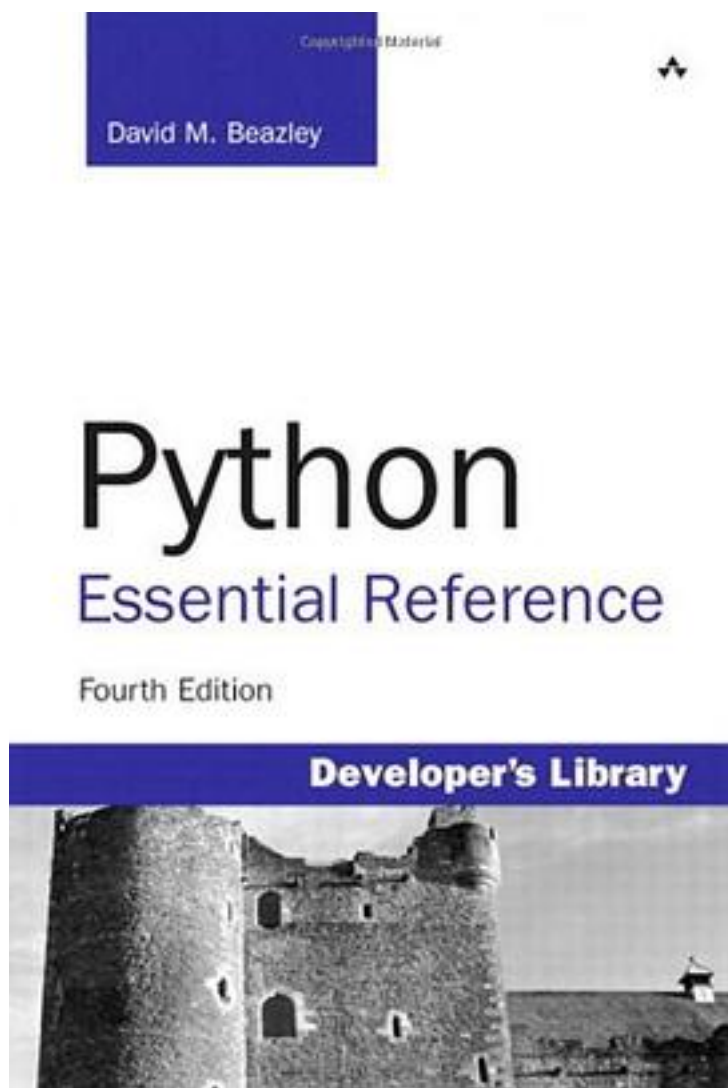


Python Essential Reference



[Python Essential Reference_ 下载链接1](#)

著者:David M. Beazley

出版者:Addison-Wesley Professional

出版时间:2009-07-19

装帧:Paperback

isbn:9780672329784

Amazon.com Review

Every so often a book comes along that makes you ask yourself, "Gee, when was the last time I had

my eyes checked?" David M. Beazley's Python: Essential Reference is just such a book. Condensing thousands of pages of Python online documentation into a compact 319-page softcover, Beazley and his editors used the old-college trick (often performed in reverse) of dicking with the font size to meet a putative page-limit requirement. The result is a truly condensed product fit for the occularly well-adjusted (*nota bene*).

Beazley's subject is Python, a full-featured, freely-redistributable, POSIX-compliant (platforms include Linux, Unix, Macintosh, and Windows) scripting language that is based on object-oriented design principles. As advertised, Beazley's source release (1.5.2) is available from an unfortunately slow server at www.python.org. The installation under Linux (Redhat 5.2) proceeded without incident.

Beazley holds true to his catalogic purpose: fully 230 pages are formatted as technical appendices and indices covering the standard litany: built-in function syntax, database features, OS-level interfaces, Internet interfaces, and compiling/profiling/debugging. All references are fully annotated and illustrated with example source code that runs from a couple of lines to a couple of pages. In lock step with competing scripting languages, Python is extensible and embeddable in C and C++, and with blitzkrieg efficiency, Beazley summarizes these crucial practical issues in the final 30 pages. Python users who are tired of chasing questions through hyperlinked online documents will benefit from the expansive random-access index.

