

# Spis treści

|                                                        |           |
|--------------------------------------------------------|-----------|
| <b>Przedmowa</b>                                       | <b>9</b>  |
| <b>O autorze</b>                                       | <b>11</b> |
| <b>O recenzencie</b>                                   | <b>12</b> |
| <b>Wprowadzenie</b>                                    | <b>13</b> |
| <b>Rozdział 1. Praca z modułami Pythona</b>            | <b>17</b> |
| Wprowadzenie                                           | 17        |
| Używanie i importowanie modułów i przestrzeni nazw     | 18        |
| Jak to zrobić?                                         | 20        |
| Jak to działa?                                         | 21        |
| Co dalej?                                              | 22        |
| Implementowanie wirtualnego środowiska Pythona         | 25        |
| Zaczynamy                                              | 26        |
| Jak to zrobić?                                         | 28        |
| Jak to działa?                                         | 29        |
| Co dalej?                                              | 29        |
| Opcje dostępne podczas instalowania pakietu Pythona    | 29        |
| Jak to zrobić?                                         | 30        |
| Jak to działa?                                         | 31        |
| Wykorzystanie pliku wymagań i rozwiązywanie konfliktów | 32        |
| Jak to zrobić?                                         | 32        |
| Jak to działa?                                         | 33        |
| Co dalej?                                              | 34        |
| Używanie lokalnych poprawek i plików ograniczeń        | 34        |
| Jak to zrobić?                                         | 34        |
| Jak to działa?                                         | 36        |
| Co dalej?                                              | 36        |
| Praca z pakietami                                      | 36        |
| Jak to zrobić?                                         | 36        |
| Jak to działa?                                         | 38        |
| Co dalej?                                              | 38        |
| Tworzenie pakietów i plików w formacie wheel           | 39        |
| Jak to zrobić?                                         | 40        |
| Jak to działa?                                         | 40        |
| Co dalej?                                              | 41        |
| Porównanie kodu źródłowego z kodem bajtowym            | 41        |
| Jak to zrobić?                                         | 41        |
| Jak to działa?                                         | 42        |
| Co dalej?                                              | 42        |
| Tworzenie pakietów modułu i odwoływanie się do nich    | 42        |

|                                                          |               |
|----------------------------------------------------------|---------------|
| Jak to zrobić?                                           | 43            |
| Jak to działa?                                           | 44            |
| Co dalej?                                                | 45            |
| Pliki binarne dla konkretnego systemu operacyjnego       | 45            |
| Jak to zrobić?                                           | 47            |
| Co dalej?                                                | 50            |
| Umieszczanie programu w repozytorium PyPI                | 51            |
| Zaczynamy                                                | 51            |
| Jak to zrobić?                                           | 52            |
| Jak to działa?                                           | 52            |
| Pakowanie projektu                                       | 55            |
| Jak to zrobić?                                           | 55            |
| Przekazanie pakietu do repozytorium PyPI                 | 56            |
| Zaczynamy                                                | 56            |
| Jak to zrobić?                                           | 57            |
| Jak to działa?                                           | 57            |
| <br><b>Rozdział 2. Zastosowanie interpretera Pythona</b> | <br><b>59</b> |
| Wprowadzenie                                             | 59            |
| Uruchamianie środowiska Pythona                          | 60            |
| Jak to zrobić?                                           | 60            |
| Jak to działa?                                           | 60            |
| Opcje polecenia python                                   | 61            |
| Jak to zrobić?                                           | 61            |
| Jak to działa?                                           | 61            |
| Zobacz również                                           | 65            |
| Praca ze zmiennymi środowiskowymi                        | 65            |
| Jak to zrobić?                                           | 66            |
| Jak to działa?                                           | 66            |
| Definiowanie skryptu jako wykonywalnego                  | 69            |
| Jak to zrobić?                                           | 69            |
| Co dalej?                                                | 69            |
| Zmiana sposobu uruchamiania interpretera interaktywnego  | 70            |
| Jak to zrobić?                                           | 70            |
| Zobacz również                                           | 70            |
| Alternatywne implementacje Pythona                       | 71            |
| Jak to zrobić?                                           | 71            |
| Co dalej?                                                | 72            |
| Instalowanie Pythona w Windowsie                         | 74            |
| Zaczynamy                                                | 74            |
| Jak to zrobić?                                           | 75            |
| Stosowanie programu uruchamiającego Pythona w Windowsie  | 75            |
| Jak to zrobić?                                           | 76            |
| Osadzanie Pythona w innych aplikacjach                   | 76            |
| Jak to zrobić?                                           | 77            |
| Jak to działa?                                           | 78            |
| Zastosowanie alternatywnej powłoki Pythona — IPython     | 78            |
| Zaczynamy                                                | 80            |

