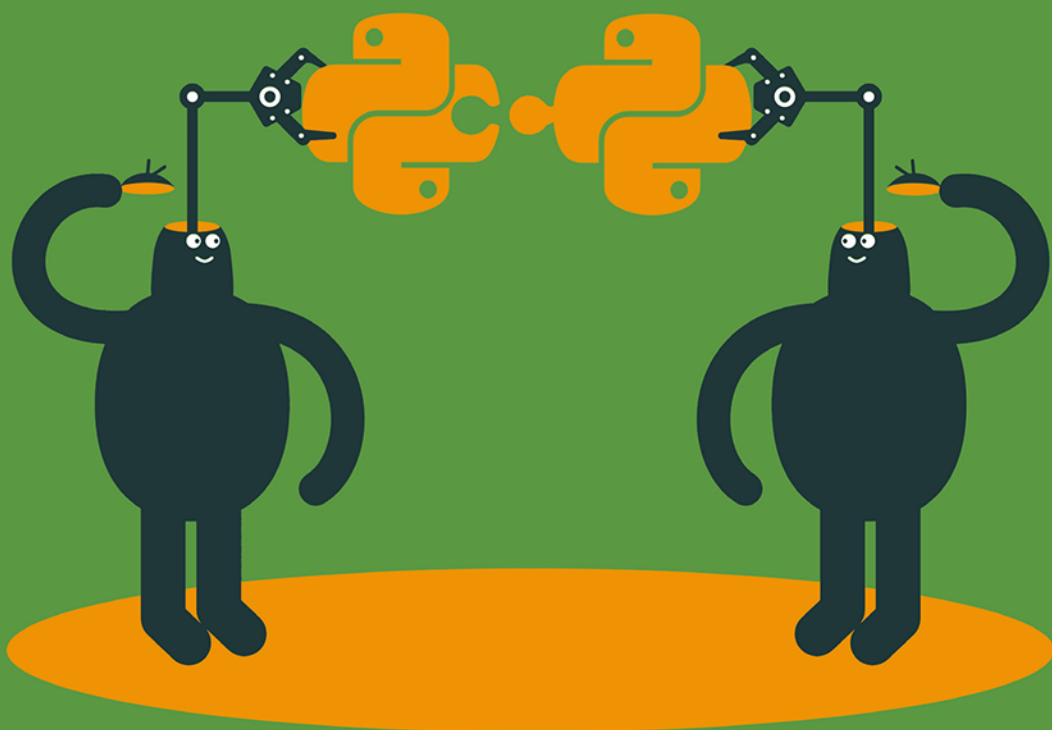


PYTHON

Uczymy się programowania



Ana Bell

Tytuł oryginału: Get Programming: Learn to code with Python

Tłumaczenie: Przemysław Gola

ISBN: 978-83-283-5315-2

Original edition copyright © 2018 by Manning Publications Co.
All rights reserved

Polish edition copyright © 2019 by HELION SA.
All rights reserved.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Helion SA dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Helion SA nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Helion SA

ul. Kościuszki 1c, 44-100 Gliwice

tel. 32 231 22 19, 32 230 98 63

e-mail: helion@helion.pl

WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Pliki z przykładami omawianymi w książce można znaleźć pod adresem:

<ftp://ftp.helion.pl/przyklady/pytusp.zip>

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/pytusp>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

SPIS TREŚCI

Przedmowa	9
Podziękowania	11
O tej książce	13
O autorce	17

Rozdział 0

NAUKA PROGRAMOWANIA

Lekcja 1	Dlaczego powinieneś nauczyć się programować?	21
Lekcja 2	Podstawowe zasady nauki języka programowania	29

Rozdział 1

ZMIENNE, TYPY, WYRAŻENIA I INSTRUKCJE

Lekcja 3	Wprowadzenie do Pythona: język programowania	45
Lekcja 4	Zmienne i wyrażenia: nadawanie rzeczom nazw i przypisywanie im wartości	56
Lekcja 5	Typy obiektów oraz instrukcje	66
Lekcja 6	Projekt podsumowujący: Twój pierwszy program w języku Python — konwerter godzin na minuty	75

