõđçŌřäčČäyNéĲä; ää
ä; äëAæõřä; ŘçŽDæYřäĲĲēſŽéĜÑCæYřäyzeğřřijÑäZäæ■d' äzŮæšæĲĲĲēŮšæĽZäĜzäijČäyŷçŽyärzäzTçŽDæ
éT'Zèřřäd' ĎçŘĚæYřä; äãĲĲçijŮčĲĲNæŮŮäſĒëäzèAèĀČèZŠçŽDäzNæČĒãĀĆ

ērČçTĲäyÄäyĲäĜ; æTřçŽyärzæĲèèõšäĲĲçõĀā■TäĀTäĀTäRĲēĲĲÄëAä; ĲçTĲ
PyObject_Call() ĲijÑ äijäyÄäyĲäRřerČçTĲärzèšäçzŽäõČãĀäyÄäyĲäRČæTřäĒČçzĎäŠNäyÄäyĲäRřéÄ
èAæĎĎäzžäRČæTřäĒČçzĎäĲŮā■ŮäĒyřijÑä; ääRřäzèä; ĲçTĲ Py_BuildValue()
,äçČäyNřijŽ

```
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);
    ...
}
```

äçČæĎĲæšæĲĲĲäĒšéT'õä■ŮäRČæTřřijÑä; ääRřäzèäijäéĀŠNULLäĀĆä; Šä; äëAæřČçTĲäĜ; æTřæŮřřijÑ
éĲĲÄëAçäõäſĲä; ĲçTĲäzĒ Py_DECREF() æĲŮèĀĒ Py_XDECREF() æyĒçŘĚäRČæTřäĀĆ
çññäzNäyĲäĜ; æTřçŽyärzäõĲäĒĲçCzřijÑäZäyžäõČäĒAèõyäijäéĀŠNULLæNĜéŠĲĲĲĲçZt' æŌèäſ; çTçäõČřij
èſŽäzšæYřäyžäzĀäzĲæĲSäzñä; ĲçTĲäõČæĲæyĒçŘĚäRřéĀĲçŽDäĒšéT'õä■ŮäRČæTřäĀĆ

ērČçTĲäyĜPythonäĜ; æTřäzNäRŌřřijÑä; ääſĒëäzæčĀæšæYřäŘæĲĲĲäijČäyŷäRŠçTšäĀĆ
PyErr_Occurred() äĜ; æTřäRřèçñçTĲäĲäAŽèſŽäzŮäzNäĀĆ
ärzärzäzŌäijČäyŷçŽDäd' ĎçŘĚäřšæĲĲĲçCzézzçČæzĒřřijNçT'säzŌæYřçTĲČèř■ēĲäĒĒçŽDřřijÑä; äæšæĲĲĲäČ
äZäæ■d' ĲijÑä; ääſĒëäzèAèõçç; õäyÄäyĲäijČäyŷçĲŮæĀAçäĀřijNæĽŠä■řäijČäyŷäſæAřæĲŮäĒäzŮčŽyäzT
ĲĲĲēſŽéĜÑřijÑæĽSäzñéĀĲæN' äzĒçõĀā■TçŽD abort()
æĲèäd' ĎçŘĚäĀĆĀŘæäd' ŮřřijNäijäçzšCçĲNäzRäSŸäRřèČ; äijŽçZt' æŌèèõĲçĲNäzRäēTæzČäĀĆ

```
...
/* Check for Python exceptions (if any) */
if (PyErr_Occurred()) {
    PyErr_Print();
    goto fail;
}
```

(continues on next page)

```
...
fail:
    PyGILState_Release(state);
    abort();
```

äzÖëŕÇçTÍPythonâĜ;æTŕçŽĐëŦâŽđâĀijäy■æRŔâRŪäŦæAŕéĂŽăyÿëAëŦŽëaŦçşzâđNæčĂæşëăŠŦă
 èeAëŦŽæăăĂŽçŽĐëŦiijŦă;ăăŦĒëqzä;ŦçTÍPythonâŕžëşqâşCăy■çŽĐâĜ;æTŕăĂĆ
 âIJlëŦŽëĜŦăĹSăznă;ŦçTlăžE PyFloat_Check() âŠŦ PyFloat_AsDouble()
 ælëæčĂæşëăŠŦăRŔâRŪPythonæŦçCzæTŕăĂĆ

æIJĀăŔŌăyĂăyĹëŬŌëcŸæŸŕăŕžăžŌPythonăĒlăşĂëŦAçŽĐçŏaçŔĒăĂĆ
 âIJlCër■ēlĂăy■ēŦēŦŬŌPythonçŽĐæŬŭăĂŽiijŦă;ăēIJĀëeAçăŏăŦlGILëcŋă■ççăŏçŽĐëŌŭăRŪăŠŦăĜĹæT;ăž
 äy■çĐŭçŽĐëŦiijŦăŦŕŕëÇ;ăijŽăŕijëĜŦëĝçëĜĹăŽlëŦâŽđëŦŽëŕŕæTŕæ■ŏăĹŪëĂĒçŽt'æŌëăēŦæžCăĂĆ
 èŕÇçTÍ PyGILState_Ensure() âŠŦ PyGILState_Release()
 âŔŕăžëçăŏăŦlăyĂăĹĜëÇ;ëÇ;æ■çăyÿăĂĆ

```
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();
}
```

äyĂæŬëëŦâŽđiijŦPyGILState_Ensure() âŔŕăžëçăŏăŦlëŕÇçTlçžŦçlŦçŦŦă■ăPythonëĝçëĜĹăŽlă
 âŕşçŏŬCăžççăAëŦŔëaŦăžŌăŔëăđ'ŬăyĂăyĹëĝçëĜĹăŽlăy■çşëéAşçŽĐçžŦçlŦăžşæşăžŦăĂĆ
 èŦŽæŬŭăĂŽiijŦCăžççăAăŔŕăžëëĜŦçŦşçŽĐă;ŦçTlăžză;ŦăŏCæÇşëeAçŽĐPython C-API
 âĜ;æTŕăĂĆ èŕÇçTlăŦŔăĹşăŔŌiijŦPyGILState_Release()ëcŋçTlălëëŏşëĝçëĜĹăŽlăAçăđ'ăĹŕăŌşăĝŦçĹă

èeAæşlăĐŔçŽĐæŸŕăŕŔăyĂăyĹ PyGILState_Ensure()
 èŕÇçTlăŦĒëqzëŭşçlĂăyĂăyĹăŦžëĒ■çŽĐ PyGILState_Release()
 èŕÇçTlăĂŦăĂŦă■şă;ŦæIJL'ëŦŽëŕŕăŔşçŦşăĂĆ âIJlëŦŽëĜŦiijŦăĹSăznă;ŦçTlăyĂăyĹ goto
 èŕ■ăŔëçIJŦăyĹăŌžæŸŕăyĹăŔŕæĂŦçŽĐëŦçëŏăiijŦă;EæŸŕăŏđëŽĒăyĹăĹSăznă;ŦçTlăŏCælëëŏşæŌĝăĹŭălČă
 âIJl fail: æăĜç■;ăŔŌëlççŽĐăžççăAăŠŦPythonçŽĐ fianl:
 âlŬçŽĐçTlăĂŦæŸŕăyĂæăŭçŽĐăĂĆ

ăeČăđIJă;ăă;ŦçTlăĹĂăIJL'ëŦŽăžŽçžăŏžălëçijŬăEŽCăžççăAŕiijŦăŦĒăŦŦăŕžGILçŽĐçŏaçŔĒăĂăij
 ä;ăăijŽăŔşçŌŕăžŌCër■ēlĂăy■ëŦÇçTÍPythonëĝçëĜĹăŽlăŸŕăŔŕëŦăçŽĐăĂŦăĂŦăŕşçŏŬăĒăđ'ăēlČçŽĐçlŦăž

17.7 15.7 äžÓCæL'f'ásTäy■éGŁæT¿áĖÍásĀéTĀ

éŮóécŸ

ä;ăæČšèŃ' CæL'f'ásTäzččăAăŠŃPythonèğćéGŁăZlăy■čŽĎăĚüăzŮèfŽčl'NăyĀètuă■ččăŃčŽĎæL'gèaŃiij
éČčăzLă;ăăřséIJĀèĕAăŮzéGŁæT¿ăžüéG■æŮřèŮăRŮăĖÍásĀèğćéGŁăZl'ēTĀiijLGILiijL'ăĀĆ

èğčăEşæŮzæaŁ

ăIJl'CæL'f'ásTäzččăAăy■iijŃGILăRfäzèéĀŽèfGăIJlăzččăAăy■æRŠăĚëăyNéÍcèfŽæăüčŽĎăŃRæĬééGŁæ

```
#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;
}
```

èŃlèőž

ăRlæIJL'ă;Šă;ăčăŃăĬăŃăšăæIJL'Python C APIăG;æTřăIJl'Căy■æL'gèaŃčŽĎæŮăăĀŽă;ăæL'■èČ;ăŃŃăĬăčŽ
GILēIJĀèĕAăèćnéGŁæT¿čŽĎăyŷèğAčŽĎăIJžæŽřæŸřăIJl'èŃăčŃŮăřĖéŽĖăđNăzččăAăy■éIJĀèĕAăIJl'CæTřčžĎ
æĬŮèĀĖăŸřèĕAăL'gèaŃéŸzăăđčŽĎl/OăŠ■ăIJæŮüiijLăřTăĕCăIJlăyĀăyĭæŮGăžŮăRŘèĕřčĕăyĽérzăRŮă

ă;ŠGILèćnéGŁæT¿ăRŮiijŃăĚüăzŮPythončžĕčl'NăL'■èćăăĖăŃŃăăIJl'èğćéGŁăZlăy■æL'gèaŃăĀĆ
Py_END_ALLOW_THREADSăŃŮăiijŽéŸzăăđăL'gèaŃčŽřăĬăřĕřČŷĬčžĕčl'NéG■æŮřèŮăRŮăžĖGILăĀĆ

17.8 15.8 CăŠŃPythonăy■čŽĎčžĕčl'NăèŮŃčTl'

éŮóécŸ

ä;ăæIJL'ăyĀăyĬčl'NăžRēIJĀèĕAăŮăăRĬă;ĕčTl'CăĀPythonăŠŃčžĕčl'NăiijŃ
æIJL'ăžŽčžĕčl'NăŸřăIJl'Căy■ăĬZăžžčŽĎiijŃëŮĖăĠžăžĖPythonèğćéGŁăZl'ēčŽĎăŮăĠŮèŃčăZřăĀĆ
ăžŮăyTăyĀăžŽčžĕčl'NèfŸă;ĕčTl'ăžĖPython C APIăy■čŽĎăG;æTřăĀĆ

èğčăEşæŮzæaŁ

ăĕČăđIJă;ăæČšăřĖCăĀPythonăŠŃčžĕčl'NăŮăăRĬăIJlăyĀètuüiijŃă;ăéIJĀèĕAăčăŃăĬă■ččăŃčŽĎăĬăăŃŃ
èĕAăČšèfŽæăüăăĀŽiijŃăRfäzèăřĖăyNăĬŮăžččăAăT¿ăĬăřă;ăčŽĎčăzččăAăy■ăžŮăčăŃăĬăŃčăZă;Tčžĕčl'Nă

```
#include <Python.h>
...
if (!PyEval_ThreadsInitialized()) {
    PyEval_InitThreads();
}
...
```

árzāžŌāzzā;TērČčTíPythonáržēsāēLŰPython C APIçŽDCāzččāAīijŇčāōāĪā;āēēŰāĒLāuščzRæ■ččāōā
ēĚZāRřāzēčTí PyGILState_Ensure() āšŇ PyGILState_Release()
æĪēāAžĀLřīijŇāēČāyŇāL'Āčd'žīijŽ

```
...
/* 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);
...
```

ærRæñæřČčTí PyGILState_Ensure() éČ;ēēAçŽyāžTčŽDērČčTí
PyGILState_Release() .

èŏlēŏž

āIJāūL'āRĹāLřCāšŇPythonçŽDénŸčžğčĪŇāžRāy■īijŇā;Ĺād'ŽāžŇāēČĒāyĀēŭāAžæŸřā;ĹāyŷēgAçŽD
ārřēČ;æŸřāržCāĀPythonāĀACžžĚčĪŇāĀPythonçžĚčĪŇçŽDæūūāRĹā;ĚčTíāĀČ
ārĪēēAā;āçāōāĪēğčēĠLāŽĪēčŇā■čçāōçŽDāĪlāgŇāŇŮīijŇāžūāyŤæūL'āRĹāLřēğčēĠLāŽĪçŽDCāzččāAæL'g

ēēAæşĹæDRçŽDæŸřērČčTí PyGILState_Ensure()
āžūāy■āijŽčŇŇāLzæĹcā■æĹŰāy■æŰ■ēğčēĠLāŽĪāĀČ āēČæđIJæIJL'āĒūāzŰāzččāAæ■čāIJāL'gēāŇīijŇēĚZ
āIJlāĒēČĪīijŇēğčēĠLāŽĪāijZæL'gēāŇāŚĹæIJšæĀğçŽDçžĚčĪŇāĠGæ■čīijŇāZāæ■d'āēČæđIJāĒūāzŰçžĚčĪŇā
ērČčTíēĀĒæIJĀçžĹĚĚŸæŸřārřāzēēĚRēāŇçŽDīijLār;çōāārřēČ;ēēAāĒĹç■L'āyĀāijŽīijL'āĀČ

17.9 15.9 çTíSWIGāŇĒēčĒCāzččāA

éŰŏéčŸ

ā;āæČšēŏŹā;āāĒZçŽDCāzččāAā;IJāyžāyĀāyĪCæL'āśŤāĪāĪŰāĪēŏŏēŰŏīijŇāČšēĀŽēĠGā;ĚčTí
SwigāŇĒēčĒçTšæĹRāŽĪ æĪēāŏŇāĒRāĀČ

ēğčĀĒşæŰzæāĹ

SwigēĀŽēĠGēğčæđRČād't æŰGāzūāžūēĠLāĹāLZāžzæL'āśŤāzččāAæĪēæŞ■ā;IJāĀČ
ēēAā;ĚčTíāŏČīijŇā;āāĒĹēēAæIJL'āyĀāyĪCād't æŰGāzūāĀČā;ŇāēČīijŇāĹSāzŇčd'žā;ŇçŽDād't æŰGāzūāē

```

/* 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);

