



Community Experience Distilled

# Python Unlocked

Become more fluent in Python—learn strategies and techniques for smart and high-performance Python programming

Arun Tigeraniya

[PACKT] open source\*  
PUBLISHING community experience distilled

# Python Unlocked

Become more fluent in Python—learn strategies and techniques for smart and high-performance Python programming

**Arun Tigeraniya**

**[PACKT]** open source   
PUBLISHING community experience distilled

BIRMINGHAM - MUMBAI

# Python Unlocked

Copyright © 2015 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, and its dealers and distributors will be held liable for any damages caused or alleged to be 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.

First published: December 2015

Production reference: 1181215

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

ISBN 978-1-78588-599-0

[www.packtpub.com](http://www.packtpub.com)

# Credits

**Author**

Arun Tigeraniya

**Project Coordinator**

Shweta H Birwatkar

**Reviewers**

Mike Driscoll

Chetan Giridhar

Sanjeev Jaiswal

Vishrut Mehta

**Proofreader**

Safis Editing

**Indexer**

Mariammal Chettiyar

**Commissioning Editor**

Veena Pagare

**Production Coordinator**

Arvindkumar Gupta

**Acquisition Editor**

Vivek Anantharaman

**Cover Work**

Arvindkumar Gupta

**Content Development Editors**

Neeshma Ramakrishnan

Deepti Thore

**Technical Editor**

Vijin Boricha

**Copy Editor**

Priyanka Ravi

# About the Author

**Arun Tigeraniya** has a BE in electronics and communication. After his graduation, he worked at various companies as a Python developer. His main professional interests are AI and Big Data. He enjoys writing efficient and testable code, and interesting technical articles. He has worked with open source technology since 2008. He currently works at Jaarvis Labs Limited, India.

---

I would like to thank my parents and elder siblings, Ashok and Asha, who have always supported me in completing this book with good quality. A special thanks to the people at Packt for being so understanding even though I missed a few deadlines.

---

# About the Reviewers

**Mike Driscoll** has been programming in Python since 2006. He enjoys writing about Python on his blog at <http://www.blog.pythonlibrary.org/>. He co-authored *Core Python Refcard for DZone*. Mike has also been a technical reviewer for *Python 3 Object Oriented Programming*, *Python 2.6 Graphics Cookbook* and *Tkinter GUI Application Development Hotshot*, all by Packt Publishing. He recently wrote the book *Python 101*, and is working on his next book.

---

I would like to thank my beautiful wife, Evangeline, for always supporting me. I would also like to thank my friends and family for all that they do to help me. Finally, I would like to thank Jesus Christ for saving me.

---

**Chetan Giridhar** is a Python developer, open source enthusiast, and start-up specialist. He has authored/reviewed books on Python, published papers in journals, and delivered talks in conferences around the world. You can get in touch with him at [cjgiridhar@gmail.com](mailto:cjgiridhar@gmail.com).

He has also reviewed *IPython Interactive Computing and Visualization Cookbook* by Packt Publishing.

---

I would like thank my mentors, friends, colleagues, and my ever-supporting family.

---

**Sanjeev Jaiswal** is a computer graduate having 6 years of industrial experience. He uses Perl, Python, and GNU/Linux for his day-to-day work. Sanjeev teaches Perl, Python, web development online as well. Sanjeev has worked closely with major clients, such as CSC, IBM, United Online, and Syniverse. He has also developed an interest in web application security since 2013.

Sanjeev loves teaching technical stuff to engineering students and IT professionals, and he has been teaching since 2008. He founded Alien Coders (<http://www.aliencoders.org/>) based on the *Learning through sharing* principle for computer science students and IT professionals in 2010, and it became a huge hit in India amongst engineering students.

He usually uploads technical videos on YouTube under the *AlienCoders* tag. One can follow him on:

- Facebook at <http://www.facebook.com/jassics>
- Twitter at <http://twitter.com/jassics>

He wrote an Instant book called *WebSpeed Optimization Techniques*, and *Learning Django Web Development* by Packt Publishing, and he is always looking forward to author or review more and more books from Packt and other publishers.

---

I would like to thank my parents and my wife, Shalini Jaiswal, for moral support at every phase of life and growth. I also give deep thanks and gratitude to my best friends Ritesh Kamal and Ranjan Pandey for their personal and professional help all the time.

It is because of them and other friends that I learned and achieved a set of impossible goals in a short time.

---

**Vishrut Mehta** is a dual degree student of IIT Hyderabad, currently in his 5th year. He is completing his MS under Dr. Vasudeva Varma, Research Dean of IIT, at Search and Information Extraction Lab (SIEL). He works in areas related to software-defined networks and cloud computing. He completed his internship at Google last summer and a research internship at INRIA, France, in 2014. During his internship, he worked on various areas of cloud computing, such as automatic reconfiguration of a multcloud application. Vishrut also participated in Google Summer of Code in 2013 and was also was the admin for the Google Code-in between 2013 and 2014 for Sahana Software Foundation.