|                                                     |          |            |
|-----------------------------------------------------|----------|------------|
| Jak to zrobić?                                      |          | 80         |
| Co dalej?                                           |          | 83         |
| Zastosowanie alternatywnej powłoki Pythona —        | bpython  | 83         |
| Zaczynamy                                           |          | 84         |
| Jak to zrobić?                                      |          | 84         |
| Co dalej?                                           |          | 85         |
| Zastosowanie alternatywnej powłoki Pythona —        | DreamPie | 85         |
| Zaczynamy                                           |          | 86         |
| Jak to zrobić?                                      |          | 86         |
| Co dalej?                                           |          | 86         |
| <b>Rozdział 3. Praca z dekoratorami</b>             |          | <b>87</b>  |
| Wprowadzenie                                        |          | 87         |
| Przegląd funkcji                                    |          | 88         |
| Jak to zrobić?                                      |          | 88         |
| Jak to działa?                                      |          | 89         |
| Wprowadzenie do dekoratorów                         |          | 90         |
| Jak to zrobić?                                      |          | 90         |
| Jak to działa?                                      |          | 92         |
| Stosowanie dekoratorów funkcji                      |          | 93         |
| Jak to zrobić?                                      |          | 94         |
| Jak to działa?                                      |          | 95         |
| Stosowanie dekoratorów klas                         |          | 97         |
| Jak to zrobić?                                      |          | 97         |
| Przykłady dekoratorów                               |          | 100        |
| Zaczynamy                                           |          | 101        |
| Jak to zrobić?                                      |          | 101        |
| Jak to działa?                                      |          | 104        |
| Co dalej?                                           |          | 105        |
| Stosowanie modułu decorator                         |          | 108        |
| Jak to zrobić?                                      |          | 108        |
| Jak to działa?                                      |          | 110        |
| Co dalej?                                           |          | 111        |
| Zobacz również                                      |          | 111        |
| <b>Rozdział 4. Zastosowanie kolekcji w Pythonie</b> |          | <b>113</b> |
| Wprowadzenie                                        |          | 113        |
| Przegląd dostępnych kontenerów                      |          | 114        |
| Jak to zrobić?                                      |          | 115        |
| Co dalej?                                           |          | 116        |
| Implementacja nazwanej krotki                       |          | 119        |
| Jak to zrobić?                                      |          | 120        |
| Co dalej?                                           |          | 122        |
| Implementacja kolejki dwustronnej                   |          | 124        |
| Jak to zrobić?                                      |          | 126        |
| Implementacja klasy ChainMap                        |          | 128        |
| Jak to zrobić?                                      |          | 129        |
| Implementacja kolekcji Counter                      |          | 132        |
| Jak to zrobić?                                      |          | 133        |
| Co dalej?                                           |          | 135        |
| Implementacja klasy OrderedDict                     |          | 136        |

|                                 |     |
|---------------------------------|-----|
| Jak to zrobić?                  | 136 |
| Implementacja klasy defaultdict | 139 |
| Jak to zrobić?                  | 140 |
| Implementacja klasy UserDict    | 142 |
| Jak to zrobić?                  | 143 |
| Implementacja klasy UserList    | 144 |
| Jak to zrobić?                  | 144 |
| Co dalej?                       | 145 |
| Implementacja klasy UserString  | 146 |
| Jak to zrobić?                  | 146 |
| Usprawnienie kolekcji Pythona   | 147 |
| Jak to zrobić?                  | 147 |
| Moduł collections-extended      | 154 |
| Zaczynamy                       | 155 |
| Jak to zrobić?                  | 155 |

## **Rozdział 5. Generatory, współprogramy i przetwarzanie równoległe**

**161**

|                                                      |     |
|------------------------------------------------------|-----|
| Sposób działania iteracji w Pythonie                 | 162 |
| Jak to zrobić?                                       | 162 |
| Stosowanie modułu itertools                          | 166 |
| Jak to zrobić?                                       | 166 |
| Stosowanie funkcji generatora                        | 183 |
| Jak to zrobić?                                       | 184 |
| Jak to działa?                                       | 184 |
| Co dalej?                                            | 186 |
| Symulowanie wielowątkowości za pomocą współprogramów | 187 |
| Jak to zrobić?                                       | 188 |
| Co dalej?                                            | 190 |
| Kiedy należy stosować przetwarzanie równoległe?      | 190 |
| Jak to zrobić?                                       | 191 |
| Co dalej?                                            | 192 |
| Rozwidlenie procesu                                  | 192 |
| Jak to zrobić?                                       | 193 |
| Jak to działa?                                       | 193 |
| Co dalej?                                            | 194 |
| Jak zaimplementować wielowątkowość?                  | 194 |
| Jak to zrobić?                                       | 195 |
| Co dalej?                                            | 199 |
| Jak zaimplementować wieloprocusowość?                | 200 |
| Jak to zrobić?                                       | 201 |
| Co dalej?                                            | 203 |

