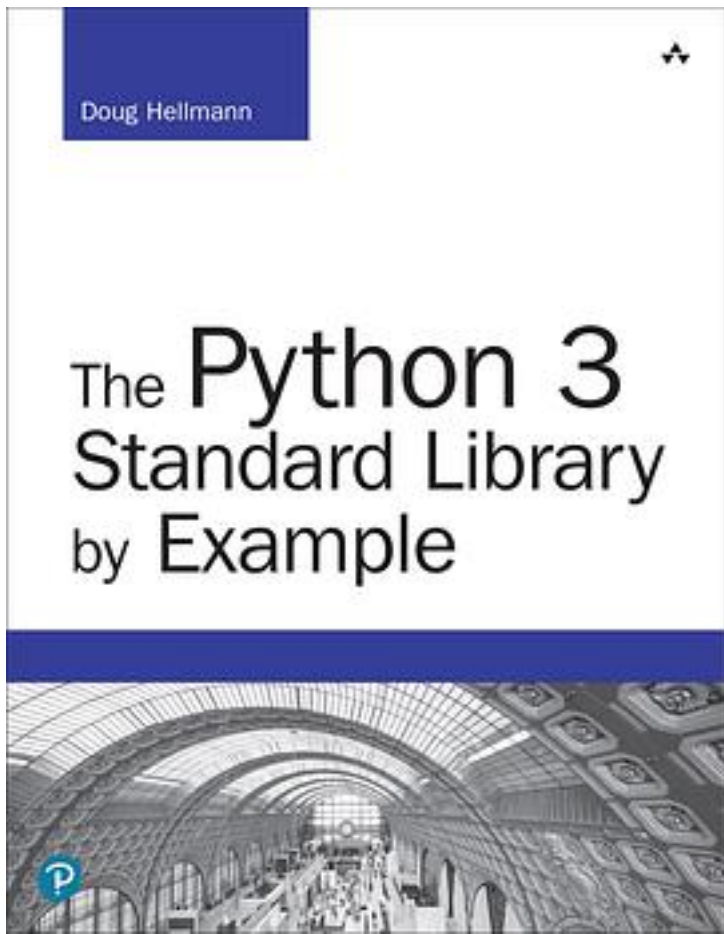


The Python 3 Standard Library by Example



[The Python 3 Standard Library by Example_ 下载链接1](#)

著者:Doug Hellmann

出版者:Imprint unknown

出版时间:2017-3-11

装帧:Paperback

isbn:9780134291055

Distributed with every copy of Python, the Standard Library contains hundreds of modules that provide tools for interacting with the operating system, interpreter, and Internet—all of them tested and ready to be used to jump-start the development of

your applications. This book presents selected examples demonstrating how to use the most commonly used features of the modules that give Python its “batteries included” slogan, taken from the popular Python Module of the Week (PyMOTW) blog series.

作者介绍:

Doug Hellmann目前是Racemi公司的一位高级开发人员，也是Python Software Foundation的信息交流主管。从1.4版开始他就一直在做Python编程，曾在大量UNIX和非UNIX平台上参与项目开发，涉及领域包括地图、医疗新闻播报、金融和数据中心自动化。为《Python Magazine》做了一年普通专栏作家后，他在2008—2009年成为这家杂志的主编。自2007年以来，Doug在他的博客上发表了颇受关注的“Python Module of the Week”系列。他居住在乔治亚州的Athens。

目录: Introduction

Chapter 1. Text

- 1.1. string — Text Constants and Templates
- 1.2. textwrap — Formatting Text Paragraphs
- 1.3. re — Regular Expressions
- 1.4. difflib — Compare Sequences

Chapter 2. Data Structures

- 2.1. enum — Enumeration Type
- 2.2. collections — Container Data Types
- 2.3. array — Sequence of Fixed-type Data
- 2.4. heapq — Heap Sort Algorithm
- 2.5. bisect — Maintain Lists in Sorted Order
- 2.6. queue — Thread-safe FIFO Implementation
- 2.7. struct — Binary Data Structures
- 2.8. weakref — Impermanent References to Objects
- 2.9. copy — Duplicate Objects
- 2.10. pprint — Pretty-print Data Structures

Chapter 3. Algorithms

- 3.1. functools — Tools for Manipulating Functions
- 3.2. itertools — Iterator Functions
- 3.3. operator — Functional Interface to Built-in Operators
- 3.4. contextlib — Context Manager Utilities

Chapter 4. Dates and Times

- 4.1. time — Clock Time
- 4.2. datetime — Date and Time Value Manipulation
- 4.3. calendar — Work with Dates

Chapter 5. Mathematics

- 5.1. decimal — Fixed and Floating Point Math
- 5.2. fractions — Rational Numbers
- 5.3. random — Pseudorandom Number Generators
- 5.4. math — Mathematical Functions
- 5.5. statistics — Statistical Calculations