He has also reviewed *Learning Python Network Programming*, *Untangle Network Security*, and *Python Network Programming Cookbook*, all by Packt Publishing.

---

I would like to thank my guides Dr. Vasudeva Varma and Dr. Reddy Raja for constantly supporting my ideas and helping me in my work. I would also like to thank my parents who never lost their faith in me.

---

# www.PacktPub.com

## Support files, eBooks, discount offers, and more

For support files and downloads related to your book, please visit [www.PacktPub.com](http://www.PacktPub.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.PacktPub.com](http://www.PacktPub.com) and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at [service@packtpub.com](mailto:service@packtpub.com) for more details.

At [www.PacktPub.com](http://www.PacktPub.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.



<https://www2.packtpub.com/books/subscription/packtlib>

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can search, access, and read Packt's entire library of books.

## Why subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print, and bookmark content
- On demand and accessible via a web browser

## Free access for Packt account holders

If you have an account with Packt at [www.PacktPub.com](http://www.PacktPub.com), you can use this to access PacktLib today and view 9 entirely free books. Simply use your login credentials for immediate access.

# Table of Contents

<b>Preface</b>	<b>v</b>
<b>Chapter 1: Objects in Depth</b>	<b>1</b>
<b>Understanding objects</b>	<b>1</b>
Identity	2
Value	3
Type	4
<b>Making calls to objects</b>	<b>5</b>
<b>How objects are created</b>	<b>8</b>
Creation of function objects	9
Creation of instances	10
Creation of class objects	12
<b>Playing with attributes</b>	<b>14</b>
Descriptors	16
Class, static, and instance methods	17
<b>Summary</b>	<b>19</b>
<b>Chapter 2: Namespaces and Classes</b>	<b>21</b>
<b>How referencing objects work – namespaces</b>	<b>21</b>
<b>Functions with state – closures</b>	<b>25</b>
<b>Understanding import and modules</b>	<b>27</b>
Customizing imports	28
<b>Class inheritance</b>	<b>29</b>
Method resolution order	29
Super's superpowers	30
<b>Using language protocols in classes</b>	<b>32</b>
Iteration protocol	33
Context manager protocol	34
<b>Using abstract classes</b>	<b>35</b>
<b>Summary</b>	<b>36</b>

---

<b>Chapter 3: Functions and Utilities</b>	<b>37</b>
Defining functions	37
Decorating callables	40
Utilities	45
Summary	49
<b>Chapter 4: Data Structures and Algorithms</b>	<b>51</b>
Python built-in data structures	51
Python library data structures	57
Third party data structures	59
Arrays/List	59
Binary tree	59
Sorted containers	60
Trie	61
Algorithms on scale	63
Summary	68
<b>Chapter 5: Elegance with Design Patterns</b>	<b>69</b>
Observer pattern	70
Strategy pattern	72
Singleton pattern	74
Template pattern	75
Adaptor pattern	77
Facade pattern	79
Flyweight pattern	80
Command pattern	82
Abstract factory	84
Registry pattern	86
State pattern	88
Summary	89
<b>Chapter 6: Test-Driven Development</b>	<b>91</b>
Mock for tests	91
Parameterization	94
Creating custom test runners	99
Testing threaded applications	103
Running test cases in parallel	104
Summary	105
<b>Chapter 7: Optimization Techniques</b>	<b>107</b>
Writing optimized code	107
Profiling to find bottlenecks	117
Using fast libraries	122

---

<b>Using C speeds</b>	<b>123</b>
SWIG	123
CFFI	124
Cython	126
<b>Summary</b>	<b>128</b>
<b>Chapter 8: Scaling Python</b>	<b>129</b>
<hr/>	
<b>Going multithreaded</b>	<b>131</b>
<b>Using multiple processes</b>	<b>133</b>
<b>Going asynchronous</b>	<b>136</b>
<b>Scaling horizontally</b>	<b>141</b>
<b>Summary</b>	<b>145</b>
<b>Index</b>	<b>147</b>

---



# Preface

Python is a versatile programming language that can be used for a wide range of technical tasks—computation, statistics, data analysis, game development, and more. Though Python is easy to learn, its range of features means there are many aspects of it that even experienced Python developers don't know about. Even if you're confident with the basics, its logic and syntax, by digging deeper you can work much more effectively with Python—and get more from the language.

*Python Unlocked* walks you through the most effective techniques and best practices for high performance Python programming—showing you how to make the most of the Python language. You'll get to know objects and functions inside and out, and will learn how to use them to your advantage in your programming projects. You will also find out how to work with a range of design patterns, including abstract factory, singleton, and the strategy pattern, all of which will help make programming with Python much more efficient. As the process of writing a program is never complete without testing it, you will learn to test threaded applications and run parallel tests.

