

# knihovna programátora

- Učebnice jazyka i referenční příručka v jednom
- Probírá konstrukce nevyžadující hlubší znalost objektového programování
- Výklad většiny konstrukcí doplňují syntaktické diagramy
- Použití konstrukcí je předvedeno na příkladech
- Rozbor příčin častých chyb a jejich řešení
- Na knihu navazuje další titul věnovaný objektovým konstrukcím



Rudolf Pecinovský

# Python 3.14

# ALGORITMICKÉ KONSTRUKCE



knihovna programátora

---

RUDOLF PECINOVSKÝ

# Python 3.14

## ALGORITMICKÉ KONSTRUKCE

GRADA  
Publishing

**Rudolf Pecinovský**

# **Python 3.14**

## **Algoritmické konstrukce**

Vydala Grada Publishing, a.s.  
U Průhonu 22, Praha 7  
obchod@grada.cz, www.grada.cz  
tel.: +420 234 264 401  
jako svou 10344. publikaci

Odpovědný redaktor Petr Somogyi  
Fotografie na obálce Depositphotos/mario7  
Grafická úprava a sazba Rudolf Pecinovský  
Počet stran 496  
První vydání, Praha 2025  
Vytisky Tiskárny Havlíčkův Brod a.s.

© Grada Publishing, a.s., 2025  
Cover Design © Grada Publishing, a. s., 2025  
Cover Photo © Depositphotos/mario7

*Upozornění pro čtenáře a uživatele této knihy:  
Všechna práva vyhrazena. Žádná část této tištěné či elektronické knihy  
nesmí být reprodukována a šířena v papírové, elektronické či jiné podobě  
bez předchozího písemného souhlasu nakladatele.  
Neoprávněné užití této knihy bude trestně stíháno.  
Názvy produktů, firem apod. použité v knize mohou být ochrannými známkami  
nebo registrovanými ochrannými známkami příslušných vlastníků.  
Automatizovaná analýza textů nebo dat ve smyslu čl. 4 směrnice 2019/790/EU  
a použití této knihy k trénování AI jsou bez souhlasu nositele práv zakázány.*

ISBN 978–80–271–8342–5 (ePub)  
ISBN 978–80–271–8341–8 (pdf)  
ISBN 978–80–271– 6061-7 (print)

*Mé ženě Jarušce a dětem  
Štěpánce, Pavlínce, Ivance a Michalovi*

# Stručný obsah

Úvod .....	21
<b>Část A Superzáklady</b> .....	<b>31</b>
Kapitola 1 <b>Startujeme</b> .....	32
Kapitola 2 <b>Zadávání jednoduchých hodnot</b> .....	57
Kapitola 3 <b>Zadávání textů – stringů</b> .....	73
Kapitola 4 <b>Volání funkcí</b> .....	90
Kapitola 5 <b>Jednoduché výrazy</b> .....	109
Kapitola 6 <b>Proměnné, výrazy, příkazy</b> .....	123
Kapitola 7 <b>Logické hodnoty a operace</b> .....	145
Kapitola 8 <b>Další jednoduché příkazy</b> .....	157
<b>Část B Složené příkazy</b> .....	<b>167</b>
Kapitola 9 <b>Moduly</b> .....	168
Kapitola 10 <b>Vytvoření vlastního modulu</b> .....	185
Kapitola 11 <b>Balíčky</b> .....	206
Kapitola 12 <b>Definice funkcí</b> .....	221
Kapitola 13 <b>Parametry, argumenty a lokální proměnné funkcí</b> .....	234
Kapitola 14 <b>Pokročilé rysy funkcí</b> .....	255
Kapitola 15 <b>Rozhodování a příkaz with</b> .....	276
Kapitola 16 <b>Opakování kódu – rekurze a cykly</b> .....	289
Kapitola 17 <b>Tvorba aplikací</b> .....	309
Kapitola 18 <b>Ošetřování chyb</b> .....	323

---

<b>Část C Kontejnery</b>	<b>347</b>
Kapitola 19 Seznamy.....	348
Kapitola 20 N-tice .....	367
Kapitola 21 Množiny.....	382
Kapitola 22 Slovníky .....	395
Kapitola 23 <b>Rozšíření definic funkcí</b> .....	409
Kapitola 24 <b>Formátování stringů</b> .....	428
Kapitola 25 Operace s kontejnery.....	453
Kapitola 26 Práce se soubory .....	464
Literatura.....	487
<b>Rejstřík</b> .....	<b>489</b>

# Podrobný obsah

Úvod .....	21
Komu kniha není určena .....	21
Komu je kniha určena .....	22
Struktura příručky .....	22
Koncepce výkladu .....	24
Jazyk identifikátorů .....	25
Potřebné vybavení .....	25
Operační systém .....	25
Doprovodné programy a jejich organizace .....	25
Použité typografické konvence .....	27
Odbočka – podšeděný blok .....	29
Zpětná vazba .....	29
<b>Část A Superzáklady</b> .....	<b>31</b>
<b>Kapitola 1 Startujeme</b> .....	<b>32</b>
1.1 Hlavní součást instalace .....	32
1.1.1 Platforma .....	32
1.1.2 Dokumentace .....	33
1.1.3 PEP .....	34
1.1.4 Pracovní režimy .....	34
1.1.5 Překladač versus interpret .....	35
1.2 Vývojová prostředí .....	35
1.2.1 PyCharm a IntelliJ IDEA .....	35
1.2.2 Visual Studio Code .....	36
1.2.3 Jupyter Notebook a JupyterLab .....	36
1.2.4 Řádkový editor a IDLE .....	37
1.3 Spuštění Pythonu .....	37
1.3.1 Spuštění ve Windows .....	37
1.3.2 Spuštění v macOS a Linuxu .....	39
1.3.3 Co je nejdůležitější .....	39
1.3.4 Spuštění Pythonu v demonstračních programech v knize .....	39
1.4 Řádkový editor – REPL .....	39
1.4.1 Komentáře .....	40
1.4.2 Odsazování .....	41
1.4.3 Víceřádkové příkazy .....	41
Fyzické a logické řádky .....	42
1.4.4 Dvě verze řádkového editoru .....	42
1.5 Spuštění skriptu napsaného v Pythonu .....	48
1.5.1 Windows .....	49
1.6 Prostředí IDLE .....	49
1.6.1 Spuštění IDLE .....	49
1.6.2 Základní popis .....	50
1.6.3 Příkazové okno .....	51