Rozdział 2

CIĄGI, KROTKI ORAZ INTERAKCJA Z UŻYTKOWNIKIEM

Lekcja 7	Wprowadzenie do ciągów — łańcuchów znaków	85
Lekcja 8	Zaawansowane operacje na ciągach	93
Lekcja 9	Podstawowe komunikaty o błędzie	100
Lekcja 10	Krotki, czyli kolekcje obiektów dowolnego typu	103
Lekcja 11	Interakcja z użytkownikiem	110
Lekcja 12	Projekt podsumowujący: maszynka do wymyślania imion	118

Rozdział 3

STEROWANIE PRZEBIEGIEM PROGRAMU

Lekcja 13	Wprowadzenie do sterowania przebiegiem programu	129
Lekcja 14	Podjęcie bardziej złożonych decyzji	142
Lekcja 15	Projekt podsumowujący: zaplanuj przygodę	156

Rozdział 4

WYKONYWANIE POWTARZALNYCH CZYNNOŚCI

Lekcja 16	Wykonywanie powtarzalnych czynności za pomocą pętli	163
Lekcja 17	Dostosowywanie pętli do własnych potrzeb	172
Lekcja 18	Wykonywanie powtarzalnych zadań po spełnieniu określonych warunków	178
Lekcja 19	Projekt podsumowujący: scrabble — wersja artystyczna	189

Rozdział 5

DZIELENIE KODU NA BLOKI WIELOKROTNEGO UŻYCIA

Lekcja 20	Tworzenie programów łatwych w rozbudowie	203
Lekcja 21	Osiągnięcie modularności i abstrakcji dzięki funkcjom	214
Lekcja 22	Zaawansowane operacje z wykorzystaniem funkcji	225
Lekcja 23	Projekt podsumowujący: sprawdź swoich przyjaciół	238

Rozdział 6**PRACA ZE ZMIENNYMI TYPMI DANYCH**

- Lekcja 24** Obiekty zmienne i niezienne 255
Lekcja 25 Praca z listami 262
Lekcja 26 Zaawansowane operacje na listach 273
Lekcja 27 Słowniki jako powiązania między obiektami 281
Lekcja 28 Tworzenie aliasów oraz kopiowanie list i słowników 292
Lekcja 29 Projekt podsumowujący: podobieństwo dokumentów 303

Rozdział 7**TWORZENIE WŁASNYCH TYPÓW OBIEKTÓW ZA POMOCĄ PROGRAMOWANIA ZORIENTOWANEGO OBIEKTOWO**

- Lekcja 30** Tworzenie własnych typów obiektów 315
Lekcja 31 Tworzenie klasy dla typu obiektu 321
Lekcja 32 Praca z własnymi typami obiektów 331
Lekcja 33 Dostosowywanie klas do własnych potrzeb 339
Lekcja 34 Projekt podsumowujący: gra w karty 347

Rozdział 8**KORZYSTANIE Z BIBLIOTEK W CELU UDOSKONALANIA WŁASNYCH PROGRAMÓW**

- Lekcja 35** Użyteczne biblioteki 357
Lekcja 36 Testowanie i debugowanie programów 367
Lekcja 37 Biblioteka do tworzenia graficznego interfejsu użytkownika 377
Lekcja 38 Projekt podsumowujący: zabawa w berka 386

Dodatek A Odpowiedzi do zadań 395

Dodatek B Ściąga z Pythona 443

Dodatek C Godne uwagi biblioteki Pythona 447
Skorowidz 451

PRZEDMOWA

Są dwa główne powody, które skłoniły mnie do napisania tej książki. Moim celem było wypełnienie luki na książkę, która uczy programowania od zera i która prezentuje programowanie jako czynność pomocną w rozwiązywaniu codziennych problemów.

Panuje błędne przekonanie, że programowanie zawsze musi wiązać się z ogromnym wysiłkiem, jako że polega na rozwiązywaniu problemów świata. W rzeczywistości to tak nie wygląda. Nauka programowania pozwoli Ci ulepszyć Twoje codzienne życie! Ja sama piszę bezustannie krótkie programy, czasem aby rozwiązać łamigłówkę, czasem aby pomóc sobie w podjęciu jakiejś decyzji. Tę ideę chciałam zawrzeć w mojej książce, by ułatwić każdemu dostęp do programowania — pokazać, że nawet z niewielką wiedzą można pisać użyteczne programy dopasowane do swoich potrzeb.

