

- 经典著作全面更新
- Python 程序员案头必备
- 涵盖 Python 2 和 Python 3 共有特性

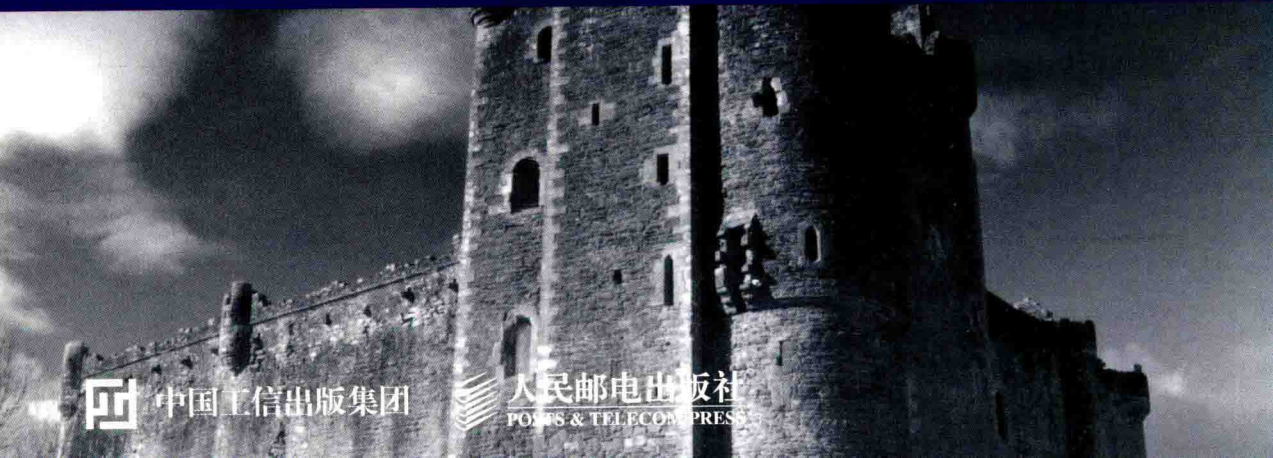
Python 参考手册

(第 4 版 · 修订版)

Python

Essential Reference Fourth Edition

[美] David M. Beazley 著
谢俊 杨越 高伟 译
宋秉金 审校





Pearson

Python参考手册

(第4版·修订版)

Python

Essential Reference Fourth Edition

[美] David M. Beazley 著

谢俊 杨越 高伟 译

宋秉金 审校

人民邮电出版社

北京

图书在版编目 (CIP) 数据

Python参考手册 : 第4版·修订版 / (美) 大卫 M. 比兹利 (David M. Beazley) 著 ; 谢俊, 杨越, 高伟译 . — 2版. — 北京 : 人民邮电出版社, 2016. 10

书名原文: Python Essential Reference, Fourth Edition

ISBN 978-7-115-39439-2

I. ①P… II. ①大… ②谢… ③杨… ④高… III. ①软件工具—程序设计—技术手册 IV. ①TP311.56-62

中国版本图书馆CIP数据核字(2016)第235862号

版 权 声 明

本书是 Python 编程语言的权威参考手册, 书中详尽讲解了 Python 核心和 Python 库中最重要的部分。全书分为三个部分, 第一部分涉及类型与对象, 运算符与表达式, 程序结构与控制流, 函数与函数式编程, 类与面向对象编程, 模块、包与分发, 输入与输出, 测试、调试、探查与调优等与 Python 语言相关的内容; 第二部分涉及内置函数和异常、运行时服务、数学运算、数据结构、算法与代码优化, 字符串和文本处理, 数据库访问, 文件和目录处理, 操作系统服务, 线程与并发, 网络编程与套接字, 网络应用程序编程, Web 编程, 网络数据处理和编码、其他库模块等与 Python 库相关的内容; 第三部分涉及扩展和嵌入 Python 等内容。此外, 书中还包括一些 Python 官方文档或其他参考资料中未提及的高级主题。

