

Spis treści

Przedmowa	11
Rozdział 1. Podstawy Pythona	13
1.1. Uruchamianie Pythona	13
1.2. Programy Pythona	14
1.3. Prymitywy, zmienne i wyrażenia	15
1.4. Operatory arytmetyczne	16
1.5. Warunki i sterowanie przepływem programu	19
1.6. Ciągi tekstowe	20
1.7. Operacje na plikach	23
1.8. Listy	24
1.9. Krotki	26
1.10. Zbiory	28
1.11. Słowniki	29
1.12. Iteracja i pętle	31
1.13. Funkcje	32
1.14. Wyjątki	34
1.15. Zakończenie programu	35
1.16. Obiekty i klasy	36
1.17. Moduły	39
1.18. Pisanie skryptów	41
1.19. Pakiety	42
1.20. Strukturyzacja aplikacji	43
1.21. Zarządzanie pakietami stron trzecich	44
1.22. Python pasuje do Twojego mózgu	45

Rozdział 2. Operatory, wyrażenia i manipulacja danymi	46
2.1. Literały	46
2.2. Wyrażenia i lokalizacje	47
2.3. Standardowe operatory	48
2.4. Modyfikacje w miejscu	49
2.5. Porównywanie obiektów	50
2.6. Operatory porównania porządkowego	51
2.7. Wyrażenia logiczne i wartości prawdziwe	52
2.8. Wyrażenia warunkowe	53
2.9. Operacje obejmujące elementy iterowalne	53
2.10. Operacje na sekwencjach	55
2.11. Operacje na mutowalnych obiektach sekwencyjnych	57
2.12. Operacje na zbiorach	58
2.13. Operacje na mapowaniach	59
2.14. Lista, zbiór i słownik	60
2.15. Wyrażenia generujące	62
2.16. Operator atrybutu (.)	63
2.17. Operator wywołania funkcji ()	63
2.18. Kolejność liczenia	64
2.19. Podsumowanie: sekretne życie danych	65
Rozdział 3. Struktura i kontrola przepływu programu	66
3.1. Struktura i wykonanie programu	66
3.2. Wykonanie warunkowe	67
3.3. Pętle i iteracje	67
3.4. Wyjątki	70
3.4.1. Hierarchia wyjątków	73
3.4.2. Wyjątki i kontrola przepływu	75
3.4.3. Definiowanie nowych wyjątków	75
3.4.4. Powiązane wyjątki	76
3.4.5. Śledzenie wyjątków	78
3.4.6. Wskazówka dotycząca obsługi wyjątków	79
3.5. Menedżery kontekstu i instrukcja with	80
3.6. Asercje i __debug__	82
3.7. Podsumowanie	83
Rozdział 4. Obiekty, typy i protokoły	85
4.1. Podstawowe pojęcia	85
4.2. Tożsamość i typ obiektu	86
4.3. Zliczanie referencji i odśmiecanie pamięci	87

4.4. Referencje i kopie	89
4.5. Reprezentacja obiektu i wyświetlanie	90
4.6. Obiekty pierwszoklasowe	91
4.7. Używanie wartości None dla opcjonalnych lub brakujących danych	92
4.8. Protokoły obiektu i abstrakcja danych	93
4.9. Protokół obiektu	94
4.10. Protokół liczbowy	95
4.11. Protokół porównania	98
4.12. Protokoły konwersji	99
4.13. Protokół kontenera	100
4.14. Protokół iteracji	102
4.15. Protokół atrybutów	103
4.16. Protokół funkcji	103
4.17. Protokół menedżera kontekstu	104
4.18. Podsumowanie: pythoniczność	104