Prowadzę na uczelni wyższej wstępny kurs informatyki oparty na języku Python. Większość studentów uczestniczących w wykładach nie ma wcześniejszych doświadczeń z programowaniem w żadnym języku. Kurs jest dość intensywny i wielu uczestników pyta o jakieś zasoby online dla ludzi, którzy wcześniej nie programowali. Prawie wszystkie zasoby, które im polecam, wymagają bazowej wiedzy programistycznej i stanowią kolejny etap w ich edukacji: muszą najpierw uchwycić ogólne zasady programowania, a potem nauczyć się, jak zastosować je w Pythonie. Zawsze staram się pamiętać, jak to jest rozpocząć naukę programowania od zera, bez względu na to, po raz który prowadzę wykład. Chciałabym, aby ta książka stanowiła łagodne wprowadzenie do programowania w jednym z najpopularniejszych współcześnie języków i pokazała, jak przystępne może być pisanie kodu programistycznego.

PODZIĘKOWANIA

Jestem bardzo wdzięczna za możliwość napisania tej książki, by pomóc tym, którzy wkraczają dopiero w świat programowania.

Najpierw chciałabym podziękować mojemu mężowi, C.J. Kiedy pisałam tę książkę, wspierał mnie nieustannie, zarówno udzielając rad, jak i opiekując się naszym synem, gdy pracowałam nad nią weekendami.

Następnie chciałabym podziękować moim rodzicom i siostrze. Mój tata nauczył mnie programować, kiedy miałam 12 lat, i nigdy nie zapomnę, ile razy musiał mi tłumaczyć, na czym polega programowanie zorientowane obiektowo, aż w końcu załapałam. Moja siostra i mama przemierzały kraj kilka razy w roku, aby pomóc mi w opiece nad synem, kiedy musiałam więcej czasu przeznaczyć na pisanie. Zwłaszcza moja mama stała się dla mnie „sekretną bronią”. Ponieważ nigdy wcześniej nie programowała, była dla mnie świetnym królikiem doświadczalnym — mogłam na niej przetestować ćwiczenia i dawać jej do sprawdzenia napisane rozdziały.

Chciałabym również podziękować moim edytorom: Kristen Watterson, Danowi Maharry’emu i Eleshy Hyde. Książka przeszła wiele zmian, zanim stała się tym, czym jest obecnie. Dziękuję im wszystkim za cierpliwość, kiedy wpisałam i przepisywałam lekcje. Ich uwagi były bardzo cenne i sprawiły, że książka jest znacznie lepsza. Wielkie podziękowania przekazuję redaktorowi technicznemu, Frances Buontempo, oraz korektorowi technicznemu, Ignacio Beltranowi Torresowi, którzy uważnie czytali poszczególne lekcje, wskazywali błędy i udzielali znakomitych rad, jak ulepszyć książkę. Chciałabym też podziękować pozostałym osobom z wydawnictwa Manning, które przyczyniły się do wydania i wypromowania książki. Oczywiście dziękuję wszystkim recenzentom, którzy poświęcili swój czas, aby czytać i oceniać książkę na każdym etapie jej powstawania. Oto ich nazwiska: Alexandria Webb, Ana Pop, Andru Estes, Angelo Costa, Ariana Duncan, Artiom Plugachev, Carlie Cornell, David Heller, David Moravec, Adnan Masood, Drew Leon, George Joseph, Gerald Mack, Grace Kacenjar, Ivo Štimac, James Gwaltney, Jeon-Young Kang, Jim Arthur, John Lehto, Joseph M. Morgan, Juston Lantrip, Keith Donaldson, Marci Kenneda, Matt Lemke, Mike Cuddy, Nestor Narvaez, Nicole E. Kogan, Nigel John, Pavol Král, Potito Colluccelli, Prabhuti Prakash, Randy Coffland, R. Udendhran Mudaliyar, Rob Morrison, Rujiraporn Pitaksalee, Sam Johnson, Shawn Bolan, Sowmy Vajjala-Balakrishna, Steven Parr, Thomas Ballinger, Tom Northwood, Vester Thacker, Warren Rust, Yan Guo, i Yves Dorfsman.