本书适合 Python 程序员以及具备其他编程语言经验的开发人员阅读和参考。

-
- ◆ 著 [美] David M. Beazley
译 谢 俊 杨 越 高 伟
审 校 宋秉金
责任编辑 杨海玲
责任印制 焦志炜
- ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京鑫正大印刷有限公司印刷
- ◆ 开本: 800×1000 1/16
印张: 33.75
字数: 879 千字 2016 年 10 月第 2 版
印数: 1—3 000 册 2016 年 10 月北京第 1 次印刷
- 著作权合同登记号 图字: 01-2010-4024 号
-

定价: 89.00 元

读者服务热线: (010)81055410 印装质量热线: (010)81055316
反盗版热线: (010)81055315

版 权 声 明

Authorized translation from the English language edition, entitled *Python Essential Reference Fourth Edition*, 9780672329784 by David M. Beazley, published by Pearson Education, Inc., publishing as Addison Wesley, Copyright © 2009 by Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD. and POSTS & TELECOM PRESS Copyright © 2016.

本书中文简体字版由Pearson Education Asia Ltd.授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

本书封面贴有Pearson Education（培生教育出版集团）激光防伪标签，无标签者不得销售。版权所有，侵权必究。

谨将本书献给Paula、Thomas和他即将出生的兄弟。

前 言

本书是Python编程语言的一份简明参考。尽管有经验的程序员也可以通过本书学习Python语言，但本书并不是一本讲述如何编程的全面教程或专著。相反，本书的目标是准确而简练地介绍Python语言核心以及Python库中最关键的部分。本书假定读者以前拥有Python或其他语言（如C或Java）的编程经验。另外，对系统编程（例如基本的操作系统概念和网络编程）有大致的了解对理解Python库参考中的部分章节可能会有一定的帮助。

Python可以从<http://www.python.org>免费下载。几乎每一种操作系统，包括UNIX、Windows和Macintosh，都有相应的版本。另外，Python网站还包含文档、指南和各种第三方软件的链接。

本书的这一版是在Python发展的关键时刻面世的。Python 2.6和Python 3.0几乎是同时发布的，但Python 3与以前的版本并不兼容。作为一名作者和程序员，我面临着一个两难问题：是直接跳到Python 3.0，还是使用大多数程序员更为熟悉的Python 2.x版本？

多年前，在应该使用哪些语言特性的问题上，身为C程序员的我经常把某些书当作终极权威。例如，使用K&R书中未曾提到过的某些特性很可能导致程序无法移植，因此得格外谨慎。作为程序员，这种方法对我非常有效，因此我决定在本书的这个版本中沿用这一做法。也就是说，这一版删除了Python 3中已经不再支持的Python 2特性。类似地，我也没有关注Python 3中不能向后兼容的特性（尽管附录中仍然会介绍这些特性）。最终，我希望，无论使用的是哪个版本的Python，本书都能够成为Python程序员案头必备的参考书。

这一版还包含自第1版出版至今10余年来最激动人心的一些变化。在过去的数年间，Python的发展重点体现在新的语言特性上——特别是与函数式编程和元编程相关的特性。因此，讲述函数式编程和面向对象编程的章节得到了极大的扩充，包括的主题有生成器、迭代器、协程、装饰器和元类。讲述库的章节则将重点转移到更加现代的模块上。整本书中的示例和代码片段都做了更新。我认为大多数程序员会对扩充之后的内容感到满意。

最后值得一提的是，Python本身已经有数千页有用的文档。本书的内容在很大程度上是基于这些文档的，但又有很多关键的区别。首先，这本书提供信息的方式更加紧凑，使用不同的示例，并对很多主题提供了不同的描述。其次，Python库参考部分的大量主题都做了扩充，包含很多外部的参考资料。底层系统和网络模块尤其如此，因为对这些模块的有效使用往往依赖于手册和外部参考中列出的种种选项。另外，为了使参考手册更加简明，这一版还删掉了很多已经废弃和相对晦涩的库模块。

