



PYTHON 3

Kolejne lekcje
dla nowych programistów

Z E D A . S H A W

Helion 

Tytuł oryginału: Learn More Python 3 the Hard Way: The Next Step for New Python Programmers

Tłumaczenie: Lech Lachowski

ISBN: 978-83-283-4237-8

Authorized translation from the English language edition, entitled: LEARN MORE PYTHON 3 THE HARD WAY: THE NEXT STEP FOR NEW PYTHON PROGRAMMERS; ISBN 0134123484; by Zed A. Shaw; published by Pearson Education, Inc., publishing as Addison-Wesley Professional. Copyright © 2018 by Zed A. Shaw.

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 Pearson Education, Inc.

Polish language edition published by HELION S.A. Copyright © 2018.

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 Wydawnictwo HELION 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 Wydawnictwo HELION nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Wydawnictwo HELION

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)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/pyt3ko>

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	11
To sprawa osobista	12
CZĘŚĆ I. Początkowa wiedza	14
A jeśli naprawdę nie podoba mi się idea Twojego głupiego procesu osobistego, Zed?	15
A jeśli się okaże, że jestem beznadziejny?	15
Ćwiczenie 0. Konfiguracja	18
Edytor programisty	18
Python 3.6	18
Działający Terminal	19
Działająca konfiguracja pip+virtualenv	19
Dziennik laboratoryjny	19
Opcjonalnie: oprogramowanie do nagrywania ekranu	19
Dalsze badania	20
Ćwiczenie 1. O procesie	22
Wyzwanie	24
Zrób to sam	24
Dalsze badania	24
Ćwiczenie 2. O kreatywności	26
Wyzwanie	26
Zrób to sam	27
Ćwiczenie 3. O jakości	28
Wyzwanie	30
Zrób to sam	30
CZĘŚĆ II. Szybkie sesje kodowania	32
Jak ćwiczyć kreatywność?	33
Proces dla początkujących programistów	34
Proces kodowania dla początkującego programisty	35
Ćwiczenie 4. Obsługiwanie argumentów wiersza poleceń	36
Wyzwanie	36
Rozwiązanie	37
Zrób to sam	38

Ćwiczenie 5. Polecenie cat	40
Wyzwanie	40
Rozwiązanie	41
Zrób to sam	41
Dalsze badania	41
Ćwiczenie 6. Polecenie find	44
Wyzwanie	45
Zrób to sam	46
Dalsze badania	46
Ćwiczenie 7. Polecenie grep	48
Wyzwanie	49
Zrób to sam	49
Dalsze badania	49
Ćwiczenie 8. Polecenie cut	50
Wyzwanie	51
Zrób to sam	51
Dalsze badania	51
Ćwiczenie 9. Polecenie sed	52
Wyzwanie	53
Zrób to sam	53
Dalsze badania	53
Ćwiczenie 10. Polecenie sort	54
Wyzwanie	54
Zrób to sam	55
Dalsze badania	55
Ćwiczenie 11. Polecenie uniq	56
Wyzwanie	56
Zrób to sam	57
Dalsze badania	57
Ćwiczenie 12. Przegląd	58
Wyzwanie	58
Zrób to sam	59
Dalsze badania	59

CZĘŚĆ III. Struktury danych	60
Uczenie się jakości poprzez struktury danych	61
Jak studiować struktury danych?	62
Ćwiczenie 13. Listy jednokierunkowe	64
Opis	64
Kontroler	66
Test	67
Wstępny audyt	69
Wyzwanie	70
Audyt	70
Zrób to sam	70
Ćwiczenie 14. Listy dwukierunkowe	72
Wprowadzamy warunki niezmienne	73
Wyzwanie	74
Zrób to sam	74
Ćwiczenie 15. Stosy i kolejki	76
Wyzwanie	76
Popsuj kod	78
Dalsze badania	78
Ćwiczenie 16. Sortowanie bąbelkowe, przez scalanie i szybkie	80
Wyzwanie	80
Zrób to sam	85
Ćwiczenie 17. Słownik	88
Wyzwanie	88
Wykonanie kopii wzorcowej kodu	88
Skopiuj kod	89
Adnotuj kod	92
Podsumuj strukturę danych	92
Zapamiętaj podsumowanie	93
Zaimplementuj z pamięci	94
Powtarzaj	95
Zrób to sam	95
Popsuj kod	95
Ćwiczenie 18. Mierzenie wydajności	96
Narzędzia	96

