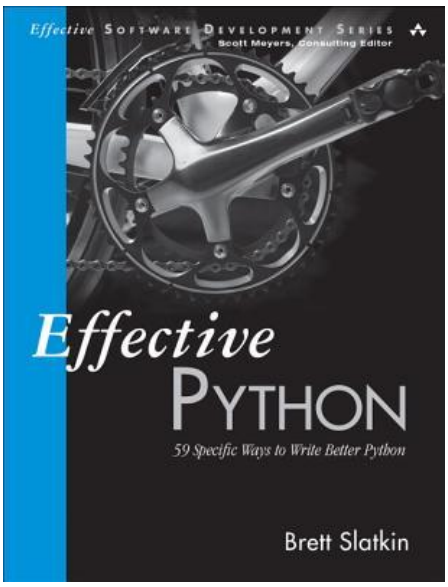

Effective Python: 59 Specific Ways to Write Better Python

Brett Slatkin

Effective Python: 59 Specific Ways to Write Better Python

Brett Slatkin

Effective Python: 59 Specific Ways to Write Better Python Brett Slatkin
Each item in Slatkin s "Effective Python" teaches a self-contained lesson with its own source code. This makes the book random-access: Items are easy to browse and study in whatever order the reader needs. I will be recommending "Effective Python" to students as an admirably compact source of mainstream advice on a very broad range of topics for the intermediate Python programmer. " Brandon Rhodes, software engineer at Dropbox and chair of PyCon 2016-2017" It s easy to start coding with Python, which is why the language is so popular. However, Python s unique strengths, charms, and expressiveness can be hard to grasp, and there are hidden pitfalls that can easily trip you up. " Effective Python " will help you master a truly Pythonic approach to programming, harnessing Python s full power to write exceptionally robust and well-performing code. Using the concise, scenario-driven style pioneered in Scott Meyers best-selling "Effective C++, " Brett Slatkin brings together 59 Python best practices, tips, and shortcuts, and explains them with realistic code examples. Drawing on years of experience building Python infrastructure at Google, Slatkin uncovers little-known quirks and idioms that powerfully impact code behavior and performance. You ll learn the best way to accomplish key tasks, so you can write code that s easier to understand, maintain, and improve. Key features include
Actionable guidelines for all major areas of Python 3.x and 2.x development, with detailed explanations and examples Best practices for writing functions that clarify intention, promote reuse, and avoid bugs Coverage of how to accurately express behaviors with classes and objects Guidance on how to avoid pitfalls with metaclasses and dynamic attributes More efficient approaches to concurrency and parallelism Better techniques and idioms for using Python s built-in modules Tools and best practices for collaborative development Solutions for debugging, testing, and optimization in order to improve quality and performance
"

Effective Python: 59 Specific Ways to Write Better Python Details

Date	:	Published March 8th 2015 by Addison-Wesley Professional (first published February 1st 2015)
ISBN	:	9780134034287
Author	:	Brett Slatkin
Format	:	Paperback 227 pages
Genre	:	Computer Science, Programming, Science, Technology, Coding, Nonfiction, Technical

From Reader Review Effective Python: 59 Specific Ways to Write Better Python for online ebook

Burke Fitzpatrick says

This is a great book. I read it once to see what it was about, but I'm going to have to reread it a few times to get the most out of it. I found several useful things on the first read though. Great info on the GIL in chapter 5.

Steve says

A good, but not perfect, text for the next level of Python programming. Unlike the Effective C++ books this is clearly based on (and which I loved), I found the book relatively slow going and the examples- particularly the trickier ones - a bit contrived. I am an experienced programmer who's new to Python and this wasn't a good fit.

Frank says

This book is a good overview of some tricks using Python 2 & 3. The author's examples, however, are typically not at all intuitive. I would recommend interested readers looking for another more intuitive book if she/he is interested in advancing their Python skill. For beginners, this is not a beginner friendly book any way, so I wouldn't bother.

Eric Casteleijn says

There is a surprising amount of meat to this book, considering how thin it is. I would recommend this book to Python developers of all experience levels, but note that if you are a beginner, some of the later chapters might be a little out of reach. Similarly to more experienced developers, nothing in the first two chapters should come as big news, except maybe where you disagree with some of the finer points ;).

