

—	Přehled kapitol	—
-1.	Co najdete v „Ponořme se do Pythonu 3“ nového — 17	—
0.	Instalujeme Python — 21	—
1.	Váš první pythonovský program — 45	—
2.	Přirozené datové typy — 61	—
3.	Generátorová notace — 91	—
4.	Řetězce — 105	—
5.	Regulární výrazy — 123	—
6.	Uzávěry a generátory — 143	—
7.	Třídy a iterátory — 159	—
8.	Iterátory pro pokročilé — 173	—
9.	Unit Testing — 193	—
10.	RefaktORIZACE — 219	—
11.	Soubory — 235	—
12.	XML — 255	—
13.	Serializace pythonovských objektů — 277	—
14.	Webové služby nad HTTP — 297	—
15.	Případová studie: Přepis chardet pro Python 3 — 329	—
16.	Balení pythonovských knihoven — 359	—
A.	Přepis kódu do Python 3 s využitím 2to3 — 377	—
B.	Jména speciálních metod — 405	—
C.	Čím pokračovat — 423	—
D.	Odstraňování problémů — 427	—

-
- 1. **Co najdete v „Ponořme se do Pythonu 3^e nového” — 17**
 - 1.1. aneb „záporná úroveň” — 19
-
- O. Instalujeme Python — 21**
 - 0.1. Ponořme se — 23
 - 0.2. Který Python je pro vás ten správný? — 23
 - 0.3. Instalace pod Microsoft Windows — 24
 - 0.4. Instalace pod Mac OS X — 29
 - 0.5. Instalace pod Ubuntu Linux — 36
 - 0.6. Instalace na jiných platformách — 40
 - 0.7. Použití Python Shell — 41
 - 0.8. Editory a vývojová prostředí pro Python — 43
-
- 1. Váš první pythonovský program — 45**
 - 1.1. Ponořme se — 47
 - 1.2. Deklarace funkcí — 48
 - 1.2.1. Nepovinné a pojmenované argumenty — 49
 - 1.3. Psaní čitelného kódu — 51
 - 1.3.1. Dokumentační řetězce — 51
 - 1.4. Vyhledávací cesta pro import — 52
 - 1.5. Všechno je objekt — 53
 - 1.5.1. Co to vlastně je objekt? — 54
 - 1.6. Odsazování kódu — 54
 - 1.7. Výjimky — 55
 - 1.7.1. Obsluha chyb importu — 57
 - 1.8. Volné proměnné — 58
 - 1.9. Vše je citlivé na velikost písmen — 58
 - 1.10. Spouštění skriptů — 59
 - 1.11. Přečtěte si — 60
-
- 2. Přirozené datové typy — 61**
 - 2.1. Ponořme se — 63
 - 2.2. Booleovský typ — 63
 - 2.3. Čísla — 64
 - 2.3.1. Vynucení převodu celých čísel na reálná a naopak — 65
 - 2.3.2. Běžné operace s čísly — 66
 - 2.3.3. Zlomky — 67
 - 2.3.4. Trigonometrie — 67
 - 2.3.5. Čísla v booleovském kontextu — 68
 - 2.4. Seznamy — 69
 - 2.4.1. Vytvoření seznamu — 69
 - 2.4.2. Vytváření podseznamů — 70
 - 2.4.3. Přidávání položek do seznamu — 71
 - 2.4.4. Vyhledávání hodnoty v seznamu — 73
 - 2.4.5. Odstraňování položek ze seznamu — 74
 - 2.4.6. Odstraňování položek ze seznamu: Bonusové kolo — 75
 - 2.4.7. Seznamy v booleovském kontextu — 75
 - 2.5. N-tice — 76
 - 2.5.1. N-tice v booleovském kontextu — 78
 - 2.5.2. Přiřazení více hodnot najednou — 78
 - 2.6. Množiny — 79
 - 2.6.1. Vytvoření množiny — 79
 - 2.6.2. Úprava množiny — 81
 - 2.6.3. Odstraňování položek z množiny — 82
 - 2.6.4. Běžné množinové operace — 83
 - 2.6.5. Množiny v booleovském kontextu — 85
 - 2.7. Slovníky — 86
 - 2.7.1. Vytvoření slovníku — 86
 - 2.7.2. Úprava slovníku — 87
 - 2.7.3. Slovníky se smíšeným obsahem — 87
 - 2.7.4. Slovníky v booleovském kontextu — 88
 - 2.8. None — 89
 - 2.8.1. None v booleovském kontextu — 90
 - 2.9. Přečtěte si — 90
-
- 3. Generátorová notace — 91**
 - 3.1. Ponořme se — 93
 - 3.2. Práce se soubory a s adresáři — 93
 - 3.2.1. Aktuální pracovní adresář — 93
 - 3.2.2. Práce se jmény souborů a adresářů — 94
 - 3.2.3. Výpis adresářů — 96
 - 3.2.4. Získání dalších informací o souboru — 97
 - 3.2.5. Jak vytvořit absolutní cesty — 98
 - 3.3. Generátorová notace seznamu — 98
 - 3.4. Generátorová notace slovníku — 100
 - 3.4.1. Další legrácky s generátorovou notací slovníků — 102
 - 3.5. Generátorová notace množin — 103
 - 3.6. Přečtěte si — 103