1.6.4	Editační okno .....	53
1.6.5	Umístění editovaných souborů .....	54
1.6.6	Problémy při práci se schránkou .....	54
1.6.7	Barevné zvýraznění textu .....	55
1.6.8	Pokročilá nastavení .....	55
1.7	Shrnutí kapitoly a doprovodné soubory .....	56
<b>Kapitola 2</b>	<b>Zadávání jednoduchých hodnot .....</b>	<b>57</b>
2.1	Objekty a objektové programování .....	58
2.1.1	Koncepce Pythonu .....	58
2.1.2	Explicitně .....	59
2.1.3	Objekt, třída, instance, odkaz .....	59
2.1.4	Kontejnery .....	61
2.2	Datový typ .....	62
2.3	Nejdůležitější zvláštnosti Pythonu .....	63
	Přísné a benevolentní programovací jazyky .....	63
2.4	Zápis celých a desetinných (reálných) čísel .....	64
2.4.1	Zpřehlednění dlouhých čísel pomocí znaku podtržení .....	65
2.4.2	Rozlišení celých a reálných čísel .....	66
2.5	Komplexní čísla .....	67
2.6	Počáteční nula u celých čísel .....	68
2.7	Zadávání čísel v jiných číselných soustavách .....	68
2.8	Logické hodnoty – platí × neplatí .....	69
2.9	Nic – None .....	69
2.10	Výpustka – Ellipsis, ... .....	70
2.11	Objekt <code>NotImplemented</code> .....	70
2.12	Literály .....	70
2.13	Důležitost přehlednosti .....	71
2.14	Shrnutí kapitoly a doprovodné soubory .....	72
<b>Kapitola 3</b>	<b>Zadávání textů – stringů .....</b>	<b>73</b>
3.1	Stringové literály .....	73
3.1.1	Víceřádkové stringy .....	74
3.1.2	Použití funkce <code>print()</code> .....	76
3.2	Komentáře .....	76
3.3	Escape sekvence .....	77
3.3.1	Zobrazení escape-sekvencí funkcí <code>print()</code> .....	81
3.3.2	Zjištění názvu znaku .....	82
3.4	Slučování sousedních textových literálů .....	82
3.5	Prefixy stringových literálů .....	83
3.6	Stringová interpolace – f-stringy .....	85
3.6.1	Samodokumentující se výrazy .....	85
3.6.2	Shrnutí zásad pro práci s f-stringy .....	85
3.7	Bajtové objekty .....	87
3.7.1	Bajtové stringy a bajtové literály .....	87
3.7.2	Třída <code>bytearray</code> – zadávání bajtových polí .....	88
3.8	Shrnutí kapitoly a doprovodné soubory .....	89
<b>Kapitola 4</b>	<b>Volání funkcí .....</b>	<b>90</b>
4.1	Volání volatelného objektu .....	90
4.1.1	Parametr versus argument .....	91
4.1.2	Syntaxe volání a návratová hodnota .....	92
4.1.3	Pořadí vyhodnocování a předávání argumentů .....	92
4.1.4	Datové × funkční (volatelné) objekty .....	92
4.2	Vytváření objektů .....	93
4.3	Vestavěné funkce s jednoduchými argumenty .....	94

4.3.1	Zápis syntaxe.....	94
4.3.2	Anotace.....	95
4.3.3	Signatura funkce.....	96
4.3.4	Přehled vestavěných funkcí s argumenty jednoduchých typů.....	97
4.4	<b>Získání nápovědy</b> .....	104
4.4.1	Argument zadán.....	104
4.4.2	Bez argumentu.....	106
4.5	<b>Rozdělení volání na více řádků</b> .....	107
4.6	<b>Shrnutí kapitoly a doprovodné soubory</b> .....	107
<b>Kapitola 5</b>	<b>Jednoduché výrazy</b> .....	<b>109</b>
5.1	<b>Trocha teorie</b> .....	109
5.1.1	Výraz.....	109
5.1.2	Operace.....	110
5.1.3	Operand.....	110
5.1.4	Operátor.....	110
5.1.5	<b>Arita operátorů</b> .....	110
5.1.6	Operátory jako volatelné objekty – funkce.....	111
5.1.7	Priorita operátorů.....	111
5.1.8	Asociativita binárních operátorů.....	113
5.2	<b>Závorky, čárka jako operátor vytvoření n-tice</b> .....	114
5.2.1	Pozor na záměnu čárky s desetinnou tečkou.....	115
5.3	<b>Numerické operace</b> .....	116
5.3.1	Tři druhy dělení.....	116
5.3.2	Umocňování.....	117
5.3.3	Nekonečna a nesmyslná čísla.....	118
5.4	<b>Operace se stringy</b> .....	119
5.4.1	Sečítání textů.....	120
5.4.2	Násobení stringu celým číslem.....	120
5.4.3	Indexace jednotlivých znaků.....	121
5.5	<b>Shrnutí kapitoly a doprovodné soubory</b> .....	122
<b>Kapitola 6</b>	<b>Proměnné, výrazy, příkazy</b> .....	<b>123</b>
6.1	<b>Co jsou to proměnné</b> .....	123
6.1.1	Proměnná versus atribut – kvalifikace.....	124
6.2	<b>Správa paměti</b> .....	124
6.2.1	Statické a dynamické typování.....	125
6.3	<b>Pravidla pro tvorbu identifikátorů</b> .....	126
6.3.1	Klíčová slova tvrdá a měkká.....	127
6.3.2	Systémové identifikátory – dundery.....	127
6.4	<b>Zavedení proměnné – přiřazovací příkaz</b> .....	128
6.5	<b>Výrazy × příkazy, výrazové příkazy</b> .....	130
6.5.1	Výrazové příkazy.....	130
6.5.2	Zadání více příkazů na řádku – oddělující středník.....	130
6.5.3	Zadání skupiny hodnot.....	131
6.5.4	Proměnné inf a nan.....	133
6.5.5	Proměnná _.....	133
6.6	<b>Vnořená volání volatelných objektů</b> .....	134
6.7	<b>Zjištění typu objektu v proměnné</b> .....	135
6.8	<b>Uložení funkce do proměnné</b> .....	136
6.9	<b>Lambda-výrazy</b> .....	137
6.9.1	Využití při snižování počtu argumentů.....	138
6.10	<b>Datové a funkční proměnné</b> .....	139
6.11	<b>Mezery ve výrazech a příkazech</b> .....	139
6.12	<b>Uložení do proměnné × propojení s názvem</b> .....	139
6.13	<b>Přiřazovací výraz</b> .....	140
6.14	<b>Pomocné proměnné</b> .....	141

6.15 F-stringy – rozšiřující informace .....	141
6.15.1 Další pravidla.....	142
6.15.2 Formátování f-stringů.....	143
6.16 Shrnutí kapitoly a doprovodné soubory.....	144
<b>Kapitola 7 Logické hodnoty a operace.....</b>	<b>145</b>
7.1 Konstanty True a False .....	145
7.2 Převod jiných hodnot na logické .....	146
7.3 Logické operátory a operace.....	147
7.3.1 Zkrácené vyhodnocení.....	147
7.3.2 Pozor na priority .....	148
7.4 Porovnávání hodnot.....	149
7.4.1 Porovnání reálných čísel .....	149
7.4.2 Porovnávání a řazení textů – stringů.....	150
7.4.3 Zřetěžené porovnávání.....	150
7.4.4 Porovnávání totožnosti objektů – operátor is .....	151
7.5 Operace s jednotlivými bity.....	152
7.6 Bitové posuny.....	153
7.6.1 Aritmetický × logický posun.....	154
7.7 Podmíněný výraz.....	154
7.8 Shrnutí kapitoly a doprovodné soubory.....	155
<b>Kapitola 8 Další jednoduché příkazy .....</b>	<b>157</b>
8.1 Přiřazovací příkaz .....	157
8.2 Rozšířený přiřazovací příkaz .....	157
8.3 Příkaz del.....	158
8.4 Příkaz assert.....	159
Návrh podle kontraktu.....	160
8.5 Příkaz pass.....	162
8.6 Složené příkazy a odsazování .....	163
8.6.1 Výhody a nevýhody koncepce <i>Pythonu</i> .....	164
8.7 Restart systému.....	165
8.8 Shrnutí kapitoly a doprovodné soubory.....	165