O TEJ KSIĄŻCE

Do kogo skierowana jest ta książka

Książka *Python. Uczymy się programowania* jest skierowana do każdego, kto jest zainteresowany programowaniem, ale niekoniecznie chce związać z tą dziedziną swoją karierę zawodową. Wcześniejsze doświadczenie w programowaniu nie jest wymagane. Wskazana jest jednak znajomość następujących koncepcji:

- *Zmienne*. Kto uczestniczył w lekcjach matematyki i zna podstawy algebry, wie, czym jest zmienna. Ta książka wyjaśnia, na czym polega specyfika zmiennych w programowaniu.
- *Przypisywanie wartości logicznych (prawda/fałsz) do wyrażeń*. Wyrażenia to zdania, które mogą być określone mianem prawdziwych lub fałszywych. Na przykład „pada deszcz” jest wyrażeniem, które jest albo prawdziwe, albo fałszywe. Powinieneś wiedzieć, jak zmieniać wartość logiczną wyrażeń za pomocą słowa not. Na przykład, jeśli „pada deszcz” jest prawdą, to „nie pada deszcz” jest fałszem.
- *Łączenie wyrażeń*. Kiedy mamy do czynienia z więcej niż jednym wyrażeniem, mogą być one łączone za pomocą słowa and lub or. Na przykład „pada deszcz” i „jestem szczęśliwy” mogą być połączone w „Pada deszcz i jestem szczęśliwy”.
- *Podejmowanie decyzji*. W przypadku wielu wyrażeń możesz podejmować decyzje na podstawie wartości logicznej jednego z nich za pomocą konstrukcji i f...then... Na przykład „Jeśli pada deszcz, ziemia jest mokra” składa się z dwóch wyrażeń: „pada deszcz” oraz „ziemia jest mokra”. Wyrażenie „ziemia jest mokra” to rezultat „pada deszcz”.
- *Umiejętność korzystania z instrukcji w czasie realizacji dowolnych czynności*. Gra w 20 pytań, posługiwanie się przepisem kulinarnym, rozwiązywanie zadań w podręczniku czy rozumienie, czym jest algorytm (algorytm to zestaw instrukcji wraz z odgałęzieniami).

Jak zorganizowana jest ta książka: mapa drogowa

Książka zawiera 8 rozdziałów złożonych z 38 lekcji. Każdy z nich kończy się projektem podsumowującym. Celem każdego rozdziału jest nauczyć Cię jednego ważnego zagadnienia z zakresu programowania, co jest realizowane za pomocą kilku krótkich lekcji:

- Rozdział 0. dostarcza odrobiny motywacji, aby wprowadzić Cię w świat programowania komputerów. Dowiesz się, że programowanie można porównać z innymi zadaniami, które czasami wykonujesz.
- Rozdział 1. wprowadzi Cię w podstawy programowania i ukaże Ci strukturę każdego programu komputerowego. Pobierzesz i zainstalujesz również środowisko, w którym będziesz pisać programy.
- Rozdział 2. pokaże Ci, jak pisać kod, który wchodzi w interakcję z użytkownikiem poprzez pobieranie od niego danych i wyświetlanie mu wyniku działania programu.
- Rozdział 3. nauczy Cię, jak pisać programy zdolne do podejmowania decyzji. Napiszesz kod, którego przebieg będzie możliwy na różne sposoby. Po uruchomieniu program zdecyduje, którą ścieżką podążać, w zależności od wartości ustalonych w miejscach rozgałęzienia.
- Rozdział 4. opiera się na założeniu, że komputery są dobre w robieniu czegoś szybko. Napiszesz kod, który wykorzystuje walory komputerów, polegające na umiejętności wykonywania określonych czynności wielokrotnie — stanie się to poprzez napisanie kodu, który automatycznie powtarza wielokrotnie zestaw poleceń.
- Rozdział 5. wprowadzi Cię na drogę pisania dobrze zorganizowanego kodu: za pomocą funkcji, które są modułami wielokrotnego użytku.
- Rozdział 6. zaprezentuje Ci złożone typy obiektów, które możesz wykorzystać w programie. Po tym rozdziale będziesz umiał tworzyć niesłychanie użyteczne i wszechstronne programy.
- Rozdział 7. pokaże Ci, jak tworzyć własne typy obiektów. Nie wszystkie języki programowania posiadają tę właściwość, jednak większość współczesnych języków ją ma.
- Rozdział 8. kończy tę książkę — pokaże Ci, jak wykorzystywać w swoich programach biblioteki napisane przez innych, jak organizować kod i jak korzystać z wcześniej napisanego kodu.

