# Python书籍推荐

从2011年5月买了第一本《Python学习手册（第四版）》开始，我阅读过大量和Python有关的纸质书和开源图书。为什么要买书来看？我认为不外乎两个原因：有趣和能学到东西。技术书肯定不会太有趣，那么最重要的就是能学到东西。市面上Python相关的书相当多，但是有些内容陈旧或者不符合国情，经常能看到并非开发第一线的人写或者翻译的书，这些书显然价值就要低一些；其次是同质化和向入门级别靠拢，市面上关于Python入门或者教授语法知识的书不少，而再深入一点的就很匮乏了。还有一点，现在为了提高书的销量，书名都起的非常有破坏力，很吸引眼球，可是内容完全不够书名的档次。

不知道一些回答推荐书籍问题的人是怎么想的，反正话说起来很廉价，又不用负责。但是会给看这些答案的人误解，甚至要白白浪费大家的钱和阅读时间，我非常乐于给「没有帮助」。

知乎上已经有一些问题，也有很多人推荐。我没有达到流畅的阅读英文原版的能力，所以本文更接地气一点，介绍的是中文书籍。本文将从如下三部分推荐技术书籍。

## Python初学者

如果你之前一点编程经验都没有，先看如下两本：

1.《简明 Python 教程》， 英文名《A Byte of Python》，现在有Python 3的版本（[Introduction ยท A Byte of Python](https://link.zhihu.com/?target=https%3A//python.swaroopch.com/)）了 。最简洁易懂的Python书了。  
2.《"笨办法"学Python》，英文名《Learn Python the Hard Way...》，个人感觉是Python书籍中看起来最轻松的，他的习题非常好，初学者一定要练练：好遗憾我初学的时候没有看到它。

如果你有其他语言的基础，初学Python首先的是要熟悉Python语法和语言的关键内容，推荐如下三本书（当然上面那有兴趣也是可以作为过渡看一看的）：

1.《Python学习手册》。应该是对Python介绍最全面的书了，包含了绝大多数Python语言的关键内容。

2. 《Python核心编程（第二版）》。 需要注意，《Python核心编程（第二版）》是经典的Python指导书，出版于2008年，有很多内容已经不再适用现在的环境，所以以了解概念和原理为主。现在已经出版了一本《Python核心编程 第3版》，**这是《Python核心编程 第一版》卷二的第三版**。所以不要以为这是再版的《Python核心编程（第二版）》， 具体的内容可以查看「[正在预售的《Python核心编程》到底是不是第三版？](https://link.zhihu.com/?target=http%3A//codingpy.com/article/core-python-name-confusion/)」。

3. 《Python语言及其应用》。感谢[@PiGu](https://www.zhihu.com/people/9fbdce82fec4e374a860fd574423a4cf)提醒，也可以作为新手入门的书籍。

PS: 《Python基础教程》。 有知友私信我，提到这本书，我之前看过一遍，觉得很一般就如实说了，想想觉得得对大家负责，这几天特意翻了一遍。在这里再唠唠，它虽然是一个经典的入门教程，写作风格也相对轻松幽默，但是由于本书写作于2010年，书中有大量内容已经过时，所以**不推荐！**

说到这里，很伤心。因为现在大部分市面上见到的书都存在过时的问题。一是翻译的书落地需要时间，另外一个是国内的互联网技术环境发展并不比国外发展慢。甚至于我们在很多领域都要超前很多，很多外来的内容对我们来说本身就是过时的。而国内优秀的Python书籍又非常匮乏。

## Python进阶者

当你对Python有了一定了解，在实际的工作中用过一段时间，那么就应该考虑进阶相关的书籍了。进阶推荐的书有如下几本：

1. **《Python高级编程》。**这本书副标题叫做「Best practices for designing, coding, and distributing your Python software」，其实感觉更适合这个名字。它是10年出版的，时间特蛮长了，但是我依然要首先推荐，因为这本书可以说对我影响非常大，以至于当时我读了好几遍，每次读完都觉得有很大收获，帮助我之后的Python之路少走了很多弯路。