我的目标是写出一本真正包含使用Python及其众多模块所需的一切内容的参考手册。尽管这并不是一本介绍Python语言的百科全书，但我希望本书能成为你未来数年里一本有用的必备编程参考书。十分欢迎读者对本书提出意见和建议。

David Beazley

2009年6月

于伊利诺伊州芝加哥

致 谢

本书能与读者见面，要感谢很多人的大力支持。首先要感谢Noah Gift参与这个项目，并提出了许多建设性意见。Kurt Grandis也对很多章节发表了中肯的见解。我还要感谢前几版的技术审稿人Timothy Boronczyk、Paul DuBois、Mats Wichmann、David Ascher和Tim Bell，他们的精彩意见和建议促成了过去几版的成功。在1999年那个炎热的夏天，Guido van Rossum、Jeremy Hylton、Fred Drake、Roger Masse和Barry Warsaw招待了我好几个星期，同时对第1版提供了极大的帮助。最后，同样重要的是，本书的问世离不开读者的热情反馈。要感谢的人实在太多，这里无法一一列出他们的名字，我已尽力采纳大家的建议来让本书变得更好。我还要感谢Addison-Wesley和Pearson Education的工作人员，他们对这个项目给予了持续的支持与帮助。Mark Taber、Michael Thurston、Seth Kerney和Lisa Thibault都对本书的顺利出版倾注了大量的心血。还要特别感谢Robin Drake，他在第3版出版的过程中做了大量的编辑工作。最后，我要感谢我了不起的妻子和好搭档Paula Kamen，感谢她给予我的所有鼓励、幽默和爱。

目 录

第一部分 Python 语言

第 1 章 Python 简介.....2

- 1.1 运行 Python.....2
- 1.2 变量和算术表达式.....3
- 1.3 条件语句.....5
- 1.4 文件输入和输出.....6
- 1.5 字符串.....7
- 1.6 列表.....8
- 1.7 元组.....9
- 1.8 集合.....10
- 1.9 字典.....11
- 1.10 迭代与循环.....12
- 1.11 函数.....13
- 1.12 生成器.....14
- 1.13 协程.....15
- 1.14 对象与类.....16
- 1.15 异常.....17
- 1.16 模块.....18
- 1.17 获得帮助.....19

第 2 章 词法约定和语法.....20

- 2.1 行结构和缩进.....20
- 2.2 标识符和保留字.....21
- 2.3 数字字面量.....21
- 2.4 字符串字面量.....22
- 2.5 容器.....23
- 2.6 运算符、分隔符及特殊符号.....24
- 2.7 文档字符串.....24
- 2.8 装饰器.....24
- 2.9 源代码编码.....25

第 3 章 类型与对象..... 26

- 3.1 术语..... 26
- 3.2 对象的标识与类型..... 26
- 3.3 引用计数与垃圾回收..... 27
- 3.4 引用与复制..... 28
- 3.5 第一类对象..... 29
- 3.6 表示数据的内置类型..... 30
 - 3.6.1 None 类型..... 30
 - 3.6.2 数值类型..... 31
 - 3.6.3 序列类型..... 31
 - 3.6.4 映射类型..... 35
 - 3.6.5 集合类型..... 36
- 3.7 表示程序结构的内置类型..... 38
 - 3.7.1 可调用类型..... 38
 - 3.7.2 类、类型与实例..... 40
 - 3.7.3 模块..... 41
- 3.8 解释器内部使用的内置类型..... 41
 - 3.8.1 代码对象..... 41
 - 3.8.2 帧对象..... 42
 - 3.8.3 跟踪对象..... 42
 - 3.8.4 生成器对象..... 43
 - 3.8.5 切片对象..... 43
 - 3.8.6 Ellipsis 对象..... 43
- 3.9 对象行为与特殊方法..... 44
 - 3.9.1 对象的创建与销毁..... 44
 - 3.9.2 对象字符串表示..... 44
 - 3.9.3 对象比较与排序..... 45
 - 3.9.4 类型检查..... 46
 - 3.9.5 属性访问..... 46
 - 3.9.6 属性包装与描述符..... 46
 - 3.9.7 序列与映射方法..... 47