- 4. Řetězce — 105**
- 4.1. Pár nudných věcí, kterým musíme rozumět dříve, než se budeme moci ponořit — 107
- 4.2. Unicode — 109
- 4.3. Ponořme se — 111
- 4.4. Formátovací řetězce — 111
 - 4.4.1. Složená jména oblastí — 113
 - 4.4.2. Specifikátory formátu — 114
- 4.5. Další běžné metody řetězců — 115
 - 4.5.1. Vykrajování podřetězců — 117
- 4.6. Řetězce vs. bajty — 117
- 4.7. Závěrečná poznámka: Kódování znaků v pythonovském zdrojovém textu — 120
- 4.8. Přečtěte si — 121

- 5. Regulární výrazy — 123**
- 5.1. Ponořme se — 125
- 5.2. Případová studie: Adresa ulice — 125
- 5.3. Případová studie: Římská čísla — 128
 - 5.3.1. Kontrola tisícovek — 128
 - 5.3.2. Kontrola stovek — 129
- 5.4. Využití syntaxe $\{n, m\}$ — 131
 - 5.4.1. Kontrola desítek a jednotek — 132
- 5.5. Víceslovné regulární výrazy — 134
- 5.6. Případová studie:
Analýza telefonních čísel — 136
- 5.7. Shrnutí — 141

5. Regulární výrazy

- 6. Uzávěry a generátory — 143**
- 6.1. Ponořme se — 145
- 6.2. Já vím jak na to! Použijeme regulární výrazy! — 146
- 6.3. Seznam funkcí — 148
- 6.4. Seznam vzorků — 150
- 6.5. Soubor vzorků — 152
- 6.6. Generátory — 154
 - 6.6.1. Generátor Fibonacciho posloupnosti — 155
 - 6.6.2. Generátor pravidel pro množné číslo — 156
- 6.7. Přečtěte si — 158

- 7. Třídy a iterátory — 159**
- 7.1. Ponořme se — 161
- 7.2. Definice tříd — 161
 - 7.2.1. Metoda `__init__()` — 162
- 7.3. Vytváření instancí tříd — 163
- 7.4. Členské proměnné — 163
- 7.5. Fibonacciho iterátor — 164
- 7.6. Iterátor pro pravidla množného čísla — 166
- 7.7. Přečtěte si — 172

- 8. Iterátory pro pokročilé — 173**
- 8.1. Ponoříme se — 175
- 8.2. Nalezení všech výskytů vzorku — 176
- 8.3. Nalezení jedinečných prvků
posloupností — 177
- 8.4. Činíme předpoklady — 178
- 8.5. Generátorové výrazy — 179
- 8.6. Výpočet permutací (pro lenochy) — 180
- 8.7. Další legrácky v modulu `itertools` — 182
- 8.8. Nový způsob úpravy řetězce — 185
- 8.9. Vyhodnocování libovolných řetězců
zachycujících pythonovské výrazy — 187
- 8.10. Spojme to všechno dohromady — 190
- 8.11. Přečtěte si — 191

8. Iterátory pro pokročilé

- 9. Unit Testing — 193**
- 9.1. (Ne)ponožme se — 195
- 9.2. Jediná otázka — 196
- 9.3. „Zastav a začni hořet“ — 202
- 9.4. Více zastávek, více ohně — 206
- 9.5. A ještě jedna věc... — 209
- 9.6. Symetrie, která potěší — 211
- 9.7. Více špatných vstupů — 215

- 10. Refaktorizace — 219**
- 10.1. Ponořme se — 221
- 10.2. Zvládání měnících se požadavků — 223
- 10.3. Refaktorizace — 228
- 10.4. Shrnutí — 232