On the whole I think it's one of those books I will reach for from time to time, for elegant examples of how to use some of the less common parts of the language, or just to be inspired by some very nice ideas.

The high points for me were the chapters on metaprogramming, with some of the clearest, most concise explanations on that topic I have so far encountered.

Conor says

I usually prefer books that give a conceptual overview versus those that are lists of specific tips, however this

book seemed to accomplish both! It is a valuable intermediate resource that immediately provides ways to structure a new project, clean up an old one, or understand deeper aspects to how python works.

Anton Antonov says

A must-have when working with Python 2/3 codebases.

Although dated by now, it's still relevant except for "Concurrency and Parallelism", if you're writing Python 3.

With the newest improvements in Python 3 regarding asyncio and futures, there needs to be a revisited version of the book.

Either way, highly recommended one-two times read.

Along with "The Python 3 Standard Library by Example (Developer's Library)" the book compliment each other to an extent, however they both fall short at "concurrency and parallelism", where the Python 3 documentation excels.

Ulas Tuerkmen says

For a long time, there was a big gap in the intermediate to advanced Python book space. There was Tarek Ziade's Expert Python Programming,a really good book that started aging as Python moved on. There were books on individual topics such as network programming or web development, but these were limited in scope and could not be recommended to people looking to advance their general knowledge of the language. Fortunately, this situation has now drastically improved, thanks to a number of new publications, among which this one can be counted. Others are Fluent Python and the Python Cookbook.

Effective Python consists of 59 concrete recipes for writing, well,effective code. The recipes are very varied, and range from the simplest things like looping and slicing, to more complex topics such as class metaprogramming. Every tip is discussed in depth with all relevant aspects, and there is ample sample code. The recipes are organized roughly around different topics such as Metaclasses and Attributes, Classes and Inheritance, Concurrency and Parallelism etc. The Python target is 3, as all code samples adhere to its syntax,but any differences to Python 2 are explicitly stated and discussed. Some recipes go more towards patterns: e.g. "Register class existence with metaclasses" explains how to use the metaclass feature to create a class registry by only subclassing. These chapters are really useful as they give concrete examples of how abstract features can be used to accomplish sophisticated functionality. Others remind the programmer to prefer certain features of the language in preference to others (e.g. when dealing with circular imports),whereas others deal with what could be called "community" aspects of Python prgoramming, as in how to write packages that are good citizens of the Python world of programming.

Despite having already read sample code, blog posts or book chapters on most of the topics, I was surprised by how useful some examples are, and the detail in which they are discussed. One example is the descriptor chapter, which explains a rarely used feature of the language. The chapter is nicely bookended by discussions of the property decorator and the special methods for attribute access,giving the user a very good

overview of the various ways of controlling attribute access. Another really well-written chapter is the one on how to use queues to coordinate between threads. This chapter nicely explains how to properly use the Queue class, by first giving an example of how a job queue can be built inefficiently, and then explaining a much more efficient and elegant counter-example.

The Queue chapter is a part of the Concurrency and Parallelism topic. This topic, despite the really nicely executed treatment of threads,unfortunately ends up being the weakest part of the book, due to the very simple reason that there was so much change in Python in this area. The new asyncio package is not discussed at all, and the coroutines that are at the heart of this package are explained at a very low level, without a clear sense of why they would be useful for concurrency and asynchronous IO. As I mentioned in the beginning, this book is a very valuable contribution to the Python bookshelf, and is highly recommended to developers of all levels who want to refresh and cement their knowledge of this wonderful language.

John says

I've observed a curious inverse relationship between the length of computer books and their quality. As with program code shorter is usually better. The canonical examples remain the original C Programming book and the first LaTeX Manual. Both of these terse masterpieces put the blundering, bloated, tree killing, screen-shot-ware that pollutes our bookshelves to shame. Hence, when picking software books I first count pages: anything above two hundred pages is suspicious. Effective Python weighs in at 216 pages. It could be a bit shorter and bit a better but overall this is an excellent book. In the age of search engines and StackOverflow we don' t need excessive detail. We need ideas and Effective Python presents a few good ones.

Isaac Lockett says

Some really good advice in here, but the learning curve is strange. It seems to start with very simplistic ideas, and then steps up quite suddenly to ways to do better decorators. It then later on explains decorators but I didn't come away with a clearer idea of what they were, but I highly doubt that many people will have a full understanding of decorators prior to the book, but don't for example know how to add a __repr__() method to a class.