3.9.8 迭代	48	6.8 使用生成器与协程	87
3.9.9 数学操作	48	6.9 列表推导	89
3.9.10 可调用接口	50	6.10 生成器表达式	90
3.9.11 上下文管理协议	50	6.11 声明式编程	91
3.9.12 对象检查与 <code>dir()</code>	51	6.12 <code>lambda</code> 运算符	92
第 4 章 运算符与表达式	52	6.13 递归	92
4.1 数值操作	52	6.14 文档字符串	93
4.2 序列操作	53	6.15 函数属性	94
4.3 字符串格式化	56	6.16 <code>eval()</code> 、 <code>exec()</code> 和 <code>compile()</code> 函数	94
4.4 高级字符串格式化	57	第 7 章 类与面向对象编程	96
4.5 字典操作	59	7.1 <code>class</code> 语句	96
4.6 集合操作	60	7.2 类实例	97
4.7 增量赋值	60	7.3 作用域规则	97
4.8 属性 (<code>.</code>) 运算符	61	7.4 继承	98
4.9 函数调用 (<code>()</code>) 运算符	61	7.5 多态动态绑定和鸭子类型	100
4.10 转换函数	61	7.6 静态方法和类方法	101
4.11 布尔表达式与真值	62	7.7 特性	102
4.12 对象等同性与标识	63	7.8 描述符	104
4.13 运算优先级	63	7.9 数据封装和私有属性	105
4.14 条件表达式	64	7.10 对象内存管理	106
第 5 章 程序结构与控制流	65	7.11 对象表示和属性绑定	108
5.1 程序结构与执行	65	7.12 <code>__slots__</code>	109
5.2 执行条件语句	65	7.13 运算符重载	110
5.3 循环与迭代	66	7.14 类型和类成员测试	111
5.4 异常	68	7.15 抽象基类	113
5.4.1 内置异常	70	7.16 元类	114
5.4.2 定义新异常	71	7.17 类装饰器	117
5.5 上下文管理器与 <code>with</code> 语句	72	第 8 章 模块、包与分发	118
5.6 断言与 <code>__debug__</code>	73	8.1 模块与 <code>import</code> 语句	118
第 6 章 函数与函数式编程	75	8.2 从模块导入选定符号	119
6.1 函数	75	8.3 以主程序的形式执行	120
6.2 参数传递与返回值	77	8.4 模块搜索路径	121
6.3 作用域规则	78	8.5 模块加载和编译	121
6.4 作为对象与闭包的函数	79	8.6 模块重新加载和卸载	122
6.5 装饰器	82	8.7 包	123
6.6 生成器与 <code>yield</code>	83	8.8 分发 Python 程序和库	125
6.7 协程与 <code>yield</code> 表达式	85	8.9 安装第三方库	127

第 9 章 输入与输出 129

- 9.1 读取命令行选项 129
- 9.2 环境变量 130
- 9.3 文件和文件对象 130
- 9.4 标准输入、输出和错误 133
- 9.5 print 语句 133
- 9.6 print() 函数 134
- 9.7 文本输出中的变量插入 134
- 9.8 生成输出 135
- 9.9 Unicode 字符串处理 136
- 9.10 Unicode I/O 137
 - 9.10.1 Unicode 数据编码 138
 - 9.10.2 Unicode 字符特性 140
- 9.11 对象持久化与 pickle 模块 140

第 10 章 执行环境 142

- 10.1 解释器选项与环境 142
- 10.2 交互式会话 144
- 10.3 启动 Python 应用程序 145
- 10.4 站点配置文件 145
- 10.5 用户站点包 146
- 10.6 启用新功能 146
- 10.7 程序终止 147