Chapter 6. The File System

- 6.1. os.path — Platform-independent Manipulation of Filenames
- 6.2. pathlib — Filesystem Paths as Objects
- 6.3. glob — Filename Pattern Matching
- 6.4. fnmatch — Unix-style Glob Pattern Matching

- 6.5. linecache — Read Text Files Efficiently
- 6.6. tempfile — Temporary File System Objects
- 6.7. shutil — High-level File Operations
- 6.8. filecmp — Compare Files
- 6.9. mmap — Memory-map Files
- 6.10. codecs — String Encoding and Decoding
- 6.11. io — Text, Binary, and Raw Stream I/O Tools
- Chapter 7. Data Persistence and Exchange
 - 7.1. pickle — Object Serialization
 - 7.2. shelve — Persistent Storage of Objects
 - 7.3. dbm — Unix Key-Value Databases
 - 7.4. sqlite3 — Embedded Relational Database
 - 7.5. xml.etree.ElementTree — XML Manipulation API
 - 7.6. csv — Comma-separated Value Files
- Chapter 8. Data Compression and Archiving
 - 8.1. zlib — GNU zlib Compression
 - 8.2. gzip — Read and Write GNU zip Files
 - 8.3. bz2 — bzip2 Compression
 - 8.4. tarfile — Tar Archive Access
 - 8.4. tarfile — Tar Archive Access 489
 - 8.5. zipfile — ZIP Archive Access
- Chapter 9. Cryptography
 - 9.1. hashlib — Cryptographic Hashing
 - 9.2. hmac — Cryptographic Message Signing and Verification
- Chapter 10. Concurrency with Processes, Threads, and Coroutines
 - 10.1. subprocess — Spawning Additional Processes
 - 10.2. signal — Asynchronous System Events
 - 10.3. threading — Manage Concurrent Operations Within a Process
 - 10.4. multiprocessing — Manage Processes Like Threads
 - 10.5. asyncio — Asynchronous I/O, event loop, and concurrency tools
 - 10.6. concurrent.futures — Manage Pools of Concurrent Tasks
- Chapter 11. Networking
 - 11.1. ipaddress — Internet Addresses
 - 11.2. socket — Network Communication
 - 11.3. selectors — I/O Multiplexing Abstractions
 - 11.4. select — Wait for I/O Efficiently
 - 11.5. socketserver — Creating Network Servers
- Chapter 12. The Internet
 - 12.1. urllib.parse — Split URLs into Components
 - 12.2. urllib.request — Network Resource Access
 - 12.3. urllib.robotparser — Internet Spider Access Control
 - 12.4. base64 — Encode Binary Data with ASCII
 - 12.5. http.server — Base Classes for Implementing Web Servers
 - 12.6. http.cookies — HTTP Cookies
 - 12.7. webbrowser — Displays web pages
 - 12.8. uuid — Universally Unique Identifiers
 - 12.9. json — JavaScript Object Notation
 - 12.10. xmlrpc.client — Client Library for XML-RPC
 - 12.11. xmlrpc.server — An XML-RPC server
- Chapter 13. Email
 - 13.1. smtplib — Simple Mail Transfer Protocol Client
 - 13.2. smtpd — Sample Mail Servers
 - 13.3. mailbox — Manipulate Email Archives
 - 13.4. imaplib — IMAP4 Client Library

Chapter 14. Application Building Blocks

- 14.1. argparse — Command-Line Option and Argument Parsing
- 14.2. getopt — Command Line Option Parsing
- 14.3. readline — The GNU readline Library
- 14.4. getpass — Secure Password Prompt
- 14.5. cmd — Line-oriented Command Processors
- 14.6. shlex — Parse Shell-style Syntaxes
- 14.7. configparser — Work with Configuration Files
- 14.8. logging — Report Status, Error, and Informational Messages
- 14.9. fileinput — Command-Line Filter Framework
- 14.10. atexit — Program Shutdown Callbacks
- 14.11. sched — Timed Event Scheduler

Chapter 15. Internationalization and Localization

- 15.1. gettext — Message Catalogs
- 15.2. locale — Cultural Localization API

Chapter 16. Developer Tools

- 16.1. pydoc — Online Help for Modules
- 16.2. doctest — Testing Through Documentation
- 16.3. unittest — Automated Testing Framework
- 16.4. trace — Follow Program Flow
- 16.5. traceback — Exceptions and Stack Traces
- 16.6. cgitb — Detailed Traceback Reports
- 16.7. pdb — Interactive Debugger
- 16.8. profile and pstats — Performance Analysis
- 16.9. timeit — Time the execution of small bits of Python code.
- 16.10. tabnanny — Indentation validator
- 16.11. compileall — Byte-compile Source Files
- 16.12. pycbr — Class Browser
- 16.13. venv — Create Virtual Environments
- 16.14. ensurepip — Install the Python Package Installer