```

äyÄæÛëjãæIJL'äZëfZäyİad't'æÜĞäzũrijNäyNäyÄæ■ëârşæYrçijŪâEŻäyÄäyİSwigâÄİæÖëâRcâÄİæŪ
æŊLçĖğçzëâŏZrijNëfZäzZæÜĞäzũäzëâÄİ.İâÄİâRÖçijÄâzũäyTçşzäijjâyNéİcëfZæäũrijŽ

```

// sample.i - Swig interface
%module sample
%{
#include "sample.h"
%}

/* Customizations */
%extend Point {
    /* Constructor for Point objects */
    Point(double x, double y) {
        Point *p = (Point *) malloc(sizeof(Point));
        p->x = x;
        p->y = y;
        return p;
    };
};

/* Map int *remainder as an output argument */
#include typemaps.i
%apply int *OUTPUT { int * remainder };

/* Map the argument pattern (double *a, int n) to arrays */
%typemap(in) (double *a, int n) (Py_buffer view) {
    view.obj = NULL;
    if (PyObject_GetBuffer($input, &view, PyBUF_ANY_CONTIGUOUS |
↳PyBUF_FORMAT) == -1) {
        SWIG_fail;
    }
    if (strcmp(view.format,"d") != 0) {
        PyErr_SetString(PyExc_TypeError, "Expected an array of doubles
↳");
        SWIG_fail;
    }
}

```

(continues on next page)

(continued from previous page)

```
$1 = (double *) view.buf;
$2 = view.len / sizeof(double);
}

%typemap(freearg) (double *a, int n) {
    if (view$argnum.obj) {
        PyBuffer_Release(&view$argnum);
    }
}

/* C declarations to be included in the extension module */

extern int gcd(int, int);
extern int in_mandel(double x0, double y0, int n);
extern int divide(int a, int b, int *remainder);
extern double avg(double *a, int n);

typedef struct Point {
    double x,y;
} Point;

extern double distance(Point *p1, Point *p2);
```

äyÄæÛëä;äâEÛäë;äzEæÖëâRçæÛGäzûiijNärsâRfäzëâIJläS;äzd'ëäNäüëâËüäy■ërCçTlSwigäzEiijŽ

```
bash % swig -python -py3 sample.i
bash %
```

swigçŽDèçŠâGžâršæYřäyd'äylæÛGäzûiijNsample_wrap.câŠNsample.pyãÄĆ
âRÖéİççŽDæÛGäzûâršæYřçTlæLüéIJÄëçAârîjâËëçŽDãÄĆ èÄNsam-
ple_wrap.cæÛGäzûæYřéIJÄëçAëçñçijÛërSâlRâR■âRñ _sample
çŽDæTræNÄæläâiÛçŽDCäzççäAãÄĆ èfZäyIâRfäzëéÄŽèfGèù\$æZóéÄŽæLl'âsTæläâiUäyÄæäüçŽDæLÄæ
äçNäçCiiijNä;äâLZäzžäzEäyÄäylæÇäyNæL'Äçd'žçŽD setup.py æÛGäzûiijŽ

```
# 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)

```
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 ‘SWIG_InitializeModule’:
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 %
```

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

SwigæYřPythonǎŎĖǎRšäy■æđĐǎžžæL'ǎsŦǎlǎǎlŮçŽĐǎIJǎǎRđ'eǎĀçŽĐǎũeǎĖũǎžNǎyǎǎǎĀĀ
SwigèĈjèĠǎĹǎĹǎNŮǎĴǎĹǎđ'ŽǎNĖèçĖĈŦšǎĹRǎŽlçŽĐǎđ'DçRĖǎǎĀĀ

589

èƒAècñà;ŠàAžæYřè;ŠàGžāĀijāĀĆ èƒZäyĭāōđēZĒäyĿæYřäyĀäyĭæĭāāijRāNžēĒēğDāĹZāĀĆ
 āĪĬæŌēäyNāēĭçŽDæĹĀæĪĬĹāčræYŌäy■ĭijNāzzā;TæŪūāĀZāRĭèçAççrāyĿ int
 *remainder ĭijNāzŪārsāijZècñā;ĪJäyžè;ŠàGžāĀĆ èƒZäyĭēĜāōŽāzĹæŪzæşTāRřāzèèŏĭ
 divide() āĜ;æTřèŧTāZđāyđ'äyĭāĀijāĀĆ

```

>>> sample.divide(42, 8)
[5, 2]
>>>
  
```

æĪJĀāRŌäyĀäyĭæŪĹāRĹāĹĭ %typemap æNĜāzd'çŽDèĜĭāōŽāzĹāRřèČ;æYřèƒZéĜNāsTçd'žçŽDæĪJĀ
 äyĀäyĭtypemapārşæYřäyĀäyĭāĪĬē;ŠāĒēäy■çĹzāōŽāRĬæTřæĭāāijRçŽDèğDāĹZāĀĆ
 āĪĬæĪNēĹCäy■ĭijNäyĀäyĭtypemapècñāōŽāzĹäyžāNžēĒē■āRĬæTřæĭāāijR (double *a,
 int n) . āĪĬtypemapāĒĒēĬĹæYřäyĀäyĭCāzççāAçĹĜæŏĭĭijNāŏČāSĹèŕĹSwigæĀŌæāūārĒäyĀäyĭPythonār
 æĪNēĹCāzççāAā;ĭçTĭāžĒPythonçŽDçijŠā■Yā■RèŏŏāŌzāNžēĒē■āzzā;TçĪJNäyĹāŌzçşzāijijāRŊçş;āžææTřçz
 ĭijĹæŕTāçCNumPyæTřçzDāĀarrayæĭāĭŪāĹZāzçŽDæTřçzDç■ĹĭijĹĭijNæŽt'ād'ŽèŕŭāRĬèĀĆ15.3ārRèĹC

āĪĬtypemapāzççāAāĒĒēĬĭijN\$1āŠN\$2èƒZæāūçŽDāRŸéĜRæZĤæ■çāijZèŌŭāRŪtypemapæĭāāijRçŽDÇ
 ĭijĹæŕTāçC\$1æYāārDäyž double *a ĭijĹāĀĆ\$inputæNĜāRŠäyĀäyĭā;ĪJäyžè;ŠāĒēçŽD
 PyObject * āRĬæTřĭijN èĀN \$argnum āŕşāzççēāĹāRĬæTřçŽDäyĭæTřāĀĆ

çijŪāĒZāŠNçRĒēğçtypemapsæYřä;ĭçTĭSwigæĪJĀāşžæĪNçŽDāĹ■æRĹāĀĆ
 äy■āžĒæYřèŕt'āzççāAæŽt'çèđçğYĭijNēĀNäyTā;æĪJĀèçAçRĒēğçPython C
 APIāŠNŠwigāŠNāŏČāzd'āžŠçŽDæŪzāijRāĀĆ SwigæŪĜæaçæĪĬ æŽt'ād'ŽèƒZæŪzēĭççŽDçzĒēĬCĭijNāRřāz

äy■èƒĜĭijNāçCæđĪJā;āæĪĬĹād'ġéĜRçŽDCāzççāAēĪJĀèçAècñæŽt'ēĪJšäyžæĹĹ'āsTæĭāāĭŪāĀĆ
 SwigæYřäyĀäyĭēĭāyŷāijžād'ğçŽDāūēāĒūāĀĆāĒşēTŏçCzāĪĬāžŌSwigæYřäyĀäyĭād'DçRĒCāçræYŌçŽDçij
 éĀZèƒĜāijžād'ğçŽDæĭāāijRāNžēĒē■āŠNēĜĭāōŽāzĹçzDāzŭĭijNāRřāzèèŏĭ'ā;āæŽt'æTžāçræYŌæNĜāōŽāŠNç;
 æŽt'ād'ŽāƒæAŕèŕŭāŌzæşèēYĒ Swigç;ŠçñŽ ĭijN èƒYæĪĬĹ
 çĹzāōŽāžŌPythonçŽDçZyāĒşæŪĜæaç

17.10 15.10 çTĭCythonāNĒèçĒCāzççāA

éŪŏécŸ

ā;āæČşā;ĭçTĭCythonæĭēāĹZāzžäyĀäyĭPythonæĹĹ'āsTæĭāāĭŪĭijNçTĭæĭēāNĒèçĒæşRäyĭāūşā■YāĪĬçŽD

èğçĀĒşæŪzæāĹ

ā;ĭçTĭCythonæđDāzžäyĀäyĭæĹĹ'āsTæĭāāĭŪçĪJNäyĹāŌzā;ĹæĹNāĒZæĹĹ'āsTæĪĬĹāžZçşzāijijĭijN
 āZāäyžā;æĪJĀèçAāĹZāzžā;Ĺād'ZāNĒèçĒēĀĜ;æTřāĀCäy■èƒĜĭijNēŭşāĹ■ēĭçäy■āRŊçŽDæYřĭijNā;āäy■ēĪJĀ

ā;ĪJäyžāĜĒād'ĜĭijNāĀĜēŏ;æĪNçñāāzNçz■ēĬāĹĒççŽDçd'zā;NāzççāAāūşçzRècñçijŪèŕSāĹĹæşRäyĭāŕ
 libsample çŽDCāĜ;æTřāžŞäy■āžĒāĀĆ èçŪāĒĹāĹZāzžäyĀäyĭāR■āŕñ csample.pxd
 çŽDæŪĜāzŭĭijNāçCäyNæĹĀçd'zĭijŽ

```

# 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 *)

```

```

    æŹZäylæŨĞäzũâĬĬCythonäy■çŽDä;ĬçŦĬârſeũşCçŽDâd't æŨĞäzũäyĂæăũăĂĆ
    âĬĬăĜŇăçræŸŎ cdef extern from "sample.h" æŇĞăôŽăžEæĹĂă■çŽĐCâd't æŨĞäzũăĂĆ
    æŎŸäyŇăĭççŽĐăçræŸŎŸĈ;æŸræĭççĜăžŎŸĈçăyĭâd't æŨĞäzũăĂĆæŨĞäzũăŔ■æŸr
    csample.pxd ĩijŇëĂŇăy■æŸr sample.pxd âĤĤăĤŦeŦŽçĈzâĭĹéĜ■èçAăĂĆ

    äyŇăyĂæ■ëĭijŇăĹŽăzzäyĂäyĭâŔ■äyž sample.pyx çŽĐéŨŏécŸăĂĆ
    èrëæŨĞäzũăijŽăôŽăzĹ'ăŇĖèçĖăŽĭijŇçŦĬæĭçæçæŎŸPythonèğççĖĜĹăŽĭăĹŸ csample.
    pxd äy■ăçræŸŎçŽĐCăzççăĂăĂĆ

```

```

# 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

```


(continued from previous page)

```
sz = a.size
with nogil:
    result = csample.avg(<double *> &a[0], sz)
return result

# Destructor for cleaning up Point objects
cdef del_Point(object obj):
    pt = <csample.Point *> PyCapsule_GetPointer(obj, "Point")
    free(<void *> pt)

# Create a Point object and return as a capsule
def Point(double x, double y):
    cdef csample.Point *p
    p = <csample.Point *> malloc(sizeof(csample.Point))
    if p == NULL:
        raise MemoryError("No memory to make a Point")
    p.x = x
    p.y = y
    return PyCapsule_New(<void *>p, "Point", <PyCapsule_Destructor>
↳ del_Point)

def distance(p1, p2):
    pt1 = <csample.Point *> PyCapsule_GetPointer(p1, "Point")
    pt2 = <csample.Point *> PyCapsule_GetPointer(p2, "Point")
    return csample.distance(pt1, pt2)
```

èrëæŮĜäzúæŽt'äd'ŽčŽĎčzEèŁCéĆlálEäijŽăIJlèólēóžéČlálEèřęçzEąśŤăijĂăĂĆ
æIJĂăŔŎijŇăyžăžEăđĎăžžæL'l'ăsŤălăăIŮijŇăČŔăyŇéİćèĚăăăăLŽăžžăyĂăył setup.
py æŮĜäzŭijŽ

```
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
)
```

èëAăđĎăžžæŁŚăžŇăŤŇërŤčŽĎčŽŏăăĜălăăIŮijŇăČŔăyŇéİćèĚăăăăĂžijŽ

```
bash % python3 setup.py build_ext --inplace
```

(continues on next page)

(continued from previous page)

```
running build_ext
cythoning sample.pyx to sample.c
building 'sample' extension
gcc -fno-strict-aliasing -DNDEBUG -g -fwrapv -O3 -Wall -Wstrict-
    prototypes
    -I/usr/local/include/python3.3m -c sample.c
    -o build/temp.macosx-10.6-x86_64-3.3/sample.o
gcc -bundle -undefined dynamic_lookup build/temp.macosx-10.6-x86_64-
    3.3/sample.o
    -L. -lsample -o sample.so
bash %
```

æCædIjæyÄäLĜeazāL'čŽDèlīijNā;āāzTèrēæIJL'āzEäyÄäyLæL'āśTælaāIŮ sample.
so iijNāRāIjIjāyNéIcā;Nā■Räv■ä;ŁçTlīijŽ

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

èóìèőž

æIJñèŁĆăŃĖăŘnăžEă; ĹLăd'ŽăL'■éİcăL'ĂëöšçŽĐénỲçgğçL'zăĂgġijNăŃĖăNňăTřczDăŞ■ă; IJăĂAăăNĚ
æfRăyĂeĆlăLEeĆ; aiJZėĂRăylećněöšēfRăĹrġijNă; EăYřăĹSăznăIJAăē; eĆ; ad' ■ăzăäyĂăyNăL'■éİcăGăârRėŁ
ăIJléquăśCġijNă; ŁçTłCythonăYřăşżăžŌCăzNăyŁăĂĆ.pxđăŪĞăzūăzĔăzĔăRłăNĚăRńCăôŻăzL'iġŁçşzăiġij.h
.pyxăŪĞăzūăNĚăRnăžEăôđçŎriJŁçşzăiġij.căŪĞăzūriJL'ăĂĆcimport
ēr■ăRėècnCythonçTłăĕărġijaĔē.pxdăŪĞăzūăy■çŽĐăőŽăzL'ăĂĆ
ăőÇeuşă; ŁçTłăŻőéĂŽçŽĐăŁăē; PythonăġăġIŬçŽĐărijaĔēēr■ăRėăYřăy■ăRŇçŽĐăĂĆ

ǎřꞥcǫǎ.pxd æŮĜǎzũǎŃĚǎŔnǎžĚǎǫŽǎzL'ijŃǎ;ĚǎǫČǎznǎžũǎy■æŸřĬlǎĭèèĜlǎLlǎLŽǎžžæL'ǎsTǎžččǎAç

āZāā■d'riiNā;āēfYāYrēAāEZāNĒēcĒāG;āTṛāĀCā;NāēCiiNāřsčŮ csample.pxd
 æŮGāzūāčræYŌāzE int gcd(int, int) āG;āTṛiiN ā;āāz■čDúéIĀēēAāIĪ sample.
 pyx āy■āyžāōCāEZāyĀāyġāNĒēcĒāG;āTṛāĀCā;NāēCiiZ