第 11 章 测试、调试、探查与调优 149

- 11.1 文档字符串和 doctest 模块 149
- 11.2 单元测试和 unittest 模块 151
- 11.3 Python 调试器和 pdb 模块 153
 - 11.3.1 调试器命令 153
 - 11.3.2 从命令行进行调试 156
 - 11.3.3 配置调试器 156
- 11.4 程序探查 156
- 11.5 调优与优化 157
 - 11.5.1 进行计时测量 157
 - 11.5.2 进行内存测量 158
 - 11.5.3 反汇编 158
 - 11.5.4 调优策略 159

第二部分 Python 库

第 12 章 内置函数和异常 164

- 12.1 内置函数和类型 164

12.2 内置异常 172

- 12.2.1 异常基类 172
- 12.2.2 异常实例 173
- 12.2.3 预定义的异常类 173

12.3 内置警告 176

12.4 future_builtins 176

第 13 章 Python 运行时服务 178

- 13.1 atexit 178
- 13.2 copy 178
- 13.3 gc 179
- 13.4 inspect 180
- 13.5 marshal 183
- 13.6 pickle 184
- 13.7 sys 186
 - 13.7.1 变量 186
 - 13.7.2 函数 189
- 13.8 traceback 191
- 13.9 types 192
- 13.10 warnings 193
- 13.11 weakref 195
 - 13.11.1 示例 196
 - 13.11.2 注意 196

第 14 章 数学运算 197

- 14.1 decimal 197
 - 14.1.1 Decimal 对象 197
 - 14.1.2 Context 对象 198
 - 14.1.3 函数和常量 200
 - 14.1.4 示例 201
 - 14.1.5 注意 202
- 14.2 fractions 203
- 14.3 math 204
- 14.4 numbers 205
- 14.5 random 206
 - 14.5.1 种子和初始化 206
 - 14.5.2 随机整数 206
 - 14.5.3 随机序列 207
 - 14.5.4 实值随机分布 207

14.5.5 注意	208	16.4.4 注意	237
第 15 章 数据结构、算法与代码简化	209	16.5 unicodedata	238
15.1 abc	209	第 17 章 Python 数据库访问	241
15.2 array	210	17.1 关系数据库 API 规范	241
15.3 bisect	212	17.1.1 连接	241
15.4 collections	213	17.1.2 Cursor	242
15.4.1 deque 和 defaultdict	213	17.1.3 生成查询	243
15.4.2 命名元组	214	17.1.4 类型对象	244
15.4.3 抽象基类	216	17.1.5 错误处理	245
15.5 contextlib	217	17.1.6 多线程	245
15.6 functools	218	17.1.7 将结果映射到字典中	246
15.7 heapq	219	17.1.8 数据库 API 扩展	246
15.8 itertools	220	17.2 sqlite3 模块	246
15.9 operator	222	17.2.1 模块级函数	246
第 16 章 字符串和文本处理	225	17.2.2 连接对象	248
16.1 codecs	225	17.2.3 游标和基本操作	250
16.1.1 低级 codecs 接口	225	17.3 DBM 风格的数据库模块	252
16.1.2 I/O 相关函数	226	17.4 shelve 模块	253
16.1.3 有用的常量	227	第 18 章 文件和目录处理	254
16.1.4 标准编码	227	18.1 bz2	254
16.1.5 注意	228	18.2 filecmp	255
16.2 re	228	18.3 fnmatch	256
16.2.1 模式语法	228	18.4 glob	257
16.2.2 函数	229	18.5 gzip	257
16.2.3 正则表达式对象	231	18.6 shutil	258
16.2.4 匹配对象	231	18.7 tarfile	259
16.2.5 示例	232	18.7.1 异常	261
16.2.6 注意	233	18.7.2 示例	262
16.3 string	233	18.8 tempfile	262
16.3.1 常量	233	18.9 zipfile	263
16.3.2 Formatter 对象	233	18.10 zlib	266
16.3.3 Template 字符串	235	第 19 章 操作系统服务	268
16.3.4 实用工具函数	235	19.1 Commands	268
16.4 struct	235	19.2 ConfigParser、configparser	269
16.4.1 打包和解包函数	236	19.2.1 ConfigParser 类	269
16.4.2 Struct 对象	236	19.2.2 示例	270
16.4.3 格式编码	236	19.2.3 注意	272

