

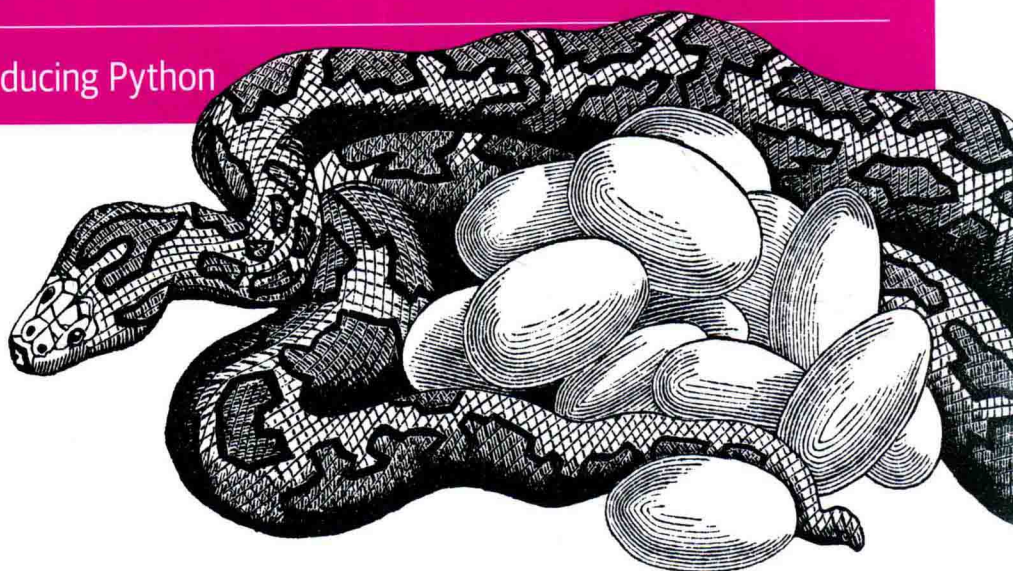
O'REILLY®

TURING

图灵程序设计丛书

# Python 语言及其应用

Introducing Python



[美] Bill Lubanovic 著  
丁嘉瑞 梁杰 禹常隆 译



中国工信出版集团



人民邮电出版社  
POSTS & TELECOM PRESS

**TURING**

图灵程序设计丛书

# Python 语言及其应用

## Introducing Python

[美] Bill Lubanovic 著  
丁嘉瑞 梁杰 禹常隆 译

**O'REILLY®**

*Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo*

O'Reilly Media, Inc. 授权人民邮电出版社出版

人民邮电出版社  
北京

## 图书在版编目 (C I P) 数据

Python语言及其应用 / (美) 卢布诺维克  
(Lubanovic, B.) 著 ; 丁嘉瑞, 梁杰, 禹常隆译. -- 北  
京 : 人民邮电出版社, 2016. 1  
(图灵程序设计丛书)  
ISBN 978-7-115-40709-2

I. ①P… II. ①卢… ②丁… ③梁… ④禹… III. ①  
软件工具—程序设计 IV. ①TP311.56

中国版本图书馆CIP数据核字(2015)第245113号

## 内 容 提 要

本书介绍 Python 语言的基础知识及其在各个领域的具体应用, 基于最新版本 3.x。书中首先介绍了 Python 语言的一些必备基本知识, 然后介绍了在商业、科研以及艺术领域使用 Python 开发各种应用的实例。文字简洁明了, 案例丰富实用, 是一本难得的 Python 入门手册。

本书适合所有编程初学者阅读。

- 
- ◆ 著 [美] Bill Lubanovic
  - 译 丁嘉瑞 梁杰 禹常隆
  - 责任编辑 岳新欣
  - 执行编辑 李艳玲 冯雪松
  - 责任印制 杨林杰
  - ◆ 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: 25.25
  - 字数: 630千字 2016年1月第1版
  - 印数: 1-5 000册 2016年1月北京第1次印刷
  - 著作权合同登记号 图字: 01-2015-3677号



---

定价: 79.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第 0021 号

# O'Reilly Media, Inc.介绍

O'Reilly Media 通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自 1978 年开始，O'Reilly 一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly 的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly 为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了 Make 杂志，从而成为 DIY 革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly 的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly 现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版、在线服务或者面授课程，每一项 O'Reilly 的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

## 业界评论

“O'Reilly Radar 博客有口皆碑。”