Analizowanie wydajności	98
Wyzwanie	100
Zrób to sam	100
Popsuj kod	100
Dalsze badania	100
Ćwiczenie 19. Poprawianie wydajności	102
Wyzwanie	103
Dalsze badania	104
Ćwiczenie 20. Binarne drzewo poszukiwań	106
Wymagania BSTree	106
Usuwanie	107
Wyzwanie	108
Zrób to sam	108
Ćwiczenie 21. Wyszukiwanie binarne	110
Wyzwanie	110
Zrób to sam	110
Dalsze badania	111
Ćwiczenie 22. Tablice sufiksowe	112
Wyzwanie	113
Zrób to sam	113
Dalsze badania	113
Ćwiczenie 23. Drzewo trójkowe	114
Wyzwanie	114
Zrób to sam	116
Ćwiczenie 24. Szybkie wyszukiwanie URL	118
Wyzwanie	118
Zrób to sam	119
Dalsze badania	119
CZĘŚĆ IV. Średnio zaawansowane projekty	120
Śledzenie błędów	121
Ćwiczenie 25. Polecenie xargs	122
Wyzwanie	122
Zrób to sam	122

Ćwiczenie 26. Polecenie hexdump	124
Wyzwanie	125
Zrób to sam	126
Dalsze badania	126
Ćwiczenie 27. Polecenie tr	128
Wyzwanie	128
Krytyka 45-minutowych bloków	129
Zrób to sam	129
Ćwiczenie 28. Polecenie sh	130
Wyzwanie	130
Zrób to sam	131
Dalsze badania	131
Ćwiczenie 29. Polecenia diff i patch	132
Wyzwanie	132
Zrób to sam	133
Dalsze badania	133
CZĘŚĆ V. Parsowanie tekstu	134
Stoień pokrycia kodu	134
Ćwiczenie 30. Automaty skończone	136
Wyzwanie	137
Zrób to sam	139
Dalsze badania	139
Ćwiczenie 31. Wyrażenia regularne	140
Wyzwanie	142
Zrób to sam	142
Dalsze badania	142
Ćwiczenie 32. Skanery	144
Skaner szczytkowego Pythona	145
Wyzwanie	147
Zrób to sam	147
Dalsze badania	147
Ćwiczenie 33. Parsery	150
Parser schodzenia rekurencyjnego	151
Gramatyka BNF	152
Szybki parser demo	153

Wyzwanie	155
Zrób to sam	156
Dalsze badania	156
Ćwiczenie 34. Analizatory	158
Wzorzec Odwiedzający	159
Krótki analizator szczytkowego Pythona	159
Porównanie parsera i analizatora	162
Wyzwanie	162
Zrób to sam	163
Dalsze badania	163
Ćwiczenie 35. Interpretery	164
Porównanie interpreterów i kompilatorów	164
Python jest jednym i drugim	165
Jak napisać interpreter?	165
Wyzwanie	166
Zrób to sam	166
Dalsze badania	166
Ćwiczenie 36. Prosty kalkulator	168
Wyzwanie	168
Zrób to sam	169
Dalsze badania	169
Ćwiczenie 37. Mały BASIC	170
Wyzwanie	170
Zrób to sam	171
CZĘŚĆ VI. SQL i mapowanie obiektowo-relacyjne	172
SQL i tabele	172
Czego się dowiesz?	173
Ćwiczenie 38. Wprowadzenie do SQL	174
Czym jest SQL?	174
Konfiguracja	175
Nauka słownictwa SQL	176
Gramatyka SQL	177
Dalsze badania	177