Also it spends way too long on APIs, which made the book feel too specific for its title. Apart from that, a worthwhile read

---

Tony Poerio says

Really good Python book. It's not as in-depth as something like *Fluent Python*, but it has a ton of helpful tips, presented in an easily digestible format.

If you're reading only *one* Python book, go with Fluent Python. But Slatkin's book is a great addition to any library.

I'd call this a true intermediate-level book on the language.

Bithika says

I've been doing Python sporadically for the last couple of years, but have mostly stuck to the shallow end. Recently, I've been trying to make some dodgy Python code a bit less dodgy and suspected that I wasn't using Python in the right way. So I picked up this book, and I really wish I'd read it sooner!

Effective Python belongs to the "Effective Software Development" series, which was conceived by Scott Meyers, author of Effective C++, to provide guides on best practices for a variety of languages. (Anyone who likes the font and formatting of Scott Meyer's C++ series, will be pleased to know that this book looks the same.)

This book explains many Python concepts, in a very simple and clear way. I guess that is the mark of a good technical book.

The highlights for me were (in no particular order):

- good ways to use try/except/else/finally
- inheriting from standard container classes (I had never even considered it)
- how best to use threads and Queue objects in a pipeline of jobs
- what is actually happening in a circular import
- an overview of the built-in algorithms
- how to use Decimal for precision
- how to use metaclasses, decorators and properties

Actually, every section has some pearls of wisdom. The whole book was a highlight for me! The items cross reference each other so you can see how to combine concepts in the best way. It is a book I can see myself referring to again and again, and has made me like Python a lot more. Highly recommended.

Pavel Karateev says

Nice short book full of Python tips and tricks. There're 59 subchapters, each one is independent. Topics vary from OOP to concurrency and parallelism.

Not intend for beginners, but a nice and easy home reading for an experience Python developer.

Pros:
- Concise
- Best practises
- Relevant topics

Cons:
- Too short
- Not realy an indepth material
- Some topics are too easy and well known (virtualenv, unittest)

Bruce says

This is a decent python book. Covers quite a few Python features and how to use them well. I enjoyed it and, after some practice, feel better-able to express myself elegantly in Python.

Where the book falls short is in having enough examples and and practice problems problems to know why/when to apply some of the more advanced Python features. Also, this seems to be target at Python 3.4/2.7, missing some newer Python 3 features.

Jascha says

If you browse Amazon's catalog for Python books, you end up with several pages of titles that introduce you to the language, those that get you from hello world up to classes and a little taste of the standard library. What a struggle for those that already know what a closure is to find something worth reading! Brett's book is one of those that experienced developers hope to see showing up as on top of the results instead: a title that covers those lesser used features that teach you to write better, robust code.

This thin book, 250 pages only, is split into 59 recipes, short and concise reviews of real life scenarios that every Python developer faces. Each focuses on a specific problem and starts with a brief overview . Next, we find an inefficient, not Pythonic solution that most of the intermediate programmers would come up with to solve it. What follows is a discussion that gets the reader, step by step, from this initial solution to an elegant and robust one. Brett clearly explains the benefits and the issues raised by each intermediate solution. Finally, he sails us to the Pythonic way introducing features and techniques.

As stated, the book is not meant to be read by a beginner. If the reader does not have a strong Python knowledge, he will struggle. Concepts like comprehension lists, closures and decorators must be well understood already.

The small size of the book could fool the reader. It's thin, but intense, plenty of things to learn. If the reader doesn't get lost, by the time he gets to the back cover, he will certainly start developing better code.

A real pearl. Every Python developer should jealously own a copy.

Suggested readings:
Learning Python Design Patterns
Learning Cython Programming

As usual, you can find more reviews on my personal blog: http://books.lostinmalloc.com Feel free to pass by and share your thoughts!

Yehia Abo el-nga says

A 200 pages summary of best practices of Python coding in production. I like the format of the book. The

author mentions 59 different practices, highlights the problems with and proposes a better approach. He doesn't just put the "best practice" out there. He takes the reader through a thought process of why that might be bad through illustration by example. I find it extremely helpful for people writing Python for anything production-level.