——*Wired*

“O'Reilly 凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——*Business 2.0*

“O'Reilly Conference 是聚集关键思想领袖的绝对典范。”

——*CRN*

“一本 O'Reilly 的书就代表一个有用、有前途、需要学习的主题。”

——*Irish Times*

“Tim 是位特立独行的商人，他不光放眼于最长远、最广阔的视野，并且切实地按照 Yogi Berra 的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去，Tim 似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——*Linux Journal*

献给Mary、Karin、Tom和Roxie。

# 前言

本书介绍 Python 编程语言，主要面向编程初学者。不过，如果你是一位有经验的程序员，想再学门 Python 编程语言，本书也很适合作为入门读物。

本书节奏适中，从基础开始逐步深入其他话题。我会结合食谱和教程的风格来解释新术语和新概念，但不会一次介绍很多。你会尽早并且常常接触到真实的 Python 代码。

虽然本书是入门读物，但我还是介绍了一些看起来比较高阶的话题，比如 NoSQL 数据库和消息传递库。之所以介绍它们，是因为在解决某类问题时它们比标准库更加合适。你需要下载并安装这些第三方 Python 包，从而更好地理解 Python “内置电池”适用于什么场景。此外，尝试新事物本身也充满乐趣。

我还会展示一些反面的例子，提醒你不要太那么去做。如果你之前使用过其他语言并且想把风格照搬到 Python 的话，要格外注意。还有，我不认为 Python 是完美的，我会告诉你哪些东西应该避免。



书中有时会出现类似本条的提示内容，主要用于解释一些容易混淆的概念或者用更合适的 Python 风格的方法来解决同一个问题。

## 目标读者

本书的目标读者是那些对世界上最流行的计算语言感兴趣的人，无论你之前是否学过编程。

## 本书结构

本书前 7 章介绍 Python 基础知识，建议按顺序阅读。后面 5 章介绍如何在不同的应用场景中使用 Python，比如 Web、数据库、网络，等等，可以按任意顺序阅读。附录 A、B、C 介绍 Python 在艺术、商业和科学方面的应用，附录 D 是 Python 3 的安装教程，附录 E 和附录 F 是每章练习题的答案和速查表。

- **第 1 章**  
程序和织袜子或者烤土豆很像。通过一些真实的 Python 程序可以了解这门语言的概貌、能力以及在真实世界中的用途。Python 和其他语言相比有很多优势，不过也有一些不完美的地方。旧版本的 Python (Python 2) 正在被新版本 (Python 3) 替代。如果你在使用 Python 2，请安装 Python 3。你可以使用交互式解释器自行尝试本书中的代码示例。
- **第 2 章**  
该章会介绍 Python 中最简单的数据类型：布尔值、整数、浮点数和文本字符串。你也会学习基础的数学和文本操作。
- **第 3 章**  
该章会学习 Python 的高级内置数据结构：列表、元组、字典和集合。你可以像玩乐高积木一样用它们来构建更复杂的结构，并学到如何使用迭代器和推导式来遍历它们。
- **第 4 章**  
该章会学习如何在之前学习的数据结构上用代码实现比较、选择和重复操作。你会学习如何用函数来组织代码，并用异常来处理错误。
- **第 5 章**  
该章会介绍如何使用模块、包和程序组织大型代码结构。你会学习如何划分代码和数据、数据的输入输出、处理选项、使用 Python 标准库并了解标准库的内部实现。
- **第 6 章**  
如果你已经在其他语言中学过面向对象编程，就可以轻松掌握 Python 的写法。该章会介绍对象和类的适用场景，有时候使用模块甚至列表和字典会更加合适。
- **第 7 章**  
该章会介绍如何像专家一样处理数据。你会学到如何处理文本和二进制数据以及 Unicode 字符和 I/O。
- **第 8 章**  
数据需要地方来存放。在该章中，你首先会学习使用普通文件、目录和文件系统，接着会学习如何处理常用文件格式，比如 CSV、JSON 和 XML。此外，你还会了解如何从关系型数据库甚至是最新的 NoSQL 数据库中存取数据。
- **第 9 章**  
该章单独介绍 Web，包括客户端、服务器、数据抓取、API 和框架。你会编写一个带请求参数处理和模板的真实网站。
- **第 10 章**  
该章会介绍系统相关内容，难度较高。你会学习如何管理程序、进程和线程，处理日期和时间，实现系统管理任务自动化。
- **第 11 章**  
该章会介绍网络相关内容：服务、协议和 API。该章示例覆盖了底层 TCP 套接字、消