```

import csample

def gcd(unsigned int x, unsigned int y):
    return csample.gcd(x, y)
  
```

ārzāžŌčŏAā■TčZDāG;āTṛiiNā;āāzūāy■éIĀēēAāŌzāAžāđ'ġāđ'ŽčZDæŮūāĀC
 CythonāijŽčTšæLŔāNĒēcĒāzččāAālēē■ččāōčZDē;ñā■cāRĀCāTṛāŠNēfTāZđāĀijāĀC
 čžŠāŏZāLŔāśđāĀgāyŁčZDĀCāTṛā■ŏčšzādNāYŕāRŕéĀLčZDāĀCāy■ēfGiiNāēCāđIĀ;āāNĒāRnāzEāŏCāzn
 ā;NāēCiiNāēCāđIĀIĪ'āžžā;ŁčTġēř šāTṛālēēřČčTġēfZāyġāG;āTṛiiNāijZāLZāGžāyĀāyġāijČāyriijZ

```

>>> 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
>>>
  
```

āēCāđIĀ;āāČšāŕzāNĒēcĒāG;āTṛāAžāRēāđ'ŮčZDæčĀæšēiiNāŔġéIĀēēAā;ŁčTġāRēāđ'ŮčZDāNĒēcĒ

```

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)
  
```

āIĪcsample.pxdæŮGāzūāy■čZD'‘in_mandel()‘‘ āčræYŌæIĪLāyġā;LāIĪL'ēūcā;EāYŕāŕTē;ČēZ;čRĒēg
 āIĪlēfZāyġāēŮGāzūāy■iiNāG;āTṛēcñāčræYŌāyžčDūāRŌāyĀāyġbintēĀNāy■æYŕāyĀāyġintāĀC
 āŏČāijZēŏl' āG;āTṛāLZāžžāyĀāyġā■ččāōčZDBooleanāĀijēĀNāy■æYŕčŏAā■TčZDæT' æTṛāĀC
 āZāā■d'riiNēfTāZđāĀijŌēāŁčđ'žFalseēĀNġēāŁčđ'žTrueāĀC

āIĪCythonāNĒēcĒāZġāy■iiNā;āāRŕāžēēĀL'æNġ'āčræYŌCāTṛā■ŏčšzādNiiNāzšāRŕāžēā;ŁčTġāL'ĀēIĪ
 ārzāžŌ divide() čZDāNĒēcĒāZġāšTčđ'žāžEēfZāāūāyĀāyġā;Nā■RriiNāŔNāŮūēfYāIĪL'āēCā;TāŌzād'Ď

```

def divide(x, y):
    cdef int rem
    quot = csample.divide(x, y, &rem)
    return quot, rem
  
```

āIĪlēfZēGNriiNrem āRŸēGRēcñāY;čđ'žčZDāčræYŌāyžāyĀāyġCāT'āđNāRŸēGRāĀC
 ā;ŠāŏČēcñāijāāēē divide() āG;āTṛčZDæŮūāĀZriiN&rem
 āLZāžžāyĀāyġēūšCāyĀāūčZDæNġāRŠāŏČčZDæNġēŠLāĀC avg()
 āG;āTṛčZDāžččāAāijTčđ'žāžEČythonāZt'ēnYčžgčZDčL'žāĀgāĀC
 ēēŮāĒL def avg(double[:] a) āčræYŌāzE avg()
 æŌēāRŮāyĀāyġāyĀčzt'čZDāRŦčš;āžēāEĒā■YēgEāZ;āĀC æIĪĀēČLāēGčZDēČġāLĒēāYŕēfTāZđčZDčžšāđ

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

āIĴlāēd'āNĒēcĒāZlāyīijNā.size0 āŠN &a[0] āLEāLāiijTçTlāTřçzDāĒČçt'āāylāTřāŠNāzTāsCāN
 ēr■āšT <double *> &a[0] æTŽā;āæĀŌæūārEæNĠēŠLē;ñæ■cāyžāy■āRŇçŽDçšzādNāĀC
 āL■āRŔæYřCāy■çŽD avg() æŌēāRŪāyĀāylāē■ççāōçšzādNçŽDæNĠēŠLāĀC
 āRČēĀCāyNāyĀēLČāĒšāžŌCythonāĒĒā■YēgEāZ;çŽDæZt'énYçžgēōšēfřāĀC

éZd'āzEād'DçŘEēĀZāyçŽDæTřçzDād'ŪiijNavg() çŽDēfZāylā;Nā■RēfYāsTçd'žāzEāēCā;Tād'DçŘ
 ēr■āRē with nogil: āčřæYŌāzEāyĀāylāy■ēIJĀēēĀGILāřsēC;æL'gēāNçŽDāzççāĀāIŪāĀC
 āIJlēfZāylāIŪāy■iijNāy■ēC;æIJL'āzzā;TçŽDæŽōēĀZPythonāržēsāāĀTāĀTāRlēC;ā;fçTlēcñāčřæYŌāyž
 cdef çŽDāržēsāāŠNāG;æTřāĀC āRēād'ŪiijNād'ŪēČlāG;æTřāfEēāzçŌřāōdçŽDāčřæYŌāōCāzñēC;āy■ā;lē
 āZāēd'iijNāIJlcsample.pxdæŪGāzūāy■iijNavg() ècñāčřæYŌāyž double avg(double
 *, int) nogil.

āržPointçzŠædDā;ŠçŽDād'DçŘEæYřāyĀāylāēNŠæLYāĀCæIJñēLCā;fçTlēcŪāZLāržēsāārEPointāržēsā
 èēAēfZæūāĀZçŽDēfIiijNāzTāsČCythonāzççāAçl■ā;ōēIJL'çCzād'■āIČāĀC
 ēēŪāĒLijNāyNēIēçŽDārijaĒēēcñçTlāēāijTāĒēCāG;æTřāzŠāŠNPython C
 APIāy■āōZāzL'çŽDāG;æTřiijZ

```
from cpython.pycapsule cimport *
from libc.stdlib cimport malloc, free
```

āG;æTřdel_Point() āŠN Point() ā;fçTlēcZāylāLšēC;ælēāLZāzzāyĀāylēČūāZLāržēsāiijN
 āōČāijZāNĒēcĒāyĀāyl Point * æNĠēŠLāĀC cdef del_Point() āRē del_Point()
 āčřæYŌāyžāyĀāylāG;æTřiijN āRlēC;ēĀZēfG CythonēōlēŪōiijNēĀNāy■ēC;āzŌPythonāy■ēōlēŪōāĀC
 āZāēd'iijNēfZāylāG;æTřāržād'ŪēČlāYřāy■āRfēgAçŽDāĀTāĀTāōČēcñçTlālēā;ŠāZāyĀāylāZdēfCāG;æ
 āG;æTřērČçTlārTāēC PyCapsule_New() āĀPyCapsule_GetPointer()
 çŽt'æŌēālēēGĤPython C APIāzūāyTāzēāRŇæāūçŽDæŪzāijRēcñā;fçTlāĀC

distance āG;æTřāzŌ Point() āLZāzžçŽDēČūāZLāržēsāy■āRŔāRŪæNĠēŠLāĀC
 ēfZēGŇēēAæšlāDRçŽDæYřā;āāy■ēIJĀēēAæNĒāfČāijCāyād'DçŘEāĀC
 āēČādIJāyĀāylēTŽēfçŽDāržēsāēcñāijāēfZālēiijNPyCapsule_GetPointer()
 āijZæLZāGžāyĀāylāijCāyīijN ā;EæYřCythonāūšçzRçšēēAšæĀŌāzLāšēæL;āLřāōČiijNāzūārEāōČāzŌ
 distance() āijāēĀŠāGžāŌzāĀC

ād'DçŘEPointçzŠædDā;ŠāyĀāylçijzçCzæYřāōČçŽDāōdçŌræYřāy■āRfēgAçŽDāĀC
 ā;āāy■ēC;ēōlēŪōāzzā;TāsđæĀgālēæšēçIJNāōČçŽDāĒēČlāĀC
 ēfZēGŇæIJL'āRēād'ŪāyĀçg■æŪzæšTāŌzāNĒēcĒāōČiijNāršæYřāōZāzLāyĀāylāL'tāsTçšzādNijNāēCāyN

```
# sample.pyx
```

(continues on next page)

(continued from previous page)

```
cimport csample
from libc.stdlib cimport malloc, free
...

cdef class Point:
    cdef csample.Point *_c_point
    def __cinit__(self, double x, double y):
        self._c_point = <csample.Point *> malloc(sizeof(csample.
↪Point))
        self._c_point.x = x
        self._c_point.y = y

    def __dealloc__(self):
        free(self._c_point)

    property x:
        def __get__(self):
            return self._c_point.x
        def __set__(self, value):
            self._c_point.x = value

    property y:
        def __get__(self):
            return self._c_point.y
        def __set__(self, value):
            self._c_point.y = value

def distance(Point p1, Point p2):
    return csample.distance(p1._c_point, p2._c_point)
```

```

aIJlEfZeGñijNcdifçsz      Point      ārEPointācrrēYŌäyžäyÄäyļæL'l'āsTçszādNāĀĆ
çszāsdaeĀg cdef csample.Point *_c_point ācrrēYŌāzEäyÄäyļaođä;NāRŸēGRrijN
æNēæIJL'äyÄäyļæNĠāRŠāzTāsĆPointçzŞædDä;ŞçZDæNĠēŚLāĀĆ
__cinit__()                  aŠN          __dealloc__()                  æŪzaesTēĀZēfG
malloc()                     aŠN          free()                          āLZāzzāzūēTāæfAāzTāsĆCçzŞædDä;ŞāĀĆ
xāŠNyāsdaeĀgçZDācrrēYŌeoŀ'ä;æeŌuāRŪāŠNēoŀ;ç;ōāzTāsĆçzŞædDä;ŞçZDāsdaeĀgāAijāĀĆ
distance()                   çZDāNĒēcEāZlEfYāRfrazēēcnaŀfoæTzrijNā;ŀā;ŪāoCēĈ;æŌēāRŪ      Point
æL'l'āsTçszādNāođä;Nā;IJäyžāRCæTrijN ēāNāijāēĀŠāzTāsCæNĠēŚLçzZCāG;æTŗāĀĆ

```

āAžZæEēfZäyŁæTřzāRŸāRŎñjNä;āäiĴZāRŚçŎřæŠ■ā;IJPointāřzēšāāřsæŸ;ā;ŮāŽř'āŁæèGłçDúāžEñjŽ

```
>>> 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
>>>

```

æIJñèŁĆăũşçzŔæijŦçd'žăžEăĹŁăd'ŹCythonçŹĐăăyăŧČçŁ'zæĂgġijŦă;ăăŔfăzëăžăæ■d'ăyžăşžăĠEăĬăæ
 äy■ēŧĠijŦă;ăæIJĂăē;ăĤĹăŬŌzéŸĖērăyŦăôŸăŮzăŮŮăăçăĬăăžĖēğçăŹŦ'ăd'ŹăŧăăŔăŦăĂĆ
 æŬŌăyŦăĬăăĠăăĤăēŧŸăijŹçzğçz■æijŦçd'žăyĂăžŹCythonçŹĐăĤăăžăŮçŁ'zæĂgăĂĆ

17.11 15.11 çŦĬCythonăEŹénŸăĂğèČĭçŹĐăŦŕçzĐăŞ■ăĬJ

éŬŌéćŸ

ăĭăăēĀăEŹénŸăĂğèČĭçŹĐăŞ■ăĬJăĬăēĠNumPyăžŦçşçzŹĐăŦŕçzĐăŮăçŮŮăĠăŦŕăĂĆ
 äĭăăũşçzŔçşēēĂşăžĖCythonēŧŹăăũçŹĐăăüăăĤăăijŹēŮŦ'ăôČăŦŸăă;ŮçŮăă■ŦġijŦă;ĖăŸŦăžăüăy■çăŮăăŮŹēŕăăĂ

èğçăEşăŮzăăĹ

ăĬJăyžăyĂăyĹă;Ŧă■ŦġijŦăyŦēĬçŹĐăžčçăĂæijŦçd'žăžEăyĂăyĹCythonăĠăŦŦġijŦçŦĬăĬăŧăŮăŦŦ'ăyĂă

```

# 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]

```

ěAçijŮerŠaŠNædĎázžèŁŻäylæL'l'ásTijjNä;æIJĀæAäyÄäylăČRäyNélcèŁŻæũçŽĎ
 setup.py æŮĞäzŭ ijLä;ŁçTl python3 setup.py build_ext --inplace
 ælæædĎázžăŏČijL'ijŽ

```

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
)

```

äjääijŽaRŠçŮŖçzŞædIJăĜ;æTřçaŏăŏďărzæTřçzĎèŁŻèaŇçŽĎăŁŏæ■cijNăzŭäyTăRřäzèéĂĆçTlăžŮăd'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;äðfYäijŽāRŠçÖrèfRèaŃçTšæLŔçzŠædIJēIdäyçŽDāfñāĀĆ
 äyNēlĆæLŠāznārEæIJñä;ŃāŠŃumpyäy■çŽDāũšā■YāIJlçŽD clip()
 āĠ;æŦrāAžāyÄäyĭæĀgèĈ;āržærŦiijŽ

```
>>> 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■čæĈä;äçIJŃāLŔçŽDiiJŃāōĈèeAāfñā;Ĺād'ŽāĀŦāĀŦèfZæYŕäyÄäyĭā;ĹæIJL'èüççŽDçzŠædIJiijŃāžā