- 11. Soubory — 235**
- 11.1. Ponořme se — 237
- 11.2. Čtení z textových souborů — 237
 - 11.2.1. Kódování znaků vystrkuje svou ošklivou hlavu — 237
 - 11.2.2. Objekty typu stream — 238
 - 11.2.3. Čtení dat z textového souboru — 239
 - 11.2.4. Zavírání souborů — 241
 - 11.2.5. Automatické zavírání souborů — 242
 - 11.2.6. Čtení dat po řádcích — 243
- 11.3. Zápis do textových souborů — 245
 - 11.3.1. A znovu kódování znaků — 246
- 11.4. Binární soubory — 246
- 11.5. Objekty typu stream z nesouborových zdrojů — 247
 - 11.5.1. Práce s komprimovanými soubory — 249
- 11.6. Standardní vstup, výstup a chybový výstup — 250
 - 11.6.1. Přesměrování standardního výstupu — 251
- 11.7. Přečtěte si — 254

- 12. XML — 255**
- 12.1. Ponoříme se — 257
- 12.2. Pětiminutový rychlokurz XML — 258
- 12.3. Struktura Atom Feed — 261
- 12.4. Analýza XML — 263
 - 12.4.1. Elementy jsou reprezentovány seznamy — 264
 - 12.4.2. Atributy jsou reprezentovány slovníky — 264
- 12.5. Vyhledávání uzlů v XML dokumentu — 265
- 12.6. lxml jde ještě dál — 268
- 12.7. Generování XML — 270
- 12.8. Analýza porušeného XML — 273
- 12.9. Přečtete si — 275

- 13. Serializace pythonovských objektů — 277**
- 13.1. Ponořme se — 279
 - 13.1.1. Stručná poznámka k příkladům v této kapitole — 279
- 13.2. Uložení dat do „pickle souboru“ — 280
- 13.3. Načítání dat z „pickle souboru“ — 281
- 13.4. „Piklení“ bez souboru — 283
- 13.5. Bajty a řetězce znovu zvedají své ošklivé hlavy — 284
- 13.6. Ladění „pickle souborů“ — 284
- 13.7. Serializace pythonovských objektů pro čtení z jiných jazyků — 286
- 13.8. Uložení dat do JSON souboru — 287
- 13.9. Zobrazení pythonovských datových typů do JSON — 289
- 13.10. Serializace datových typů, které JSON nepodporuje — 289
- 13.11. Načítání dat z JSON souboru — 293
- 13.12. Přečtěte si — 295

-
- 4. Řetězce — 105**
- 4.1. Pár nudných věcí, kterým musíme rozumět dříve, než se budeme moci ponořit — 107
- 4.2. Unicode — 109
- 4.3. Ponořme se — 111
- 4.4. Formátovací řetězce — 111
- 4.4.1. Složená jména oblastí — 113
- 4.4.2. Specifikátory formátu — 114
- 4.5. Další běžné metody řetězců — 115
- 4.5.1. Vykrajování podřetězců — 117
- 4.6. Řetězce vs. bajty — 117
- 4.7. Závěrečná poznámka: Kódování znaků v pythonovském zdrojovém textu — 120
- 4.8. Přečtěte si — 121
-
- 5. Regulární výrazy — 123**
- 5.1. Ponořme se — 125
- 5.2. Případová studie: Adresa ulice — 125
- 5.3. Případová studie: Římská čísla — 128
- 5.3.1. Kontrola tisícovek — 128
- 5.3.2. Kontrola stovek — 129
- 5.4. Využití syntaxe $\{n,m\}$ — 131
- 5.4.1. Kontrola desítek a jednotek — 132
- 5.5. Víceslovné regulární výrazy — 134
- 5.6. Případová studie: Analýza telefonních čísel — 136
- 5.7. Shrnutí — 141
-
- 6. Uzávěry a generátory — 143**
- 6.1. Ponořme se — 145
- 6.2. Já vím jak na to! Použijeme regulární výrazy! — 146
- 6.3. Seznam funkcí — 148
- 6.4. Seznam vzorků — 150
- 6.5. Soubor vzorků — 152
- 6.6. Generátory — 154
- 6.6.1. Generátor Fibonacciho posloupnosti — 155
- 6.6.2. Generátor pravidel pro množné číslo — 156
- 6.7. Přečtěte si — 158
-
- 7. Třídy a iterátory — 159**
- 7.1. Ponořme se — 161
- 7.2. Definice tříd — 161
- 7.2.1. Metoda `__init__()` — 162
- 7.3. Vytváření instancí tříd — 163
- 7.4. Členské proměnné — 163
- 7.5. Fibonacciho iterátor — 164
- 7.6. Iterátor pro pravidla množného čísla — 166
- 7.7. Přečtěte si — 172
-
- 8. Iterátory pro pokročilé — 173**
- 8.1. Ponořme se — 175
- 8.2. Nalezení všech výskytů vzorku — 176
- 8.3. Nalezení jedinečných prvků posloupnosti — 177
- 8.4. Činíme předpoklady — 178
- 8.5. Generátorové výrazy — 179
- 8.6. Výpočet permutací (pro lenochy) — 180
- 8.7. Další legrácky v modulu `itertools` — 182
- 8.8. Nový způsob úpravy řetězce — 185
- 8.9. Vyhodnocování libovolných řetězců zachycujících pythonovské výrazy — 187
- 8.10. Spojme to všechno dohromady — 190
- 8.11. Přečtěte si — 191
-
- 9. Unit Testing — 193**
- 9.1. (Ne)ponořme se — 195
- 9.2. Jediná otázka — 196
- 9.3. „Zastav a začni hořet“ — 202
- 9.4. Více zastávek, více ohně — 206
- 9.5. A ještě jedna věc... — 209
- 9.6. Symetrie, která potěší — 211
- 9.7. Více špatných vstupů — 215
-
- 10. Refaktorizace — 219**
- 10.1. Ponořme se — 221
- 10.2. Zvládání měnících se požadavků — 223
- 10.3. Refaktorizace — 228
- 10.4. Shrnutí — 232