19.3	datetime	272	19.11.3	文件与目录	313
19.3.1	date 对象	272	19.11.4	进程管理	316
19.3.2	time 对象	273	19.11.5	系统配置	320
19.3.3	datetime 对象	274	19.11.6	异常	321
19.3.4	timedelta 对象	275	19.12	os.path	321
19.3.5	涉及日期的数学运算	276	19.13	signal	323
19.3.6	tzinfo 对象	277	19.13.1	例子	325
19.3.7	日期与时间解析	278	19.13.2	注意	325
19.4	errno	278	19.14	subprocess	325
19.4.1	POSIX 错误代码	278	19.14.1	例子	327
19.4.2	Windows 错误代码	279	19.14.2	注意	328
19.5	fcntl	280	19.15	time	328
19.5.1	示例	281	19.16	winreg	330
19.5.2	注意	281	第 20 章	线程与并发	334
19.6	io	282	20.1	基本概念	334
19.6.1	基本 I/O 接口	282	20.2	并发编程与 Python	335
19.6.2	原始 I/O	282	20.3	multiprocessing	336
19.6.3	缓存二进制 I/O	283	20.3.1	进程	336
19.6.4	文本 I/O	285	20.3.2	进程间通信	337
19.6.5	open() 函数	285	20.3.3	进程池	343
19.6.6	抽象基类	286	20.3.4	共享数据与同步	345
19.7	logging	286	20.3.5	托管对象	347
19.7.1	日志记录级别	286	20.3.6	连接	352
19.7.2	基本配置	286	20.3.7	各种实用工具函数	353
19.7.3	Logger 对象	288	20.3.8	多进程处理的一般建议	353
19.7.4	处理器对象	292	20.4	threading	354
19.7.5	消息格式化	294	20.4.1	Thread 对象	354
19.7.6	各种实用工具函数	296	20.4.2	Timer 对象	356
19.7.7	日志记录配置	296	20.4.3	Lock 对象	356
19.7.8	性能考虑	299	20.4.4	RLock 对象	356
19.7.9	注意	299	20.4.5	信号量与有边界的信号量	357
19.8	mmap	299	20.4.6	事件	358
19.9	msvcrt	301	20.4.7	条件变量	358
19.10	optparse	303	20.4.8	使用 Lock	359
19.10.1	例子	305	20.4.9	线程终止与挂起	360
19.10.2	注意	306	20.4.10	实用工具函数	361
19.11	os	307	20.4.11	全局解释器锁	361
19.11.1	进程环境	307	20.4.12	使用线程编程	361
19.11.2	文件创建与文件描述符	309			