## Část B Složené příkazy

167

<b>Kapitola 9 Moduly.....</b>	<b>168</b>
9.1 Základní principy modulů .....	168
9.1.1 Vše je součástí nějakého modulu .....	168
9.1.2 Dva názvy objektů.....	168
9.1.3 Zdrojový soubor.....	169
9.1.4 Přeložený soubor.....	169
9.2 Další trocha teorie OOP .....	170
9.2.1 Atributy .....	170
9.2.2 Práce s objekty – kvalifikace .....	170
9.2.3 Další podrobnosti o tvorbě objektů .....	171
Terminologická vsuvka: co je konstruktor.....	172
9.3 Příkaz <b>import</b> .....	173
9.3.1 Čistý import jiného modulu .....	173
9.3.2 Import modulu pod jiným názvem .....	175
9.3.3 Přímý import vyjmenovaných objektů.....	176
9.3.4 Import objektů modulu nezahrnuje import jejich modulu .....	177
9.3.5 Argumentem příkazů <b>import</b> a <b>from ... import</b> nesmí být výraz.....	178
9.3.6 Hromadný import všech atributů modulu – hvězdičkový import .....	178
9.3.7 Systémové identifikátory .....	180
9.3.8 Syntaktické diagramy příkazu <b>import</b> .....	180

9.4 Modul jako objekt .....	181
9.4.1 Modul <code>builtins</code> a zdánlivě neobjektové programování.....	182
9.5 Postup systému při importu modulu .....	182
9.6 Shrnutí kapitoly a doprovodné soubory.....	183
<b>Kapitola 10 Vytvoření vlastního modulu .....</b>	<b>185</b>
10.1 Demonstrační modul <code>m10a_Modul</code> .....	185
10.1.1 Kódová stránka.....	187
10.1.2 Dokumentační komentář a atribut <code>__doc__</code> .....	187
10.1.3 Kontrolní tisky načítání modulu.....	188
10.1.4 Definice datových atributů .....	188
10.1.5 Neveřejné atributy .....	188
10.1.6 Reakce na zadané příkazy .....	189
10.1.7 Alternativní zveřejnění atributů.....	189
10.2 Modul <code>dbg</code> a kontrolní tisky.....	190
10.2.1 Co modul <code>dbg</code> není .....	191
10.3 Průběh importu vytvořeného modulu.....	192
10.3.1 Použitelné názvy.....	193
10.4 Hromadný (hvězdičkový) import.....	194
10.5 Zveřejnění atributů pomocí <code>__all__</code> .....	196
10.6 Import jako přiřazovací příkaz – shrnutí.....	198
10.7 Reimport již importovaného modulu .....	199
10.7.1 Pozor na přímo importované proměnné .....	201
10.7.2 Specifika funkce <code>importlib.reload()</code> .....	203
10.7.3 Rozbor chybového hlášení.....	204
10.7.4 Důsledky chybného zavedení modulu .....	204
10.7.5 Syntaktické chyby.....	204
10.8 Shrnutí kapitoly a doprovodné soubory.....	205
<b>Kapitola 11 Balíčky.....</b>	<b>206</b>
11.1 Balíčky (packages).....	206
11.1.1 Initor standardního balíčku .....	207
11.1.2 Zobrazování začátku a konce inicializace balíčků a modulů .....	207
11.1.3 Uspořádání balíčků pro demonstraci.....	208
11.1.4 Demonstrace importu podbalíčků .....	210
11.2 Relativní import.....	212
11.2.1 Relativní import z rodičovských balíčků.....	212
11.2.2 Relativní import z podbalíčků .....	213
11.2.3 Demonstrace relativního importu .....	213
11.2.4 Cyklický import.....	215
11.3 Co dělat, když interpret na modul nevidí.....	215
11.3.1 Detekce a změna pracovní složky .....	216
11.4 Kde hledat moduly.....	217
11.4.1 Prohledávané složky.....	217
11.5 Přidání složky mezi prohledávané .....	218
11.6 Shrnutí kapitoly a doprovodné soubory.....	219
<b>Kapitola 12 Definice funkcí.....</b>	<b>221</b>
12.1 Pověšně o funkcích.....	221
12.1.1 Funkce je standardní objekt.....	222
12.2 Definice funkce je jen zvláštní přiřazovací příkaz .....	222
12.3 Definice vlastní funkce.....	222
12.3.1 Jednořádková definice .....	223
12.3.2 Víceřádková definice.....	224
12.3.3 Interaktivní režim versus zdrojový kód modulu.....	225
12.3.4 Prázdné funkce .....	226
12.4 Zadávání stringů zabírajících více řádků.....	227

12.4.1 Použití funkce <code>textwrap.dedent()</code> .....	229
12.5 Definice funkcí v modulu .....	230
12.6 Kdy se projeví chyby v definici funkce.....	231
12.7 Shrnutí kapitoly a doprovodné soubory.....	233
<b>Kapitola 13 Parametry, argumenty a lokální proměnné funkcí .....</b>	<b>234</b>
13.1 Parametry, argumenty a lokální proměnné .....	234
13.1.1 Definice .....	234
13.1.2 Lokální proměnné .....	235
13.1.3 Volání funkcí s parametry .....	236
13.1.4 Povinné pojmenované argumenty .....	237
13.1.5 Povinné poziční argumenty .....	238
13.1.6 Mix pozičních a pojmenovaných argumentů .....	239
13.2 Implicitní hodnoty argumentů .....	240
13.2.1 Ještě jednou porovnání reálných čísel.....	242
13.3 Konstantnost předdefinovaných hodnot .....	243
13.4 Funkce s vedlejším efektem .....	244
13.5 Funkce vracějící hodnotu a příkaz <code>return</code> .....	244
13.6 Přetěžování funkcí .....	245
13.6.1 Něco přetížít jde .....	246
13.7 Anotace.....	247
13.8 Dekorátory.....	248
13.9 Pomocné funkce a dekorátory z modulu <code>dbg</code> .....	249
13.9.1 Základní funkce modulu <code>dbg</code> .....	249
13.9.2 Přehled nabízených dekorátorů .....	250
13.10 Shrnutí kapitoly a doprovodné soubory.....	253
<b>Kapitola 14 Pokročilé rysy funkcí .....</b>	<b>255</b>
14.1 Vnitřní funkce.....	255
14.1.1 Odsazování.....	257
14.1.2 Přístup z okolního kódu .....	257
14.1.3 Import uvnitř funkce .....	258
14.2 IDLE a nastavení <b>Show Code Context</b> .....	258
14.3 Jmenné prostory.....	259
14.4 Oblast/rozsah platnosti, působnost (scope) .....	259
14.4.1 Zanoření oblastí platnosti .....	260
14.5 Lokalita použitých proměnných.....	261
14.5.1 Volná proměnná.....	261
14.5.2 Příkaz <code>global</code> .....	263
14.5.3 Příkaz <code>nonlocal</code> .....	264
14.6 Vnoření funkce versus vnoření volání funkcí .....	267
14.7 Zobrazování kontextu.....	267
14.8 Vnořená volání funkcí .....	268
14.9 Funkce vyššího řádu .....	269
14.10 Atributy funkcí .....	270
14.11 Nelokální proměnné a uzávěry (closures).....	271
14.12 Zákaz hvězdičkového importu uvnitř funkcí .....	272
14.13 Možné řešení cyklického importu .....	272
14.14 Shrnutí kapitoly a doprovodné soubory.....	274
<b>Kapitola 15 Rozhodování a příkaz <code>with</code> .....</b>	<b>276</b>
15.1 Rozhodovací příkazy .....	276
15.2 Jednoduchý podmíněný příkaz .....	277
15.3 Úplný podmíněný příkaz .....	277
15.4 Rozšířený podmíněný příkaz .....	279
15.5 Přepínač <code>match</code> .....	279