14. Webové služby nad HTTP — 297

- 14.1. Ponoříme se — 299
- 14.2. Vlastnosti HTTP — 300
 - 14.2.1. Používání mezipaměti — 300
 - 14.2.2. Kontrola Last-Modified — 301
 - 14.2.3. Kontrola ETag — 303
 - 14.2.4. Komprese — 304
 - 14.2.5. Přesměrování — 304
- 14.3. Jak se nedostat k datům přes HTTP — 305
- 14.4. Co že to máme na drátě? — 306
- 14.5. Představujeme `httplib2` — 309
 - 14.5.1. Krátká odbočka vysvětlující, proč `httplib2` vrací bajty místo řetězců — 311
 - 14.5.2. Jak `httplib2` zachází s mezipamětí — 312
 - 14.5.3. Jak `httplib2` zachází s hlavičkami Last-Modified a ETag — 315
 - 14.5.4. Jak `httplib2` pracuje s kompresí — 318
 - 14.5.5. Jak `httplib2` řeší přesměrování — 318
- 14.6. Za hranicemi HTTP GET — 322
- 14.7. Za hranicemi HTTP POST — 326
- 14.8. Přečtěte si — 328

- 15. Případová studie:
Přepis chardet pro Python 3 — 329**
- 15.1. Ponořme se — 331
- 15.2. Co se rozumí autodetekcí znakového kódování? — 331
 - 15.2.1. Není to náhodou neproveditelné? — 331
 - 15.2.2. Existuje takový algoritmus? — 332
- 15.3. Úvod do modulu `chardet` — 332
 - 15.3.1. UTF-N s BOM — 332
 - 15.3.2. Kódování escape sekvencemi — 333
 - 15.3.3. Vícebajtová kódování — 333
 - 15.3.4. Jednobajtová kódování — 334
 - 15.3.5. `windows-1252` — 334
- 15.4. Spouštíme 2to3 — 335
- 15.5. Krátká odbočka k vícesouborovým modulům — 338
- 15.6. Opravme, co 2to3 neumí — 340
 - 15.6.1. `False` je syntaktická chyba — 340
 - 15.6.2. Nenalezen modul `constants` — 341
 - 15.6.3. Jméno `'file'` není definováno — 342
 - 15.6.4. Řetězcový vzorek nelze použít pro bajtové objekty — 343
 - 15.6.5. Objekt typu `'bytes'` nelze implicitně převést na `str` — 345
 - 15.6.6. Nepodporované typy operandů pro `+`: `'int'` a `'bytes'` — 348
 - 15.6.7. Funkce `ord()` očekávala řetězec o délce 1, ale byl nalezen `int` — 350
 - 15.6.8. Neuspořádatelné datové typy: `int()` `>=` `str()` — 352
 - 15.6.9. Globální jméno `'reduce'` není definováno — 355
- 15.7. Shrnutí — 357

- 16. Balení pythonovských knihoven — 359**
- 16.1. Ponořme se — **361**
- 16.2. Věci, které za nás Distutils neudělají — **362**
- 16.3. Struktura adresáře — **363**
- 16.4. Píšeme svůj instalační skript — **364**
- 16.5. Přidáváme klasifikaci našeho balíčku — **366**
- 16.5.1. Příklady dobrých klasifikátorů balíčků — **367**
- 16.6. Určení dalších souborů prostřednictvím manifestu — **368**
- 16.7. Kontrola chyb v našem instalačním skriptu — **369**
- 16.8. Vytvoření distribuce obsahující zdrojové texty — **369**
- 16.9. Vytvoření grafického instalačního programu — **371**
- 16.9.1. Tvorba instalačních balíčků pro jiné operační systémy — **373**
- 16.10. Přidání našeho softwaru do Python Package Index — **373**
- 16.11. Více možných budoucností balení pythonovských produktů — **375**
- 16.12. Přečtěte si — **375**