20.5	queue、Queue	362
20.6	协程与微线程	364
第 21 章 网络编程和套接字		365
21.1	网络编程基础	365
21.2	asynchat 模块	367
21.3	asyncore 模块	370
21.4	select	374
21.4.1	高级模块功能	375
21.4.2	高级异步 I/O 示例	375
21.4.3	异步联网的时机	381
21.5	socket	383
21.5.1	地址族	383
21.5.2	套接字类型	383
21.5.3	寻址	384
21.5.4	函数	385
21.5.5	异常	395
21.5.6	示例	395
21.5.7	注意	396
21.6	ssl	396
21.7	SocketServer	399
21.7.1	处理程序	399
21.7.2	服务器	400
21.7.3	定义自定义服务器	401
21.7.4	自定义应用服务器	403
第 22 章 网络应用程序编程		404
22.1	ftplib	404
22.2	http 包	407
22.2.1	http.client (httplib)	408
22.2.2	http.server (BaseHTTPServer、CGIHTTPServer 和 SimpleHTTPServer)	412
22.2.3	http.cookies (Cookie)	416
22.2.4	http.cookiejar (cookiejarlib)	418
22.3	smtplib	418
22.4	urllib 包	419
22.4.1	urllib.request (urllib2)	419
22.4.2	urllib.response	423
22.4.3	urllib.parse	424
22.4.4	urllib.error	426
22.4.5	urllib.robotparser (robotparser)	427
22.4.6	注意	427
22.5	xmlrpc 包	427
22.5.1	xmlrpc.client (xmlrpclib)	427
22.5.2	xmlrpc.server (SimpleXMLRPCServer 和 DocXMLRPCServer)	430
第 23 章 Web 编程		433
23.1	cgi	435
23.1.1	CGI 编程建议	438
23.1.2	注意	439
23.2	cgitb	440
23.3	wsgiref	440
23.3.1	WSGI 规范	440
23.3.2	wsgiref 包	442
23.4	webbrowser	444
第 24 章 网络数据处理和编码		445
24.1	base64	445
24.2	binascii	447
24.3	CSV	447
24.3.1	方言	449
24.3.2	示例	450
24.4	email 包	450
24.4.1	解析电子邮件	450
24.4.2	编写电子邮件	453
24.4.3	注意	456
24.5	hashlib	456
24.6	hmac	456
24.7	HTMLParser	458
24.8	json	460
24.9	mimetypes	462
24.10	quopri	463

24.11	xml 包	463	26.1.4	从 Python 到 C 语言的类型转换 ..	488
24.11.1	XML 示例文档	464	26.1.5	从 C 到 Python 的类型转换	492
24.11.2	xml.dom.minidom	465	26.1.6	给模块添加值	493
24.11.3	xml.etree.ElementTree	468	26.1.7	错误处理	494
24.11.4	xml.sax	474	26.1.8	引用计数	495
24.11.5	xml.sax.saxutils	476	26.1.9	线程	496
第 25 章	其他库模块	477	26.2	嵌入 Python 解释器	496
25.1	Python 服务	477	26.2.1	嵌入模板	497
25.2	字符串处理	478	26.2.2	编译与链接	497
25.3	操作系统模块	478	26.2.3	基本的解释器操作与设置	497
25.4	网络	478	26.2.4	在 C 语言中访问 Python	498
25.5	网络数据处理	478	26.2.5	将 Python 对象转换为 C 对象 ..	499
25.6	国际化	478	26.3	ctypes	500
25.7	多媒体服务	479	26.3.1	加载共享库	500
25.8	其他	479	26.3.2	外来函数	500
			26.3.3	数据类型	501
			26.3.4	调用外来函数	502
			26.3.5	其他类型构造方法	503
			26.3.6	实用工具函数	504
			26.3.7	示例	505
第三部分	扩展与嵌入		26.4	高级扩展与嵌入	506
第 26 章	扩展与嵌入 Python	482	26.5	Jython 和 IronPython	507
26.1	扩展模块	482	附录	Python 3	508
26.1.1	扩展模块原型	484			
26.1.2	命名扩展模块	486			
26.1.3	编译与打包扩展	486			

Part 1

第一部分

Python 语言

本部分内 容

- 第 1 章 Python 简介
- 第 2 章 词汇和语法约定
- 第 3 章 类型与对象
- 第 4 章 运算符与表达式
- 第 5 章 程序结构与控制流
- 第 6 章 函数与函数编程
- 第 7 章 类与面向对象编程
- 第 8 章 模块、包与分发
- 第 9 章 输入与输出
- 第 10 章 执行环境
- 第 11 章 测试、调试、探查与调优



本章将快速介绍Python这门语言，目标是在阐明Python的大部分基本特性的同时，又不会太过纠缠于特殊的规则或细节。为此，本章简要讲述一些基本概念，如变量、表达式、控制流、函数、生成器、类和输入/输出。本章不追求大而全，但有经验的程序员应该能够把本章中的资料推而广之，创建出更加高级的程序。鼓励初学者多尝试一些示例，找到对这门语言的感觉。如果你对Python不熟悉也没有使用过Python 3，可以使用Python 2.6来学习本章内容。本章介绍的几乎所有主要概念同时适用于这两个版本，但在Python 3中有少数关键语法变化（其中大多数与打印和I/O有关），可能会使本章中介绍的许多示例无法运行。请参考附录A，以了解详细信息。