Kod programistyczny

Kod programistyczny zawarty w tej książce jest zgodny z Pythonem w wersji 3.5, który jest najnowszą wersją w chwili pisania tej książki.

Próbki kodu zawarte w książce prezentują praktyczne zastosowanie koncepcji wyłożonych w danej lekcji, tak abyś potrafił je zastosować w codziennym życiu. W miarę postępów przykłady kodu stają się coraz dłuższe, a to samo zadanie może być rozwiązane na różne sposoby.

Pod koniec każdego rozdziału projekt podsumowujący zbiera kluczowe koncepcje, które przewinęły się w poszczególnych lekcjach. Najpierw dowiesz się, na czym polega problem, a następnie zapoznasz się z jego możliwymi rozwiązaniami. Odkryjesz, jak przetłumaczyć opis zadania wyrażony w języku naturalnym na kod programistyczny.

Książka zawiera wiele przykładów kodu źródłowego, zarówno w postaci listingów, jak i linii w obrębie normalnego tekstu. W obu przypadkach kod źródłowy jest sformatowany czcionką o stałej szerokości znaków, aby bez trudu można było go odróżnić od reszty tekstu. Czasami kod jest także **pogrubiony**, aby zaznaczyć, co zmieniło się w nim w porównaniu z poprzednią jego wersją, w sytuacji gdy dodano jakiś fragment.

W wielu przypadkach oryginalny kod źródłowy został przeformatowany; dodaliśmy znaki przełamania wiersza i zmieniliśmy układ akapitu, aby zmieścić kod w przestrzeni dostępnej na stronie książki. W nielicznych sytuacjach nawet to okazywało się niewystarczające i konieczne było użycie w listingach znaków kontynuacji wiersza (↪). Dodatkowo komentarze kodu często bywały usuwane z listingów, zwłaszcza gdy kod został omówiony w tekście. W listingach odnajdziesz uwagi do kodu, które podkreślają istotne zagadnienia.

Forum książki

Dzięki zakupowi książki *Python. Uczymy się programowania* uzyskujesz bezpłatny dostęp do prywatnego forum uruchomionego przez wydawnictwo Manning Publications. Możesz tam pozostawić swoją opinię na temat książki, zadać pytanie techniczne i uzyskać pomoc od autorki lub innych użytkowników. Aby uzyskać dostęp do forum, przejdź na stronę <https://forums.manning.com/forums/get-programming>. Dodatkowe informacje na temat forum wydawnictwa Manning możesz uzyskać na stronie <https://forums.manning.com/forums/about>.

Manning dąży do tego, aby stworzyć przestrzeń, w której mogłaby być prowadzona merytoryczna dyskusja między czytelnikami i autorką. Nie oznacza to, że zaangażowanie autorki ma być nieograniczone, zwłaszcza że jej uczestnictwo na forum jest oparte na zasadach wolontariatu (nie jest opłacane). Sugerujemy zatem, abyś zwracał się do autorki w przypadku bardziej wymagających pytań, które mogą również jej wydać się interesujące. Forum wraz z historią wątków dyskusji będzie dostępne na stronie wydawcy tak długo, jak długo książka będzie dostępna w sprzedaży.

O AUTORCE



Dr Ana Bell jest wykładowcą w Massachusetts Institute of Technology, na Wydziale Elektrotechniki i Informatyki. Prowadziła w ciągu ostatnich pięciu lat dwa kursy wprowadzające do Pythona: jeden przeznaczony dla studentów bez wcześniejszego doświadczenia w programowaniu, drugi rozszerzający wiedzę uzyskaną na pierwszym roku. Autorce sprawia radość uczenie innych podstaw programowania i obserwowanie, jak wraz z postępami w nauce wzrasta ich pewność siebie. Szczególnie satysfakcjonujące dla autorki jest ukazywanie tego samego problemu z różnych perspektyw, co skutkuje głębszym zrozumieniem tematu przez studentów.

Autorka zapoznała się z Pythonem w czasie studiów na Princeton University, gdzie używała go do parsowania i formatowania dużych plików. Doceniła wówczas intuicyjność i prostotę tego języka.

