

# Pro Python 3

Features and Tools for  
Professional Development

Third Edition

**J. Burton Browning**  
**Marty Alchin**

Apress®

## *Pro Python 3: Features and Tools for Professional Development*

J. Burton Browning  
Oak Island, NC, USA

Marty Alchin  
Agoura Hills, CA, USA

ISBN-13 (pbk): 978-1-4842-4384-8  
<https://doi.org/10.1007/978-1-4842-4385-5>

ISBN-13 (electronic): 978-1-4842-4385-5

Library of Congress Control Number: 2019936454

Copyright © 2019 by J. Burton Browning and Marty Alchin

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr  
Acquisitions Editor: Todd Green  
Development Editor: James Markham  
Coordinating Editor: Jill Balzano

Cover designed by eStudioCalamar

Cover image designed by Freepik ([www.freepik.com](http://www.freepik.com))

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail [orders-ny@springer-sbm.com](mailto:orders-ny@springer-sbm.com), or visit [www.springeronline.com](http://www.springeronline.com). Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail [rights@apress.com](mailto:rights@apress.com), or visit <http://www.apress.com/rights-permissions>.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at [www.apress.com/9781484243848](http://www.apress.com/9781484243848). For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

*This edition is dedicated to Champion Suyaki Mamma Mia of Misibo and her sister Champion Sienna of Olympia of Misibo, two of the best Siamese cats who ever owned me.*

# Table of Contents

<b>About the Authors</b> .....	<b>xv</b>
<b>About the Technical Reviewer</b> .....	<b>xvii</b>
<b>Acknowledgments</b> .....	<b>xix</b>
<b>Introduction</b> .....	<b>xxi</b>
<b>Chapter 1: Principles and Philosophy</b> .....	<b>1</b>
The Zen of Python .....	2
Beautiful Is Better Than Ugly .....	3
Explicit Is Better Than Implicit.....	4
Simple Is Better Than Complex .....	5
Complex Is Better Than Complicated.....	6
Flat Is Better Than Nested .....	6
Sparse Is Better Than Dense .....	8
Readability Counts.....	9
Special Cases Aren't Special Enough to Break the Rules.....	10
Practicality Beats Purity .....	10
Errors Should Never Pass Silently .....	11
Unless Explicitly Silenced.....	14
In the Face of Ambiguity, Refuse the Temptation to Guess.....	15
There Should Be One—and Preferably Only One—Obvious Way to Do It .....	15
Although That Way May Not Be Obvious at First, Unless You're Dutch .....	17
Now Is Better Than Never .....	17
Although Never Is Often Better Than <i>Right Now</i> .....	18
If the Implementation Is Hard to Explain, It's a Bad Idea .....	18
If the Implementation Is Easy to Explain, It May Be a Good Idea.....	19
Namespaces Are One Honking Great Idea: Let's Do More of Those! .....	19

TABLE OF CONTENTS

- Don't Repeat Yourself..... 20
- Loose Coupling..... 21
- The Samurai Principle..... 22
- The Pareto Principle..... 22
- The Robustness Principle..... 23
- Backward Compatibility ..... 25
- Taking It With You..... 26
- Chapter 2: Advanced Basics..... 27**
- General Concepts..... 27
- Iteration ..... 27
- Caching..... 29
- Transparency ..... 30
- Control Flow ..... 32
- Catching Exceptions ..... 32
- Exception Chains ..... 37
- When Everything Goes Right ..... 39
- Proceeding Regardless of Exceptions ..... 41
- Optimizing Loops ..... 43
- The *with* Statement..... 44
- Conditional Expressions ..... 46
- Iteration..... 49
- Sequence Unpacking..... 50
- List Comprehensions ..... 52
- Generator Expressions..... 53
- Set Comprehensions ..... 55
- Dictionary Comprehensions..... 56
- Chaining Iterables Together ..... 56
- Zipping Iterables Together ..... 57
- Collections ..... 58
- Sets ..... 58
- Named Tuples..... 65

