

Python 3 Object-Oriented Programming

Third Edition

Build robust and maintainable software with object-oriented design patterns in Python 3.8

Dusty Phillips

Packt>

BIRMINGHAM - MUMBAI

Python 3 Object-Oriented Programming

Third Edition

Copyright © 2018 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing or its dealers and distributors, will be held liable for any damages caused or alleged to have been caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

Commissioning Editor: Richa Tripathi
Acquisition Editor: Chaitanya Nair
Content Development Editor: Rohit Kumar Singh
Technical Editor: Ketan Kamble
Copy Editor: Safis Editing
Project Coordinator: Vaidehi Sawant
Proofreader: Safis Editing
Indexer: Mariammal Chettiyar
Graphics: Alishon Mendonsa
Production Coordinator: Aparna Bhagat

First published: July 2010
Second edition: August 2015
Third edition: October 2018

Production reference: 2051118

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham
B3 2PB, UK.

ISBN 978-1-78961-585-2

www.packt.com



`mapt.io`

Mapt is an online digital library that gives you full access to over 5,000 books and videos, as well as industry leading tools to help you plan your personal development and advance your career. For more information, please visit our website.

Why subscribe?

- Spend less time learning and more time coding with practical eBooks and Videos from over 4,000 industry professionals
- Improve your learning with Skill Plans built especially for you
- Get a free eBook or video every month
- Mapt is fully searchable
- Copy and paste, print, and bookmark content

Packt.com

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.packt.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at customercare@packtpub.com for more details.

At www.packt.com, you can also read a collection of free technical articles, sign up for a range of free newsletters, and receive exclusive discounts and offers on Packt books and eBooks.

Contributors

About the author

Dusty Phillips is a Canadian software developer and author currently living in New Brunswick. He has been active in the open source community for two decades and has been programming in Python for nearly as long. He holds a master's degree in computer science and has worked for Facebook, the United Nations, and several start-ups. He's currently researching privacy-preserving technology at beanstalk.network.

Python 3 Object-Oriented Programming was his first book. He has also written *Creating Apps in Kivy*, and self-published *Hacking Happy*, a journey to mental wellness for the technically inclined. A work of fiction is coming as well, so stay tuned!

About the reviewers

Yogendra Sharma is a developer with experience of the architecture, design, and development of scalable and distributed applications. He was awarded a bachelor's degree from Rajasthan Technical University in computer science. With a core interest in microservices and Spring, he also has hands-on experience technologies such as AWS Cloud, Python, J2EE, Node.js, JavaScript, Angular, MongoDB, and Docker. Currently, he works as an IoT and cloud architect at Intelizign Engineering Services, Pune.

Josh Smith has been coding professionally in Python, JavaScript, and C# for over 5 years, but has loved programming since learning Pascal over 20 years ago. Python is his default language for personal and professional projects. He believes code should be simple, goal-oriented, and maintainable. Josh works in data automation and lives in St. Louis, Missouri, with his wife and two children.

Packt is searching for authors like you

If you're interested in becoming an author for Packt, please visit authors.packtpub.com and apply today. We have worked with thousands of developers and tech professionals, just like you, to help them share their insight with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Table of Contents

Preface	1
Chapter 1: Object-Oriented Design	7
Introducing object-oriented	7
Objects and classes	9
Specifying attributes and behaviors	11
Data describes objects	12
Behaviors are actions	13
Hiding details and creating the public interface	15
Composition	17
Inheritance	20
Inheritance provides abstraction	22
Multiple inheritance	23
Case study	24
Exercises	32
Summary	33
Chapter 2: Objects in Python	35
Creating Python classes	35
Adding attributes	37
Making it do something	38
Talking to yourself	38
More arguments	40
Initializing the object	41
Explaining yourself	43
Modules and packages	46
Organizing modules	48
Absolute imports	49
Relative imports	50
Organizing module content	51
Who can access my data?	54
Third-party libraries	56
Case study	58
Exercises	66
Summary	67
Chapter 3: When Objects Are Alike	69
Basic inheritance	70
Extending built-ins	72
Overriding and super	73

Multiple inheritance	75
The diamond problem	77
Different sets of arguments	82
Polymorphism	85
Abstract base classes	88
Using an abstract base class	88
Creating an abstract base class	90
Demystifying the magic	92
Case study	93
Exercises	99
Summary	100
Chapter 4: Expecting the Unexpected	101
Raising exceptions	101
Raising an exception	103
The effects of an exception	104
Handling exceptions	106
The exception hierarchy	111
Defining our own exceptions	113
Case study	117
Exercises	127
Summary	128
Chapter 5: When to Use Object-Oriented Programming	129
Treat objects as objects	129
Adding behaviors to class data with properties	133
Properties in detail	136
Decorators – another way to create properties	138
Deciding when to use properties	139
Manager objects	142
Removing duplicate code	145
In practice	147
Case study	149
Exercises	157
Summary	158
Chapter 6: Python Data Structures	159
Empty objects	159
Tuples and named tuples	161
Named tuples	163
Dataclasses	164
Dictionaries	168
Dictionary use cases	172
Using defaultdict	173
Counter	175

Lists	176
Sorting lists	178
Sets	181
Extending built-in functions	185
Case study	188
Exercises	195
Summary	196
Chapter 7: Python Object-Oriented Shortcuts	197
Python built-in functions	198
The len() function	198
Reversed	199
Enumerate	200
File I/O	201
Placing it in context	204
An alternative to method overloading	205
Default arguments	207
Variable argument lists	209
Unpacking arguments	213
Functions are objects too	214
Using functions as attributes	219
Callable objects	220
Case study	221
Exercises	228
Summary	229
Chapter 8: Strings and Serialization	231
Strings	231
String manipulation	232
String formatting	235
Escaping braces	235
f-strings can contain Python code	236
Making it look right	238
Custom formatters	241
The format method	241
Strings are Unicode	242
Converting bytes to text	242
Converting text to bytes	243
Mutable byte strings	245
Regular expressions	247
Matching patterns	247
Matching a selection of characters	249
Escaping characters	250
Matching multiple characters	250
Grouping patterns together	251
Getting information from regular expressions	252

Making repeated regular expressions efficient	254
Filesystem paths	255
Serializing objects	258
Customizing pickles	259
Serializing web objects	261
Case study	265
Exercises	270
Summary	272
Chapter 9: The Iterator Pattern	273
Design patterns in brief	273
Iterators	274
The iterator protocol	275
Comprehensions	277
List comprehensions	277
Set and dictionary comprehensions	279
Generator expressions	281
Generators	282
Yield items from another iterable	285
Coroutines	287
Back to log parsing	290
Closing coroutines and throwing exceptions	293
The relationship between coroutines, generators, and functions	294
Case study	295
Exercises	305
Summary	307
Chapter 10: Python Design Patterns I	309
The decorator pattern	309
A decorator example	310
Decorators in Python	314
The observer pattern	316
An observer example	317
The strategy pattern	319
A strategy example	320
Strategy in Python	322
The state pattern	322
A state example	323
State versus strategy	329
State transition as coroutines	329
The singleton pattern	329
Singleton implementation	330
Module variables can mimic singletons	331
The template pattern	334
A template example	334