SKOROWIDZ

A

abstrakcja, 214, 354
 kodu, 209
 szczegółów, 212
aktualizacja zmiennej, 63
algorytm, 34
aliasy obiektów, 293, 301
 niezmiennych, 293
 zmiennych, 294
Anaconda Python Distribution, 46
analiza
 informacji, 246
 kodu, 373
argument, 216
atrybut, 66, 323

B

biblioteka, 355, 357
 basemap, 448
 math, 360, 447
 matplotlib, 448
 nlTK, 448
 numpy, 447
 pandas, 449
 pillow, 448
 pygame, 448
 pyqt, 448
 random, 362, 363, 447
 scipy, 448
 scrapy, 448
 smtplib, 448
 sqlalchemy, 448
 time, 364, 447
 tkinter, 379, 447
 unittest, 368, 372, 447
 wxpython, 448

biblioteki
 importowanie, 359
bigramy, 311
blok wyrażień, 71
błąd, 51
brak wartości, 70
bug, 51

C

ciągi znaków, 70, 85, 443
 dzielenie, 88
 indeksowanie, 87
 odnajdywanie słów, 306
 operacje, 90, 93
 tworzenie, 87
 zamiana wielkości liter, 91
 zwracanie liczby znaków, 90
czarna skrzynka, 207

D

dane wejściowe, 241
debugowanie, 35, 52, 244, 367–372
definiowanie obiektu stosu, 332
docstring, 209, 222
dodawanie widżetów, 379
dokumentacja funkcji, 222
drukowanie obiektu, 343
działania, 59
dzielenie zadania, 77, 204

E

edytor plików, 53

F

falsz, 130
 figura na kanwie, 391
 funkcje, 201, 215, 225

- inicjalizacja zmiennej, 229
- obiekty zmienne, 296
- parametry, 296
- pomocnicze, 247
- przekazywanie, 233
- skrótów, 285
- wielokrotne używanie, 245
- zagnieżdżone, 232
- zakres, 227, 230
- zwracanie, 234

G

gra

- w karty, 347
- Zaplanuj przygodę, 159

 graficzny interfejs użytkownika, GUI, 377
 gry losowe, 362

H

hierarchia operatorów, 145

I

IDE, 47

- Spyder, 48

 implementacja, 36

- algorytmu, 34
- metod, 333

 import bibliotek, 358
 inicjalizacja

- metody, 332
- zmiennej, 62

 instalacja Pythona, 45
 instrukcja, 53, 71, 129

- break, 184, 187
- continue, 186, 187
- elif, 148, 153
- else, 148, 149
- if, 132, 134, 140, 145, 149
- if-elif-else, 148–151
- import, 348

print, 222
 return, 220
 instrukcje warunkowe, 131, 139

- zagnieżdżone, 135

 IPython, 49

K

kanwa, 389
 klasa, 321, 354

- CardDeck, 350
- Movie, 345
- Player, 349
- random, 348
- Stack, 333
- Time, 345

 klasy

- implementacja nowego typu, 322
- notacja kropkowa, 328
- tworzenie, 327
- użycie, 345

 klucz, 283
 kod wielokrotnego użycia, 210, 211
 kolejki, 279
 kolekcje

- danych, 104
- obiektów, 103

 kolizja figur, 392
 komentarz, 39
 komunikat o błędzie, 100, 110
 konfiguracja środowiska pracy, 49
 konkatenacja, 98
 konsola IPython, 49
 konwersja, 71

- ciągu znaków, 277
- danych, 115
- godzin, 75
- minut, 80, 81

 kopiowanie

- list, 292
- obiektów zmiennych, 297
- słowników, 292

 krotki, 103, 104, 263

- odwoływanie się do elementów, 106
- operacje, 105
- podmienianie obiektów, 107
- tworzenie, 104
- ustalenie długości, 105

L

- liczby
 - całkowite, 68
 - dziesiętne, 69
 - losowe, 362
 - zmiennoprzecinkowe, 69
- listy, 262, 264
 - dodawanie elementów, 266
 - konwersja ciągu znaków, 277
 - kopiowanie, 292
 - modyfikowanie wartości elementu, 270
 - odwracanie kolejności elementów, 274
 - sortowanie, 274
 - usuwanie elementów, 268
 - zastosowania, 278
 - zawierające inne listy, 275
 - zmiana wartości elementu, 270

Ł

- łańcuchy znaków, 70, 85
- łączenie
 - słów, 307
 - warunków, 143
 - wyrażeń, 24