èóĭèőž

æIJñèLĆāLŦçŦlāžEçCythonçszādŃçŽDāEĒā■YēgEāZ;iiJŃædAād'gçŽDçōĀāNŪāžEæŦŕçzDçŽDæŠ■ā;I
 cpdef clip() āčŕæYŌāžE clip() āŦŦæUūāyžCçžgāĹnāĠ;æŦŕāžēāŦŦPythonçžgāĹnāĠ;æŦŕāĀĆ
 āIJlCythonāy■iijNēfZāyĭæYŕā;ĹéĠ■èeAçŽDiiJŃāžāyžāōĈēāĹçd'žæ■d'āĠ;æŦŕèŕĈçŦĭlèeAæŕŦāĒūāžŪCytho
 iijLærŦæĈā;āæĈsāIJlāŦēād'ŪāyÄäyĭāy■āŦŦçŽDCythonāĠ;æŦŕāy■èŕĈçŦlclip()iijLāĀĆ

çszādŃāŦŦæŦŦ double[:] a āŦŦ double[:] out
 āčŕæYŌèfZāžZāŦŦæŦŕāyžāyĀçzt'çŽDāŦŦçs;āžæŦŕçzDāĀĆ
 ā;IJāyžē;ŠāĒēiijŃāōĈāznāijŽèōfēŪōāžzā;ŦāōđçŌŕāžEāEĒā■YēgEāZ;æŌēāŦŦçŽDæŦŕçzDāržēsaiijNēfZāyĭ
 3118æIJL'èŕççzEāōŽāžL'āĀĆ āŦŦæŦŦāžEŦNumPyāy■çŽDæŦŕçzDāŦŦāEĒç;ōçŽDarrayāžŠāĀĆ

ā;Šā;äçijŪāEŽçŦšæLŦŦçzŠædIJāyžæŦŕçzDçŽDāžççāAæŪiijŃā;āāžŦèŕēeAŦā;ĭāyĹēlĆçd'žā;ŦéĈçæūē
 āōĈāijŽārEāLZāžžē;ŠāĠžæŦŕçzDçŽDèŦ'čāžžçzŽèŦçŦĭlèĀēiijŃāy■ēIJĀēeAçšēeAŠā;āæŠ■ā;IJçŽDæŦŕçzDç
 iijLāōĈāžEāžEāAĠēō;æŦŕçzDāũšçzŦāĠEāđ'Ġāē;āžEiijŃāŦŦlēIJĀēeAāAžāyĀāžZārŦçŽDæĈĀæšærŦāēĈçā
 āIJlāĈŦNumPyāžŦçszçŽDāžŠāy■iijŃā;fçŦŦ numpy.zeros() æĹŪ numpy.
 zeros_like() āĹZāžžē;ŠāĠžæŦŕçzDçŽDāŦŦæŦŦēĀæŦŦē;ĈāōžæYŠāĀĆĀŦēād'ŪiijNēeAāĹZāžžæIJlāL
 ā;āāŦŦāžēā;fçŦŦ numpy.empty() æĹŪ numpy.empty_like() .
 āēĈādIJā;āæĈšēeEçŽŪæŦŕçzDāEĒāōžā;IJāyžçzŠædIJçŽDèŦlēĀL'æŦŦ'èfZāyđ'āyĭāijžærŦē;ĈāfñçĈzāĀĆ

āIJlā;äçŽDāĠ;æŦŦāōđçŌŕāy■iijŃā;āāŦŦlēIJĀēeAçōĀā■ŦçŽDēĀžēfĠāyŦæāĠēfŦŦçōŪāŦŦæŦŕçzDæšēæ
 CythonāijžèŦ'sèŦ'čāyžā;äçŦšæLŦŦēŦYæŦŦçŽDāžççāAāĀĆ

clip() āōžāžL'āžŦāL'■çŽDāyđ'āyĭēēēēŦāŽlāŦŦāžēāijYāNŪāyŦæĀgèĈ;āĀĆ
 @cython.boundscheck(False) çIJĀāŌžāžEæL'ĀæIJL'çŽDæŦŕçzDèŦŦçŦŦŦæĈĀæšēiijŦ
 ā;Šā;äçšēeAŠāyŦæāĠēōfēŪōāy■āijžēŦŦçŦŦçŽDæŦŦāĀžāŦŦāžēā;fçŦŦlāōĈāĀĆ
 @cython.wraparound(False) æŦŦŦēZd'āžEçŽyāržæŦŕçzDāržēĈlçŽDèŦ'sæŦŦāyŦæāĠçŽDād'ĐçŦŦiijŦ
 āijŦāĒēēfZāyđ'āyĭēēēēŦāŽlāŦŦāžēāđAād'gçŽDæŦŦŦā■ĠæĀgèĈ;iiJLætŦŦèŦŦēfZāyĭā;Ŧā■ŦçŽDæŦŦāĀžād'

āžžā;ŦæŦŦāĀžād'ĐçŦŦæŦŦçŽDæŦŦiijŦçāŦçŦŦūāžæŦŦāŦŦāžēāđAād'gçŽ
 ā;ŦāēĈiijŦēĀĈēŽŠārž clip() āĠ;æŦŦçŽDæĈāyŦāfōæ■çiiJŦā;fçŦŦlāāžūēāĹē;āijŦiijŽ

```
@cython.boundscheck(False)
@cython.wraparound(False)
cpdef clip(double[:] a, double min, double max, double[:] out):
    (continues on next page)
```


(continued from previous page)

```
if min > max:
    raise ValueError("min must be <= max")
if a.shape[0] != out.shape[0]:
    raise ValueError("input and output arrays must be the same_
↪size")
for i in range(a.shape[0]):
    out[i] = (a[i] if a[i] < max else max) if a[i] > min else_
↪min
```

ãððéZĖætNërTçzŞædIJæYřijNëfZäylçL'ŁæIJñçZĎäzççăAèfRëaÑéĂşăžèèçAăñ50%ăžăyŁrijL2.44ç
timeit() æTjNërTçZĎ3.76çğŠijL'ăĂĆ

ăŁrëfZëGŃäyza■cuijNă;ăăRrëČ;æČşçşëéAŞëfZçğ■ăžççăAæĂŌăzLëČ;èŭşæL'ŃăEŻCër■élĂPKăŚcuijş
ăĵŃăçCuijNă;ăăRrëČ;ăEŻăžEăçCăyNçZĎCăG;æTřăžŭă;ĤçTlăL'■élăăGăèLČçZĎăŁĂæIJrăelăæL'ŃăEŻæL'P

```
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);
    }
}
```

ăĹŚăžnăşşæIJL'ăşTçd'žèfZäylçZĎăL'P'ăşTăžççăArijNă;EæYřerTçtNăžŃăRŌrijNăĹŚăžnăŔŚçŌřăyĂă
æIJĂăžTăyNçZĎăyĂëaNærTă;ăæČşëşăçZĎëfRëaNçZĎăñăĴăd'ZăĂĆ

ăĵăăRřăžăŕzăôđăĴNăžççăAæđĎăžzăđ'ZăylæL'P'ăşTăĂĆăŕzăžŌăşŔăžZæTřçzĎăŞ■ă;IJrijNăIJĂăç;èèA
èèAèfZăăŭăĂZçZĎërIrijNëIJĂëçAăĴŏăTăžççăArijNă;ĤçTl with nogil: èŕ■ăRërijZ

```
@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
```

ăçCădIJă;ăæČşăEŻăyĂăyłæŞ■ă;IJăžNçzt'æTřçzĎçZĎçL'ŁæIJñrijNăyNéİcăYřăRřăžăŔCëĂČăyŃrijZ

```
@cython.boundscheck(False)
@cython.wraparound(False)
cpdef clip2d(double[:, :] a, double min, double max, double[:, :]_
↪out):
    if min > max:
```

(continues on next page)

(continued from previous page)

```
raise ValueError("min must be <= max")
for n in range(a.ndim):
    if a.shape[n] != out.shape[n]:
        raise TypeError("a and out have different shapes")
for i in range(a.shape[0]):
    for j in range(a.shape[1]):
        if a[i, j] < min:
            out[i, j] = min
        elif a[i, j] > max:
            out[i, j] = max
        else:
            out[i, j] = a[i, j]
```

äyÑæIJZerzèÄËäy■èeAâfYäzEæIJñèLCæL'ÄæIJL'äzççäAéÇ;äy■äijZçzSåóZåLræ§Räy!çL'zåóZæTřçzD
èfZæäüäzççäAâræZt'æIJL'çAæt'zæÄgäÄÇ äy■èfGüijÑèeAæslæDRçZDæYräeCædIJäd'DçREæTřçzDèeAæ
èfZäzZäEËEäózáuščzRéüEäGzæIJñèLCèNÇäZt'ijjÑæZt'äd'ZäfaæAferüaRCèÄÇ PEP
3118 üijÑ äRÑæÜü CythonæÜGæaçäy■äËşäzÖäÄIJçszädNäEËä■YègEäZ;âÄI
çrGäzşäÄijä;ÜäyÄerzäÄÇ

17.12 15.12 äŖEäG;æTřæNGéŠLè;ñæ■cäyžâRrèrČçTlâržèsa

éÜöécY

ä;äâüščzRéÖüä;ÜäzEäyÄäy!ècñcijÜerSâG;æTřçZDäEËä■YäIJräIÄrijÑæCşârEäóCè;ñæ■cæL'RäyÄäy!F
èfZæäüçZDèrlä;äârşâRräzèârEäóCä;IJäyžäyÄäy!æL'äsTâG;æTřä;ççTlâzEäÄÇ

èğcâEşæÜzæaĹ

ctypes ælaäIÜâRfècñçTlæIèaLZäzžâNÈècEäzzæDRäEËä■YäIJräIÄçZDPythonâRrèrČçTlâržèsaÄÇ
äyÑéIççZDä;Nä■RæijTçd'zäzEæÄÖæäüèÖüâRÜCâG;æTřçZDäÖşägNäÄAäzTâsCâIJräIÄrijÑäzèâRŁæçCä;

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

èõléõž

èeAædDåzzäyÄäyläRfèrÇçTlärzèsajNä;æeUäÉLÉIJÄeAäLZäzzäyÄäyl
CFUNCTYPE äõä;NäÄÇ CFUNCTYPE() çŽDçñnäyÄäyläRCæTæYfèTäZdçszädnäÄÇ
æÕäyNäIèçŽDäRCæTæYfäRCæTæYfäszädnäÄÇäyÄæUä;ääöZäZL'äZÈäG;æTæYfäszädnäijNä;äärseÇ;ärEäöÇ
çTšæLRçŽDärfzèsæcñä;ŠäAZæZóéÄZçŽDäRféÄZèfG ctypes
èöféUöçŽDäG;æTæYfäIä;fçTlärÄÇ

æIJnèLCçIJNäyLäÖZäRfèÇ;æIJL'çCzçèdçgYijNäAÄRäZTäsCäyÄçCzäÄÇ
ä;EäYfijNä;EäYfäöCècñäZfæšZä;fçTlärZÖäRDçg■énYçžgäzççäAçTšæLRæLÄæIJfærTäeCä■sæUüçijÜerS

ä;NäeCijNäyNéIcæYfäyÄäylä;fçTl 11vmpy æL'l'äsTçŽDçóÄä■Tä;Nä■RijNçTlärIædDåzzäyÄäyläR
äZüärEäEüè;nä■cäyZäyÄäyPythonäRfèrÇçTlärzèsäÄÄÇ

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

äZüäy■æYfèr'äIJlèfZäyIäsCéIççLräZÈäZä;TéTŽèrrärsäijZärijeGtPythonègçcèGLäZlæNÇæÖL'äÄÇ
èeAèöä;ÜçŽDæYfä;æYfäIJçZt'æÕèèušæIJZäZlçžgäLncŽDäEä■YäIJräIÄäŠNæIJnäIJæIJZäZlççäAæL'Šä

17.13 15.13 äijäéÄŠNULLčzŠärčžŽDā■ŮčņēäyščzžŽCāĢ;æTŗāžŠ

éŮóécŸ

äjäëeAâEŽäyÄäyŁæL'ŕāsTæÍaāIŮiijNěIJÄëeAäijäéÄŠäyÄäyI NULLčzŠärčžŽDā■ŮčņēäyščzžŽCāĢ;æTŗāžŠ
äy■èfĢiijNä;äy■æŸřā;ŁčāōāōŽæÄŌæāüā;ččTÍPythončŽDUnicodeā■ŮčņēäyšāŌžāōččŌřāōČāĀĆ

èġčāEşæŮzæaĹ

èöyād'ŽCāĢ;æTŗāžŠāNĚāRñäyÄžŽæS■ä;IJNULLčzŠärčžŽDā■ŮčņēäyšiiijNěcñāčřæŸŌčšzādNäyž
char *.èÄČèŽŠāčCäyNčžŽDCāĢ;æTŗiijNæŁSāžņčTlæIěāAžæijTčd'žāŠNætNèrTčTlčžŽDiiijŽ

```
void print_chars(char *s) {  
    while (*s) {  
        printf("%2x ", (unsigned char) *s);  
  
        s++;  
    }  
    printf("\n");  
}
```

æ■d'āĢ;æTŗaiijŽæL'Sā■rēcñäijäèfŽæIěā■ŮčņēäyščžŽDæřRäyŁā■ŮčņēčžŽDā■AāĚ■èfŽāŁüēāčd'žiiijNèfŽ

```
print_chars("Hello");    // Outputs: 48 65 6c 6c 6f
```

āržāžŌāIJÍPythonäy■èrČčTlèfŽæāüčžŽDCāĢ;æTŗiijNä;äæIJL'āĢčğ■éĀL'æNl'āĀĆ
éčŮāĚĹiijNä;āāRřāžééÄžèfĢèrČčTl
PyArg_ParseTuple()
āžūæNĢāōžŽāĀIyāĀIJè;ñæ■ččāAāIěéŽŘāŁüāōČāRlèČ;æS■ä;IJā■ŮèŁČiijNæCäyNiiijŽ

```
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;  
}
```

čzŠædIJāĢ;æTŗčžŽDā;ččTlæŮzæšTæCäyNāĀĆžTčzEèġČāršā;NāĚčāžENULLā■ŮèŁČčžŽDā■Ůčņēäyš

```
>>> print_chars(b'Hello World')  
48 65 6c 6c 6f 20 57 6f 72 6c 64  
>>> print_chars(b'Hello\x00World')  
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
TypeError: must be bytes without null bytes, not bytes  
>>> print_chars('Hello World')
```

(continues on next page)

(continued from previous page)

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'str' does not support the buffer interface
>>>
```

```
PyObject *
PyObject *PyArg_ParseTuple(
char *s;
```

```
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;
}
```

```
PyObject *
PyObject *PyArg_ParseTuple(
char *s;
```

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