Ćwiczenie 39. Tworzenie za pomocą SQL	178
Tworzenie tabel	178
Tworzenie wielotabelowych baz danych	178
Wstawianie danych	179
Wstawianie danych referencyjnych	180
Wyzwanie	180
Dalsze badania	181
Ćwiczenie 40. Odczytywanie za pomocą SQL	182
Wybieranie z wielu tabel	182
Wyzwanie	183
Dalsze badania	184
Ćwiczenie 41. Aktualizacja za pomocą SQL	186
Aktualizacja złożonych danych	186
Podmienianie danych	187
Wyzwanie	187
Dalsze badania	188
Ćwiczenie 42. Usuwanie za pomocą SQL	190
Usuwanie przy użyciu innych tabel	190
Wyzwanie	191
Dalsze badania	192
Ćwiczenie 43. Administrowanie bazą danych SQL	194
Niszczanie i zmienianie tabel	194
Migracja i ewoluowanie danych	195
Wyzwanie	196
Dalsze badania	196
Ćwiczenie 44. Korzystanie z API baz danych Pythona	198
Nauka API	198
Wyzwanie	199
Dalsze badania	199
Ćwiczenie 45. Tworzenie ORM	200
Wyzwanie	200
Dalsze badania	201

CZĘŚĆ VII. Projekty końcowe	202
Jaki jest Twój proces?	203
Ćwiczenie 46. Narzędzie blog	204
Wyzwanie	204
Zrób to sam	205
Ćwiczenie 47. Język bc	206
Wyzwanie	206
Zrób to sam	207
Ćwiczenie 48. Narzędzie ed	208
Wyzwanie	208
Zrób to sam	209
Ćwiczenie 49. Narzędzie sed	210
Wyzwanie	211
Zrób to sam	211
Ćwiczenie 50. Narzędzie vi	212
Wyzwanie	212
Zrób to sam	213
Ćwiczenie 51. lessweb	214
Wyzwanie	214
Popsuj kod	214
Zrób to sam	215
Ćwiczenie 52. moreweb	216
Wyzwanie	216
Popsuj kod	217
Dalsze badania	217
Skorowidz	219

O kreatywności

Kreatywność to nic specjalnego. Jeśli jesteś osobą o co najmniej średniej inteligencji, to jesteś kreatywny. Posiadanie przemyśleń i pomysłów, które można potem wcielić w życie, jest po prostu aspektem ludzkiej inteligencji i myśli. Problem polega na tym, że kreatywność stała się wizytówką specjalnej, magicznej kasty ludzi zwanych Kreatywnymi. Istnieją całe książki opisujące tego mitycznego kapłana świata sztuki, który potrafi wyobrazić sobie ideę i lekkim ruchem niesamowicie twórczych, złotych dłoni tworzyć czyste, emocjonalnie inteligentne i empatyczne dzieła sztuki, sprawiające, że aniołki w niebie ronią platynowe łezki. Szczerze mówiąc, słowo „kreatywność” jest nadużywanym komunałem, wykorzystywanym do segregowania ludzi na podstawie realizacji ich pomysłów, ale nie mam innego wyboru, jak użyć tego słowa w tej książce.

W mojej książce słowo „kreatywność” oznacza jedynie „przekładanie idei na świat rzeczywisty”. Nie sugeruję tym słowem żadnej wyższości ani magicznej mocy ludziom, którzy są biegli w urzeczywistnianiu własnych myśli. Jedyna różnica między mną (rzekomo bardzo twórczą osobą) a Tobą polega na tym, że ja ćwiczyłem wcielanie swoich pomysłów w życie. Mam notes z pomysłami i staram się je regularnie implementować. Studiuję malarstwo, muzykę, pisarstwo i programowanie jako środki realizacji i przekuwania myśli w rzeczywistość. Starając się jedynie tworzyć coś regularnie, stałem się w tym biegły i nie ma w tym żadnej magii. Po prostu próbuję, aż uda mi się to zrobić.