15.5.1	Trocha terminologie.....	280
15.5.2	Postup vyhodnocení.....	281
15.5.3	Sdružování hodnot ve vzorech.....	281
15.5.4	Klíčové slovo <code>_</code> je jen symbol.....	282
15.5.5	Další možnosti.....	282
15.6	<b>Představení knihovny <code>Karelcz7475</code></b> .....	283
15.6.1	Práce se světem – dvorkem.....	283
15.6.2	Zadatelné příkazy.....	284
15.6.3	Testovatelné podmínky.....	284
15.6.4	Skrytí aktuálního stavu robota.....	285
15.7	<b>Příkaz <code>with</code> a správce kontextu</b> .....	285
15.7.1	Pravidla používání příkazu <code>with</code> .....	286
15.7.2	Aplikace na příkladu s robotem.....	286
15.8	Shrnutí kapitoly a doprovodné soubory.....	288
<b>Kapitola 16 Opakování kódu – rekurze a cykly</b> .....		289
16.1	Rekurze.....	289
16.1.1	Nekonečná rekurze.....	289
16.1.2	Rádně ukončená rekurze.....	290
	Zásobník návratových adres – ZNA.....	291
16.1.3	Rekurzivní definice faktoriálu a dodatečná dekorace funkce.....	291
16.1.4	Nevhodnost použití rekurze pro hluboké volání.....	293
16.2	<b>Příkaz <code>while</code> – cyklus se vstupní podmínkou</b> .....	293
16.3	<b>Nekonečný cyklus</b> .....	294
16.4	<b>Příkaz <code>break</code> – cyklus s podmínkou uprostřed</b> .....	295
16.5	<b>Cyklus s ukončovací podmínkou</b> .....	296
16.6	<b>Přiřazení v hlavičce cyklu</b> .....	296
16.7	<b>Větev <code>else</code></b> .....	298
16.8	<b>Příkaz <code>continue</code></b> .....	299
16.9	<b>Účel a syntaxe cyklu <code>for</code></b> .....	300
16.10	<b>Vyjmenování hodnot parametru cyklu</b> .....	301
16.11	<b>Využití objektů typu <code>range</code></b> .....	302
16.11.1	Důvod použití větve <code>else</code> .....	302
16.12	<b>Větev <code>else</code></b> .....	303
16.13	<b>Použití indexů</b> .....	304
16.14	<b>Použití stringu jako zdroje</b> .....	305
16.15	<b>Využití objektů typu <code>enumerate</code></b> .....	305
16.16	<b>Vnořování cyklů</b> .....	306
16.17	<b>Postupné použití několika zdrojů</b> .....	306
16.18	Shrnutí kapitoly a doprovodné soubory.....	307
<b>Kapitola 17 Tvorba aplikací</b> .....		309
17.1	<b>Vytvoření aplikace či knihovny</b> .....	309
17.2	<b>Přímé spuštění versus <code>import</code> modulu</b> .....	309
17.2.1	Rozpoznání režimu, v němž byl modul spuštěn.....	310
17.2.2	Demonstrace rozpoznání způsobu spuštění.....	310
17.2.3	Spuštění modulu umístěného v balíčku.....	311
17.3	<b>Argumenty příkazového řádku</b> .....	312
17.4	<b>Rozsáhlejší aplikace</b> .....	315
17.4.1	Umístění kódu aplikace.....	316
17.4.2	Initiator balíčku a změny záhlaví modulů balíčku.....	316
17.4.3	Modul <code>io</code> pro ošetření vstupu.....	316
17.4.4	Modul <code>game</code> s jádrem hry.....	318
17.4.5	Řídící modul <code>main</code> .....	319
17.5	<b>Vytvoření spustitelné aplikace</b> .....	320

17.5.1 Soubor typu <code>pyz</code> .....	321
17.5.2 Vytváříme aplikaci „ručně“ .....	321
17.5.3 Alternativy .....	322
17.6 Shrnutí kapitoly a doprovodné soubory.....	322
<b>Kapitola 18 Ošetřování chyb .....</b>	<b>323</b>
18.1 Tři druhy chyb.....	323
18.1.1 Syntaktické chyby.....	323
18.1.2 Běhové chyby.....	324
18.1.3 Logické chyby.....	325
18.1.4 Varování.....	325
18.2 Chybové zprávy .....	325
18.2.1 Syntaktické chyby při práci v interaktivním režimu .....	326
18.2.2 Syntaktické chyby při překladu importovaného modulu.....	326
18.2.3 Běhové chyby.....	327
Odchytky v hlášeném prvotním zdroji .....	328
18.3 I chyby jsou objekty – výjimky .....	329
18.4 Rozdělení výjimek.....	330
18.4.1 Výjimky versus varování.....	331
18.5 Zachycení a ošetření výjimek .....	331
18.5.1 Více větví <code>except</code> .....	332
18.5.2 Větev <code>else</code> .....	333
18.5.3 Větev <code>finally</code> .....	333
18.5.4 Syntaktický diagram příkazu <code>try</code> .....	333
18.5.5 Příklad s kompletní verzí příkazu <code>try</code> .....	334
18.6 Praktický příklad .....	336
18.7 Zdánlivé záludnosti větve <code>finally</code> .....	336
18.8 Vyvolání výjimky.....	339
18.9 Problematika oddělených tisků .....	341
18.10 Příkaz <code>assert</code> .....	342
18.11 Jak vyvolat a zachytit varování .....	343
18.11.1 Vyvolání varovného hlášení .....	343
18.11.2 Zachycení varovného hlášení.....	344
18.11.3 Režimy zachytávání.....	345
18.12 Shrnutí kapitoly a doprovodné soubory.....	346