M

- menu
 - File, 55
 - Start, 47
- metody, 324, 340
 - implementacja, 333
 - nadpisywanie, 340
- modularność, 208, 214, 354
- modulo, 74
- moduł kodu, 208

N

- nadpisywanie metod, 340
- nazwa, 39
 - obiektu, 60
 - zmiennej, 57, 443
- niezależność zadań, 205
- notacja kropkowa, 319, 328

O

- obiekt, 59
 - Player, 349
 - Stack, 334
 - stosu, 332
 - typu funkcja, 235
- obiekty
 - atrybuty, 323
 - działania, 59
 - metody, 324
 - nazwy, 59
 - niezmiennie, 256
 - typy, 59, 67
 - własne typy, 315, 321, 331
 - właściwości, 318, 323
 - zachowania, 318
 - zmienne, 260
 - kopiowanie, 297
- obsługa
 - list, 266
 - zdarzeń, 381
- odśmiecianie pamięci, 258
- okno
 - debugowania, 374
 - Spydera, 50
- opakowanie kodu, 211
- operacja, 67
 - przypisania, 58
- operacje
 - dotyczące podciągów, 94
 - matematyczne, 98
 - na ciągach, 87, 93
 - na krotkach, 105
- operator [], 106
- operatory, 146
 - logiczne, 144

P

- para klucz – wartość, 284
- parametr, 216
 - __init__, 327, 329
 - formalny, 226
 - self, 328
- parametry
 - aktualne, 219, 227
 - funkcji, 296

pętla, 163
 for, 166–175, 183
 while, 178–183
 pętle nieskończone, 181
 pliki
 odczyt danych, 239, 305
 zapis danych, 306
 pobieranie
 informacji, 112
 kopii list, 299
 pozycji elementów, 265
 podciąg, 88
 podejmowanie decyzji, 24, 130
 polecenie
 append, 266, 267
 capitalize, 91, 125
 count, 96, 265
 extend, 266, 267
 find, 94, 95, 122
 index, 265
 input, 113, 117
 insert, 266, 267
 len, 90, 91, 105
 lower, 91, 125
 pop, 268, 286
 print, 110, 112
 replace, 97
 rfind, 95
 songs.values, 287
 swapcase, 91
 upper, 91
 porównywanie dokumentów, 308
 potęgowanie, 73
 prawda, 130
 program
 Gra w karty, 347
 Scrabble, 189, 196
 Zabawa w berka, 386
 Zaczytywanie pliku, 239
 Zaplanuj przygodę, 156
 programowanie, 21, 29
 zorientowane obiektowo, 313
 prymityw, 67
 przechowywanie danych, 115, 241
 przekazywanie funkcji, 233
 przypisanie, 58
 pseudokod, 38, 120
 Python, 45

R

rekalkulacja, 57
 rodzaje testów, 372
 rzutowanie, 116

S

scenariusz testowy, 77
 schemat blokowy, 24, 31, 33
 skalar, 67
 składnia języka, 133
 słowniki, 283
 częstotliwość wystąpień, 289
 kopiowanie, 292
 niekonwencjonalne, 290
 para klucz – wartość, 284
 usuwanie par klucz – wartość, 286
 wystąpienia słów, 309
 słowo kluczowe
 break, 184
 continue, 186
 class, 322
 elif, 148
 else, 148
 if, 150
 print, 54
 range, 172
 return, 218
 Spyder, 47
 stos, 278, 331–336

T

test, 52
 jednostkowy, 370
 testowanie, 101, 244, 367
 tworzenie
 figur, 387
 klasy, 327
 krotek, 104
 list, 264
 obiektów, 352
 słowników, 283, 308
 zmiennej, 62
 typ danych, 253
 float, 73
 int, 73

typy

- logiczne, 68, 69
- obiektów, 59, 72, 317
- wbudowane, 339

U

użycie klas, 345

W

wartość, 66, 283

widżet, 378

- Button, 382
- Canvas, 382
- Checkbutton, 382
- Entry, 382
- Frame, 382
- Label, 382
- LabelFrame, 382
- Listbox, 382
- Menu, 382