2. **《Python高手之路》。**本书的作者是开源项目OpenStack的PTL（项目技术负责人）之一的[Julien Danjou](https://link.zhihu.com/?target=https%3A//julien.danjou.info/)。我一直说，一定要看书及其技术评审的作者的经历和经验，这本书虽然只有2百多页，但是本书对项目的结构设计，对模块和库的管理，如何编写文档，如何分发，以及如何通过虚拟环境对项目进行测试都有自己的见解。此外，本书还涉及了很多高级主题，如性能优化、插件化结构的设计与架构、Python 3的支持策略等。值得一提的，本书的评审包含Python核心开发者Nick Coghlan、Doug Hellmann以及Victor Stinner等，所以书的作者是`圈内人`。

3. **《Python编程实战：运用设计模式、并发和程序库创建高质量程序》。**这本书有Doug Hellmann作序。书中的设计模式、并发变成和扩充Python都是非常通用的技术，是难得深入讲解如此深入的一本好书。

4. **《Python标准库》。**这本书是Doug Hellmann写的，介绍Python 2标准库的使用，最初是[PYMOTW](https://link.zhihu.com/?target=https%3A//pymotw.com/2/py-modindex.html)网站上的一系列文章，后来成书。现在作者已经出版了Python 3版本的《The Python Standard Library by Example》。这是学好Python的必经之路，如果英文不过关，买本书花一段时间好好看这些标准库的使用方法，对于未来写代码和阅读开源项目以及其他同事写的项目都有非常大的帮助。

5. **《Effective Python:编写高质量Python代码的59个有效方法》**。 这本书虽然不太知名，但是相信大部分开发者看完都会有所收获，可以提高Python代码质量的。

6. 《[Python Web开发实战](https://link.zhihu.com/?target=https%3A//www.amazon.cn/Python-Web%25E5%25BC%2580%25E5%258F%2591%25E5%25AE%259E%25E6%2588%2598-%25E8%2591%25A3%25E4%25BC%259F%25E6%2598%258E/dp/B01L8NVIC6)》。好吧，这是一个广告！这是我写的一本书，不说太多，有兴趣的可以去看介绍，尤其是看作序和推荐的人都有谁就好了 ^.^

## Python 3相关

Python 3是未来这毋庸置疑，市面上也有一些书以Python 3作为卖点。但是目前为止，我还没有发现一本好的中文书籍。期待中...

不要纠结 [Python 2 or Python 3](https://zhuanlan.zhihu.com/p/22405782)。

最后分享一个我2年前在公司做过的《[Python高级编程](https://link.zhihu.com/?target=http%3A//www.dongwm.com/archives/fen-xiang-%255B%253F%255D-ge-zhun-bei-gei-gong-si-jiang-pythongao-ji-bian-cheng-de-slide/)》的视频。

正文完

# 教你阅读Python开源项目代码

[[https://pic4.zhimg.com/258463ec3_xs.jpg](https://www.zhihu.com/people/dongweiming)董伟明](https://www.zhihu.com/people/dongweiming)· 11 天前

**注：本专栏文章未经允许请勿转载。**

知乎上有不少人问和关注阅读开源项目代码这个问题，比如「[初学 Python，有哪些 Pythonic 的源码推荐阅读？](https://www.zhihu.com/question/20336475)」、「[github中适合新人阅读的Python项目有哪些?](https://www.zhihu.com/question/30398596)」、「[如何去阅读并学习一些优秀的开源框架的源码？](https://www.zhihu.com/question/26766601)」等。本文将从推荐可阅读的Python项目和如何阅读者2个方面聊聊这个话题。

阅读Python开源项目代码主要有如下三个原因：

1. 在工作过程中遇到一些问题，Google和StackOverFlow等网站找不到解决办法，只能去翻源码。

2. 对某些项目或者方向非常感兴趣，希望深入。

3. 学习遇到瓶颈需要汲取开源项目的经验和用法来做提高。

**没有目的的阅读开源项目就是耍流氓。**浪费了时间，但是能学到的东西也很少。怎么样根据自身情况去阅读呢？

1. 和兴趣以及工作契合。举个例子，工作中没有机会用到Celery又不是想自己造个轮子，读它的源码做什么？所以要从平时能接触到的那些项目中选取。比如我，我肯定不去看Django的代码，因为日常工作基本遇不到，遇到了现翻就好了。

2. 一个方向只看一两个典型的就可以了。比如Web框架，我只看过Bottle和Flask的源码（其实之前也看过Django的，只是浅尝辄止），而且看Bottle已经是好几年前的事情了。并不是堆的多了更好，有时候反而选择太多会懵。