Proces nauki tworzenia tego, co mam w głowie, wygenerował monstrialne sterty śmieci, ale na ich czubku znajduje się kilka prac, z których jestem dumny. Jeśli chcesz pracować nad swoimi zdolnościami tworzenia, też będziesz musiał tworzyć własne sterty śmieci. Nie możesz jednak po prostu tworzyć stert z przypadkowych śmieci i mieć nadzieję, że gdy osiągniesz szczyt, staniesz się niesamowity. Sztuka bycia *produktywnie* kreatywną osobą polega na uczeniu się implementowania własnych pomysłów w ramach procesu lub zestawu ograniczeń prowadzących ścieżką uczenia się, ale jednocześnie pozwalających unikać pułapek rygorystycznego procesu, który zabija kreatywność. Pomysłowa osoba balansuje na linii pomiędzy procesem, który ją prowadzi, a procesem, który zabija jej pomysły. Mam nadzieję, że w tej książce znajdziesz tę słodką równowagę.

Wyzwanie

Aby popracować nad procesem twórczym, najpierw musisz popracować nad własnymi przypadkowymi zachowaniami. Myślę, że jedną z moich głównych mocnych stron jest umiejętność wzięcia dwóch pozornie przypadkowych pomysłów i przekształcenia ich w coś interesującego lub przydatnego. Pracę na tym możesz zacząć od codziennego wykonywania tego małego ćwiczenia:

1. Zapisz połączenie co najmniej trzech losowo wybranych słów. Idiotyczne lasy mają iguany. Symbolizm rodzi naleśniki. Python może przywoływać kosmitów.

2. Następnie poświęć 10 minut na napisanie o tych trzech słowach (lub o jednym z nich) eseju poprzez pryzmat różnych zmysłów — wzroku, słuchu, smaku, zapachu czy poczucia równowagi. Sprawdź, ile zmysłów tak naprawdę mają ludzie, aby uzyskać wyobrażenie o tym, co możesz napisać. Nie cenzuruj się, po prostu pozwól słowom płynąć. Możesz także ten pomysł narysować, namalować lub napisać o nim wiersz.
3. Podczas wykonywania tego ćwiczenia mogą nagle przyjść Ci do głowy rzeczywiste pomysły związane z programowaniem lub innymi interesującymi Cię tematami. Zapisz je w jakimś „poważniejszym” miejscu, aby je później zbadać, lub nawet narysuj, jeśli potrafisz.

Wierz lub nie, ale to jedno proste ćwiczenie poprawi bardzo wiele rzeczy, gdy usiądziesz do implementowania oprogramowania:

1. Nauczy Cię, jak pozwolić pomysłom swobodnie płynąć i ich nie cenzurować.
2. Wyszkoli Cię, jak swobodnie kojarzyć pozornie niepowiązane pomysły, aby znajdować możliwe połączenia.
3. Otworzy Twój umysł na możliwość myślenia bez samokrytyki.
4. Poprawi Twoją zdolność do wyrażania myśli w pisaniu lub rysowaniu, co zwykle jest pierwszym krokiem do przekształcania pomysłów w rzeczywistość.
5. Zmusi Cię do wyobrażenia sobie, jak działają Twoje zmysły, a także jak działają u innych ludzi, co pomoże Ci wdrażać pomysły w realnym świecie.
6. Spowoduje także, że ludzie zaczną uważać Cię za osobę wybitną, wręcz Artystę. Teraz możesz równie dobrze kupić sobie beret i przeprowadzić się do Paryża.

Ten proces losowego pisania i myślenia o absurdalnych koncepcjach może być trudny dla ludzi, którzy są przyzwyczajeni do zadreżczania się szczegółami oprogramowania i do przejmwania się jakością. To całkowicie zrozumiałe i na pewno *nadal* potrzebujesz tego poczucia dostarczanej jakości. Kreatywność bez poczucia krytycznych cech jakości wytwarza tylko śmieci. Jednak jakość bez kreatywności jest pozbawiona wyobraźni niezbędnej, żeby zobaczyć, co może pójść nie tak z rezultatami Twojej twórczości. Potrzebujesz tej mieszanki kreatywności i jakości, pomaga ona bowiem tworzyć oprogramowanie i zapewnia jego solidność.