## Část C Kontejnery

347

<b>Kapitola 19 Seznamy.....</b>	<b>348</b>
19.1 Proměnné a neměnné objekty.....	348
19.2 Základní informace o seznamech .....	349
19.3 Vytváření seznamů .....	350
19.3.1 Použití literálu .....	350
19.3.2 Využití konstruktoru <code>list()</code> .....	351
19.3.3 Sčítání a násobení .....	352
19.3.4 Generátorová notace seznamů .....	353
19.3.5 Tovární funkce a funkce vracující seznam.....	355
19.4 Modifikace seznamů.....	356
19.4.1 Metody <code>append()</code> a <code>extend()</code> .....	356
19.4.2 Rizika práce s odkazy na proměnné objekty .....	357
19.4.3 Postupné budování seznamu .....	358
19.4.4 Přičítání jiných zdrojů.....	359
19.4.5 Indexace prvků seznamu .....	360
19.4.6 Metody pracující s indexy.....	361
19.4.7 Metody pracující s celým seznamem.....	362

19.5	Vícerozměrné seznamy .....	363
19.6	Souhrnný příklad .....	364
19.7	Anotace odkazující na seznamy .....	365
19.8	Shrnutí kapitoly a doprovodné soubory .....	366
<b>Kapitola 20</b>	<b>N-tice .....</b>	<b>367</b>
20.1	Základní informace o n-ticích .....	367
20.2	Vytváření n-tic .....	368
20.2.1	Vytváření n-tic pomocí literálů .....	368
20.2.2	Využití konstrukturu tuple() .....	369
20.2.3	Sčítání a násobení .....	370
20.2.4	Přičítání n-tic .....	371
20.2.5	Balení a rozbalování n-tic .....	372
20.2.6	Prohazování proměnných .....	372
20.2.7	Hvězdičkové pravidlo .....	373
20.3	Generátorová notace n-tic .....	373
20.4	Problematika neměnnosti n-tic .....	374
	Hešovatelné objekty .....	375
20.5	Přístup k prvkům n-tic .....	375
20.6	Sčítání seznamů a n-tic .....	376
20.7	Proměnné a neměnné prvky n-tice .....	377
20.8	Pojmenované n-tice .....	378
20.9	Anotace odkazující na n-tice .....	380
20.10	Shrnutí kapitoly a doprovodné soubory .....	381
<b>Kapitola 21</b>	<b>Množiny .....</b>	<b>382</b>
21.1	Základní informace o množinách .....	382
21.2	Vytváření množin .....	382
21.2.1	Vytváření množin pomocí literálů .....	383
21.2.2	Vytváření množin pomocí konstrukturu set() .....	383
21.2.3	Vytváření množin prostřednictvím množinových operací .....	385
21.3	Generátorová notace množin .....	388
21.4	Zmrazené množiny .....	388
21.5	Modifikace množin .....	389
21.5.1	Modifikace pracující s jedním prvkem .....	390
21.5.2	Množinové operátory a rozšířené operace .....	391
21.5.3	Porovnávání množin .....	393
21.6	Anotace odkazující na množiny .....	394
21.7	Shrnutí kapitoly a doprovodné soubory .....	394
<b>Kapitola 22</b>	<b>Slovníky .....</b>	<b>395</b>
22.1	Mapovací objekty a slovníky .....	395
22.2	Vytváření slovníků .....	396
22.2.1	Vytváření slovníků pomocí literálů .....	396
22.2.2	Vytváření slovníků pomocí konstrukturu dict() .....	397
22.2.3	Ekvivalence slovníků .....	398
22.2.4	Vytváření slovníků pomocí metody fromkeys() .....	398
22.3	Generátorová notace slovníků .....	399
22.4	Operace se slovníkem .....	401
22.4.1	Práce s hodnotami pomocí „indexace“ klíčem .....	401
22.4.2	Další metody pro práci s jednotlivými položkami .....	402
22.4.3	Modifikace slovníku daty ze zadaného zdroje .....	403
22.4.4	Slovník jako generátor .....	404
22.5	Pohledy .....	404
22.5.1	Pohledy jako zdroje dat .....	405
22.5.2	Operace s pohledy .....	406
22.6	Anotace odkazující na slovníky .....	407

22.7 Shrnutí kapitoly a doprovodné soubory.....	407
<b>Kapitola 23 Rozšíření definic funkcí .....</b>	<b>409</b>
23.1 Předávání argumentů odkazem a hodnotou .....	409
23.1.1 Dvě hodnoty odkazu.....	410
23.1.2 Proč jde o předávání hodnotou.....	410
23.2 Pomocná funkce <code>gr()</code> .....	412
23.3 Proměnný počet pozičních argumentů .....	412
23.3.1 Hvězdičkový parametr.....	413
23.3.2 Hvězdičkový argument.....	414
23.4 Proměnný počet pojmenovaných argumentů.....	416
23.4.1 Dvuhvězdičkový parametr .....	416
23.4.2 Dvuhvězdičkový argument .....	416
23.5 Stručný souhrn .....	417
23.5.1 Podivné chování .....	418
23.5.2 Použití v definicích literálů .....	418
23.6 Vestavěné funkce pracující s kontejnery .....	419
23.7 Funkce modulu <code>dbg</code> pracující s kontejnery .....	425
23.8 Shrnutí kapitoly a doprovodné soubory.....	426
<b>Kapitola 24 Formátování stringů.....</b>	<b>428</b>
24.1 Formátovací operátor <code>%</code> .....	428
24.2 Pokročilejší metody formátování .....	430
24.2.1 Metody <code>format()</code> a <code>format_map()</code> versus f-stringy .....	430
24.2.2 Formátovací string.....	431
24.2.3 Syntaxe nahrazovacího pole .....	432
24.2.4 Vyhodnocení nahrazovaného textu .....	432
24.2.5 Formátování nahrazujícího textu .....	434
24.2.6 Konverze.....	435
24.3 Specifikace formátu.....	436
24.3.1 Počet zabraných pozic .....	436
24.3.2 Přesnost.....	438
24.3.3 Typ hodnoty .....	438
24.3.4 Skupiny číslic .....	441
24.3.5 Alternativní formát a vedoucí nuly .....	442
24.3.6 Znaménko .....	443
24.3.7 Zarovnání a plnění .....	443
24.3.8 Vnořená nahrazovací pole ve specifikaci formátu .....	445
24.3.9 Formátování samodokumentujících se nahrazovacích polí.....	446
24.4 Příklad: Pascalův trojúhelník.....	446
24.5 Formátování instancí dalších typů.....	447
24.6 Šablony stringů.....	448
24.6.1 Instanční metody třídy <code>string.Template</code> .....	449
24.6.2 Ukázka použití stringových šablon.....	451
24.7 T-stringy jako šablony stringů .....	451
24.8 Shrnutí kapitoly a doprovodné soubory.....	452
<b>Kapitola 25 Operace s kontejnery.....</b>	<b>453</b>
25.1 Přítomnost prvku v kontejneru.....	453
25.2 Proměnné objekty jako implicitní hodnoty parametrů .....	454
25.3 Kopírování.....	454
Mělké a hluboké kopie objektů .....	454
25.3.1 Zdánlivé kopie neměnných kontejnerů .....	455
25.3.2 Alternativní způsob tvorby mělkých kopií.....	456
25.3.3 Nebezpečí hlubokých kopií.....	456
25.4 Rozdělení doposud probraných kontejnerů .....	456
25.5 Řazení prvků posloupnosti.....	457