Ordered Dictionaries.....	66
Dictionaries with Defaults .....	67
Importing Code.....	68
Fallback Imports.....	68
Importing from the Future .....	70
Using <code>__all__</code> to Customize Imports .....	71
Relative Imports .....	74
The <code>__import__()</code> Function .....	74
The <code>importlib</code> Module.....	78
Exciting Python Extensions: Random Number Beacon at NIST .....	79
How to Install the NIST Beacon Library .....	80
Simple Example to Get a Value.....	80
Example to Simulate Rolling Coin Flipping a Certain # Times and Display Heads or Tails ....	81
Taking It With You.....	81
<b>Chapter 3: Functions .....</b>	<b>83</b>
Arguments.....	84
Planning for Flexibility .....	84
Variable Positional Arguments.....	85
Variable Keyword Arguments .....	87
Combining Different Kinds of Arguments .....	89
Invoking Functions with Variable Arguments .....	92
Passing Arguments.....	93
Introspection.....	95
Example: Identifying Argument Values .....	96
Example: A More Concise Version.....	100
Example: Validating Arguments .....	102
Decorators.....	105
Closures.....	107
Wrappers .....	109
Decorators with Arguments.....	111
Decorators with—or without—Arguments .....	114

TABLE OF CONTENTS

- Example: Memoization ..... 116
- Example: A Decorator to Create Decorators ..... 118
- Function Annotations ..... 120
  - Example: Type Safety..... 121
  - Factoring Out the Boilerplate..... 131
  - Example: Type Coercion..... 134
  - Annotating with Decorators..... 137
  - Example: Type Safety as a Decorator ..... 137
- Generators ..... 143
- Lambdas ..... 146
- Introspection ..... 148
  - Identifying Object Types ..... 149
  - Modules and Packages..... 150
  - Docstrings ..... 151
- Exciting Python Extensions: Statistics ..... 154
  - Install Pandas and Matplotlib ..... 154
  - Make a Text File of Data ..... 155
  - Use Pandas to Display Data..... 155
  - Running Some Data Analysis..... 156
  - Plotting with Matplotlib ..... 157
  - Types of Charts ..... 158
  - Combine Matplotlib with Pandas..... 158
- Taking It with You..... 159
- Chapter 4: Classes ..... 161**
  - Inheritance ..... 161
    - Multiple Inheritance..... 165
    - Method Resolution Order..... 166
    - Example: C3 Algorithm ..... 171
    - Using super() to Pass Control to Other Classes..... 179
  - Introspection..... 184

How Classes Are Created .....	186
Creating Classes at Runtime .....	188
Metaclasses .....	190
Example: Plugin Framework.....	192
Controlling the Namespace .....	196
Attributes .....	198
Properties .....	198
Descriptors .....	201
Methods .....	204
Unbound Methods .....	204
Bound Methods .....	205
Class Methods .....	207
Static Methods .....	209
Assigning Functions to Classes and Instances.....	210
Magic Methods .....	211
Creating Instances.....	211
Example: Automatic Subclasses.....	213
Dealing with Attributes .....	215
String Representations.....	218
Exciting Python Extensions: Iterators.....	221
Taking It With You.....	223
<b>Chapter 5: Common Protocols.....</b>	<b>225</b>
Basic Operations.....	226
Mathematical Operations .....	228
Bitwise Operations .....	234
Variations.....	236
Numbers .....	240
Sign Operations .....	242
Comparison Operations .....	243



TABLE OF CONTENTS

- Iterables ..... 244
  - Example: Repeatable Generators ..... 248
- Sequences ..... 250
- Mappings ..... 257
- Callables ..... 259
- Context Managers ..... 260
- Exciting Python Extensions: Scrapy ..... 262
  - Installation ..... 263
  - Running Scrapy ..... 263
  - Project Setup ..... 263
  - Retrieve Web Data with Scrapy ..... 265
  - View a Web Page via Scrapy ..... 265
  - Shell Options ..... 265
- Taking It With You ..... 267
- Chapter 6: Object Management ..... 269**
  - Namespace Dictionary ..... 270
    - Example: Borg Pattern ..... 270
    - Example: Self-Caching Properties ..... 275
  - Garbage Collection ..... 280
    - Reference Counting ..... 281
    - Cyclical References ..... 283
    - Weak References ..... 287
  - Pickling ..... 289
  - Copying ..... 296
    - Shallow Copies ..... 297
    - Deep Copies ..... 299
  - Exciting Python Extensions: Beautiful Soup ..... 301
    - Installing Beautiful Soup ..... 302
    - Using Beautiful Soup ..... 302
  - Taking It With You ..... 303