3. 不同技术阶段的选择代码量、复杂度不一样的项目，下面会具体推荐。

4. 清楚自己看代码的目的。就是你看代码是想了解人家怎么设计、调试BUG、还是只是想学习正确的编程用法呢？其实没有必要细抠每个代码细节，有时候当黑盒看，知道输入输出就可以了。

## 我个人的喜好

和工作中看别人代码差不多，基本每个人、每个项目、每个团队都有自己写代码的风格，比如变量命名风格、某些语言特性使用方式、代码规范要求、目录风格等，其实开源项目的作者也是一样。看代码，如看人（团队）。 首先介绍下我的喜好（排名分先后）：

1. [kennethreitz](https://link.zhihu.com/?target=https%3A//github.com/kennethreitz)。requests和python-guide作者。他还有一个非常励志的故事，有兴趣的可以看 [谁说程序员不是潜力股？](https://zhuanlan.zhihu.com/p/22332669)

2. [mitsuhiko](https://link.zhihu.com/?target=https%3A//github.com/mitsuhiko)。flask、Jinja2、werkzeug和flask-sqlalchemy作者。

3. [sigmavirus24](https://link.zhihu.com/?target=https%3A//github.com/sigmavirus24)。flake8、pycodestyle（原pep8）、requests、urllib3等项目的主要贡献者和维护者。

4. [ask](https://link.zhihu.com/?target=https%3A//github.com/ask)。Celery及相关依赖的作者。

5. [ajdavis](https://link.zhihu.com/?target=https%3A//github.com/ajdavis)。mongo-python-driver（pymongo）、tornado等项目的主要贡献者。

6. [bitprophet](https://link.zhihu.com/?target=https%3A//github.com/bitprophet)。fabric、paramiko（Python的ssh库）作者。

前2个是公认的Python领域代码写的最好的、最有创意的工程师。

## 初学者推荐阅读项目

初学者可以先阅读一些代码量比较少的，最好是单文件的项目：

1. [GitHub - kennethreitz/pip-pop: Tools for managing requirements files.](https://link.zhihu.com/?target=https%3A//github.com/kennethreitz/pip-pop)

2. [GitHub - kennethreitz/envoy: Python Subprocesses for Humans™.](https://link.zhihu.com/?target=https%3A//github.com/kennethreitz/envoy)

3. [GitHub - kennethreitz/records: SQL for Humans™](https://link.zhihu.com/?target=https%3A//github.com/kennethreitz/records)

4. [GitHub - mitsuhiko/pluginbase: A simple but flexible plugin system for Python.](https://link.zhihu.com/?target=https%3A//github.com/mitsuhiko/pluginbase)

5. [GitHub - mitsuhiko/pipsi: pip script installer](https://link.zhihu.com/?target=https%3A//github.com/mitsuhiko/pipsi/)

6. [GitHub - mitsuhiko/unp: Unpacks things.](https://link.zhihu.com/?target=https%3A//github.com/mitsuhiko/unp)

7. [GitHub - chrisallenlane/cheat](https://link.zhihu.com/?target=https%3A//github.com/chrisallenlane/cheat/)

8. [GitHub - jek/blinker: A fast Python in-process signal/event dispatching system.](https://link.zhihu.com/?target=https%3A//github.com/jek/blinker)

9. [GitHub - mitsuhiko/platter: A useful helper for wheel deployments.](https://link.zhihu.com/?target=https%3A//github.com/mitsuhiko/platter/)

10. [GitHub - kennethreitz/tablib: Python Module for Tabular Datasets in XLS, CSV, JSON, YAML, &c.](https://link.zhihu.com/?target=https%3A//github.com/kennethreitz/tablib)

看代码主要是了解别人写代码的方式，语法实践这些内容。看完之后，你可以针对这些项目能解决的问题自己写个项目，写完之后和上述项目去对比一下，看看哪些方面做的不好。

## 进阶阅读项目

进阶的时候就要阅读一些相对复杂的项目，它们能帮助你提升Python编程技巧：

1. [faif/python-patterns](https://link.zhihu.com/?target=https%3A//github.com/faif/python-patterns)。使用Python实现一些设计模式的例子。

2. [pallets/werkzeug](https://link.zhihu.com/?target=https%3A//github.com/pallets/werkzeug)。flask的WSGI工具集。其中包含了实现非常好的LocalProxy、cached\_property、import\_string、find\_modules、TypeConversionDict等。