```
PyObject *
PyObject *PyArg_ParseTuple(
char *s;
```

```
/* 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);
}

```

āL■ēīcāyđ'çğ■ē;ñā■céĈ;āRřāzēçāōā£īāYřNULLçz\$āř;çŽDæTřæ■ōīijŇ
 ä;EæYřāōĈāzñāžūāy■ācĀæšēā■Ůçņēāyšāy■ēŮt'æYřāRēāŧŇāĒēāzĒNULLā■ŮēĽĈāĀĆ
 āŽāæ■d'īijŇāēĈāđIJē£Žāyġā;ĽēĜ■ēēAçŽDēřīijŇēĈčā;āēIJāēēAēĜġāūsāŌzāAŽācĀæšēāzĒāĀĆ

ēōīēōž

āēĈāđIJāRřēĈ;çŽDēřīijŇā;āāžTēřēēA£āĒ■āŌzāēZāyĀāzŽā;ĪēŧŮāžŌNULLçz\$āř;çŽDā■ŮçņēāyšīijŇ
 æIJāāē;çz\$āRĽā;£çŧĪāyĀāyġāēŇĜēŚĽāSŇēŧ£āžēāĀijāēĪēād'ĎçRĒā■ŮçņēāyšāĀĆ
 āy■ē£ĜīijŇāēIJĽ'æŮūāĀŽā;āā£ĒēēāzāŌzād'ĎçRĒĈēr■ēĪāēAŮçŧŽāzčçāAæŮūāřsæšāā;ŮēĀĽ'æŇĪ'āžĒāĀĆ

āř;çōāā;ĽāōžæYšā;£çŧĪīijŇā;EæYřā;ĽāōžæYšā;ēğEçŽDāyĀāyġēŮōēćYæYřāIJĪ
 PyArg_ParseTuple() āy■ā;£çŧĪāĀIJšāĀĪāēījāijRāŇŮçāĀāijŽæIJĽ'āēĒā■Yæ■šēĀŮāĀĆ
 ā;Eā;āēIJāēēAā;£çŧĪē£çğ■ē;ñā■ćçŽDæŮūāĀŽīijŇāyĀāyġUTF-
 8ā■ŮçņēāyšēćnāĽāžžāžūāēryāžĒēŽDāĽāāIJġāŌšāğŇā■ŮçņēāyšāřzēsāyĽēīcāĀĆ
 āēĈāđIJāŌšāğŇā■ŮçņēāyšāŇĒāRņēĪđASCIIā■ŮçņēçŽDēřīijŇāřsāijŽāřijēĜr'ā■ŮçņēāyšçŽDāřžāryācđāĽrāy

```

>>> 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
>>>

```

āēĈāđIJā;āāIJġāžŌē£ŽāyġāēĒā■YçŽDæ■šēĀŮīijŇā;āæIJāāē;ēĜ■āēZā;āçŽDCæĽĪ'āsŧāzčçāĀīijŇēōĪ'ā
 PyUnicode_AsUTF8String() āĜ;æTřāĀĆāēĈāyŇīijŽ

```
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;
}
```

éĀŽèĤĞèĤŽäyĤäĤōæŤzĭjŇäyĀäyĤUTF-8çijŮčăAçŽĎăŮçñęäyşæăżæőéIJĀèĕAèĕnáĤZăżzĭjŇçDŭăŔĈ

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

ăĕĈăđIJă;ăērŤçĬĀăĭjăĕĂŞNULLçzŞăŕĭăŮçñęäyşçzŽctypesăŇĒèĕĒèĤĞçŽĎăĜjăŤŕĭĭjŇ
èĕAęşĬăĐŔçŽĎăŸŕctypesăŔĤèĈjăĕĒAèőyăĭjăĕĂŞăŮèĤĈĭjŇăzŭăyŤăőĈăyăăĭjŽăĕĂăşęăyăĕŮŕăŤŇăĒĕçŽĎă

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

ăĕĈăđIJă;ăăĈşăĭjăĕĂŞăŮçñęäyşĕĂŇäyăŸŕăŮèĤĈĭjŇăjăĕIJĀèĕAăĒĤăĤĝëăŇăĤŇăĤĤçŽĎUTF-
8çijŮčăAăĀĈăjŇăĕĈĭjŽ

```
>>> print_chars('Hello World'.encode('utf-8'))
48 65 6c 6c 6f 20 57 6f 72 6c 64
>>>
```

ărzăžŌăĒŭăzŮăĤŕăşŤăŭĕăĒŭĭjĤăŕŤăĕĈSwigăĂĂCythonĭjĤŕĭjŇ

āīĴā;āā;£çŦlāōČāznāijāēĀŠā■ŮčņēäyščzžČāzččāAæŮűēēAāĒĹāē;āē;ā■ēāzāçŽyāžŦçžĐäyĪēēāzĒāĀĆ

17.14 15.14 äijäéĀŠUnicodeā■ŮčņēäyščzžČāĜ;æŦrāžŠ

éŮőécŸ

ä;äēēAāĒŽäyĀäyĴæĴ'āśŦāĴāāĪŮiijŅēĪĀēēAāŦĒäyĀäyĴPythonā■ŮčņēäyšāijāēĀŠçzžČçžĐæšŦäyĴāžŠ

èğčāĒşæŮzæāĴ

èĒŽéĜŅāĴŠāžñēĪĀēēAēĀČēŽŠāĴĴād'ŽçžĐēŮőécŸiijŅā;ĒæŸŦæĪĀäyžēēAçžĐēŮőécŸæŸŦçŦŦā■Ÿç
āŽāæ■d'iijŅā;āçžĐæŅŠāĴŸæŸŦāŦĒPythonā■Ůčņēäyšē;ñæ■čäyžäyĀäyĴēČ;ēēŦČçŦĒēğççžĐā;čāijŦāĀĆ

äyžāžĒæijŦçd'žçžĐçžōçžĐriijŅäyŅēĴæĪĴäy d'äyĴČāĜ;æŦriijŅçŦĴāĴēæš■ā;Īā■ŮčņēäyšæŦŦæ■ōāžűē
äyĀäyĴā;£çŦlā;čāijŦäyž char *, int ā;čāijŦçžĐā■ŮēĴČiijŅ
ēĀŅāŦēäyĀäyĴā;£çŦlā;čāijŦäyž wchar_t *, int çžĐāō;ā■Ůčņēā;čāijŦiijž

```
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");
}
```

āržāžŦēĴāŦŦā■ŮēĴČçžĐāĜ;æŦŦprint_chars() iijŅā;āēĪĀēēAāŦĒPythonā■Ůčņēäyšē;ñæ■čäyžā.
8. äyŅēĴæŸŦäyĀäyĴēĒæāüçžĐæĴ'āśŦāĜ;æŦŦŦä;Ŧā■Ŧiijž

```
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;
}
```

ärzäžŎéĆčžŽēIJĀēēAāđ'ĐčŘĚæIJžāŽlæIJñāIJř wchar_t
 çšzādŇčŽĎžšSāG;æTřijŇä;ääRřāžēāČŘāyŇélcēfZæăüçijŮāĚZæL'āśTāžččāAřijŽ

```
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;
}
```

äyŇélcæYřāyĀäyłāžd'āžŠāijŽēřlælēæijTčd'žēfZāyłāG;æTřæYřāēCā;Tāüēä;IJčŽĎřijŽ

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

äžTčzĚēgČāršēfZāyłēlcāŘŠā■ŮēŁčŽĎāG;æTř print_chars()
 æYřæĀŎæăüæŎēāRŮUTF-8çijŮčāAæTřæ■ŏčŽĎřijŇ äžēāŘŁ print_wchars()
 æYřæĀŎæăüæŎēāRŮUnicodeçijŮčāAāĀijçŽĎ

èóìèőž

āIJłçžgçž■æIJñēŁČāžŇāL■řijŇä;ääžTēřēēŮāĚŁā■ēāžāä;æēŏēŮŏçŽĎCāG;æTřāžšçŽĎçL'žā;AāĀĆ
 ärzäžŎā;Łād'ŽCāG;æTřāžšřijŇēĀžāyāijāēĀŠā■ŮēŁČēĀŇāy■æYřā■ŮçñēäyšāijZærTē;Čāē;āžZāĀĆēēAēf

```
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;
}
```

āēĆādIJā;ääž■čĎūēfYæYřæČšēēAāijāēĀŠā■ŮçñēäyšřijŇ
 ä;āēIJĀēēAçšēēAšPython 3āRřā;fçTlāyĀäyłāŘŁēĀĆçŽĎā■Ůçñēäyšēāłčd'žřijŇ

ǎŏČázűäy■çŽt' æŎēæ ŸăăŕĎăĹŕă;£çŦlæăĜăĜEçśzăđN char * æĹŪ
 wchar_t * iijĹæŽt'ăđ'ŽçzEēĹĈăŔĈcèĂĈPEP 393iijĹ'çŽĎCăĜ;æŦŕăžŦăĂĈ
 ăŽăă■đ' iijŊēæAăIJĹCăy■ēăĹçđ' žēĹŽăyĹă■ŪçņęăyŝæŦŕæ■ŏiijŊăyĂăžŽē;ŋæ■cēĹŸæŸŕăĹĒēăžēēAçŽĎăĂĈ
 ăIJĹ PyArg_ParseTuple() äy■ă;£çŦlăĂİs#ăĂİ ăŦŊăĂİu#ăĂİăiijăijŔăŊŪçăAăŔŕăžēăŏĹ'ăĒĹçŽĎăĹ'ğēă
 äy■ēĹĜēĹŽçğ■ē;ŋæ■cæIJĹ'äyĹçijžçĈzăŕŝæŸŕăŏĈăŔŕēĈ;ăijŽăŕijēĜt'ăŎŝăĝŊă■ŪçņęăyŝăŕžēŝăçŽĎăŕžăŕy
 äyĂæŪēē;ŋæ■cēĹĜăŔŎiijŊăijŽæIJĹ'äyĂăyĹē;ŋæ■cæŦŕæ■ŏçŽĎăđ'■ăĹŭéŽĎăĹăăĹŕăŎŝăĝŊă■Ūçņęăyŝăŕžēŝă
 ä;ăăŔŕăžēēĜĈăŕŝăyŊēĹŽçğ■æŦĹăđIJiijŽ

```

>>> 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
>>>
  
```

ărzăžŎărŦéĜŔçŽĎă■ŪçņęăyŝăŕžēŝăiijŊăŔŕēĈ;æŝăăžĂăžĹă;ŝăŦiijŊ
 ä;EæŸŕăēĈăđIJă;ăēIJĂēēAăIJĹăĹĹ'ăŦăŦăy■ăđ'ĎçŔĒăđ'ğēĜŔçŽĎăŪĜăĹŊiijŊă;ăăŔŕēĈ;æĈŝéAăăăē■ēĹŽăyĹ
 äyŊēĹcæŸŕăyĂăyĹăĹŏēŏçĈĹĹăIJăăŔŕăžēēAăăăē■ēĹŽçğ■ăĒĹă■Ÿæ■ŝēĂŪiijŽ

```

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;
}
  
```

ēĂŊăŕž wchar_t çŽĎăđ'ĎçŔĒæŪŭæĈŝēēAēăăăē■ăĒĹă■Ÿæ■ŝēĂŪăŕŝæŽt'ăĹăēŽ;ăĹđăžĒăĂĈ
 ăIJĹăĒĒēĈiijŊPythonă;£çŦlăIJăēŋŸæŦĹçŽĎăĹçđ'žăĹēă■ŸăĈĹă■ŪçņęăyŝăĂĈ
 ä;ŊăēĈiijŊăŔŕăŊăŊăŔŋASCIIçŽĎă■Ūçņęăyŝēcŋă■ŸăĈĹăyžă■ŪēĹĈăŦŕçzĎiijŊ
 ēĂŊăŊăŔŋēŊĈăŽt'ăžŎŪ+0000ăĹŕŪ+FFFFçŽĎă■ŪçņęçŽĎă■Ūçņęăyŝă;£çŦlăŔŊă■ŪēĹĈăĹçđ'žăĂĈ
 çŦŝăžŎărzăžŎăŦŕæ■ŏçŽĎăĹçđ'žă;ăăijŔăy■æŸŕă■ŦăyĂçŽĎiijŊă;ăăy■ēĈ;ăŕĒăĒēĈĹăŦŕçzĎē;ŋæ■căyž
 wchar_t * çĎŭăŔŎăIJŝæIJžăŏĈēĈ;æ■cçăŏçŽĎăŭēă;IJăĂĈ ä;ăăžŦēŕăăĹŽăžžăyĂăyĹ
 wchar_t æŦŕçzĎăžűăŔŦăĒŭăy■ăđ'■ăĹŭăŪĜăĹŊăĂĈ PyArg_ParseTuple()
 çŽĎăĂİu#ăĂİăiijăijŔçăAăŔŕăžēăyŏăĹĹ'ă;ăēŋŸæŦĹçŽĎăŏŊăĹŔăŏĈiijĹăŏĈăŕĒăđ'■ăĹŭçzŝăđIJēŽĎăĹăăĹ

æĊædIJä;äæĈséAġäĖ■ēTġæŮüēŮr'āĖĖä■Ÿæ■şēÄŮiijŃä;ääŤräyĀçŽĎēĀL'æŃl'ārsæŸřäd'■āLŭUnicode
 āŖĖāōĈäijäēĀŞçžŽCāĠ;æŤřiiŃŃçĎŭāŖŌāŽdæŤŭēŁŻäyŁæŤřçžĎçŽĎāĖĖä■ŸāĀCäyŃéİcæŸřäyĀäyŁāŖēĈçž

```
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;
}
```

āIJlēŁŻäyŁāōđçŌřäy■iijŃPyUnicode_AsWideCharString()
 āLŽāžžäyĀäyŁäyŤæŮüçŽĎwchar_ŧçijŞāĖşāžŭād'■āLŭāŤŤräæ■ōēŁŽāŌžāĀĆ
 ēŁŻäyŁçijŞāĖşēēñäijäēĀŞçžŽCçĎŭāŖŌēēñéĠŁæŤĴæŌL'āĀĆ äĴæŸřæŁŚāĖŽēŁŽæIJñäžççŽĎæŮŭāĀŽiijŃē

æĊædIJä;äçşēēAŞCāĠ;æŤřāžŞéIJĀēēAçŽĎā■ŮēŁĈçijŮçāAāžŭäy■æŸřUTF-8iijŃ
 ä;ääŖřäžēäijžāLŭPythonä;ŁçŤlæLl'āsŤçāAāĪēæL'ġēāŃæ■ĈçāōçŽĎē;ñæ■ćiijŃārsāĈŖäyŃéİcēŁŽæäŭiijŽ