<b>Chapter 7: Strings</b> .....	<b>305</b>
Bytes .....	305
Simple Conversion: chr() and ord().....	306
Complex Conversion: The Struct Module.....	307
Text .....	312
Unicode.....	312
Encodings .....	313
Simple Substitution.....	316
Formatting .....	319
Looking Up Values Within Objects .....	321
Distinguishing Types of Strings .....	321
Standard Format Specification.....	322
Example: Plain Text Table of Contents .....	324
Custom Format Specification .....	326
Exciting Python Extensions: Feedparser .....	327
Feedparser .....	327
How to Install.....	328
How to Use .....	328
Taking It With You.....	329
<b>Chapter 8: Documentation</b> .....	<b>331</b>
Proper Naming .....	332
Comments.....	333
Docstrings.....	334
Describe What the Function Does.....	335
Explain the Arguments.....	335
Don't Forget the Return Value .....	336
Include Any Expected Exceptions .....	336
Documentation Outside the Code .....	336
Installation and Configuration .....	337
Tutorials.....	337
Reference Documents .....	337

TABLE OF CONTENTS

- Documentation Utilities..... 338
  - Formatting..... 339
  - Links..... 340
  - Sphinx..... 343
- Exciting Python Extensions: NumPy..... 343
  - Install NumPy..... 344
  - Using NumPy..... 344
  - Working With NumPy Arrays..... 346
  - Statistical Measures..... 347
- Taking It With You..... 347
- Chapter 9: Testing..... 349**
  - Test-Driven Development..... 349
  - Doctests..... 350
    - Formatting Code..... 350
    - Representing Output..... 351
    - Integrating With Documentation..... 353
    - Running Tests..... 354
  - The unittest Module..... 356
    - Setting Up..... 356
    - Writing Tests..... 357
    - Other Comparisons..... 363
    - Testing Strings and Other Sequence Content..... 363
    - Testing Exceptions..... 364
    - Testing Identity..... 366
    - Tearing Down..... 367
  - Providing a Custom Test Class..... 368
    - Changing Test Behavior..... 368
  - Exciting Python Extensions: Pillow..... 369
    - How to Install Pillow (PIL)..... 369
    - Image Display: Determine File Size, Type, and Display It..... 369
    - Image Processing: Crop a Portion of an Image..... 370

Image Processing: Changing Image Orientation.....	370
Image Processing: Filters .....	371
Taking It With You.....	371
<b>Chapter 10: Distribution .....</b>	<b>373</b>
Licensing.....	373
GNU General Public License .....	373
Affero General Public License .....	375
GNU Lesser General Public License.....	376
Berkeley Software Distribution License .....	376
Other Licenses.....	378
Packaging .....	378
setup.py.....	379
MANIFEST.in .....	382
The sdist Command.....	384
Distribution.....	386
Exciting Python Extensions: Secrets Module .....	387
Random Numbers.....	388
Password Generation .....	388
Taking It With You.....	390
<b>Chapter 11: Sheets: A CSV Framework.....</b>	<b>391</b>
Building a Declarative Framework.....	392
Introducing Declarative Programming.....	393
To Build or Not to Build?.....	394
Building the Framework.....	396
Managing Options.....	397
Defining Fields.....	401
Attaching a Field to a Class .....	403
Adding a Metaclass .....	405
Bringing It Together .....	409
Ordering Fields.....	410

TABLE OF CONTENTS

- DeclarativeMeta.\_\_prepare\_\_() ..... 411
- Column.\_\_init\_\_() ..... 414
- Column.\_\_new\_\_() ..... 418
- CounterMeta.\_\_call\_\_() ..... 419
- Choosing an Option ..... 421
- Building a Field Library ..... 421
  - StringField ..... 423
  - IntegerColumn ..... 424
  - FloatColumn ..... 424
  - DecimalColumn ..... 425
  - DateColumn ..... 426
- Getting Back to CSV ..... 431
  - Checking Arguments ..... 433
  - Populating Values ..... 436
  - The Reader ..... 439
  - The Writer ..... 444
- Taking It With You..... 448
- Index..... 449**