- A. Přepis kódu do Pythonu 3 s využitím 2to3 — 377**
- A.1. Ponořme se — 379
 - A.2. Příkaz `print` — 379
 - A.3. Literály Unicode řetězců — 380
 - A.4. Globální funkce `unicode()` — 380
 - A.5. Datový typ `long` — 380
 - A.6. Porovnání `<>` — 381
 - A.7. Slovníková metoda `has_key()` — 381
 - A.8. Slovníkové metody, které vrací seznamy — 382
 - A.9. Moduly, které byly přejmenovány nebo reorganizovány — 383
 - A.9.1. `http` — 383
 - A.9.2. `urllib` — 384
 - A.9.3. `dbm` — 385
 - A.9.4. `xmlrpc` — 385
 - A.9.5. Ostatní moduly — 386
 - A.10. Relativní `importy` uvnitř balíčku — 387
 - A.11. Metoda iterátoru `next()` — 388
 - A.12. Globální funkce `filter()` — 388
 - A.13. Globální funkce `map()` — 389
 - A.14. Globální funkce `reduce()` — 390
 - A.15. Globální funkce `apply()` — 390
 - A.16. Globální funkce `intern()` — 390
 - A.17. Příkaz `exec` — 391
 - A.18. Příkaz `execfile` — 391
 - A.19. `repr` literály (zpětné apostrofy) — 392
 - A.20. Příkaz `try...except` — 392
 - A.21. Příkaz `raise` — 393
 - A.22. Metoda generátorů `throw` — 393
 - A.23. Globální funkce `xrange()` — 394
 - A.24. Globální funkce `raw_input()` a `input()` — 395
 - A.25. Atributy funkcí `func_*` — 395
 - A.26. Metoda `xreadlines()` V/V objektů — 396
 - A.27. `lambda` funkce, které akceptují `n`-tici místo více parametrů — 396
 - A.28. Atributy speciálních metod — 397
 - A.29. Speciální metoda `__nonzero__` — 397
 - A.30. Oktalové literály — 398
 - A.31. `sys.maxint` — 398
 - A.32. Globální funkce `callable()` — 399
 - A.33. Globální funkce `zip()` — 399
 - A.34. Výjimka `StandardError` — 399
 - A.35. Konstanty modulu `types` — 400
 - A.36. Globální funkce `isinstance()` — 400
 - A.37. Datový typ `basestring` — 401
 - A.38. `itertools` module — 401
 - A.39. `sys.exc_type`, `sys.exc_value`, `sys.exc_traceback` — 401
 - A.40. Generátory seznamů nad `n`-ticemi — 402
 - A.41. Funkce `os.getcwdu()` — 402
 - A.42. Metařádky — 402
 - A.43. Věci týkající se stylu — 403
 - A.43.1. Množinové literály (`set()`; explicitně) — 403
 - A.43.2. Globální funkce `buffer()` (explicitně) — 403
 - A.43.3. Bílé znaky kolem čárek (explicitně) — 404
 - A.43.4. Běžné obraty (explicitně) — 404

- B. Jména speciálních metod — 405**
- B.1. Ponořme se — 407
- B.2. Základy — 407
- B.3. Třídy, které se chovají jako iterátory — 407
- B.4. Vypočítávané atributy — 408
- B.5. Třídy, které se chovají jako funkce — 411
- B.6. Třídy, které se chovají jako množiny — 412
- B.7. Třídy, které se chovají jako slovníky — 413
- B.8. Třídy, které se chovají jako čísla — 414
- B.9. Třídy, které se dají porovnávat — 417
- B.10. Třídy, které podporují serializaci — 418
- B.11. Třídy, které mohou být použity
v bloku with — 418
- B.12. Opravdu esoterické věci — 420
- B.13. Přechtěte si — 420

- C. Čím pokračovat — 423**
- C.1. Doporučuji k přečtení — 425
- C.2. Kde hledat kód kompatibilní s Pythonem 3 — 426

- D. Odstraňování problémů — 427**
- D.1. Ponořme se — 429
- D.2. Jak se dostat k příkazovému řádku — 429
- D.3. Spuštění Pythonu z příkazového řádku — 429

-
- 11. Soubory — 235**
 - 11.1. Ponořme se — 237
 - 11.2. Čtení z textových souborů — 237
 - 11.2.1. Kódování znaků vystrkuje svou ošklivou hlavu — 237
 - 11.2.2. Objekty typu stream — 238
 - 11.2.3. Čtení dat z textového souboru — 239
 - 11.2.4. Zavírání souborů — 241
 - 11.2.5. Automatické zavírání souborů — 242
 - 11.2.6. Čtení dat po řádcích — 243
 - 11.3. Zápis do textových souborů — 245
 - 11.3.1. A znovu kódování znaků — 246
 - 11.4. Binární soubory — 246
 - 11.5. Objekty typu stream z nesouborových zdrojů — 247
 - 11.5.1. Práce s komprimovanými soubory — 249
 - 11.6. Standardní vstup, výstup a chybový výstup — 250
 - 11.6.1. Přesměrování standardního výstupu — 251
 - 11.7. Přečtěte si — 254