## **Rozdział 6. Praca z modułem math Pythona**

**205**

|                                          |     |
|------------------------------------------|-----|
| Stosowanie stałych i funkcji modułu math | 206 |
| Jak to zrobić?                           | 206 |
| Praca z liczbami zespolonymi             | 219 |
| Jak to zrobić?                           | 219 |

|                                                      |     |
|------------------------------------------------------|-----|
| Usprawnienie pracy z liczbami typu decimal           | 221 |
| Jak to zrobić?                                       | 222 |
| Zwiększenie dokładności za pomocą ułamków            | 225 |
| Jak to zrobić?                                       | 226 |
| Praca z liczbami losowymi                            | 227 |
| Jak to zrobić?                                       | 227 |
| Stosowanie modułu secrets                            | 231 |
| Jak to zrobić?                                       | 231 |
| Implementowanie podstawowych operacji statystycznych | 233 |
| Jak to zrobić?                                       | 233 |
| Poprawa funkcjonalności za pomocą modułu comath      | 237 |
| Zaczynamy                                            | 237 |
| Jak to zrobić?                                       | 237 |

## **Rozdział 7. Poprawa wydajności działania Pythona za pomocą PyPy**

**241**

|                                |     |
|--------------------------------|-----|
| Wprowadzenie                   | 241 |
| Co to jest PyPy?               | 242 |
| Zaczynamy                      | 244 |
| Jak to zrobić?                 | 244 |
| Co dalej?                      | 247 |
| Co to jest RPython?            | 248 |
| Jak to zrobić?                 | 249 |
| Co dalej?                      | 251 |
| Kilka rzeczywistych przykładów | 251 |
| Jak to zrobić?                 | 252 |
| Co dalej?                      | 256 |

## **Rozdział 8. Dokumenty PEP**

**257**

|                                                              |     |
|--------------------------------------------------------------|-----|
| Wprowadzenie                                                 | 257 |
| Co to jest PEP?                                              | 257 |
| Jak to zrobić?                                               | 258 |
| Co dalej?                                                    | 260 |
| PEP 556 — mechanizm usuwania nieużytków wykorzystujący wątki | 261 |
| Zaczynamy                                                    | 261 |
| Jak to zrobić?                                               | 263 |
| Co dalej?                                                    | 266 |
| PEP 554 — wiele podinterpreterów                             | 267 |
| Jak to zrobić?                                               | 267 |
| Jak to działa?                                               | 271 |
| Co dalej?                                                    | 272 |
| PEP 551 — większe bezpieczeństwo                             | 272 |
| Zaczynamy                                                    | 272 |
| Jak to zrobić?                                               | 274 |
| PEP 543 — ujednolicone API TLS                               | 276 |
| Jak to zrobić?                                               | 277 |
| Co dalej?                                                    | 278 |

|                                                              |            |
|--------------------------------------------------------------|------------|
| <b>Rozdział 9. Dokumentowanie kodu za pomocą LyX</b>         | <b>279</b> |
| Wprowadzenie                                                 | 279        |
| Techniki i narzędzia Pythona związane z dokumentowaniem kodu | 280        |
| Jak to zrobić?                                               | 280        |
| Komentarze osadzone i wywołanie dir()                        | 282        |
| Stosowanie komentarzy typu docstring                         | 284        |
| Jak to zrobić?                                               | 284        |
| Co dalej?                                                    | 289        |
| Stosowanie narzędzia PyDoc                                   | 292        |
| Jak to zrobić?                                               | 293        |
| Raporty w formacie HTML                                      | 294        |
| Jak to zrobić?                                               | 294        |
| Stosowanie plików w formacie reStructuredText                | 299        |
| Zaczynamy                                                    | 299        |
| Jak to zrobić?                                               | 299        |
| Stosowanie LaTeX i LyX do przygotowania dokumentacji         | 302        |
| Zaczynamy                                                    | 303        |
| Jak to zrobić?                                               | 304        |
| Co dalej?                                                    | 307        |
| <b>Skorowidz</b>                                             | <b>311</b> |