息库以及队列系统、云端部署。

- 第 12 章

该章会介绍 Python 相关的小技巧，比如安装、使用 IDE、测试、调试、日志、版本控制和文档，还会介绍如何寻找并安装有用的第三方包、打包自己的代码以供重用，以及如何寻找更多有用的信息。祝你好运。

- 附录 A

附录 A 会介绍 Python 在艺术领域的应用：图像、音乐、动画和游戏。

- 附录 B

Python 在商业领域也有应用：数据可视化（图表、图形和地图）、安全和管理。

- 附录 C

Python 在科学领域应用得尤其广泛：数学和统计学、物理科学、生物科学以及医学。附录 C 会介绍 NumPy、SciPy 和 Pandas。

- 附录 D

如果你还没有安装 Python 3，附录 D 会介绍 Windows、Mac OS/X、Linux 和 Unix 下的安装方法。

- 附录 E

附录 E 包含每章结尾的练习答案。请在亲自尝试解答之后再查看答案。

- 附录 F

附录 F 包含一些有用的速查内容。

## Python 版本

开发者会不断向计算机语言中加入新特性、修复问题，因此计算机语言一直在变化。本书中的代码示例在 Python 3.3 中编写和测试。在本书编辑期间 Python 3.4 发布了，我会介绍一些新版本的内容。如果你想了解相关信息和特性的发布时间，可以阅读 [What's New in Python](https://docs.python.org/3/whatsnew/) 页面 (<https://docs.python.org/3/whatsnew/>)。这个页面技术性比较强，对于 Python 初学者来说难度较大，不过如果你之后想研究 Python 的兼容性，可以阅读它。

## 排版约定

本书使用了下列排版约定。

- 楷体

表示新术语。

- 等宽字体 (constant width)

表示程序片段，以及正文中出现的变量、函数名、数据库、数据类型、环境变量、语句和关键字等。



- 加粗等宽字体 (**constant width bold**)  
表示应该由用户输入的命令或其他文本。



该图标表示一般注记。



该图标表示警告或警示。

## 使用代码示例

补充材料（代码示例、练习等）可以从 <https://github.com/madscheme/introducing-python> 下载。

本书是要帮你完成工作的。一般来说，如果本书提供了示例代码，你可以把它用在你的程序或文档中。除非你使用了很大一部分代码，否则无需联系我们获得许可。比如，用本书的几个代码片段写一个程序就无需获得许可，销售或分发 O'Reilly 图书的示例光盘则需要获得许可；引用本书中的示例代码回答问题无需获得许可，将书中大量的代码放到你的产品文档中则需要获得许可。

我们很希望但并不强制要求你在引用本书内容时加上引用说明。引用说明一般包括书名、作者、出版社和 ISBN。比如：“*Introducing Python* by Bill Lubanovic(O'Reilly). Copyright 2015 Bill Lubanovic, 978-1-449-35936-2.”

如果你觉得自己对示例代码的用法超出了上述许可的范围，欢迎你通过 [permissions@oreilly.com](mailto:permissions@oreilly.com) 与我们联系。

## Safari<sup>®</sup> Books Online



Safari Books Online (<http://www.safaribooksonline.com>) 是应运而生的数字图书馆。它同时以图书和视频的形式出版世界顶级技术和商务作家的专业作品。技术专家、软件开发人员、Web 设计师、商务人士和创意专家等，在开展调研、解决问题、学

习和认证培训时，都将 Safari Books Online 视作获取资料的首选渠道。

对于组织团体、政府机构和个人，Safari Books Online 提供各种产品组合和灵活的定价策略。用户可通过一个功能完备的数据库检索系统访问 O'Reilly Media、Prentice Hall Professional、Addison-Wesley Professional、Microsoft Press、Sams、Que、Peachpit Press、Focal Press、Cisco Press、John Wiley & Sons、Syngress、Morgan Kaufmann、IBM

Redbooks、Packt、Adobe Press、FT Press、Apress、Manning、New Riders、McGraw-Hill、Jones & Bartlett、Course Technology 以及其他几十家出版社的上千种图书、培训视频和正式出版之前的书稿。要了解 Safari Books Online 的更多信息，我们网上见。