 - 12. XML — 255**
 - 12.1. Ponořme se — 257
 - 12.2. Pětiminutový rychlokurz XML — 258
 - 12.3. Struktura Atom Feed — 261
 - 12.4. Analýza XML — 263
 - 12.4.1. Elementy jsou reprezentovány seznamy — 264
 - 12.4.2. Atributy jsou reprezentovány slovníky — 264
 - 12.5. Vyhledávání uzlů v XML dokumentu — 265
 - 12.6. lxml jde ještě dál — 268
 - 12.7. Generování XML — 270
 - 12.8. Analýza porušeného XML — 273
 - 12.9. Přečtěte si — 275

 - 13. Serializace pythonových objektů — 277**
 - 13.1. Ponořme se — 279
 - 13.1.1. Stručná poznámka k příkladům v této kapitole — 279
 - 13.2. Uložení dat do „pickle souboru“ — 280
 - 13.3. Načítání dat z „pickle souboru“ — 281
 - 13.4. „Piklení“ bez souboru — 283
 - 13.5. Bajty a řetězce znovu zvedají své ošklivé hlavy — 284
 - 13.6. Ladění „pickle souborů“ — 284
 - 13.7. Serializace pythonových objektů pro čtení z jiných jazyků — 286
 - 13.8. Uložení dat do JSON souboru — 287
 - 13.9. Zobrazení pythonových datových typů do JSON — 289
 - 13.10. Serializace datových typů, které JSON nepodporuje — 289
 - 13.11. Načítání dat z JSON souboru — 293
 - 13.12. Přečtěte si — 295

 - 14. Webové služby nad HTTP — 297**
 - 14.1. Ponořme se — 299
 - 14.2. Vlastnosti HTTP — 300
 - 14.2.1. Používání mezipaměti — 300
 - 14.2.2. Kontrola Last-Modified — 301
 - 14.2.3. Kontrola ETag — 303
 - 14.2.4. Komprese — 304
 - 14.2.5. Přesměrování — 304
 - 14.3. Jak se nedostat k datům přes HTTP — 305
 - 14.4. Co že to máme na drátě? — 306
 - 14.5. Představujeme httpLib2 — 309
 - 14.5.1. Krátká odbočka vysvětlující, proč httpLib2 vrací bajty místo řetězců — 311
 - 14.5.2. Jak httpLib2 zachází s mezipamětí — 312
 - 14.5.3. Jak httpLib2 zachází s hlavičkami Last-Modified a ETag — 315
 - 14.5.4. Jak http2lib pracuje s kompresí — 318
 - 14.5.5. Jak httpLib2 řeší přesměrování — 318
 - 14.6. Za hranicemi HTTP GET — 322
 - 14.7. Za hranicemi HTTP POST — 326
 - 14.8. Přečtěte si — 328

 - 15. Případová studie: Přepis chardet pro Python 3 — 329**
 - 15.1. Ponořme se — 331
 - 15.2. Co se rozumí autodetekcí znakového kódování? — 331