# About the Authors

**Dr. J. Burton Browning** earned his doctorate from North Carolina State University. He has conducted research in areas including distance learning, programming, and instructional technology. As a lifelong learner and someone who has interests in topics such as programming, photography, robotics, car restoration, woodworking, hunting, reading, fishing, and archery, he is never at a loss for something to do. The art and joy of serving as a professor suits his inquisitive nature. Dr. Browning's previous publications include works on Cross-Functional Learning Teams (CFLT), the Utopian School (teacher-led school model), computer programming (several languages), open-source software, healthcare statistics and data mining, CNC plasma cutter operation, educational technology, biography, mobile learning, online teaching, and more.

By day, **Marty Alchin** works as a senior software engineer at Heroku, and after that, he writes and codes for fun and community. His blog can be found at <http://martyalchin.com> and he has profiles on many other services under the name Gulopine. In particular, his code can be found on GitHub and his random thoughts are on Twitter. He also accepts tips for his open source work at <https://gittip.com/gulopine>.

# About the Technical Reviewer



**Michael Thomas** has worked in software development for over 20 years as an individual contributor, team lead, program manager, and vice president of engineering. Michael has over 10 years of experience working with mobile devices. His current focus is in the medical sector, using mobile devices to accelerate information transfer between patients and healthcare providers.

# Acknowledgments

This third edition covers some exciting library features available in the Python 3 language. Try the samples and expand on your own projects with what you learn. Do not be afraid to experiment and have fun!

J. Burton Browning

I wouldn't have even started this project if not for the endless encouragement from my lovely wife, Angel. She's been my sounding board, my task manager, my copyeditor, and my own personal cheerleader. There's no way I could do anything like this without her help and support.

I'd also like to thank my technical reviewer, George, for everything he's done to help me out. He's gone above and beyond the limits of his role, helping with everything from code to grammar and even a good bit of style. After enjoying his help on *Pro Django*, I wouldn't have even signed on for another book without him by my side.

Lastly, I never would've considered a book like this if not for the wonderful community around Python. The willingness of Python programmers to open their minds and their code is, I believe, unrivaled among our peers. It's this spirit of openness that encourages me every day, leading me to discover new things and push myself beyond the limits of what I knew yesterday.

We learn by doing and by seeing what others have done. I hope that you'll take the contents of this book and do more with it than what I've done. There's no better reward for all this hard work than to see better programmers writing better code.

Marty Alchin



# Introduction

This third edition expands on Marty's original work. Found in each chapter of this third edition are useful libraries that any Python programmer will find of value. Use what you learn for your own projects and enjoyment!

J. Burton Browning

When I wrote my first book, *Pro Django*, I didn't have much of an idea what my readers would find interesting. I had gained a lot of information I thought would be useful for others to learn, but I didn't really know what would be the most valuable thing they'd take away. As it turned out, in nearly 300 pages, the most popular chapter in the book barely mentioned Django at all. It was about Python.

The response was overwhelming. There was clearly a desire to learn more about how to go from a simple Python application to a detailed framework like Django. It's all Python code, but it can be hard to understand based on even a reasonably thorough understanding of the language. The tools and techniques involved require some extra knowledge that you might not run into in general use.

This gave me a new goal with *Pro Python*: to take you from proficient to professional. Being a true professional requires more experience than you can get from a book, but I want to at least give you the tools you'll need. Combined with the rich philosophy of the Python community, you'll find plenty of information to take your code to the next level.

Marty Alchin

## Who This Book Is For

The goal is to bring intermediate programmers to a more advanced level; we wrote this book with the expectation that you'll already be familiar with Python on some basic level. You should be comfortable using the interactive interpreter, writing control structures, and [using] a basic object-oriented approach.

That's not a very difficult prerequisite. If you've tried your hand at writing a Python application—even if you haven't released it into the wild, or even finished it—you likely have all the necessary knowledge to get started. The rest of the information you'll need is contained in these pages.