Python the book captures the orderliness of Python the language. Beazley begins with an 86-page précis of Python in the fashion of Kernighan and Ritchie: too brief for a newbie tutorial but enough to propel old hands into a scripting language that aspires to the elegance of a compiled language.

Indeed, it is a byte-compiling language. The line `bytecode=compile("some_python_script","", 'exec')` creates 'bytecode' as a token executed by `exec` bytecode. But a five-minute investigation through Beazley's book does not describe how 'bytecode' can be written into a separate executable file. If writing the byte-compiled code to a file is not possible, Python suffers from the limitations of other scripting languages: the executable is the source and cannot be hidden from the user, at least not without some difficulty. Despite its extensibility, embeddability, and pleasing architecture, Python is like other scripting languages: appropriate for solving small nonproprietary problems.

Those familiar with more established scriptors like Perl may ask, "Why Python?" Unlike Perl, Python is a product of the fully object-oriented (OO) era, and its constructs reflect design principles that aspire beyond keystroke shortcuts of the succinct-but-often-arcane Perl. Python creator Guido van Rossum cleansed Perl's idiosyncracies and objectified basic data structure, data manipulations, and I/O. With Python, OO is so intrinsic that learning Python is equivalent to learning OO. The same cannot be said of Perl.

Unfortunately, comparisons with other languages are missing from Beazley's book. Van Rossum, in an embarrassingly self-serving foreword, preemptively asserts that we readers need "neither evangelizing nor proselytizing"--after all, we already own the book--but we do need galvanizing and we don't find it. Specifically, we need a response to the oft-repeated wisdom that new computer languages are only worth learning if they teach us to organize our thinking along new lines.

Scripting languages, however, are for quick and dirty projects: quick to write, easy to hack, and ultimately disposable. The essential tension created by van Rossum and friends is between the elegance of object-oriented principles and the utility of a quick-hacked script. Sadly, the tension remains unresolved in Beazley's reference. There is little to convince us that Python has earned its place in the firmament by changing our thinking. But Beazley has given us much to get us going if we have already taken the leap of faith. --Peter Leopold --This text refers to an out of print or unavailable edition of this title.

From Library Journal

Though Python is a relatively new programming language, it has quite a significant audience owing to its sensible syntax. An active user of Python since 1996, Beazley provides ample information on the fundamentals of versions 2.0 and 2.1, including syntax, functions, operators, classes, and libraries. This is first and foremost a reference, so he avoids lengthy discussions of Python's superiority. Peppered with good code samples and featuring a companion web site with more extensive pieces, this title should be on hand in larger libraries.

Copyright 2001 Reed Business Information, Inc.

作者介绍:

David M. Beazley

早在1996年就开始使用Python编程。在洛斯阿莫斯国家实验室工作期间，他教会很多志愿者用Python编写科学计算软件。他创办的Dabeaz公司
提供软件开发、培训和咨询服务，专长于Python、Ruby、Perl等动态编程语言的实际应用。他是Python软件基金会的会员。

目录:

[Python Essential Reference_ 下载链接1](#)

标签

Python

programming

编程

计算机

编程语言

python

开发技术

程序开发

评论

A concise python introductory book for experienced developers.