- 15.2.1. Není to náhodou neproveditelné? — 331
 - 15.2.2. Existuje takový algoritmus? — 332
 - 15.3. Úvod do modulu `charset` — 332
 - 15.3.1. UTF-N s BOM — 332
 - 15.3.2. Kódování escape sekvencemi — 333
 - 15.3.3. Vícebajtová kódování — 333
 - 15.3.4. Jednobajtová kódování — 334
 - 15.3.5. `windows-1252` — 334
 - 15.4. Spouštíme `2to3` — 335
 - 15.5. Krátká odbočka k vícesouborovým modulům — 338
 - 15.6. Opravme, co `2to3` neumí — 340
 - 15.6.1. `False` je syntaktická chyba — 340
 - 15.6.2. Nenalezen modul `constants` — 341
 - 15.6.3. Jméno `'file'` není definováno — 342
 - 15.6.4. Řetězcový vzorek nelze použít pro bajtové objekty — 343
 - 15.6.5. Objekt typu `'bytes'` nelze implicitně převést na `str` — 345
 - 15.6.6. Nepodporované typy operandů pro `+`: `'int'` a `'bytes'` — 348
 - 15.6.7. Funkce `ord()` očekávala řetězec o délce 1, ale byl nalezen `int` — 350
 - 15.6.8. Neuspořádatelné datové typy: `int() >= str()` — 352
 - 15.6.9. Globální jméno `'reduce'` není definováno — 355
 - 15.7. Shrnutí — 357
-
- 16. Balení pythonovských knihoven — 359**
 - 16.1. Ponořme se — 361
 - 16.2. Věci, které za nás `Distutils` neudělají — 362
 - 16.3. Struktura adresáře — 363
 - 16.4. Píšeme svůj instalační skript — 364
 - 16.5. Přidáváme klasifikaci našeho balíčku — 366
 - 16.5.1. Příklady dobrých klasifikátorů balíčků — 367
 - 16.6. Určení dalších souborů prostřednictvím manifestu — 368
 - 16.7. Kontrola chyb v našem instalačním skriptu — 369
 - 16.8. Vytvoření distribuce obsahující zdrojové texty — 369
 - 16.9. Vytvoření grafického instalačního programu — 371
 - 16.9.1. Tvorba instalačních balíčků pro jiné operační systémy — 373
 - 16.10. Přidání našeho softwaru do `Python Package Index` — 373
 - 16.11. Více možných budoucností balení pythonovských produktů — 375
 - 16.12. Přečtěte si — 375
-
- A. Přepis kódu do Pythonu 3 s využitím `2to3` — 377**
 - A.1. Ponořme se — 379
 - A.2. Příkaz `print` — 379
 - A.3. Literály `Unicode` řetězců — 380
 - A.4. Globální funkce `unicode()` — 380
 - A.5. Datový typ `long` — 380
 - A.6. Porovnání `<>` — 381
 - A.7. Slovníková metoda `has_key()` — 381
 - A.8. Slovníkové metody, které vrací seznamy — 382
 - A.9. Moduly, které byly přejmenovány nebo reorganizovány — 383
 - A.9.1. `http` — 383
 - A.9.2. `urllib` — 384
 - A.9.3. `dbm` — 385
 - A.9.4. `xmlrpc` — 385
 - A.9.5. Ostatní moduly — 386
 - A.10. Relativní importy uvnitř balíčku — 387
 - A.11. Metoda iterátoru `next()` — 388
 - A.12. Globální funkce `filter()` — 388
 - A.13. Globální funkce `map()` — 389
 - A.14. Globální funkce `reduce()` — 390
 - A.15. Globální funkce `apply()` — 390
 - A.16. Globální funkce `intern()` — 390
 - A.17. Příkaz `exec` — 391
 - A.18. Příkaz `execfile` — 391
 - A.19. `repr`-literály (zpětné apostrofy) — 392
 - A.20. Příkaz `try...except` — 392

- A.21. Příkaz `raise` — 393
 - A.22. Metoda generátorů `throw` — 393
 - A.23. Globální funkce `xrange()` — 394
 - A.24. Globální funkce `raw_input()`
a `input()` — 395
 - A.25. Atributy funkcí `func_*` — 395
 - A.26. Metoda `xreadlines()`
V/V objektů — 396
 - A.27. `lambda` funkce, které akceptují
n-tici místo více parametrů — 396
 - A.28. Atributy speciálních metod — 397
 - A.29. Speciální metoda `__nonzero__` — 397
 - A.30. Oktalové literály — 398
 - A.31. `sys.maxint` — 398
 - A.32. Globální funkce `callable()` — 399
 - A.33. Globální funkce `zip()` — 399
 - A.34. Výjimka `StandardError` — 399
 - A.35. Konstanty modulu `types` — 400
 - A.36. Globální funkce `isinstance()` — 400
 - A.37. Datový typ `basestring` — 401
 - A.38. `itertools` module — 401
 - A.39. `sys.exc_type`, `sys.exc_value`,
`sys.exc_traceback` — 401
 - A.40. Generátory seznamů nad n-ticemi — 402
 - A.41. Funkce `os.getcwd()` — 402
 - A.42. Metatřídy — 402
 - A.43. Věci týkající se stylu — 403
 - A.43.1. Množinové literály (`set()`;
explicitně) — 403
 - A.43.2. Globální funkce `buffer()`
(explicitně) — 403
 - A.43.3. Bílé znaky kolem čárek
(explicitně) — 404
 - A.43.4. Běžné obraty (explicitně) — 404
-
- B. Jména speciálních metod — 405**
 - B.1. Ponořme se — 407
 - B.2. Základy — 407
 - B.3. Třídy, které se chovají
jako iterátory — 407
 - B.4. Vypočítávané atributy — 408
 - B.5. Třídy, které se chovají jako funkce — 411
 - B.6. Třídy, které se chovají
jako množiny — 412
 - B.7. Třídy, které se chovají
jako slovníky — 413
 - B.8. Třídy, které se chovají jako čísla — 414
 - B.9. Třídy, které se dají porovnávat — 417
 - B.10. Třídy, které podporují serializaci — 418
 - B.11. Třídy, které mohou být použity
v bloku `with` — 418
 - B.12. Opravdu esoterické věci — 420
 - B.13. Přečtěte si — 420
-
- C. Čím pokračovat — 423**
 - C.1. Doporučuji k přečtení — 425
 - C.2. Kde hledat kód kompatibilní
s Pythonem 3 — 426
-
- D. Odstraňování problémů — 427**
 - D.1. Ponořme se — 429
 - D.2. Jak se dostat k příkazovému řádku — 429
 - D.3. Spuštění Pythonu z příkazového
řádku — 429