```
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;
}
```

æIJĀāŖŌiijŃäæĊædIJä;äæĈşçŽŤ æŌēād'ĎçŖĖUnicodeā■ŮçñēäyşiiŃŃäyŃéİcçŽĎæŸřä;Ńā■ŖiijŃæijŤç

```
static PyObject *py_print_wchars(PyObject *self, PyObject *args) {
    PyObject *obj;
    int n, len;
    int kind;
    void *data;

    if (!PyArg_ParseTuple(args, "U", &obj)) {
        return NULL;
    }
    if (PyUnicode_READY(obj) < 0) {
        return NULL;
    }
}
```

(continues on next page)

(continued from previous page)

```
len = PyUnicode_GET_LENGTH(obj);
kind = PyUnicode_KIND(obj);
data = PyUnicode_DATA(obj);

for (n = 0; n < len; n++) {
    Py_UCS4 ch = PyUnicode_READ(kind, data, n);
    printf("%x ", ch);
}

printf("\n");
Py_RETURN_NONE;
}
```

[illegible]

ěĚŸæIJL'æIJAãRÕãGããRëijŽã;ŞãžÕPythonäijäéĂŞUnicodeã■ŮçņäÿşczŽCçŽĎæŮuãĂŽiijNã;ääžTĕrĕ
 äĉĈæđIJæIJLUTF-8ãŠNãŏ;ã■Ůçņäÿđ'çġéĂL'æNĬ'iijNĕrŭéĂL'æNĬ'UTF-8. ģŗŹUTF-
 8çŽĎæTĕræNĂæŽt'ãĽãæŽŏéA■äÿĂäžŽiijNãžşäÿ■ãŏžæŸŞçĽrĕTŽiijNĕğĉĉĖĽãŽiãžşĕĈ;æTĕræNĂçŽĎæŽt'äĉ
 æIJAãRÕiijNçãŏãĽiã;ääžTĉçEĕŸĖĕrżãŽĖ äĖşãžÕãđ'ĎĉRĖUnicodeçŽĎçŽÿãĖŞşãŮĜæäĉ

17.15 15.15 Că■Ůčņęäŷšë;ñæ■căŷžPythonă■Ůčņęäŷš

éŮóécŸ

æAŒæăăărĚCäy■cŽDă■Üçñäyşè;ñæ■cävžPythonă■UèŁĆæLŰăYĂäyİă■Üçñäyşärzèsajijş

èġċăẸsæŮzæąŁ

Cā■Ūçņäyšä;ŁçŦlāyĀārĶ	char *	āšŅ	int	æļēēāłc'd'zījŅ
ä;āēIJĀēēAāEšāōZā■ŪçņäyšāŁrāžŦæYŗçŦlāyĀäyĭāŌšāģNā■ŪēŁCā■ŪçņäyšēŁYāYŗāyĀäyŦUnicodeā■Ūç				
ā■ŪēŁCārĶzēšāāŦrāžēāČŦāyŦēīcēŁZāūūä;ŁçŦlĶ	Py_BuildValue()	æļēēāłDāzzījŹ		

```
char *s;      /* Pointer to C string data */
int  len;     /* Length of data */

/* Make a bytes object */
PyObject *obj = Py_BuildValue("y#", s, len);
```

æĈæđIJä;æèAålZázžäYÄäyUnicodeå■ŬçņęăÿšiiJÑáúăYŤaj;ăçşéeAŞ
æŃĞăRŠsäzEUTF-8cijŮčAcŽĐæTræ■öiijNâRfrazëä;ŁcTlăyNéİccŽĐæÚzaijRüijŻ

```
PyObject *obj = Py_BuildValue("s#", s, len);
```

æĈæđIJ s ä;ŁćŦlăĔŭăzŮćijŮćăAæŮzăijRiijŇéĈcăzŦlăRřăzěăĈRăyŇéİcă;ŁćŦl
PyUnicode_Decode() æĬæđĐăžžăyĂăylă■ŮĉņăyşiiŹ

```
PyObject *obj = PyUnicode_Decode(s, len, "encoding", "errors");
```

```
/* Examples */
```

```
obj = PyUnicode_Decode(s, len, "latin-1", "strict");
```

```
obj = PyUnicode_Decode(s, len, "ascii", "ignore");
```

æĈæđIJăăæAřăē;æIJLăyĂăylćŦl wchar_t *, len řřzēăłćđ'žćŽĐăő;ă■ŮĉņăyşiiŹ
æIJLăĜăĉğ■ćĀLæŇŦæĀğăĀĈćēŮăĔLă;ăăRřăzěă;ŁćŦl Py_BuildValue() iijŹ

```
wchar_t *w; /* Wide character string */
```

```
int len; /* Length */
```

```
PyObject *obj = Py_BuildValue("u#", w, len);
```

ăRēăđ'ŮiijŇă;ăēŁŸăRřăzěă;ŁćŦl PyUnicode_FromWideChar() :

```
PyObject *obj = PyUnicode_FromWideChar(w, len);
```

ăřžăžŎăő;ă■ŮĉņăyşiiŹŇăžŭăşşăæIJLăřžă■ŮĉņăTřă■őēŁŹăŇēĝćăđŘăĀŦăĀŦăőĈćcŇăĀĜăőŽăŸřăŎă

ěőĬěőž

ăřĚCăy■ćŽĐă■Ůĉņăyşē;Ňă■ćăyžPythonă■ŮĉņăyşēAŦă;ŦăŠŇŦ/OăRŇăăŭćŽĐăŎşăĹŽăĀĈ
ăžşăřşăŸřēŦ'iiŇăĬēĜŦCăy■ćŽĐăTřă■őăĬĔēăzăăžăæ■őăyĂăžŹēĝćăĀăŽĬēcŇăŸ;ăijRćŽĐēĝćăĀăyžăyĂă
éĂŽăyŷćijŮćăAæăijăijRăŇĔăŇŇASCIIăĀĂLatin-1ăŠŇUTF-8.
æĈæđIJăăăžŭăy■ćăőăőŽćijŮćăAæŮzăijRăĹŮēĂĔăTřă■őăŸřăžŇēŁŹăĹŮćŽĐiiŇă;ăăIJĂăē;ăřĔă■Ůĉņăyş
ă;ŞăđĐēĂăăyĂăylăřžēşăćŽĐăŮŭăĂŽiiŇPythonéĂŽăyŷăijŽăđ'■ăĹŭă;ăăRŘă;ŽćŽĐă■ŮĉņăyşăTřă■őăĀĈ
æĈæđIJăIJLăĤĔēēAćŽĐēŦiiŇă;ăēIJăēēAăIJăŦŎēĬăŎžēĜĹăŦ;Că■ŮĉņăyşăĀĈ
ăRŇăŮŭiiŇăyžăžĔēŦ'ćĬŇăžRăŽŦăĹăăAēăőiiŇă;ăăžŦēŦăRŇăŮŭă;ŁćŦlăyĂăylăŇĜēŚĹăŠŇăyĂăylăđ'ğ
ēĂŇăy■ăŸřă;ĬēŦŮNULLćžŞăř;ăTřă■őăĬăĹŽăžžă■ŮĉņăyşăĀĈ

17.16 15.16 äy■ćăőăőŽćijŮćăAæăijăijRćŽĐCă■Ůĉņăyş

éŮőéćŸ

ă;ăēēAăIJĬăŠŇPythonćŽŦ'æŎēăĬăăžđē;Ňă■ćă■ŮĉņăyşiiŇă;ĔăŸŦCăy■ćŽĐćijŮćăAæăijăijRăžŭăy■ćă
ă;ŇăēĈiiŇăRřēĈ;Căy■ćŽĐăTřă■őăIJşăIJŽăŸŦUTF-8iiŇă;ĔăŸŦăžŭăşşăæIJLăijžăĹŭăőĈăĤĔēăzăŸřăĀĈ
ă;ăăĈşćijŮăĔăžăžćăAæĬăăžăyĂăĝğ■ăijŸēŹĔćŽĐăŮzăijRăđ'ĐćŘĔēŁŹăžŽăy■ăŘĹăăijăTřă■őiiŇŇēŁŹăăŭă

èğçàEşæŮzæąŁ

äyÑéÍcæYřäyÄäzŻCçŻĐæŤræ■óăŠŇäyÄäyŁăĜjæŤræİēæijŤçd'zèŁŻäyŁéŮóécYřijŻ

```
/* Some dubious string data (malformed UTF-8) */
const char *sdata = "Spicy Jalape\x3\xbo\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");
}
```

ăİJİēŁŻäyŁäzççăAäy■řijŇă■Ůçņäyš sdata äŇĚăŘnăžĚUTF-
8ăŠŇäy■ăŖŁæăijæŤræ■óăŠŇäy äy■ēŁĜřijŇăçĈæđİçŤİæŁuăİJİCäy■ērĈçŤİ
print_chars(sdata, slen) řijŇăđĈçijžēĈjæ■čäyŷăüēă;IJăĂĈ
çŎřăİJİăAĜēđ;ăjăăĈşărĚ sdata çŻĐăĚĚăđzē;Ňă■čäyžäyÄäyŁPythonă■ŮçņäyšăĂĈ
ēŁŻäyÄæ■ēăAĜēđ;ăjăăİJİăŖŎéÍcèŁYæĈşéĂŻēŁĜäyÄäyŁăL'ăśŤăŕĚēĈçäyŁă■ŮçņäyšăijăäyŁ
print_chars() âĜjæŤrăĂĈ äyÑéÍcæYřäyĂçĝ■çŤİæİēăfİæŁd'ăŎşăĝŇăŤræ■óçŻĐæŮzæşŤřijŇăŕşçđŮă

```
/* Return the C string back to Python */
static PyObject *py_retstr(PyObject *self, PyObject *args) {
    if (!PyArg_ParseTuple(args, "")) {
        return NULL;
    }
    return PyUnicode_Decode(sdata, slen, "utf-8", "surrogateescape");
}

/* Wrapper for the print_chars() function */
static PyObject *py_print_chars(PyObject *self, PyObject *args) {
    PyObject *obj, *bytes;
    char *s = 0;
    Py_ssize_t len;

    if (!PyArg_ParseTuple(args, "U", &obj)) {
        return NULL;
    }

    if ((bytes = PyUnicode_AsEncodedString(obj, "utf-8",
↪ "surrogateescape"))
        == NULL) {
        return NULL;
    }
    PyBytes_AsStringAndSize(bytes, &s, &len);
    print_chars(s, len);
}
```

(continues on next page)

(continued from previous page)

```
Py_DECREF (bytes) ;
Py_RETURN_NONE;
}
```

æĈæđIJä;ääIJPythonäy■ārīērTēfZāZāG;æTrijNāyNéíċæYrēfRēāNæTLæđIJijZ

```
>>> s = retstr()
>>> s
'Spicy JalapeÃso\uudcae'
>>> print_chars(s)
53 70 69 63 79 20 4a 61 6c 61 70 65 c3 b1 6f ae
>>>
```

āzTçzEëġĈāršçzŞæđIJä;ääijZāRŚçŌrijNāy■āRLæāijā■ŪçņēäyšēcñcijŪçāAālRāyÄäyPythonā■Ūçņēäy
āzūāyTā;ŞāōCēcāZđāijāçzZCşZDæUūāĀZrijNēcñē;ñæ■cāyZāŞNāzNāL■āŌşāġŃCā■ŪçņēäyšāyÄæāūçZDā

ëöleöž

æIJñēLĈāsTçd'zāzEāIJāL'āſTæīāāIŪäy■ād'DçRĒā■ŪçņēäyšæŪūāijZēĒ■āLrçZDāyÄäyīæčYæLŊāRī
āzşārşæYrēf'rijNāIJāL'āſTāy■çZDcā■ŪçņēäyşārřēČ;äy■āijZāyēæāijéAţā;IPythonæL'ĀæIJşæIJZçZDUn
āZāæ■d'rijNā;LāRrēČ;äyÄāzZāy■āRLæāijCæTřæ■ōāijāéĀŞāLřPythonäy■āŌzāĀĆ
äyÄäyīā;Lāē;çZDā;Nā■RārşæYræūL'āRLāLrāzTāſCşçzçzşērČçTīæfTāēCæŪĠāzūāR■ēfZæāūçZDā■Ūçņēäy
ā;NāēČrijNāēCæđIJäyÄäyīçşçzçzşērČçTīēfTāZđçzZēġčēĠāZīāyÄäyīæ■şāIRçZDā■ŪçņēäyşrijNāy■ēČ;ēcñ

äyÄēLñæīēēōrijNāRřāzēēĀZēfĠāLūāōZāyÄāzZēTŻērrç■ŪçTēæfTāēCāyēæāijāĀĀāf;çTēāĀĀæZēāzç
äy■ēfĠrijNēfZāzZç■ŪçTēçZDāyÄäyīçijzçCzæYrāōČāznæryāzĒæĀġçāt āIRāzEāŌşāġNā■ŪçņēäyşçZDāĒē
ā;NāēČrijNāēCæđIJä;Nā■Rāy■çZDāy■āRLæāijæTřæ■ōā;ççTīēfZāzZç■ŪçTēāzNāyÄēġççāArijNā;ääijZā;Ū

```
>>> raw = b'Spicy Jalape\xc3\xb1o\xae'
>>> raw.decode('utf-8', 'ignore')
'Spicy JalapeÃso'
>>> raw.decode('utf-8', 'replace')
'Spicy JalapeÃso?'
>>>
```

surrogateescape ēTŻērrād'DçRĒç■ŪçTēāijZārEæL'ĀæIJL'äy■āRřēġççāĀā■ŪēLÇē;ñāNŪāyžāyĀ
ā;NāēČrijZ

```
>>> raw.decode('utf-8', 'surrogateescape')
'Spicy JalapeÃso\uudcae'
>>>
```

ā■TçNñçZDā;Ōā;■āzççRĒā■ŪçņēæfTāēĆ \udcae āIJUni-
codeäy■æYrēīđæşTçZDāĀĆ āZāæ■d'rijNēfZāyīā■ŪçņēäyşārşæYrāyÄäyīēīđæşTēāīçd'zāĀĆ
āōđēZĒäyLūijNāēCæđIJä;āārEāōČāijāäyīāyÄäyīæL'ġēāNē;ŞāĠzçZDāG;æTrijNā;ääijZā;ŪāLrāyÄäyīēTŻērr