25.5.1	<code>reversed(seq)</code> .....	458
25.5.2	<code>sorted(iterable, *, key=None, reverse=False)</code> .....	458
25.6	Vykrajování (slicing) .....	459
25.7	Indexování a vykrajování u rozsahů .....	460
25.8	Nahrazování hodnot .....	461
25.9	„Úprava“ neměnných objektů .....	463
25.10	Shrnutí kapitoly a doprovodné soubory .....	463
<b>Kapitola 26</b>	<b>Práce se soubory</b> .....	<b>464</b>
26.1	Soubory: bleskové opakování .....	464
26.1.1	Soubor, souborový systém, cesta .....	465
26.1.2	Absolutní a relativní cesta .....	466
26.1.3	Substituované disky ve Windows .....	466
26.2	Koncepce vstupu a výstupu dat v <i>Pythonu</i> .....	466
26.2.1	Starší koncepce souborů v jazycích C nebo Pascal .....	467
26.2.2	Novější koncepce datových proudů .....	467
26.2.3	Koncepce Pythonu .....	467
26.2.4	Shrnutí používané terminologie .....	468
26.3	Dva způsoby práce se souborovým systémem .....	469
26.4	Moduly <code>os</code> a <code>os.path</code> .....	471
26.5	Pracovní složka .....	472
26.6	Skládání a rozkládání cest .....	473
26.7	Vytváření a mazání složek .....	474
26.7.1	Mazání .....	475
26.8	Získání informací o souborech .....	476
26.9	Zápis a čtení dat .....	477
26.9.1	Problematika kódování ve Windows .....	477
26.9.2	Otevírání souborů a datových proudů .....	478
26.9.3	Zápis dat, splachování a zavírání proudů a přidružených souborů .....	480
26.10	Automatické zavírání proudů použitím <code>with</code> .....	482
26.11	Čtení ze souborů .....	483
26.12	Shrnutí kapitoly a doprovodné soubory .....	485
Literatura	.....	487
<b>Rejstřík</b>	.....	<b>489</b>

# O autorovi

**Ing. Rudolf Pecinovský, CSc.** studoval teoretickou kybernetiku na *Fakultě jaderné a fyzikálně inženýrské ČVUT* a poté technickou kybernetiku na *Fakultě Elektrotechnické ČVUT*, kterou ukončil v roce 1979. Titul CSc. získal v Ústavu teorie informace a automatizace ČSAV v roce 1983. Od počátku 80. let učí a publikuje, přičemž svůj výzkum soustředí především na oblast vstupních kurzů moderního programování a výuku objektivě orientovaného paradigmatu. V současné době učí na *Fakultě jaderné a fyzikálně inženýrské ČVUT* a na *Fakultě informatiky a statistiky Vysoké školy ekonomické v Praze*. Vedle toho učí ještě v řadě kurzů pro začátečníky i profesionální programátory. Doposud mu vyšlo přes 70 knih, které byly přeloženy do pěti jazyků. Většina jeho knih je zaměřena na výuku moderního programování a návrh objektivě orientované architektury.



# Poděkování

Vím, že se v českých knížkách většinou neděkuje, ale tvorba knih je spojena s takovými oběťmi řady lidí z mého blízkého i vzdálenějšího okolí, že bych měl velkou újmu na duši, kdybych tak neučinil.

Chtěl bych především nesmírně poděkovat své ženě Jarušce, která byla po celou dobu mojí největší oporou a jejíž nekonečná trpělivost a vstřícnost mi pomohla dokončit knihu v termínu, který se příliš nelišil od toho, jež jsme původně s nakladatelem dohodli, a ne až někdy za rok po něm. Stále marně přemýšlím, kde má schovanou tu svatozář.

Původně jsem se domníval, že s dalším vydáním nebude moc práce. Šeredně jsem se zmýlil, protože veškeré úpravy, jež jsem do knihy zanesl, jsou vykoupeny hodinami studia a experimentování, které rodina s neuvěřitelnou trpělivostí snášela.

Na vylepšování textu nového vydání se podílela řada dalších lidí. Mezi nimi musím poděkovat především těm, kteří si dali práci s odhalováním případných chyb ve vznikajícím rukopisu. Mezi nimi pak především Luďkovi Šťastnému, který po celou přípravu rukopis pročetl a odhaloval v něm pasáže, jež by si zasloužily vylepšit. Neméně velkou zásluhu na současné podobě má i Jirka Kofránek, který mne průběžně upozorňoval na některé problémy s výukou podle běžně používaných postupů. Řadu podnětných myšlenek přinesl i Michal Palas, jenž pracuje v oblasti analýzy a zpracování dat a přispěl tak řadou poznatků z praxe.

V neposlední řadě patří můj dík redakci, především redaktoru Petru Somogyimu, který trpělivě snášel mé neustálé modifikace již zkorigovaného textu, a Radku Matulíkovi, který mne k napsání jednotlivých knih z posledních let vyhecovoal a byl pak ochoten týden či dva počkat, když se mi nepodařilo přesně dodržet původně dohodnutý termín.

# Úvod

*Python* je moderní programovací jazyk, který umožňuje velmi jednoduše navrhovat jednoduché programy, ale na druhou stranu nabízí mocné prostředky k tomu, abyste mohli s přiměřeným úsilím navrhovat i programy poměrně rozsáhlé. Je pro něj vyvinuto obrovské množství knihoven, které uživatelům umožňují soustředit se na řešený úkol a nerozptylovat se vývojem nejrůznějších pomocných podprogramů.

*Python* je v současné době nejlepším jazykem pro ty, kteří se nechtějí živit jako programátoři, ale jejich profese či zájem je nutí jednou za čas něco naprogramovat. Potřebují proto jazyk, který se mohou rychle naučit a v němž budou moci rychle vytvářet jednoduché programy řešící jejich problémy nebo alespoň pomáhající při jejich řešení.

*Python* je vynikajícím nástrojem i pro ty, kteří vyvíjejí rozsáhlejší programy a ocení jeho obrovské možnosti, které běžné jazyky neposkytují. Oblíbí si ho všichni, kteří nepotřebují pracovat v jazyku plném pravidel zabraňujících častým chybám začínajících programátorů a poskytujícím nepřetržitý dozor přísného překladače kontrolujícího dodržování těchto pravidel.

Pro takový druh uživatelů je určena tato příručka. Určitě ji však přivítají i čtenáři, kteří řeší složitější problémy a potřebují proto znát jazyk do větší hloubky. Najdou zde výklad leckterých konstrukcí, na něž v běžných učebnicích nezbylo místo.

## Komu kniha není určena

Dopředu musím upozornit, že tato kniha není koncipována pro naprosté začátečníky, kteří se s programováním teprve seznamují. Ti potřebují poněkud jiný postup výkladu a jiný výběr příkladů. Pro ně je určena kniha *Začínáme programovat v jazyku Python* ([15]), která vedle používání základních konstrukcí učí své čtenáře také řadu zásad moderního programování, jejichž zvládnutí je nutnou podmínkou pro všechny, kteří nehodlají zůstat u malých žákovských programů, ale chtějí se naučit efektivně vyvíjet robustní středně rozsáhlé aplikace, jejichž údržba nepovede uživatele k chrlení nepublikovatelných výroků na adresu jejich autorů.