## 联系我们

请把对本书的评价和问题发给出版社。

美国：

O'Reilly Media, Inc.  
1005 Gravenstein Highway North  
Sebastopol, CA 95472

中国：

北京市西城区西直门南大街 2 号成铭大厦 C 座 807 室 (100035)  
奥莱利技术咨询 (北京) 有限公司

O'Reilly 的每一本书都有专属网页，你可以在那儿找到本书的相关信息，包括勘误表、示例代码以及其他信息。本书的网站地址是：

<http://shop.oreilly.com/product/0636920028659.do>

对于本书的评论和技术性问题，请发送电子邮件到：

[bookquestions@oreilly.com](mailto:bookquestions@oreilly.com)

要了解更多 O'Reilly 图书、培训课程、会议和新闻的信息，请访问以下网站：

<http://www.oreilly.com>

我们在 Facebook 的地址如下：

<http://facebook.com/oreilly>

请关注我们的 Twitter 动态：

<http://twitter.com/oreillymedia>

我们的 YouTube 视频地址如下：

<http://www.youtube.com/oreillymedia>

## 致谢

非常感谢那些阅读本书初稿并给予反馈的人。我尤其要感谢仔细审阅本书的 Eli Bessert、Henry Canival、Jeremy Elliott、Monte Milanuk、Loïc Pefferkorn 和 Steven Wayne。

# 目录

前言	xiv
第 1 章 Python 初探	1
1.1 真实世界中的 Python	5
1.2 Python 与其他语言	5
1.3 为什么选择 Python	7
1.4 何时不应该使用 Python	8
1.5 Python 2 与 Python 3	8
1.6 安装 Python	9
1.7 运行 Python	9
1.7.1 使用交互式解释器	9
1.7.2 使用 Python 文件	10
1.7.3 下一步	11
1.8 禅定一刻	11
1.9 练习	11
第 2 章 Python 基本元素：数字、字符串和变量	13
2.1 变量、名字和对象	13
2.2 数字	16
2.2.1 整数	17
2.2.2 优先级	20
2.2.3 基数	21
2.2.4 类型转换	22
2.2.5 一个 int 型有多大	23
2.2.6 浮点数	24

2.2.7	数学函数 .....	24
2.3	字符串 .....	24
2.3.1	使用引号创建 .....	25
2.3.2	使用 <code>str()</code> 进行类型转换 .....	27
2.3.3	使用 <code>\</code> 转义 .....	27
2.3.4	使用 <code>+</code> 拼接 .....	28
2.3.5	使用 <code>*</code> 复制 .....	28
2.3.6	使用 <code>[]</code> 提取字符 .....	28
2.3.7	使用 <code>[start:end:step]</code> 分片 .....	29
2.3.8	使用 <code>len()</code> 获得长度 .....	31
2.3.9	使用 <code>split()</code> 分割 .....	32
2.3.10	使用 <code>join()</code> 合并 .....	32
2.3.11	熟悉字符串 .....	32
2.3.12	大小写与对齐方式 .....	33
2.3.13	使用 <code>replace()</code> 替换 .....	34
2.3.14	更多关于字符串的内容 .....	35
2.4	练习 .....	35
<b>第 3 章</b>	<b>Python 容器：列表、元组、字典与集合</b> .....	<b>36</b>
3.1	列表和元组 .....	36
3.2	列表 .....	37
3.2.1	使用 <code>[]</code> 或 <code>list()</code> 创建列表 .....	37
3.2.2	使用 <code>list()</code> 将其他数据类型转换成列表 .....	37
3.2.3	使用 <code>[offset]</code> 获取元素 .....	38
3.2.4	包含列表的列表 .....	39
3.2.5	使用 <code>[offset]</code> 修改元素 .....	39
3.2.6	指定范围并使用切片提取元素 .....	40
3.2.7	使用 <code>append()</code> 添加元素至尾部 .....	40
3.2.8	使用 <code>extend()</code> 或 <code>+=</code> 合并列表 .....	40
3.2.9	使用 <code>insert()</code> 在指定位置插入元素 .....	41
3.2.10	使用 <code>del</code> 删除指定位置的元素 .....	41
3.2.11	使用 <code>remove()</code> 删除具有指定值的元素 .....	42
3.2.12	使用 <code>pop()</code> 获取并删除指定位置的元素 .....	42
3.2.13	使用 <code>index()</code> 查询具有特定值的元素位置 .....	42
3.2.14	使用 <code>in</code> 判断值是否存在 .....	42
3.2.15	使用 <code>count()</code> 记录特定值出现的次数 .....	43
3.2.16	使用 <code>join()</code> 转换为字符串 .....	43
3.2.17	使用 <code>sort()</code> 重新排列元素 .....	44
3.2.18	使用 <code>len()</code> 获取长度 .....	44
3.2.19	使用 <code>=</code> 赋值，使用 <code>copy()</code> 复制 .....	45