```
>>> s = raw.decode('utf-8', 'surrogateescape')
>>> print(s)
```

(continues on next page)

(continued from previous page)

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
UnicodeEncodeError: 'utf-8' codec can't encode character '\udcaE'
in position 14: surrogates not allowed
>>>
```

çĐüēĀŇrijŇĀĔĀēōyāzčċŘĚē;ñæ■ćçŽĎăĔĚŝēŤōċĆzāIJlāžŌāžŌCāijăçžŽPythonăŖĹăŽđäijăçžŽCçŽĎăy■
å;ŞēĒŽăylă■ŬçñēăyşăĔ■æñăä;ĲçŦĬ surrogateescape çijŬçăĀæŬŦrijŇăzčċŘĚă■ŬçñēăijŽē;ñæ■ćăŽđăŌ

```
>>> s
'Spicy JalapeĀso\udcaE'
>>> s.encode('utf-8', 'surrogateescape')
b'Spicy Jalape\xc3\xbl\xae'
>>>
```

ä;IJăyžăyĀēĹŇăĠĔĲĲŲijŇæIJĀăē;éĀăĔĔăžčċŘĚçijŬçăĀăĀŦăĀŦăēĆăđIJă;ăæ■ćçăōçŽĎă;ĲçŦĬăžĚçij
ăy■ēĒĠrijŇæIJĹæŬŦăĀŽçăōăđăijŽăĠžçŌŖă;ăăžŭăy■ēĆ;æŌġăĹŭæŦŖæ■ōçijŬçăĀăžŭăyŦă;ăăŖĹăy■ēĆ;ăĲ;
éĆčăžĹăŖşăŖăžăä;ĲçŦĬăIJñēĹĆçŽĎăĹăĔŖăžĔăĀĆ

æIJĀăŖŌăyĀçĆçēēĀæşĹæĎŖçŽĎăŲŖijŇPythonăy■ēōyăđ'ŽēĬăŖŖŖşçşçzçşçŽĎăĠă;æŦŖijŇçĹzăĹŇæŲŖă
éĆ;ăijŽă;ĲçŦĬăžčċŘĚçijŬçăĀăĀĆă;ŇăēĆŲijŇăēĆăđIJă;ăă;ĲçŦĬăĈŖ os.listdir()
ēĒŽăăŭçŽĎăĠă;æŦŖijŇăijăăĔēăyĀăylăŇĔăŖŇăžĔăy■ăŖŖēġççăĀæŬĠăžŭăŖ■çŽĎçŽŏă;ŦçŽĎŖŖijŇăŏCăijŽ
ăŖĆēĀĆ5.15çŽĎçŽyăĔşçñăēĹĆăĀĆ

PEP 383 äy■æIJĹæŽŦăđ'ŽăĔşăžŌæIJŇæIJæŖŖăĹŖçŽĎăžăŖĹăŖŖŖsurroga-
teescapeēŦŽŖŖăđ'ĎçŘĚçŽyăĔşçŽĎăĲăæĀŖăĀĆ

17.17 15.17 äijăēĀŖæŬĠăžŭăŖ■çžŽCæĹŦăŖŦ

éŬŏéçŲ

ä;ăēIJĀēēĀăŖŖŖCăžŖăĠă;æŦŖăijăēĀŖæŬĠăžŭăŖ■rijŇă;ĔæŲŖēIJĀēēĀçăŏăĬăŬĠăžŭăŖ■ăăžăæ■ōçşççşş

ēġçăĔşşæŬžæăĹ

ăĔŽăyĀăylăŌēăŖŬăyĀăylăŬĠăžŭăŖ■ăyžăŖĆăŦŖçŽĎăĹŦăŖŦăĠăĠă;æŦŖijŇăēĆăyŇēĒŽăăŭrijŽ

```
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)


```
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);
```

äžčāRfçgžæd'■æŨžaijRælēād'DčŘEæŨĠžūāŘ■æŸfäyÄäyIāŁŁæçŸæL'NçŽĐēŨŌēcŸiijNæIJĀāŘŌāžd'
 æĈædIJājäāIJāæL'L'āšTžžčçAäy■ajŁçTīæIJnēŁĈçŽĐæŁĀæIJriijNæŨĠžūāŘ■çŽĐād'DčŘEæŨžaijRāšNāš
 āNĒæNñcijŨčāA/çTŊēIcā■ŨēŁĈiijNād'DčŘEāIRā■ŨçñēiijNāžçčŘEjēnæ■cāšNāēŨūāžŨād'■æIĈæĈēāEſāA

éŮőécŸ

617

èeAårEäyÄäylæŨGäzûe;ñæ■cäyžäyÄäylæTı'ăđNçŽDæŨGäzûæRRèłřçñēiijNä;łçTł
 PyFile_FromFd() iijNăçCăyNiiJŽ

```
PyObject *fobj;          /* File object (already obtained somehow) */
int fd = PyObject_AsFileDescriptor(fobj);
if (fd < 0) {
    return NULL;
}
```

```

çzŞædIJæŨĜäzúæRRèfřçņæYřéĀŽèfĜèrĈçTÍ      fobj      äy■çŽD      fileno()
æŨzæşTèŌuâĬŨçŽDăĀCăZăæ■d'ijNăzză;TăžèèfŽçg■ăŨzâijRăŽt'ēIJSççZăyĂăylæRRèfřăZlçŽDăřzèşæĈ
ăyĂăŨçă;ăæIJLăžEèfZăylæRRèfřăZlĭijNăŏĈăřsèĈ;ècnăijăeĂŞççZăd'Žăylă;ŌçžgçŽDăRăd'DçRĖæŨĜäzú
ăeĈædIJă;ăeIJĂèçAè;ñă■căyĂăylæTĭădNăŨĜäzúæRRèfřçņăyăyăĂăylPythonăřzèşăijNăĀĈçTlăyNă
PyFile_FromFd() :

```

```
int fd;          /* Existing file descriptor (already open) */
PyObject *fobj = PyFile_FromFd(fd, "filename", "r", -1, NULL, NULL, NULL,
    ↪ 1);
```

PyFile_FromFd() çŽĎāŔĆæŦřáržāŦĥĚř;őçŽĎ open() āĠ;æŦřāĀĆ NUL-
Lēālcđ'žcijŪčāĀāĀēŦŽēřāŦŠŅæ■cēāŦāŔĆæŦřā;ŕçŦĹéžŸēōđ'āĀijāĀĆ

æĈædIĴæRPythonäy■ĈŽDæŮĜäzûärzèšäĵijăĉzŻCĳNæIJL'äyÄăžZæšlæĐRăžNéazăĂĆ
 éĕŮăĒĹĳijŃPythonéĂžēĜ i o ælăăĹŮăL'gëăNëĜlăûšĉŽĐI/OĉijŞăĒşăĂĆ
 âIĴăĳăéĂşăžză;ŢĉşşăđŃĉŽDæŮĜäzûăŔŔēŕĉņĉĉzŻCăžNăL'■ĳijNă;ăĉĈ;ĕĖĂĕĕŮăĒĹăIJĴĉŽyăžŢăŮĜäzûăr
 äy■ĈĐŭĉŽĐĕŕĳijNă;ăăĳiŽăL'ŞăžşăŮĜäzŭĉşĉzĉşăyĹéĬĉĉŽDæŢŕă■ăăĂĆ

æĖũæñajijNā;æēIJĀðeAçL'zálŋæşlæĐRæŨĠgāzūçŽĐā;ŠāsdeĀĖāzeāRLāĖŞeŨæŨĠgāzūçŽĐeAŋŋet'čāĀC
 æeCædIJāyĀāylæŨĠgāzūæRRēfřçñeēcnāijāçzŽCijNā;EæŸřāIJĪPythonā■ēŸĀIJĪecnā;ĚçTĪçĪĀijNā;æēIJĀðe
 çşzāijijçŽĐrijNāeCædIJāyĀāylæŨĠgāzūæRRēfřçñeēcnē;ñæ■cāyžāyĀāylPythonæŨĠgāzūārfzeşajijNā;æēIJĀðeç
 PyFile_FromFd() çŽĐeIJĀāRŌāyĀāylāRCæTřecnēō;ç;ōāLRĪijNçTĪlæĪeāNĠGāGžPythonāžTēřeāĖŞeŨ

```

æċCædIJä;æéIÄëeAäzÖCæāGăGEI/OāžŞäy■ä;fçTlæĆăĂĂfdopen()
ăĜ;ætTræIēālZāzzäy■ăRNŇşzādNŇcŽDæŮĞazūārżēsəærTāeĆ      FILE *      árzésajijŃ
ä;äéIÄëeAçL'zālŋārRāfÇāzEāĀĆefŻæuûaAZājJZāIJl/OāāEæāLāy■āzğçTŞayd'äyláoŃāÉlāy■ăRNŇcŽDI/Oç
iijLāyĀäylæYřæIēèGHPythonçŽD      io      ælqāIŮiijŃāRęäyĀäylæIēèGtCçŽD      stdio
iijL'ăĀĆ      āČRCäy■çŽD      fclose()      äijZāEşēŮ■PythonèeÄä;fçTlčŽDæŮĞazūāĀĆ
æċCædIJeđl'ä;æéĀL'çŽDērliijŃä;ääžTerēaijZeĀL'æŃl'ăŌzædĐāzzäyĀäylæL'l'āsTāzçčāAælēād'DçŘEāžTāsĆ
èĀŃäy■æYřä;fçTlæIēèGt<stdio.h>çŽDéŋYāsĆæŁ;ēsaaŁšèČ;ăĀĆ

```

17.19 15.19 äžŒCèr■èlĀäy■èrżàRŪçşşæŪĠäzŭârżèşą

èŬóécŸ

äĵäèèAâEŽCæLl'âśŦæĭèèrżàRŪæĭèèĠäzżäŦPythonçşşæŪĠäzŭârżèşąäy■çŽDæŦŕæ■ōiijĹæŦŦæĆæŽŌ

èğčâEşşæŪzæąĹ

èèAèŕżàRŪäyĀäyĭçşşæŪĠäzŭârżèşąçŽDæŦŕæ■ōiijNăĵăèĬĬAèèAéĠ■ăd'■èŕČçŦĬ
read() æŪzæşŦiijNçDŭâRŌæ■ççăŏçŽDèğççăAèŌŭăĹŪçŽDæŦŕæ■ŏăĂĆ

äyNéĭcæŸŕäyĀäyĭCæLl'âśŦăĠæŦŕăĹNă■RiijNăžĚäžĚăŦĹæŸŕèŕżàRŪäyĀäyĭçşşæŪĠäzŭârżèşąäy■çŽD

```
#define CHUNK_SIZE 8192

/* Consume a "file-like" object and write bytes to stdout */
static PyObject *py_consume_file(PyObject *self, PyObject *args) {
    PyObject *obj;
    PyObject *read_meth;
    PyObject *result = NULL;
    PyObject *read_args;

    if (!PyArg_ParseTuple(args, "O", &obj)) {
        return NULL;
    }

    /* Get the read method of the passed object */
    if ((read_meth = PyObject_GetAttrString(obj, "read")) == NULL) {
        return NULL;
    }

    /* Build the argument list to read() */
    read_args = Py_BuildValue("(i)", CHUNK_SIZE);
    while (1) {
        PyObject *data;
        PyObject *enc_data;
        char *buf;
        Py_ssize_t len;

        /* Call read() */
        if ((data = PyObject_Call(read_meth, read_args, NULL)) == NULL)
            ↪{
                goto final;
            }

        /* Check for EOF */
        if (PySequence_Length(data) == 0) {
            Py_DECREF(data);
            break;
        }
    }
}
```

(continues on next page)

(continued from previous page)

```
}

/* Encode Unicode as Bytes for C */
if ((enc_data=PyUnicode_AsEncodedString(data, "utf-8", "strict
↪")) == NULL) {
    Py_DECREF(data);
    goto final;
}

/* Extract underlying buffer data */
PyBytes_AsStringAndSize(enc_data, &buf, &len);

/* Write to stdout (replace with something more useful) */
write(1, buf, len);

/* Cleanup */
Py_DECREF(enc_data);
Py_DECREF(data);
}
result = Py_BuildValue("");

final:
/* Cleanup */
Py_DECREF(read_meth);
Py_DECREF(read_args);
return result;
}
```

èĕAætĦNèrTefZäyłazçčăAġijŃăĔŁăđĐēĀăăyĂăyłçşzæŮĠăzũărzèsăæŕTăĕCăyĂăyłStringIOăđăġŃġijŃç

```
>>> import io
>>> f = io.StringIO('Hello\nWorld\n')
>>> import sample
>>> sample.consume_file(f)
Hello
World
>>>
```

ëöİëőž

ăŠŃăŽőéĂŽçşçzşæŮĠăzũăy■ăŔŇçŽĐăŸŕġijŃăyĂăyłçşzæŮĠăzũărzèsăăzũăy■éIJăĕĕAăġçTłăġŌçžğ
ăŽăă■đ'ġijŃăġăăy■ĕġăġçTłăŽőéĂŽçŽĐCăžŠăĠăġæŦŕăġēōŁēŮőăőČăĂĆ
ăġăĕIJăĕĕAăġçTłPythonçŽĐC APIăġăăČŔăŽőéĂŽăŮĠăzũçşzăġijġçŽĐēĆĕăăăă\$■ăġIJçşzæŮĠăzũărzèsăă

ăIJăĔŁSăžŋçŽĐēğčăĖşăŮžăăĹăy■ġijŃŕead() æŮžăşŦăžŌĕćăġġăĕĂŠçŽĐărzèsăăy■ăŕŔăŕŮăĠăžăġē
ăyĂăyłăŔĆăŦŕăĹŮăġĕćăăđĐăžžçĐăăŔŌăy■ăŮ■çŽĐĕćăġġăçžŽ PyObject_Call()
ăġēĕŕČçTłēŁZăyłăŮžăşŦăĂĆ èĕAăĕĂăşĕăŮĠăzũăIJăŋăġġijĹEOŦġijŃăġçTłăžĖ
PySequence_Length() æġăşĕçIJăŸŕăŔĕĕŦăŽđărzèsăĕŦăžăyž0.

árzäžŌæL'ÄæIJL'çŽĐI/OæŞ■ä;IJiijŇä;æéIJÄèeAäĖşæşlăžTăşĆçŽĐçijŮčăAæăijăijŔiijŇëŔYæIJL'ă■ŮèŁæIJñèŁĆæijTçd'žăžEăĉCă;TăžèæŮĜæIJñălăăijŔèržăŔŮăyĂăyIæŮĜăžŭăžŭăŔĖçzŞæđIJæŮĜæIJñèĝççăAăyžăĉCăđIJă;ăæCşăžèăžŇëŔŽăĹŭălăăijŔèržăŔŮăŮĜăžŭiijŇăŔlėIJÄèeAăĤŏæTžăyAçCžçCžă■şăŔŕiijŇăĹŇăĉC