- 0. Instalujeme Python — 21**
 - 0.1. Ponořme se — 23
 - 0.2. Který Python je pro vás ten správný? — 23
 - 0.3. Instalace pod Microsoft Windows — 24
 - 0.4. Instalace pod Mac OS X — 29
 - 0.5. Instalace pod Ubuntu Linux — 36
 - 0.6. Instalace na jiných platformách — 40
 - 0.7. Použití Python Shell — 41
 - 0.8. Editory a vývojová prostředí pro Python — 43

- 1. Váš první pythonovský program — 45**
- 1.1. Ponoříme se — 47
- 1.2. Deklarace funkcí — 48
- 1.2.1. Nepovinné a pojmenované argumenty — 49
- 1.3. Psaní čitelného kódu — 51
- 1.3.1. Dokumentační řetězce — 51
- 1.4. Vyhledávací cesta pro import — 52
- 1.5. Všechno je objekt — 53
- 1.5.1. Co to vlastně je objekt? — 54
- 1.6. Odsazování kódu — 54
- 1.7. Výjimky — 55
- 1.7.1. Obsluha chyb importu — 57
- 1.8. Volné proměnné — 58
- 1.9. Vše je citlivé na velikost písmen — 58
- 1.10. Spouštění skriptů — 59
- 1.11. Přečtěte si — 60

- 2. Půrodníé datové typy — 61**
- 2.1. Ponořme se — 63
- 2.2. Booleovský typ — 63
- 2.3. Čísła — 64
 - 2.3.1. Vynucení převodu celých čísel na reálná a naopak — 65
 - 2.3.2. Běžné operace s čísly — 66
 - 2.3.3. Zlomky — 67
 - 2.3.4. Trigonometrie — 67
 - 2.3.5. Čísła v booleovském kontextu — 68
- 2.4. Seznamy — 69
 - 2.4.1. Vytvoření seznamu — 69
 - 2.4.2. Vytváření podseznamů — 70
 - 2.4.3. Přidávání položek do seznamu — 71
 - 2.4.4. Vyhledávání hodnoty v seznamu — 73
 - 2.4.5. Odstraňování položek ze seznamu — 74
 - 2.4.6. Odstraňování položek ze seznamu: Bonusové kolo — 75
 - 2.4.7. Seznamy v booleovském kontextu — 75
- 2.5. N-tice — 76
 - 2.5.1. N-tice v booleovském kontextu — 78
 - 2.5.2. Přifazení více hodnot najednou — 78
- 2.6. Množiny — 79
 - 2.6.1. Vytvoření množiny — 79
 - 2.6.2. Úprava množiny — 81
 - 2.6.3. Odstraňování položek z množiny — 82
 - 2.6.4. Běžné množinové operace — 83
 - 2.6.5. Množiny v booleovském kontextu — 85
- 2.7. Slovníky — 86
 - 2.7.1. Vytvoření slovníku — 86
 - 2.7.2. Úprava slovníku — 87
 - 2.7.3. Slovníky se smíšeným obsahem — 87
 - 2.7.4. Slovníky v booleovském kontextu — 88
- 2.8. None — 89
 - 2.8.1. None v booleovském kontextu — 90
- 2.9. Přečtěte si — 90

- 3. Generátorová notace — 91**
- 3.1. Ponořme se — 93
- 3.2. Práce se soubory a s adresáři — 93
 - 3.2.1. Aktuální pracovní adresář — 93
 - 3.2.2. Práce se jmény souborů a adresářů — 94
 - 3.2.3. Výpis adresářů — 96
 - 3.2.4. Získání dalších informací o souboru — 97
 - 3.2.5. Jak vytvořit absolutní cesty — 98
- 3.3. Generátorová notace seznamu — 98
- 3.4. Generátorová notace slovníku — 100
 - 3.4.1. Další legrácky s generátorovou notací slovníků — 102
- 3.5. Generátorová notace množin — 103
- 3.6. Přečtěte si — 103

3. Generátorová notace