Zrób to sam

Jeśli nie podoba Ci się pomysł zapisywania połączeń przypadkowych słów, takich jak „unitarianie przeważnie latają omlętami”, możesz po prostu losowo wybrać wyraz ze słownika i napisać o nim z perspektywy Twoich zmysłów. Sprawdza się to równie dobrze, ale nie wydaje się tak niepoważne, choć zachęcam Cię, żebyś był trochę niepoważny. Nikt jeszcze nie został zwolniony z pracy za pisanie wierszy o złotych pszczołach na wybrzeżu pereł. Kolejną opcją jest wyrażenie swoich odczuć z punktu widzenia wszystkich zmysłów. To również może pomóc Ci być kreatywnym, a poza tym ma działanie terapeutyczne.

Skorowidz

A

administrowanie bazą danych, 194
 adnotowanie kodu, 92
 aktualizacja złożonych danych, 186
 algorytm sortowania
 bąbelkowego, 82
 przez scalanie, 83
 szybkiego, 80, 85
 analizator, 158, 162
 szczątkowego Pythona, 159
 analizowanie wydajności, 98
 API baz danych, 198
 argumenty wiersza poleceń, 36
 audyt wstępny, 69
 automaty skończone, 136

B

BASIC, 170
 baza danych, 178
 BDD, behavior-driven development, 129
 binarne drzewo poszukiwań, BST, 106
 blog, 204
 bloki czasu, 129
 błędy, 121
 semantyczne, 158
 BST, binary search tree, 106
 BSTree, 106

C

cProfile, 97
 CRUD, 175

D

dane referencyjne, 180
 drzewo
 binarne poszukiwań, 106
 sufiksowe, 11102
 trójkowe, 114
 dziennik laboratoryjny, 19

E

edytor programisty, 18
 ewoluowanie danych, 195

G

gramatyka
 BNF, 152
 SQL, 177

I

implementowanie z pamięci, 94
 instalacja SQLite3, 176
 instrukcja
 CREATE, 176
 DELETE, 176, 190
 DROP, 176, 194
 FROM, 176
 IN, 176
 INSERT, 176, 179
 REPLACE, 187
 SELECT, 176, 182
 SET, 176
 UPDATE, 176, 186
 WHERE, 176
 interpreter, 164, 165
 BASIC-a, 170

J

jakość, 28
 język
 bc, 206
 SQL, 174

K

kalkulator, 168
 kodowanie, 32
 kolejka, 76
 kompilatory, 164

konfiguracja pip+virtualenv, 19
 kontroler, 64, 66
 kopia wzorcowa kodu, 88
 kopiowanie kodu, 89
 krawędź, 64
 kreatywność, 26, 33

L

lessweb, 214
 lista
 dwukierunkowa, 72
 jednokierunkowa, 64

M

mapowanie obiektowo-relacyjne, 172
 migracja danych, 195
 moduł http.server, 214
 moreweb, 216

N

nagrywanie ekranu, 19
 narzędzie
 blog, 204
 cProfile, 97
 ed, 208
 patch, 132
 profile, 97
 sed, 210
 timeit, 96
 vi, 212
 niezmiennik, 74
 notacja Backusa-Naura, 152

O

ORM, object-relational mapping, 200

P

parser, 150, 162
 schodzenia rekurencyjnego, RDP, 151,
 153
 parsowanie tekstu, 134
 podsumowanie struktury danych, 92
 pokrycie kodu, 134

polecenie, *Patrz także* instrukcja
 cat, 40
 cut, 50
 diff, 132
 find, 44
 grep, 48
 hexdump, 124
 patch, 132
 sed, 52
 sh, 130
 sort, 54
 tr, 128
 uniq, 56
 xargs, 122

poprawianie wydajności, 102
 powtarzanie procesu, 95
 proces
 CASMIR, 88
 kodowania, 34, 35
 XP, 22
 profile, 97
 programowanie
 ekstremalne, XP, 22
 oparte na testach, 128
 Python 3.6, 18

R

RDP, recursive descent parser, 151
 relacyjna baza danych, 200

S

serwer WWW, 214, 216
 skaner, 144
 szczątkowego Pythona, 145
 skanowanie kodu, 145
 słownik, 88
 sortowanie
 bąbelkowe, 80, 82
 przez scalanie, 80, 83
 szybkie, 80, 85
 SQL, Structured Query Language, 172
 administrowanie bazą danych, 194
 aktualizacja danych, 186
 gramatyka, 177
 niszczenie tabel, 194
 odczytywanie danych, 182