Chapter 17. Runtime Features

- 17.1. site — Site-wide Configuration
- 17.2. sys — System-specific Configuration
- 17.3. os — Portable access to operating system specific features
- 17.4. platform — System Version Information
- 17.5. resource — System Resource Management
- 17.6. gc — Garbage Collector
- 17.7. sysconfig — Interpreter Compile-time Configuration

Chapter 18. Language Tools

- 18.1. warnings — Non-fatal Alerts
- 18.2. abc — Abstract Base Classes
- 18.3. dis — Python Bytecode Disassembler
- 18.4. inspect — Inspect Live Objects

Chapter 19. Modules and Packages

- 19.1. importlib — Python's Import Mechanism
- 19.2. pkgutil — Package Utilities
- 19.3. zipimport — Load Python Code from ZIP Archives

Appendix A Porting Notes

- A.1. References
- A.2. New Modules
- A.3. Renamed Modules
- A.4. Removed Modules
- A.5. Deprecated Modules
- A.6. Summary of Changes to Modules

Appendix B Outside of the Standard Library

B.1. Text

B.2. Algorithms

B.3. Dates and Times

B.4. Mathematics

B.5. Data Persistence and Exchange

B.6. Cryptography

B.7. Concurrency with Processes, Threads, and Coroutines

B.8. The Internet

B.9. Email

B.10. Application Building Blocks

B.11. Developer Tools

• • • • • ([收起](#))

[The Python 3 Standard Library by Example_下载链接1](#)

标签

Python

标准库

计算机

python

编程

开发

py

en

评论

[The Python 3 Standard Library by Example_ 下载链接1](#)

书评

我想这本书作为一本字典的存在可能必作为平常书籍存在的意义更大。1300多页（英文版），详尽地介绍了标准库中绝大多数模块，并且难能可贵的是每一个模块都有相应的例子。但是冗长的内容加上内容本身的特殊性，使得本书更适合作为一本字典。我想一个python程序员不可能仅仅通过...

一直希望手里有本上千页的Python大全书（c,c++,delphi,java都有过....）
这本书过了一千页 ----- 啊，找到了能长期占据书桌的宝典了 但是 读了一小时之后
我只能说：这是一本充分体现现有机器翻译能力的作品

最近双十一气氛弥漫在整个互联网，不买点东西总觉得缺了什么。在逛某东的时候无意中发现了这本刚出版没多久的书，一时心血来潮立即加入购物车，这不对啊，价格这么贵。去当当一看，五折，99.5块，果断拿下。
Python的库太庞大的，确实需要一本书来梳理一下，最好是按库的功能来...

<http://pycoders-weekly-chinese.readthedocs.org/en/latest/issue4/a-python-reading-list.html>
如果你希望有一本拥有很多代码样例的参考书，那么这本书就是你想要的，这本书不仅仅包含了所有的标准库的样例，还包含了许多现在开发中经常用到的库的使用样例，这本书基于作者...

好处:有例子，可以参考，diy
坏处：例子有错误，没有说明函数原型，不太适合新手，如果能配合工程经验和实际的函数解释就更好了，建议参考这里<http://docs.python.org/2/library/re.html?highlight=re.compile#re.compile>

1.方便查询常见的python的各种类型模块,根据不同内容分成了不同的章节，对于初学者很有好处。 2，实例丰富，可以作为工具书使用

3.除了书本内的内容，还有一些参考内容供自己学习 4.强烈推荐给刚学python的盆友

这本书来源作者博客的一个专栏，Python Module of the Week。讲了python标准库中很多模块的使用方法，而且是以完整的代码来讲解模块的应用。

对于我这种喜欢一边看书，一边敲代码的人而言，晚上拿着这本放在腿上，选上一节主题，敲上那么100来行（2至3个程序的量）的代...

Python之所以流行，原因在于：1.
语言本身优美，表达力强。适合做快速原型开发。且学习曲线平坦，上手快。 2.
batteries-included

Python标准库正是应对了这第二点。丰富的库实现得以让python程序员迅速完成各种编程任务，将关注的重点集中在问题本身，而不是繁杂的实现细节上...

我看的是网络版的Python Standard Library,中文名为python标准库中文版，翻译是：python江湖群，没有出版社。
出彩的地方：

- 1.是一本查阅python库的工具书，涉及的库很多，不仅有标准库还有其他常用库
- 2.署名作者是：python江湖群。好像是一群网友热心翻译的作品，很佩服这种...

[The Python 3 Standard Library by Example 下载链接1](#)