Kniha, kterou právě čtete, předpokládá alespoň minimální programátorské znalosti a zkušenosti, což jí umožní soustředit se na výklad konstrukcí a rysů jazyka včetně těch, na které v učebnicích pro začátečníky již obvykle nezbyvá místo.

## Komu je kniha určena

Je určena pro tři typy uživatelů:

- Autory, jejichž programátorské zkušenosti nejsou příliš hluboké, protože je využívají pro tvorbu jednoduchých programů řešících jejich každodenní problémy. Pro ty jsem výklad konstrukcí jazyka občas proložil výklady základů programování, které mohou ti pokročilejší přeskočit.

U těchto čtenářů bych byl rád, kdyby jim kniha pomohla v postupném prohlubování jejich znalostí a dovedností. Narazí-li někde na obtížnější pasáž, jejíž znalost právě nepotřebují, mohou ji klidně přeskočit a vrátit se k ní, až se jejich zkušenosti prohloubí a budou potřebovat probíranou vlastnost použít.

- Programátory, kteří doposud vyvíjeli v jiném programovacím jazyce a chtějí rozšířit své dovednosti o programování v jazyce *Python*. Pro ty jsou určena různá upozornění na rysy, jimiž se *Python* může odlišovat od toho, na co jsou ze své dosavadní praxe zvyklí.

Z ohlasů na předchozí verze publikace odhaduji, že těchto čtenářů je 60 až 80 %, takže se jim budu snažit vyjít vstříc a vždy včas upozornit na potřebu odlišného přístupu k probírané konstrukci při návrhu programu.

Předpokládám navíc, že tyto čtenáři budou chtít tuto knihu používat spíše jako referenční příručku pro situace, kdy si nebudou zcela jisti, jak se některá konstrukce používá nebo jak přesně funguje.

- Uživatelé, kteří již v *Pythonu* programují, ale potřebují znát jazyk do větší hloubky, než jim poskytly absolvované kurzy a prostudované učebnice, nebo se potřebují seznámit s novými konstrukcemi a rysy, které se mezi tím v *Pythonu* objevily.

K dokonalému využití platformy *Python* je však potřeba i znalost základních knihoven. V této příručce vám představím pouze ty nejzákladnější funkce a třídy. Těm, kteří touží po podrobnějším seznámení s knihovnou a jejími možnostmi v oblasti práce s daty, poslouží publikace [\[13\]](#).

## Struktura příručky

Tato kniha je prvním dílem dvoudílné příručky. V tomto dílu se čtenář seznámí se vestavěnými jednoduchými datovými typy a základními algoritmickými konstrukcemi jazyka. Dozví se základní informace o práci s objekty a naučí se pracovat s nejdůležitějšími strukturovanými datovými typy.

Druhý díl nazvaný [Python 3.14 – Objektové konstrukce](#) je pak zaměřen na konstrukce podporující objektově orientované programování a na hlubší pochopení některých pokročilých programových konstrukcí.

Příručka se nesnaží probrat rozsáhlou knihovnu, která je součástí standardní konfigurace, nicméně většinu jejích funkcí typický uživatel nevyužije. Seznamuje proto

opravdu jen s jejími nejdůležitějšími součástmi, které použije prakticky každý. O použití některých nadstavbových součástí vyšly nebo vyjdou samostatné publikace, přičemž spektrum těchto navazujících učebnic mohou ovlivňovat sami čtenáři.

Tento přístup umožňuje probrat jazyk do větší hloubky, takže se dostane i na oblasti, které běžné učebnice opomíjejí a očekávají, že si některá témata čtenáři osvojí metodou pokus-omyl.

### **První díl**

První díl se věnuje především algoritmickým konstrukcím a základním datovým strukturám. Je rozdělen do tří částí doprovázených samostatně distribuovanými přílohami:

- První část seznamuje s platformou Python a s vývojovými prostředími, která používám pro definici ukázkových příkladů a demonstraci jejich funkce. Poté vám ukáže, jak zadávat hodnoty jednoduchých datových typů, seznámí vás s jejich použitím ve výrazech a s jednoduchými příkazy, které s takovými daty pracují. Jejich zvládnutí je podmínkou pro studium dalších částí.
- Druhá část probírá složené příkazy. Postupně probereme definice modulů, funkcí a řady složených příkazů. Po jejím zvládnutí budete schopni vytvářet jednoduché programky, i když se bude jednat spíše o programky na hraní.
- Třetí část rozšiřuje množinu používaných dat o kontejnery, což jsou objekty určené k uchování jiných objektů. Snažil jsem se v ní uvést i pár příkladů, které již nejsou pouze demonstrační, a měly by mít i praktické využití. Po jejím zvládnutí byste měli být schopni vytvářet v *Pythonu* užitečné programy, které budou moci používat jiní.

### **Druhý díl**

Druhý díl se soustředí na konstrukce pro podporu objektově orientovaného programování, které rozebírá do nadstandardní hloubky. Je rozdělený do čtyř částí.

- První část se zaměřuje na výklad základních konstrukcí umožňujících a podporujících objektově orientované programování. Po jejím zvládnutí budete schopni vytvářet v *Pythonu* středně složité programy.
- Druhá část probírá tyto konstrukce do větší hloubky a ukazuje, jak definovat ekvivalenty některých konstrukcí a datových struktur, které se do té doby v příručce jenom používaly.
- Třetí část se pak věnuje pokročilejším objektovým konstrukcím, jejichž znalost umožňuje navrhovat efektivnější architekturu vytvářených programů. Podrobně seznamuje s náročnými tématy jako přetěžování operátorů, správa paměti a použití metatříd.
- Čtvrtá část se pak soustředí na podporu souběžného programování, při němž počítač provádí několik úloh současně. Po stručném seznámení s problematikou

pak postupně představí a na praktických příkladech ukáže nejprve návrh programů pracujících s více vlákny, poté návrh programů pracujících s více procesy, poté představí možnosti asynchronního programování a na závěr seznámí s novinkou verze 3.14, kterou je možnost vytváření subinterpretů.

Samostatnou část knihy tvoří přílohy. Ty ale nejsou součástí tištěného vydání – čtenáři papírových knih je najdou webových stránkách příručky. V elektronických vydáních je ale najdete.

## Koncepce výkladu

Snažil jsem se, aby tato kniha mohla sloužit současně jako učebnice jazyka *Python* i jako referenční příručka. Tyto dva druhy publikací si ale ve svých požadavcích poněkud odporují.

- Dobrá učebnice vyžaduje, aby se výklad obešel bez dopředných odkazů. Musí proto maximálně omezit (nejlépe zcela zrušit) používání čehokoliv, co ještě nebylo vysvětleno, a to i za cenu toho, že výklad mnoha témat bude třeba rozdělit do několika částí, aby v něm mohly být vždy použity jen doposud probrané rysy jazyka.

V opačném případě čtenáři vnímají danou konstrukci jako obrázek, který okopírují a používají, aniž by znali význam jeho jednotlivých součástí. (To je ostatně nectnost řady učebnic začínajících oblíbeným příkladem „*Hello World*“.) Při takovémto přístupu totiž čtenáři občas nechápou důsledky některých obrátů a akcí a příčiny ohlášených chyb, jejichž odstranění je pak stojí nemalé úsilí.