Rozdział 5. Funkcje106

5.1. Definicje funkcji	106
5.2. Argumenty domyślne	106
5.3. Argumenty wariadyczne (zmienna liczba argumentów)	107
5.4. Argumenty słów kluczowych	108
5.5. Wariadyczne argumenty słów kluczowych	109
5.6. Funkcje akceptujące wszystkie dane wejściowe	109
5.7. Argumenty tylko pozycyjne	110
5.8. Nazwy, wpisy dokumentacyjne i wskazówki dotyczące typów	111
5.9. Zastosowanie funkcji i przekazywanie parametrów	112
5.10. Zwracane wartości	113
5.11. Obsługa błędów	114
5.12. Zasady określania zakresu	115
5.13. Rekurencja	118
5.14. Wyrażenie lambda	118
5.15. Funkcje wyższego rzędu	119
5.16. Przekazywanie argumentów w funkcjach zwrotnych	121
5.17. Zwracanie wyników z wywołań zwrotnych	125
5.18. Dekoratory	127
5.19. Funkcje map, filter i reduce	130
5.20. Przegląd funkcji, atrybutów i sygnatur	131
5.21. Inspekcja środowiska	133
5.22. Dynamiczne wykonywanie i tworzenie kodu	135

5.23. Funkcje asynchroniczne i await	136
5.24. Podsumowanie: przemyślenia na temat funkcji i kompozycji	139
Rozdział 6. Generatory	140
6.1. Generatory i yield	140
6.2. Generatory z możliwością ponownego uruchomienia	143
6.3. Delegowanie generatora	143
6.4. Używanie generatorów w praktyce	144
6.5. Ulepszone generatory i wyrażenia yield	147
6.6. Zastosowania ulepszonych generatorów	148
6.7. Generatory i obsługa await	151
6.8. Podsumowanie: krótka historia generatorów i patrzenie w przyszłość	152
Rozdział 7. Klasy i programowanie obiektowe	153
7.1. Obiekty	153
7.2. Wyrażenie class	154
7.3. Instancje	155
7.4. Dostęp do atrybutów	156
7.5. Zasady ustalania zakresu	157
7.6. Przeciążanie operatora i protokoły	158
7.7. Dziedziczenie	159
7.8. Unikanie dziedziczenia poprzez kompozycję	162
7.9. Unikanie dziedziczenia poprzez funkcje	164
7.10. Wiązanie dynamiczne i technika kaczego typowania	165
7.11. Niebezpieczeństwo dziedziczenia po typach wbudowanych	166
7.12. Zmienne i metody klasy	167
7.13. Metody statyczne	170
7.14. Słowo na temat wzorców projektowych	173
7.15. Enkapsulacja danych i atrybuty prywatne	174
7.16. Wskazówka typu	176
7.17. Właściwości	177
7.18. Typy, interfejsy i klasy abstrakcyjne	180
7.19. Wielokrotne dziedziczenie, interfejsy i domieszki	183
7.20. Dyspozycja oparta na typie	188
7.21. Dekoratory klas	189
7.22. Nadzorowane dziedziczenie	192
7.23. Cykl życia obiektu i zarządzanie pamięcią	193
7.24. Słabe referencje	197
7.25. Wewnętrzna reprezentacja obiektów i wiązanie atrybutu	199
7.26. Proxy, wrapper i delegacje	201