3.3	元组	46
3.3.1	使用 () 创建元组	46
3.3.2	元组与列表	47
3.4	字典	47
3.4.1	使用 {} 创建字典	48
3.4.2	使用 dict() 转换为字典	48
3.4.3	使用 [key] 添加或修改元素	49
3.4.4	使用 update() 合并字典	50
3.4.5	使用 del 删除具有指定键的元素	51
3.4.6	使用 clear() 删除所有元素	51
3.4.7	使用 in 判断是否存在	51
3.4.8	使用 [key] 获取元素	52
3.4.9	使用 keys() 获取所有键	52
3.4.10	使用 values() 获取所有值	53
3.4.11	使用 items() 获取所有键值对	53
3.4.12	使用 = 赋值, 使用 copy() 复制	53
3.5	集合	53
3.5.1	使用 set() 创建集合	54
3.5.2	使用 set() 将其他类型转换为集合	54
3.5.3	使用 in 测试值是否存在	55
3.5.4	合并及运算符	56
3.6	比较几种数据结构	58
3.7	建立大型数据结构	59
3.8	练习	60
第 4 章 Python 外壳: 代码结构		61
4.1	使用 # 注释	61
4.2	使用 \ 连接	62
4.3	使用 if、elif 和 else 进行比较	63
4.4	使用 while 进行循环	66
4.4.1	使用 break 跳出循环	66
4.4.2	使用 continue 跳到循环开始	67
4.4.3	循环外使用 else	67
4.5	使用 for 迭代	68
4.5.1	使用 break 跳出循环	69
4.5.2	使用 continue 跳到循环开始	69
4.5.3	循环外使用 else	69
4.5.4	使用 zip() 并行迭代	70
4.5.5	使用 range() 生成自然数序列	71
4.5.6	其他迭代方式	71

4.6	推导式	72
4.6.1	列表推导式	72
4.6.2	字典推导式	74
4.6.3	集合推导式	74
4.6.4	生成器推导式	74
4.7	函数	75
4.7.1	位置参数	79
4.7.2	关键字参数	79
4.7.3	指定默认参数值	79
4.7.4	使用 * 收集位置参数	80
4.7.5	使用 ** 收集关键字参数	81
4.7.6	文档字符串	82
4.7.7	一等公民：函数	82
4.7.8	内部函数	84
4.7.9	闭包	84
4.7.10	匿名函数：lambda() 函数	85
4.8	生成器	86
4.9	装饰器	87
4.10	命名空间和作用域	89
4.11	使用 try 和 except 处理错误	91
4.12	编写自己的异常	93
4.13	练习	94
<b>第 5 章</b>	<b>Python 盒子：模块、包和程序</b>	<b>95</b>
5.1	独立的程序	95
5.2	命令行参数	96
5.3	模块和 import 语句	96
5.3.1	导入模块	96
5.3.2	使用别名导入模块	98
5.3.3	导入模块的一部分	98
5.3.4	模块搜索路径	98
5.4	包	99
5.5	Python 标准库	99
5.5.1	使用 setdefault() 和 defaultdict() 处理缺失的键	100
5.5.2	使用 Counter() 计数	101
5.5.3	使用有序字典 OrderedDict() 按键排序	103
5.5.4	双端队列：栈 + 队列	103
5.5.5	使用 itertools 迭代代码结构	104
5.5.6	使用 pprint() 友好输出	105