3. [bottlepy/bottle](https://link.zhihu.com/?target=https%3A//github.com/bottlepy/bottle)。阅读一个Web框架对Web开发就会有更深刻的理解，flask太大，bottle就4k多行，当然如果你有毅力和兴趣直接看flask是最好了的。

4. [msiemens/tinydb](https://link.zhihu.com/?target=https%3A//github.com/msiemens/tinydb)。了解用Python实现数据库。

5. [coleifer/peewee](https://link.zhihu.com/?target=https%3A//github.com/coleifer/peewee)。了解ORM的实现。

6. [pallets/click](https://link.zhihu.com/?target=https%3A//github.com/pallets/click)。click已经内置于在flask 0.11里，提供命令行功能，值得阅读。

7. [mitsuhiko/flask-sqlalchemy](https://link.zhihu.com/?target=https%3A//github.com/mitsuhiko/flask-sqlalchemy)。了解一个flask插件是怎么实现的。

除此之外Web开发者可以阅读一些相关的项目：

1. [runscope/httpbin](https://link.zhihu.com/?target=https%3A//github.com/Runscope/httpbin)。使用flask，网站是[httpbin(1): HTTP Client Testing Service](https://link.zhihu.com/?target=http%3A//httpbin.org/)。

2. [jahaja/psdash](https://link.zhihu.com/?target=https%3A//github.com/Jahaja/psdash)。使用flask和psutils的获取Linux系统信息的面板应用。

3. [pallets/flask-website](https://link.zhihu.com/?target=https%3A//github.com/pallets/flask-website)。 flask官方网站应用。

4. [pypa/warehouse](https://link.zhihu.com/?target=https%3A//github.com/pypa/warehouse)。如果你使用pyramid，这个[新版的PYPI网站](https://link.zhihu.com/?target=https%3A//pypi.org/)，可以帮助你理解很多。

当然，2个学习flask重要的资源必须爆一爆：

1. [GitHub - realpython/discover-flask: Full Stack Web Development with Flask](https://link.zhihu.com/?target=https%3A//github.com/realpython/discover-flask)。

2. [The Flask Mega-Tutorial](https://link.zhihu.com/?target=http%3A//blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world)。 这个就是《Flask Web开发：基于Python的Web应用开发实战》的原始博客。

## 500lines

推荐一个非常厉害的项目 [GitHub - aosabook/500lines: 500 Lines or Less](https://link.zhihu.com/?target=https%3A//github.com/aosabook/500lines), 它里面包含了22个由该领域的专家完成，用不到500行的代码实现一个特定功能的子项目。连Guido van Rossum都亲自来写基于asyncio爬虫了，Nick Coghlan、ajdavis也出场了。更具体的介绍可以看[Python 的练手项目有哪些值得推荐？ - 小小搬运工的回答](https://www.zhihu.com/question/29372574/answer/88624507)。

## 怎么阅读开源项目源码

我基于个人兴趣和日常开发需要阅读过一些开源项目，我列一些比较通用的经验和感悟：

1. **不要畏惧**。我发现大家都经常会感叹XXX强大，YYY流行，无形中你会把它放在一个不可触及到的地位，感觉它很难，而令自己不敢去挑战它。其实是人就会产出bug，假如你发现它有问题，就应该抓住机会去验证它。这个过程中，它的神秘感也就消失了，有过这么几次经验你就有信心了。其次是不要怕你提交的PR被拒绝。这是非常正常的，我有很多PR是被拒绝的，尤其是给标准库提交的Patch，绝大多数都被拒绝了。

2. **带着问题去阅读代码**，这也是我认为最有效的方式。这会让你在阅读时候有个主线，比较有针对性。

3. **断点调试。**在Python代码中使用pdb一般不太好使，因为代码复杂的话，这种断点需要你使用多个n跳到对应的位置，我一般都是先抛出异常，然后使用pdb的up/down/n等命令调试。当然在目标位置添加一些print日志或注释部分代码然后直接使用exit()退出也是可以的。

4. **善用文档，官网例子**。阅读一个项目一开始会有点无从下手，那么就先好好这些内容，它们一般都是作者表达这个项目的第一个入口。quickstart、tutorial等内容中的最小化的例子其实就是最好的阅读入口，先去看这些引用的模块和调用的对应方法或者函数的对应实现，从下至上去阅读。