7.27. Zmniejszenie wykorzystania pamięci za pomocą __slots__	203
7.28. Deskryptory	204
7.29. Proces definicji klasy	207
7.30. Dynamiczne tworzenie klas	208
7.31. Metaklasy	209
7.32. Obiekty wbudowane dla instancji i klas	213
7.33. Podsumowanie: zachowaj prostotę	214
Rozdział 8. Moduły i pakiety	215
8.1. Moduły i wyrażenie import	215
8.2. Buforowanie modułów	217
8.3. Importowanie wybranych nazw z modułu	218
8.4. Importy cykliczne	220
8.5. Ponowne ładowanie i zwolnienie modułu	221
8.6. Kompilacja modułów	222
8.7. Ścieżka wyszukiwania modułów	223
8.8. Wykonanie jako program główny	224
8.9. Pakiety	225
8.10. Import wewnątrz pakietu	226
8.11. Uruchamianie podmodułu pakietu jako skryptu	227
8.12. Kontrolowanie przestrzeni nazw pakietu	228
8.13. Kontrolowanie eksportu pakietów	229
8.14. Dane pakietu	230
8.15. Obiekty modułu	231
8.16. Wdrażanie pakietów Pythona	232
8.17. Przedostatnie słowo: zacznij od pakietu	233
8.18. Podsumowanie: zachowaj prostotę	234
Rozdział 9. Obsługa operacji wejścia-wyjścia	235
9.1. Reprezentacja danych	235
9.2. Kodowanie i dekodowanie tekstu	236
9.3. Formatowanie tekstu i bajtów	238
9.4. Czytanie opcji wiersza poleceń	241
9.5. Zmienne środowiskowe	243
9.6. Pliki i obiekty plików	243
9.6.1. Nazwy plików	244
9.6.2. Tryby plików	245
9.6.3. Buforowanie operacji wejścia-wyjścia	245
9.6.4. Kodowanie w trybie tekstowym	246
9.6.5. Obsługa wiersza w trybie tekstowym	247

9.7. Warstwy abstrakcyjne wejścia-wyjścia	247
9.7.1. Metody plików	248
9.8. Standardowe wejście, wyjście i błąd	250
9.9. Katalogi	251
9.10. Funkcja print()	252
9.11. Generowanie wyjścia	252
9.12. Pobieranie danych wejściowych	253
9.13. Serializacja obiektów	254
9.14. Operacje blokujące i współbieżność	255
9.14.1. Nieblokujące operacje wejścia-wyjścia	256
9.14.2. Odpytywanie operacji wejścia-wyjścia	257
9.14.3. Wątki	257
9.14.4. Równoczesne wykonywanie z asyncio	258
9.15. Standardowe moduły biblioteczne	259
9.15.1. Moduł asyncio	259
9.15.2. Moduł binascii	260
9.15.3. Moduł cgi	261
9.15.4. Moduł configparser	261
9.15.5. Moduł csv	262
9.15.6. Moduł errno	263
9.15.7. Moduł fcntl	263
9.15.8. Moduł hashlib	264
9.15.9. Pakiet http	264
9.15.10. Moduł io	265
9.15.11. Moduł json	265
9.15.12. Moduł logging	266
9.15.13. Moduł os	266
9.15.14. Moduł os.path	267
9.15.15. Moduł pathlib	267
9.15.16. Moduł re	268
9.15.17. Moduł shutil	269
9.15.18. Moduł select	269
9.15.19. Moduł smtplib	270
9.15.20. Moduł socket	270
9.15.21. Moduł struct	272
9.15.22. Moduł subprocess	273
9.15.23. Moduł tempfile	273
9.15.24. Moduł textwrap	274
9.15.25. Moduł threading	275
9.15.26. Moduł time	276

9.15.27. Pakiet urllib	277
9.15.28. Moduł unicodedata	278
9.15.29. Pakiet xml	279
9.16. Podsumowanie	279
Rozdział 10. Funkcje wbudowane i biblioteka standardowa	281
10.1. Funkcje wbudowane	281
10.2. Wyjątki wbudowane	297
10.2.1. Klasy bazowe wyjątków	297
10.2.2. Atrybuty wyjątków	298
10.2.3. Predefiniowane klasy wyjątków	299
10.3. Biblioteka standardowa	301
10.3.1. Moduł collections	302
10.3.2. Moduł datetime	302
10.3.3. Moduł itertools	302
10.3.4. Moduł inspect	302
10.3.5. Moduł math	302
10.3.6. Moduł os	302
10.3.7. Moduł random	302
10.3.8. Moduł re	302
10.3.9. Moduł shutil	303
10.3.10. Moduł statistics	303
10.3.11. Moduł sys	303
10.3.12. Moduł time	303
10.3.13. Moduł turtle	303
10.3.14. Moduł unittest	303
10.4. Podsumowanie: korzystaj z wbudowanych elementów	303