学过一门OO语言的直接翻这本就ok

包含语法、系统库、嵌入与扩展的全面内容，与官方文档范围和叙事结构挺像，不错的线下手册。基于Python2.6和3.0，内容准确而丰富，提及了MRO的C3算法、各类Encoding的详细区别等等。

不错的参考书。从基本语法到标准库都有，而且有实用例子。

Very clear

是本好书，清明3天要读完

一本书可作两本读，“Probably the best second book on Python”。

后面没啥读的，前面的半部算是可以称为essential纲领的东西。但对我来说不值这个价。。。。

都出到第五版了，可见其质量。是很好的reference。

非入门书，对Python这门技术进行非常有条理的阐述；英文看的有些吃力，过段时间再来回顾体会。

读了 WeiZhong 2006-01-17 翻译的第二版电子书

勉强算读了吧，后面的类库详解覆盖面太广，如果不是专注于某个专门的领域的话，这么多的函数列表读了也记不住，还是姑且放下吧。前面8章写的实在是太好了，清晰明了，对于机制的介绍详细而且易懂，受益匪浅啊，打算从头再把这几章回顾一遍

被拯救了

前半部分写得很好，后面讲库的跳着翻了点。觉得可以用作入门到流畅之间的过渡。

工具书，随手翻。

中文翻译质量果然很烂，<http://book.douban.com/subject/5401851/>，不过书的确不错

谁把我书借走了。。。

小而全 可以多看几遍

很详细，可以当字典查

想深入学习下python，虽然用起来很顺手，但是有时候会有种使不上劲的感觉，要深入学习会儿 ----
库大概过了一遍，前面的仔细看了，感觉合pro python似乎都不错～

[Python Essential Reference 下载链接1](#)

书评

第13章，第2节 copy copy The copy module provides functions for making shallow and deep copies of compound objects, including lists, tuples, dictionaries, and instances of user-defined objects.
copy(x) Makes a shallow copy of x by creating a new compound ob...

推荐直接下载英文版阅读，该版翻译质量较差，会造成理解上的错误和不畅，翻译者在很多地方都很明显没有明白原书想要表达的意思。建议大家不要浪费钱财在这种不负责任的书上。直接看原版吧。
谢俊、杨越、高伟，这几个名字值得记住，太差了。怪我买书的时候没有细看，不过也...

1：主从句关系混乱，主谓宾衔接关系混乱。
2：英语即使他们懂，他们也不会用很流畅的汉语逻辑反应到书面上 3：英语其实他们不是特别懂
4：汉语水平也很一般
5：对他们的学识水平强烈怀疑，技术思维很一般，没有敏感度，否则书不会翻译成这样。

这本书买来之后翻看了几章，翻译的水平太差了，严重影响对书的内容的理解。以后坚决买英文原版的，即便是翻译的也要买那些口碑好的译者的书。像这种一本书由好几个人翻译的，而且在书中找不到任何译者介绍的，坚决不买！

很全面，结构很清晰，没有废话，第一部分必须从头到尾认真看，全部都有用，帮整你肯定都会用上。有些例如lamda表达式、yield与生成式讲的不够具体，但其它python也不过如此，这其实是函数式编程的内容。第二部分是Python库，可用以参考

页码32页，电子书45页，表3-2,lens (s) 应为len (s) 页码33页，电子书46页，表3-4，
原文：s.pop ([i]) 返回元素i并从列表中删除它，如果省略i，则返回列表中最后一个元素 表述有误。
应为：s.pop ([i]) 返回下标为i的元素并从列表中删除它，如果省略i，则返回列表中最后一...

每天睡觉前看看，有助于睡眠，老大的英文书，带回来治疗失眠的哈哈。
这本书，还不错啊？不知道中文的翻译是什么样子的，反正英文的看着还行，后面的高级部分主要是介绍原理和模块的，例子也很少，想看例子的同学需要移步python cookbook

书是好书，中文版把Index去掉了，唉。图灵的书大都保留Index的，这本不知道为什么去掉了，可惜。为什么还说我的评论太短了～为什么还说我的评论太短了～为什么还说我的评论太短了～为什么还说我的评论太短了～为什么还说我的评论太短了～

如果已经掌握了几种编程语言了，学Python就用这本书，其它Python的书都可以烧掉了。这本书不仅将语法和API组织的井然有序，更宝贵的，是他在整部书中穿插了很多best practices.

[Python Essential Reference 下载链接1](#)