- operacje, 176
- podmienianie danych, 187
- tworzenie tabel, 178
- usuwanie danych, 190
- wstawianie danych, 179

SQLite3, 175

stopień pokrycia kodu, 134

stos, 76

struktura danych, 60

- binarne drzewo poszukiwań, 106
- drzewo sufiksowe, 112
- drzewo trójkowe, 114
- kolejka, 76
- lista dwukierunkowa, 72
- lista jednokierunkowa, 64
- słownik, 88
- stos, 76
- tablica sufiksowa, 112

szpikulec, 36

szybki parser demo, 153

Ś

śledzenie błędów, 121

T

tabele, 172

tablica sufiksowa, 112

TDD, test-driven development, 128

Terminal, 19

test, 67

timeit, 96

tworzenie

- ORM, 200
- tabel, 178
- wielotabelowych baz danych, 178

U

URL, 118

W

węzeł, 64

wiersz poleceń, 36

wstawianie danych, 179

wydajność, 96

- analizowanie, 98
- poprawianie, 102

wyrażenia regularne, 140

wyszukiwanie

- binarne, 110
- URL, 118

wzorzec Odwiedzający, 159

X

XP, eXtreme Programming, 22

Z

zapamiętywanie podsumowania, 93

PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ HELION



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

Zmień swoją stronę WWW
w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

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

GRUPA WYDAWNICZA

 **Helion SA**

PROCES. KREATYWNOŚĆ. JAKOŚĆ. PYTHON.

Jeśli masz już za sobą pisanie pierwszych programów w Pythonie, to wiesz, jak wszechstronny jest ten język i jak imponujące możliwości oferuje. Python nadaje się do bardzo wielu zastosowań, jeśli jednak chcesz go wykorzystać w pełni, musisz wyjść poza podstawy. Efektywny programista korzysta z wiedzy wykraczającej poza znajomość struktur języka, poza tym jest zdolny do obiektywnej refleksji nad własnymi możliwościami i stara się cały czas doskonalić swój warsztat.

Dzięki tej książce, zanim się spostrzeżesz, wykonasz 52 interesujące i świetnie przygotowane projekty, dzięki którym zyskasz kluczowe praktyczne umiejętności i pogłębisz rozumienie sedna pracy programisty. Nauczysz się analizować problem i projektować sposób jego implementacji w programie. Później zaczniesz tworzyć konkretne rozwiązania, dbając o ich prostotę i elegancję. Wystarczy, że wykażesz się dyscypliną, zaangażowaniem i wytrwałością, a wkrótce zrozumiesz znaczenie procesu, rozwiniesz kreatywność — i przede wszystkim zadbasz o jakość kreowanych rozwiązań. Twoim celem nie będzie już tylko pisanie „kodu, który działa”. Będziesz tworzyć znakomity kod i staniesz się prawdziwym programistą, biegłym w Pythonie.

W tej książce:

- praca ze złożonymi projektami
- korzystanie ze struktur danych
- algorytmy i przetwarzanie struktur danych
- techniki parsowania i przetwarzania tekstu
- modelowanie danych za pomocą języka SQL



ZED A. SHAW jest programistą od ponad 20 lat. Chętnie angażuje się w różne projekty open source, jest również uznanym autorem książek i artykułów dotyczących technik programowania — jego publikacje są czytane i dyskutowane przez miliony czytelników na całym świecie. Posiada niezwykłą umiejętność pisania o trudnych zagadnieniach w sposób przystępny, żywy i interesujący, a równocześnie zmuszający czytelnika do myślenia. W wolnym czasie studiuje malarstwo i historię sztuki.

Helion

hellion.pl

HELION SA
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 98 63
hellion@hellion.pl

INFORMATYKA W NAJLEPSZYM WYDANIU

Sprawdź nasze szkolenia!

SZKOLENIA

AKADEMIA IT & BUSINESS

WWW.SZKOLENIA.HELION.PL

KOD KORZYŚCI
Sięgnij po więcej!

ISBN 978-83-283-4237-8

9 788328 342378

Cena: 39,90 zł

Pearson
Addison-Wesley