- Naproti tomu referenční příručky by měly umožňovat co nejsnazší a nejrychlejší dohledání potřebných informací. Mohou být proto koncipovány tak, že každé téma je v nich plně probráno na jednom místě se všemi detaily, a to i za cenu dopředných odkazů, protože při výkladu často potřebují použít konstrukci, která teprve bude vysvětlena. Mohou si to dovolit, protože jsou určeny především zkušenějším čtenářům, u nichž se předpokládají základní znalosti a danou referenční příručku používají většinou k tomu, aby si ozřejmili některé detaily.

Jak jsem řekl, chtěl jsem, aby tato kniha mohla sloužit k oběma účelům, tedy jak jako základní příručka, podle které se dá učit, tak jako referenční příručka, kde lze rychle nalézt pasáž zabývající se diskutovaným tématem. Její struktura je proto navržena tak, aby jednotlivé kapitoly probraly dané téma pokud možno komplexně a bez potřeby se k němu vracet, ale aby se v nich na druhou stranu pokud možno neobjevovaly dopředné odkazy i za cenu toho, že bude třeba výklad některých pasáží rozdělit.

Veškerý výklad je prostoupen hojnými AHA-příklady, tedy příklady, jejichž hlavním účelem není vytvoření nějakého zdánlivě užitečného programu, ale kódu, jehož cílem je co nejjednodušeji a co nejsrozumitelněji demonstrovat probíranou konstrukci, aby čtenář pochopil její princip – aby u něj nastal kýžený AHA-efekt.

Vedle nich se občas objeví i nějaký příklad, který by byl v praxi užitečný. Protože se ale nejedná o učebnici programování, ale především o učebnici jazyka, bude těchto praktických příkladů poskrovnu.

## Jazyk identifikátorů

Jak jsem řekl, doprovodné programy v první části jsou převážně AHA-příklady vysvětlující probíranou konstrukci. V nich budu pro větší názornost používat české identifikátory. V příkladech, které zavánějí praktickou aplikovatelností, dám – jak bývá v programování zvykem – přednost identifikátorům anglickým.

## Potřebné vybavení

Pro úspěšné studium této knihy je vhodné mít instalovanou platformu *Python*. Tu lze stáhnout na adrese <https://www.python.org/downloads/>. Na dané adrese najdete i stručný návod, jak *Python* stáhnout a instalovat.

Kniha je psána pro verzi 3.14, ale drtivá většina příkladů poběží i na verzi 3.10. Protože ale vím, že řada čtenářů pracuje se staršími verzemi (např. proto, že to jejich zaměstnavatel požaduje z důvodu kompatibility), snažím se upozorňovat na všechna vylepšení, která se objevila v novějších verzích počínaje verzí 3.5, abyste byli včas upozorněni na to, co by vám nemuselo na počítači chodit bez vašeho zavinění.

## Operační systém

Při práci v *Pythonu* vám může být většinou zcela jedno, nad jakým operačním systémem je instalovaný. V některých výjimečných situacích na tom ale záleží. Protože asi 85 % mých studentů i účastníků mých kurzů používá operační systém *Windows*, jsou mé knihy primárně zaměřeny na ně. Pokusím se ale nezapomínat ani na ten zbytek.

## Doprovodné programy a jejich organizace

Text knihy je prostoupen řadou doprovodných programů. Budete-li si je chtít spustit a ověřit jejich funkci, potřebujete je nejprve stáhnout. Najdete je na stránce knihy na adrese<sup>1</sup> [http://knihy.pecinovsky.cz/74\\_Python314bas](http://knihy.pecinovsky.cz/74_Python314bas).

---

<sup>1</sup> Číslem 74 na počátku poslední složky se nevzrušujte, je to pouze moje interní označení pořadí vytvářené knihy, protože bych v nich jinak bloudil. V textu příručky se s ním setkáte ještě mnohokrát.

Soubory s doprovodnými programy jsou uloženy v ZIP-souboru, jehož název začíná `74_Python_314bas_PGM_`. Za tímto začátkem následuje číslo verze a datum vytvoření. Když tento ZIP-soubor rozbalíte, vytvoří se následující složky:

- 74\_APP** Složka určená pro samostatné aplikace, které budeme v průběhu výkladu vytvářet.
- 74\_ILB** Složka se spustitelným souborem `Karelcz7475.pyz` obsahujícím knihovnu, se kterou budeme v některých kapitolách pracovat.
- 74\_ILS** Složka se zdrojovými kódy knihovny `Karel7475`
- 74\_INP** Složka s doprovodnými programy. K její vnitřní struktuře se za chvíli ještě vrátím.
- 74\_IWD** Texty výpisů programů v příručce vysazené spolu s čísly řádků, abyste si mohli daný soubor zobrazit vedle knihy např. v textovém editoru a při pročítání popisu daného programu nemuseli listovat v knize, abyste zjistili, jak vypadá řádek s daným číslem, o němž se v textu hovoří. Soubory odpovídají stejnojmenným souborům ze složky `74_INP/lessons`.

Pojďme se nyní podívat na vnitřní strukturu složky `74_INP`. Složka představuje pro doprovodné programy kořenový balíček. V něm je klíčový modul v souboru `dbg.py`, který používám při ladění a k názornému zobrazení průběhu některých procesů. kromě něj zde naleznete pár dalších souborů s příponou `py`, které potřebují být z nej-různějších důvodů v kořenové složce.

Kromě toho jsou tu tři podsložky představující balíčky (o balíčcích budeme hovořit v kapitole [11 Balíčky](#) na straně ). Každý z nich je schránkou na jeden druh souborů:

- lessons** Balíček (složka) s doprovodnými programy obsahujícími příkazy zadávané v průběhu kapitol většinou řádkovému interpretu označovanému REPL, občas ale operačnímu systému. Zadání příkazu i odpověď systému jsou pak zobrazeny ve výpisech v knize. Názvy těchto souborů začínají vždy písmenem `m` následovaným dvojmístným číslem kapitoly, dvěma podtržítky a textem naznačujícím téma kapitoly – např. `m01__Startujeme.py`.
- modules** Balíček se samostatnými moduly, které se vztahují ke konkrétním kapitolám. Názvy zde umístěných souborů začínají vždy písmenem `m` následovaným dvojmístným číslem kapitoly, písmenem určujícím pořadí vzniku souboru v průběhu kapitoly, podtržítkem a textem naznačujícím obsah souboru – například `m10a_Modul.py`.
- util** Balíček s užitečnými programy, které se občas použijí.
- DATA** Pomocná složka, kterou budeme používat, až budeme v závěrečné kapitole probírat práci se soubory.

Stažený ZIP-soubor si rozbalte, kam uznáte za vhodné. Jak napovídají adresy souborů v jejich úvodních komentářích, já mám celý ZIP rozbalený do kořenové složky substitovaného disku `R:`, takže složka `R:/74_INP` je kořenovou složkou doprovodných programů.

Vážení čtenáři, právě jste dočetli ukázkou z knihy ***Python 3.14 - Algoritmické konstrukce***.  
Pokud se Vám ukáзка líbila, na našem webu si můžete zakoupit celou knihu.