5.6 获取更多 Python 代码	105
5.7 练习	106
<b>第 6 章 对象和类</b>	<b>107</b>
6.1 什么是对象	107
6.2 使用 class 定义类	108
6.3 继承	109
6.4 覆盖方法	111
6.5 添加新方法	112
6.6 使用 super 从父类得到帮助	112
6.7 self 的自辩	113
6.8 使用属性对特性进行访问和设置	114
6.9 使用名称重整保护私有特性	117
6.10 方法的类型	118
6.11 鸭子类型	119
6.12 特殊方法	120
6.13 组合	123
6.14 何时使用类和对象而不是模块	124
6.15 练习	126
<b>第 7 章 像高手一样玩转数据</b>	<b>127</b>
7.1 文本字符串	127
7.1.1 Unicode	127
7.1.2 格式化	134
7.1.3 使用正则表达式匹配	137
7.2 二进制数据	144
7.2.1 字节和字节数组	144
7.2.2 使用 struct 转换二进制数据	145
7.2.3 其他二进制数据工具	148
7.2.4 使用 binascii() 转换字节 / 字符串	149
7.2.5 位运算符	149
7.3 练习	149
<b>第 8 章 数据的归宿</b>	<b>152</b>
8.1 文件输入 / 输出	152
8.1.1 使用 write() 写文本文件	153
8.1.2 使用 read()、readline() 或者 readlines() 读文本文件	154
8.1.3 使用 write() 写二进制文件	156
8.1.4 使用 read() 读二进制文件	157
8.1.5 使用 with 自动关闭文件	157

8.1.6	使用 seek() 改变位置	157
8.2	结构化的文本文件	159
8.2.1	CSV	159
8.2.2	XML	161
8.2.3	HTML	163
8.2.4	JSON	163
8.2.5	YAML	165
8.2.6	安全提示	166
8.2.7	配置文件	167
8.2.8	其他交换格式	168
8.2.9	使用 pickle 序列化	168
8.3	结构化二进制文件	169
8.3.1	电子数据表	169
8.3.2	层次数据格式	169
8.4	关系型数据库	170
8.4.1	SQL	170
8.4.2	DB-API	172
8.4.3	SQLite	172
8.4.4	MySQL	174
8.4.5	PostgreSQL	174
8.4.6	SQLAlchemy	174
8.5	NoSQL 数据存储	179
8.5.1	dbm family	180
8.5.2	memcached	180
8.5.3	Redis	181
8.5.4	其他的 NoSQL	189
8.6	全文数据库	189
8.7	练习	190
<b>第 9 章</b>	<b>剖析 Web</b>	<b>191</b>
9.1	Web 客户端	192
9.1.1	使用 telnet 进行测试	193
9.1.2	Python 的标准 Web 库	194
9.1.3	抛开标准库: requests	195
9.2	Web 服务端	196
9.2.1	最简单的 Python Web 服务器	196
9.2.2	Web 服务器网关接口	198
9.2.3	框架	198
9.2.4	Bottle	198
9.2.5	Flask	201



9.2.6	非 Python 的 Web 服务器	204
9.2.7	其他框架	206
9.3	Web 服务和自动化	207
9.3.1	webbrowser 模块	207
9.3.2	Web API 和表述性状态传递	208
9.3.3	JSON	209
9.3.4	抓取数据	209
9.3.5	用 BeautifulSoup 来抓取 HTML	209
9.4	练习	210
<b>第 10 章</b>	<b>系统</b>	<b>212</b>
10.1	文件	212
10.1.1	用 open() 创建文件	212
10.1.2	用 exists() 检查文件是否存在	213
10.1.3	用 isfile() 检查是否为文件	213
10.1.4	用 copy() 复制文件	213
10.1.5	用 rename() 重命名文件	214
10.1.6	用 link() 或者 symlink() 创建链接	214
10.1.7	用 chmod() 修改权限	214
10.1.8	用 chown() 修改所有者	214
10.1.9	用 abspath() 获取路径名	215
10.1.10	用 realpath() 获取符号的路径名	215
10.1.11	用 remove() 删除文件	215
10.2	目录	215
10.2.1	使用 mkdir() 创建目录	215
10.2.2	使用 rmdir() 删除目录	215
10.2.3	使用 listdir() 列出目录内容	216
10.2.4	使用 chdir() 修改当前目录	216
10.2.5	使用 glob() 列出匹配文件	216
10.3	程序和进程	217
10.3.1	使用 subprocess 创建进程	218
10.3.2	使用 multiprocessing 创建进程	219
10.3.3	使用 terminate() 终止进程	219
10.4	日期和时间	220
10.4.1	datetime 模块	221
10.4.2	使用 time 模块	223
10.4.3	读写日期和时间	225
10.4.4	其他模块	227
10.5	练习	228