```
...
/* 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);
...
```

æIJñèŁĆæIJÄéŽ;çŽĐăIJŕæŮžăIJlăžŌăĉCă;TèŔZèaŇæ■čçăŏçŽĐăĖĖă■YçŏăçŔĖăĂĆă;Şăđ'ĐçŔĖPyObject *ăŔYéĜŔçŽĐæŮŭăĂŽiijŇëIJÄèeAæşlăĎŔçŏăçŔĖăijTçTlėŏăæTŕăžèăŔĹăIJăy■ăŕžPy_DECREF() çŽĐèŕCçTlăŕşæYŕæIėăAŽèŔŽăyIçŽĐăĂĆ

æIJñèŁĆăžççăAăžèăyĂçĝ■éĂŽçTlăŮžăijŔçijŮăĖŽiijŇăŽăæ■d'ăžŮăžşèČ;éĂĆçTlăžŌăĖŭăžŮçŽĐæŮŮăĹŇăĉCŕiijŇëeAăĖŽæTŕæ■ŏiijŇăŔlėIJÄèeAăŖŭăŔŮçşzæŮĜăžŭăŕžèşçŽĐ write()æŮžæşTŕiijŇăŕĖæTŕæ■ŏè;Ňæ■căyžăŔĹéĂĆçŽĐPythonăŕžèşă iijĹă■ŮèŁĆæĹŮUni-codēiijL'iijŇçĐŭăŔŖèŕCçTlėŕēæŮžæşTŕăŕĖē;ŞăĖăĖŽăĖĖăĹŕæŮĜăžŭăĂĆ

æIJăăŔŖŌiijŇăŕ;çŏăçşzæŮĜăžŭăŕžèşæĂŽăyŕēŔYæŔŔă;ŽăĖŭăžŮæŮžæşTŕiijĹăŕTăĉCreadline(), read_info())iijL'iijŇ æĹSăžŇæIJĂăē;ăŔlă;ŔçTlăşşzæIJñçŽĐ read()ăŦŇ write()æŮžæşTăĂĆăIJlăĖŽCæL'ŕăşTçŽĐæŮŭăĂŽiijŇëČ;çŏĂă■Tŕşăŕ;éĜŔçŏĂă■TăĂĆ

17.20 15.20 ad'ĐçŔĖCèrñelĀăy■çŽĐăŔŕèŔ■ăžčăŕžèşă

éŮŏéćY

ă;ăæČşăĖŽCæL'ŕăşTăžççăAăđ'ĐçŔĖæIėèĜlăžžă;TăŔŕèŔ■ăžčăŕžèşăăĉCăĹŮèălăĂăăĖČçzĐăĂăæŮĜă

èĝçăĖşæŮžæăĹ

ăyŇéIćæYŕăyĂăyIçæL'ŕăşTăĜ;æTŕă;Ňă■ŔiijŇăijTçd'žăžĖæĂŖăŭăđ'ĐçŔĖăŔŕèŔ■ăžčăŕžèşăăy■çŽĐă

```

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("");
}

```

ěóíěőž

æIJñèŁĆäy■çŽDžččăAăŠŇPythonäy■árfžăŽTăžččăAçşzäijijăĂĆ
 PyObject_GetIter() çŽDèrČçTíăŠŇèrČçTí iter()
 äyĂăăüăRřèŎăŭŮäyĂäyłëf■ăžčăŽíăĂĆ PyIter_Next() ăĠæTřèrČçTí next
 æŮžæşTřTăŽďäyŇäyĂäyłăĚČť'ăăĹŮNULL(ăĚČăđIJăşăæIJĹăĚČť'ăăžĚ)ăĂĆ
 èĚAăşłăĎRă■čçăŏçŽĎăĚĚă■ŸçŏăçŘĚăĂŤăĂŤ Py_DECREF()
 éIJĂèĚAăRŇăŮŭăIJłăžğçTřçŽĎăĚČť'ăăŠŇèf■ăžčăŽíărfžèşăæIJñèžŇäyŁăRŇăŮŭèćŇèrČçTíijŇ
 äžééAłăĚăăĠžçŎřăĚĚă■ŸăşĎéIJşăĂĆ

17.21 15.21 èřŁæŮ■ăĹĚæŏťéŤŽèrr

éŮŏécŸ

èğćéĠŁăŽíăŽăäyžăşŘăyłăĹĚæŏťéŤŽèrrăĂAăĂžçžĚéŤŽèrrăĂAăèŏĚéŮŏèŭŁçTŇăĹŮăĚŭăžŮèĠt'ăŚ;éŤ
 äjăăĚçşèŎăăŭŮPythonăăĚăăĹăăăăAřijŇăžŎèĂŇăĹ;ăĠžăăIJłăRŚçTřéŤŽèrrçŽĎăŮŭăĂžăjăçŽĎćíŇăžŘèĚ

èğćăĚşăĚžăăĹ

faulthandler æłăăĹŮèČ;èćŇçŤíăĹăăyŏăjăèğćăĚşşéĚăyłéŮŏécŸăĂĆ
 ăIJłăjăçŽĎćíŇăžŘăy■ăijTăĚĚăyŇăĹŮăžččăAřijŽ

```
import faulthandler
faulthandler.enable()
```

āRēād' ŪēfYāRfāzēāČRāyNélcēfZæuā;fçTl -Xfaulthandler
ælēēfRēāNPythonijZ

```
bash % python3 -Xfaulthandler program.py
```

æIJāRŌijNā;āāRfāzēēō;ō PYTHONFAULTHANDLER çŌfācČāRŸéGRāĀĆ āijĀāRf-
faulthandlerāRŌijNāIJĪCæL'l'āsTāy■çZDēGt'āS;éTŽēfāijZāfijēGt'āyĀāyIPythonéTŽēfāāEæāLēcāæL'Sā■r

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
```

ār;çōæēfZāyIāzūāy■ēČ;āSŁēfL'ā;āCāzčçāAāy■āSŁēGNāGžēTŽāžEijNā;EæYrēGšārSēČ;āSŁēfL'ā;āPyth

ēōlēōž

faulthandlerāijZāIJĪPythonāzčçāAæL'gēāNāGžēTŽçZDæŪūāĀZāRŠā;āāsTçd'žēūšēyIāfæAfrāĀĆ
ēGšārSrijNāōČāijZāSŁēfL'ā;āāGžēTŽæŪūēcñērČçTlçZDæIJĀēāūçžgæL'l'āsTāG;æTfræYrāSŁāyIāĀĆ
āIJĪpdbāSŌNāĒūāzŪPythonērČērTāZlçZDāyōāL'l'āyNrijNā;āārSēČ;ēf;æāzæžræžRæL'āLrēTŽēfāæL'ĀāIJĪçZD

faulthandlerāy■āijZāSŁēfL'ā;āāzžā;TČēr■ēlĀāy■çZDēTŽēfāfæAfrāĀĆ
āZāæ■d'rijNā;āēIJĀēēAā;fçTlāijāçzšçZDČērČērTāZlrijNārTāēCgdbāĀĆ
āy■ēfGrijNāIJĪfaulthandlerēf;ēyIāfæAfrāRfāzēēōl'ā;āāŌzāLd'æŪ■āzŌāSŁēGNçIĀæL'NāĀĆ
ēfYēēAæşlæDRçZDæYrāIJĪCāy■æšRāžZçszādNçZDēTŽēfāRrēČ;āy■ād'IāōzæYšæAcād'■āĀĆ
ā;NāēČrijNāēCādIJāyĀāyIæL'l'āsTāyčāijČāžEçlNāžRāāEæāLāfæAfrīijNāōČāijZēōl'faulthandlerāy■āRççT
ēČcāzLā;āāzšā;Ūāy■āLrāzžā;Tē;šāGžrijLēZd'āžEçlNāžRāēTæžČād'ŪrijL'āĀĆ

18 éZDā;TA

18.1 ālJlçZŒējDæžR

<http://docs.python.org>

āēČādIJā;āēIJĀēēAæūsāĒēāžEēgčæŌççl'ūēr■ēlĀāSŌNāēlāāIŪçZDçzEēLČrijNéCčāzLāy■āfĒērt'rijNPyth
3 çZDæŪGæāçēāNāy■æYrāzēāL'■çZDēĀAçL'LæIJñ

<http://www.python.org/dev/peps>

āēČādIJā;āāRŠçRĒēgčāyžpythonēr■ēlĀæūsāLāæŪrçL'zæĀgçZDāLlæIJzāzēāRĒāōdçŌrçZDçzEēLČrijN
Enhancement ProposalsāĀT-PythonāijĀāRŠçijŪçāAēgDēNČrijL'çzIārzæYrēlāyāōlēr'ççZDējDæžRāĀĆārd'

<http://pyvideo.org>

èfŽéGŇæIJL'æIëèGłæIJĀèĤŚçŽĎPyConăd'găijŽăĀAçTłæLũçžDègAéIcâijŽçL'çŽĎăd'gèGRègEéçŚæi
3ăy■æûzăŁăçŽĎçŽĎăŮřçL'žæĀgăĀĆ

<http://code.activestate.com/recipes/langs/python>

éTfæIJšăžèæIëiijŇActiveStateçŽĎPythonçL'ŁăiŮăũšçžRæŁRăyžăyĀăylæL'ăŁræTřăžěă■CèôaçŽĎéŚŁ

<http://stackoverflow.com/questions/tagged/python>

Stack Overflow çŽôăL'■æIJL'èŮĒèĤG175,000ăyléŮôécŸècŇæăGèôřăyžPythonçŽyăĒsrijLèĀŇăĒŮăy■ăd'
3çŽĎiijL'ăĀĆăřçôăéŮôécŸăŤŇăŽđç■TçŽĎèťléGRăy■ăRŇiijŇă;EæŸřăž■çĎŮèC;ăRŚçŌřăŁăd'Žăë;ăijŸçg

18.2 Pythonă■ęăžăăžęçs■

ăyŇéIcèĤŽăžŽăžęçs■æRRă;ŽăžEăřžPythonçijŮçłŇçŽĎăĒéŮłăžŇçz■iijŇăyťéG■çĆăť;ăIJłăžEPython
3ăyŁăĀĆ

- *Learning Python*iijŇçŇăăŽŽçL'Ł iijŇă;IJèĀĒ Mark LutziijŇ ŌăĀŽReilly & Associates
ăĜžçL'Ł (2009)ăĀĆ
- *The Quick Python Book*iijŇă;IJèĀĒ Vernon CederiijŇ Manning äĜžçL'Ł(2010)ăĀĆ
- *Python Programming for the Absolute Beginner*iijŇçŇăăyL'çL'ŁiijŇă;IJèĀĒ Michael
DawsoniijŇCourse Technology PTR äĜžçL'Ł(2010).
- *Beginning Python: From Novice to Professional*iijŇçŇăăžŇçL'ŁiijŇ ä;IJèĀĒ Magnus
Lie HetăĀR landiijŇ Apress äĜžçL'Ł(2008).
- *Programming in Python 3*iijŇçŇăăžŇçL'ŁiijŇă;IJèĀĒ Mark SummerfieldiijŇAddison-
Wesley äĜžçL'Ł (2010).

18.3 éŇŸçžgăžęçs■

ăyŇéIcçŽĎèĤŽăžŽăžęçs■æRRă;ŽăžEăžťăd'ŽénŸçžgçŽĎèŇĆăžť iijŇăžšăŇĒăRŇPython
3ăŮžéIcçŽĎăĒăôžăĀĆ

- *Programming Python*iijŇçŇăăŽŽçL'Ł, by Mark Lutz, ŌăĀŽReilly & Associates
ăĜžçL'Ł(2010).
- *Python Essential Reference*iijŇçŇăăŽŽçL'ŁiijŇă;IJèĀĒ David Beazley, Addison-Wesley
ăĜžçL'Ł(2009).
- *Core Python Applications Programming*iijŇçŇăăyL'çL'ŁiijŇă;IJèĀĒ Wesley Chun,
Prentice Hall äĜžçL'Ł(2012).
- *The Python Standard Library by Example* iijŇ ä;IJèĀĒ Doug HellmanniijŇAddison-
Wesley äĜžçL'Ł(2011).
- *Python 3 Object Oriented Programming*iijŇă;IJèĀĒ Dusty Phillips, Packt Publishing
ăĜžçL'Ł(2010).

- *Porting to Python 3* by Lennart Regebro, CreateSpace (2011), <http://python3porting.com>.

19 附录

附录

- 附录
- 附录
- Email: yidao620@gmail.com
- 附录 <https://www.xncoding.com/>
- GitHub: <https://github.com/yidao620c>



扫描上面的QR Code，加我WeChat。

20 Roadmap

2014/08/10 - 2014/08/31:

	github 仓库
	read the docs

2014/09/01 - 2014/10/31:

	附录
--	----

2014/11/01 - 2015/01/31:

	åL'■8çnáçfzèrSåõÑæLŘ
2015/02/01 - 2015/03/31:	
	åL'■9çnáçfzèrSåõÑæLŘ
2015/04/01 - 2015/05/31:	
	10çnáçfzèrSåõÑæLŘ
2015/06/01 - 2015/06/30:	
	11çnáçfzèrSåõÑæLŘ
2015/07/01 - 2015/07/31:	
	12çnáçfzèrSåõÑæLŘ
2015/08/01 - 2015/08/31:	
	13çnáçfzèrSåõÑæLŘ
2015/09/01 - 2015/11/30:	
	14çnáçfzèrSåõÑæLŘ
2015/12/01 - 2015/12/20:	
	15çnáçfzèrSåõÑæLŘ
2015/12/21 - 2015/12/31:	
	årzáĚléČlçfzèrSèfZèaÑæäaårzáÿĂæñą
2016/01/01 - 2016/01/10:	
	<p>årzáđ'ŮăĚñăi jĂăRŚăÿČăõÑæt'çL'Íl.</p> <p>↪0ïi jŇăŇĚăŇñè; ñæ■căŘŎçŽĎPDFæŮĞăžú</p>