1.1 运行Python

Python程序是由解释器来执行的。通常，只要在命令shell中输入python即可启动解释器。然而，解释器和Python开发环境存在多种实现（如Jython、IronPython、IDLE、ActivePython、Wing IDE、pydev等），因此需要参考相应文档中的启动说明。解释器启动后将出现一个提示符，在此可以开始输入程序，进入简单的读入-求值循环。例如，在下面的输出中，解释器显示了版权消息和>>>提示符，用户可以在提示符后输入熟悉的打印“Hello World”命令：

```
Python 2.6rc2 (r26rc2:66504, Sep 19 2008, 08:50:24)
[GCC 4.0.1 (Apple Inc. build 5465)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print "Hello World"
Hello World
>>>
```

注意 如果在尝试前述例子时出现了语法错误，你使用的很可能就是Python 3。遇到这种情况并不妨碍你继续学习本章的内容，但要注意print语句在Python 3中已经变为一个函数。在下面的例子中，只要在要打印的内容两边加上括号即可正常运行，例如：

```
>>> print("Hello World")
Hello World
>>>
```

如果要打印的内容只有一项，在要打印内容两边放置括号的方法在Python 2中同样有效。然而，这种语法在现有的Python代码中并不常见。在后面的章节中，这种语法有时会用在与打印无直接关系的展示特性的例子中，但这些例子应该同时适用于Python 2和3。

Python的交互模式是它最有用的功能之一。在交互式shell中，可以输入任意合法的语句或语句序列，然后立即查看结果。很多人甚至使用交互式Python作为桌面计算器，作者本人也是如此。例如：

```
>>> 6000 + 4523.50 + 134.12
10657.620000000001
>>> _ + 8192.32
18849.940000000002
>>>
```

以交互模式使用Python时，特殊变量`_`保存最后一次运算的结果。如果要在后续语句中保存或使用最后一次运算的结果，使用此变量十分方便。但要强调一点，此变量只有在以交互模式编程时才会被定义。

如果要创建可以重复运行的程序，可将语句放到一个文件中：

```
# helloworld.py
print "Hello World"
```

Python源文件是普通的文本文件，后缀通常是`.py`。`#`字符表示该行直至行尾的内容都是注释。

要执行`helloworld.py`文件，可通过如下方式将文件名提供给解释器：

```
% python helloworld.py
Hello World
%
```

在Windows中，双击一个`.py`文件或者在Windows开始菜单的“运行”命令中输入程序名称，均可启动Python程序。这会启动解释器，并在控制台窗口中运行程序。但要注意，当程序执行完成后，控制台窗口将立即消失（通常你来不及看清楚输出）。要进行调试，最好是在像IDLE这样的Python开发工具中运行程序。

在UNIX中，可以在程序的首行中使用`#!`，如下所示：

```
#!/usr/bin/env python
print "Hello World"
```

解释器不断运行语句，直到到达输入文件的结尾。如果是交互模式运行，有两种方法可以退出解释器，一种是输入EOF（end of file，文件结束）字符，另一种是从Python IDE的下拉菜单中选择Exit。在UNIX中，EOF是`Ctrl+D`，而在Windows中则是`Ctrl+Z`。程序可以通过抛出`SystemExit`异常来请求退出。

```
>>> raise SystemExit
```

1.2 变量和算术表达式

程序清单1-1中的程序通过执行一次简单的复利计算，说明变量和表达式的用法。

程序清单 1-1 简单的复利计算

```
principal = 1000          # 初始金额
rate = 0.05               # 利率
numyears = 5              # 年数
year = 1
while year <= numyears:
    principal = principal * (1 + rate)
    print year, principal  # 注意：在Python 3中应写成print(year, principal)
    year += 1
```