If you want the edge when it comes to Python, use this book to unlock the secrets of smarter Python programming.

## What this book covers

*Chapter 1, Objects in Depth*, discusses object properties, attributes, creation and how calling objects work.

*Chapter 2, Namespaces and Classes*, discusses namespaces, how imports work, class multiple inheritance, MRO, Abstract classes, and protocols.

*Chapter 3, Functions and Utilities*, teaches function definitions, decorators, and some utilities.

*Chapter 4, Data Structures and Algorithms*, discusses in-built, library, third party data structures and algorithms.

*Chapter 5, Elegance with Design Patterns*, covers many important design patterns.

*Chapter 6, Test-Driven Development*, discusses mock objects, parameterization, creating custom test runners, testing threaded applications, and running testcases in parallel.

*Chapter 7, Optimization Techniques*, covers optimization techniques, profiling, using fast libraries, and compiling C modules.

*Chapter 8, Scaling Python*, covers multithreading, multiprocessing, asynchronization, and scaling horizontally.

## What you need for this book

You should have a working installation of Python, preferably greater than 3.4. You can use this with Python 2 as well but the book uses Python 3 and introduces its many new features.

## Who this book is for

If you are a Python developer and you think that you do not fully understand the language, then this book is for you. This book will unlock mysteries and reintroduce the hidden features of Python to write efficient programs, making optimal use of the language.

## Conventions

In this book, you will find a number of text styles that distinguish between different kinds of information. Here are some examples of these styles and an explanation of their meaning.

Code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows: "So, we can change an object's type by changing its `__class__` attribute."

A block of code is set as follows:

```
def __init__(self, name):
    self.name = name
    self._observers = weakref.WeakSet()

def register_observer(self, observer):
    """attach the observing object for this subject
    """
    self._observers.add(observer)
    print("observer {0} now listening on {1}".format(
        observer.name, self.name))
```

When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:

```
self.assertFalse(assign_if_free(mworker, {}))

def test_worker_free(self,):
    mworker = create_autospec(IWorker)
    mworker.configure_mock(**{'is_busy.return_value':False})
    self.assertTrue(assign_if_free(mworker, {}))
```

**New terms** and **important words** are shown in bold. Words that you see on the screen, for example, in menus or dialog boxes, appear in the text like this: "Let's take an example of an object **iC** instance of the **C** class with the **str** and **lst** attributes."

 Warnings or important notes appear in a box like this.

 Tips and tricks appear like this.

## Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book – what you liked or disliked. Reader feedback is important for us as it helps us develop titles that you will really get the most out of.

To send us general feedback, simply e-mail [feedback@packtpub.com](mailto:feedback@packtpub.com), and mention the book's title in the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide at [www.packtpub.com/authors](http://www.packtpub.com/authors).

## Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

## Downloading the example code

You can download the example code files from your account at <http://www.packtpub.com> for all the Packt Publishing books you have purchased. If you purchased this book elsewhere, you can visit <http://www.packtpub.com/support> and register to have the files e-mailed directly to you.

## Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books – maybe a mistake in the text or the code – we would be grateful if you could report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting <http://www.packtpub.com/submit-errata>, selecting your book, clicking on the **Errata Submission Form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded to our website or added to any list of existing errata under the Errata section of that title.

To view the previously submitted errata, go to <https://www.packtpub.com/books/content/support> and enter the name of the book in the search field. The required information will appear under the **Errata** section.

## **Piracy**

Piracy of copyrighted material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works in any form on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at [copyright@packtpub.com](mailto:copyright@packtpub.com) with a link to the suspected pirated material.

We appreciate your help in protecting our authors and our ability to bring you valuable content.

## **Questions**

If you have a problem with any aspect of this book, you can contact us at [questions@packtpub.com](mailto:questions@packtpub.com), and we will do our best to address the problem.



# 1

## Objects in Depth

In this chapter, we will dive into Python objects. Objects are the building blocks of the language. They may represent or abstract a real entity. We will be more interested in factors affecting such behavior. This will help us understand and appreciate the language in a better way. We will cover the following topics:

- Object characteristics
- Calling objects
- How objects are created
- Playing with attributes

### Understanding objects

**Key 1: Objects are language's abstraction for data. Identity, value, and type are characteristic of them.**

All data and items that we work on in a program are objects, such as numbers, strings, classes, instances, and modules. They possess some qualities that are similar to real things as all of them are uniquely identifiable just like humans are identifiable by their DNA. They have a type that defines what kind of object it is, and the properties that it supports, just like humans of type cobbler support repairing shoes, and blacksmiths support making metal items. They possess some value, such as strength, money, knowledge, and beauty do for humans.

Name is just a means to identify an object in a namespace similar to how it is used to identify a person in a group.