wizualizacja zadania, 36

własne typy obiektów, 315, 321, 331

właściwości obiektu, 318

wprowadzanie danych, 112

wykrywanie kolizji figur, 392

wymyślanie imion, 118

wyrażenie, 50, 56, 71, 111

- boolowskie, 131

wyświetlanie

- komunikatów, 110
- obiektów, 112
- wyrażeń, 111

Z

zachowania obiektów, 318

zacztywanie pliku, 305

zadania zależne, 205

zagnieżdżanie funkcji, 232

zakres funkcji, 227

zamiana wielkości liter, 91

zapis z kropką, 90

zapisywanie plików, 54

zdanie logiczne, 23

zdarzenia, 381

zegar, 365

zestaw testów, 368

ziarno generatora, 363

zintegrowane środowisko deweloperskie, IDE,
47, 48

zmiana wartości elementu, 270

zmienna, 23, 39, 56–63, 114

- aktualizacja, 63

- inicjalizacja, 62

- nazwy, 443

- typy obiektów, 67

zmiennie globalne, 228

znak

- przełamania wiersza, 191, 240

- przypisania, 135

- równości, 135

zwracanie

- funkcji, 234

- krotki, 220

PROGRAM PARTNERSKI

— GRUPY HELION —

1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

GRUPA
Helion 

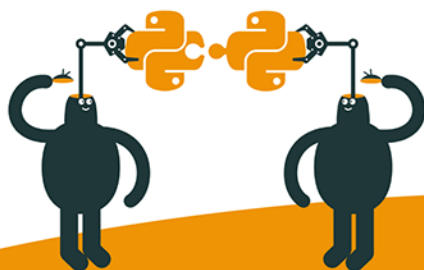
Nie musisz być zawodowym programistą, by zabawić się z Pythonem!

Programowanie wielu ludziom kojarzy się z czymś niewiarygodnie skomplikowanym i zarezerwowanym tylko dla profesjonalistów. Tymczasem jest to umiejętność, którą można porównać do pisania, liczenia czy gotowania. Niezależnie od tego, że na świecie istnieją pisarze tworzący bestsellery, światowej sławy matematycy czy mistrzowie kuchni, ich umiejętności można wykorzystywać na co dzień do napisania listu, sprawdzenia domowych wydatków czy ugotowania zdrowego posiłku. Analogicznie każdy może nauczyć się programowania i zacząć pisać kod, który pomoże mu w rozwiązywaniu codziennych, drobnych problemów. Idealnie zacząć naukę właśnie od Pythona!

Dzięki tej książce szybko się przekonasz, że programowanie nie jest wiedzą tajemną, a całkiem praktyczną umiejętnością, która dodatkowo może dać sporo satysfakcji. Znajdziesz tu przystępnie podane podstawy programowania w Pythonie — zamiast rozbudowanej teorii mnóstwo ćwiczeń do samodzielnego wykonania wraz ze szczegółowymi instrukcjami i wskazówkami. Książka składa się z 38 dobrze zaplanowanych lekcji oraz 7 interesujących projektów. Dzięki przejrzystym ilustracjom, ćwiczeniom weryfikującym stopień zrozumienia lekcji oraz wskazówkom co do dalszej pracy wkrótce zaczniesz myśleć jak programista. W efekcie będziesz mógł dalej rozwijać swoje umiejętności z Pythonem lub wybrać jakiś inny język programowania — a może wykorzystasz nowe kompetencje do stworzenia czegoś zupełnie innego?

Dzięki tej książce:

- przygotujesz środowisko pracy i zaprojektujesz strukturę programu
- zaczniesz posługiwać się instrukcjami, pętlami, zmiennymi, obiektami
- zrozumiesz, jak działają funkcje
- przetestujesz i zdebugujesz swój kod
- nauczysz się korzystać z bibliotek



Dr Ana Bell wykłada w MIT. Prowadzi niezwykle popularne zajęcia z zakresu programowania i informatyki oraz nauki o danych. Specjalizuje się w stosowaniu technik obliczeniowych w naukach biologicznych, szczególnie w genetyce.

	Sprawdź nasze szkolenia!	KOD KORZYŚCI Sięgnij po więcej! ▶	
helion.pl		ISBN 978-83-283-5315-2	
HELION SA ul. Kościuszki 1c 44-100 Gliwice tel.: 32 230 98 63 helion@helion.pl	AKADEMIA IT & BUSINESS WWW.SZKOLENIA.HELION.PL		9 788328 353152
INFORMATYKA W NAJLEPSZYM WYDANIU		Cena: 77,00 zł	