5. **理解作者的思考方式。**不同的项目要有不同的思考方式来阅读，不要拧着自己的习惯去阅读，这样会很累，得尝试接受别人的观点，甚至于改变自己。

6. **阅读项目的早期版本**。一些项目随着时间演进已经非常复杂了，读起来有难度，那么你可以回到项目的早期版本上，先去看相对简单地版本，然后设置几个时间点或者版本节点，渐进的来阅读。

7. **记忆并绘制项目架构。**项目就是一堆代码的组合，除了学习编程技巧，还要了解项目的架构决策，这对于未来自己写大型项目非常用用。这种理解越补充，你会对它就越来越清晰。

8. **参与**。如果只是看和思考而不参与，其实效果大打折扣。我建议要参与一些issue讨论、向作者提问、提PR添加新的功能、帮助改善文档等等。很多时候你理解的其实和作者是有冲突的，无论是你说服他和被他说服，对自己的成长都是很有帮助的。

# 我的Python订阅列表

[[https://pic4.zhimg.com/258463ec3_xs.jpg](https://www.zhihu.com/people/dongweiming)董伟明](https://www.zhihu.com/people/dongweiming)· 4 天前

除了经常看[Github Trending](https://link.zhihu.com/?target=https%3A//github.com/trending)，我还会通过订阅列表等渠道看一些程序员做的有意思的事和分享。之前这个专栏聊过「[Python书籍推荐](https://zhuanlan.zhihu.com/p/22198827)」和「[教你阅读Python开源项目代码](https://zhuanlan.zhihu.com/p/22275595)」，今天我们继续，聊聊我的订阅列表和经常去的网站。

## 1. [Planet Python](https://link.zhihu.com/?target=http%3A//planetpython.org/)

搜集了非常多的Python相关的博客，其中有很多Python的核心开发者 如Georg Brandl、Nick Coghlan、Ajdavis、Benjamin Peterson、David Goodger等。首页也展示了很多有深度的、非常核心的文章。我平时还会关注它上面的[stackoverflow python report](https://link.zhihu.com/?target=http%3A//python-weekly.blogspot.com/2016/09/xxxvii-stackoverflow-python-report.html)，里面有一些比较有特点和深度的Python相关问题。

## 2. [Import Python Weekly](https://link.zhihu.com/?target=http%3A//importpython.com/newsletter/)/[Python Weekly](https://link.zhihu.com/?target=http%3A//www.pythonweekly.com/)/[Pycoder’s Weekly](https://link.zhihu.com/?target=http%3A//www.pycoders.com/)

这是3个Python的订阅列表，每周都会总结一些最近热点的Python新闻、项目和讨论，也会包含一些挖掘到的好的Python相关的文章，经常可以翻到宝贝。

## 3. [开发者头条](https://link.zhihu.com/?target=http%3A//toutiao.io/)

我认为这是国内做的最好的程序员分享的平台，建议同学们都去关注一下，里面有一些独家号/或者团队号都有干货分享。内容不仅限于Python。我个人关注了「架构之美」、「运维就精选」、「前端外刊评论」、「腾讯AlloyTeam」等独家号。每天它的精选内容都可以作为学习的参考，我最近看的「[大型web系统数据缓存设计](https://link.zhihu.com/?target=http%3A//toutiao.io/posts/324imi)」、「[使用纯 CSS 实现 500px 照片列表布局](https://link.zhihu.com/?target=http%3A//toutiao.io/j/z7484t)」和「[各大互联网公司架构演进之路汇总](https://link.zhihu.com/?target=http%3A//toutiao.io/j/r2b31c)」等文章都让我收获颇多。

## 4. [Python 酷站导航](https://link.zhihu.com/?target=http%3A//simple-is-better.com/sites/)

很多人不知道怎么找Python老鸟的博客，我给你们一个秘诀。这个网站虽然不咋运营了，但是这个导航页有个「[Python 技术博客](https://link.zhihu.com/?target=http%3A//simple-is-better.com/sites/blogs) 」的板块，里面这些博客添加的越早就越'老'，你看我才排到86位。这些博客里面很多都很有名，我也会经常去看看他们的状态（虽然不少网站都不能访问了），比如「[The5fire的技术博客](https://link.zhihu.com/?target=http%3A//www.the5fire.com/)」和「[编程派](https://link.zhihu.com/?target=http%3A//codingpy.com/)」。