

Join the discussion @ [p2p.wrox.com](http://p2p.wrox.com)



Wrox Programmer to Programmer™



Beginning Python: Using Python 2.6 and Python 3.1

# Python

# 编程入门经典

(美) James Payne 著  
张春晖 译

清华大学出版社

# Python 编程入门经典

(美) James Payne 著

张春晖 译

清华大学出版社

北 京

James Payne

Beginning Python: Using Python 2.6 and Python 3.1

EISBN: 978-0-470-41463-7

Copyright © 2010 by Wiley Publishing, Inc., Indianapolis, Indiana

All Rights Reserved. This translation published under license.

本书中文简体字版由 Wiley Publishing, Inc. 授权清华大学出版社出版。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。

北京市版权局著作权合同登记号 图字：01-2010-1678

本书封面贴有 Wiley 公司防伪标签，无标签者不得销售。

版权所有，侵权必究。举报电话：010-62782989 13701121933

图书在版编目(CIP)数据

Python 编程入门经典/(美) 佩恩(Payne, J.) 著; 张春晖 译. —北京: 清华大学出版社, 2011.7

书名原文: Beginning Python: Using Python 2.6 and Python 3.1

ISBN 978-7-302-25709-7

I. P… II. ①佩… ②张… III. 软件工具—程序设计 IV. TP311.56

中国版本图书馆 CIP 数据核字(2011)第 106864 号

责任编辑: 王 军 赵利通

装帧设计: 孔祥丰

责任校对: 胡雁翎

责任印制: 王秀菊

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 清华大学印刷厂

装 订 者: 三河市溧源装订厂

经 销: 全国新华书店

开 本: 185×260 印 张: 34.75 字 数: 846 千字

版 次: 2011 年 7 月第 1 版 印 次: 2011 年 7 月第 1 次印刷

印 数: 1~4000

定 价: 68.00 元

---

产品编号: 033391-01

# 作者简介

James Payne 居住在佛罗里达州马盖特市，他是 Developer Shed 公司的主编，从 7 岁起就开始写作和编程。James 精通多门语言，已经撰写了 400 余篇文章，囊括了几乎每一种主流编程语言。James 的工作是使用 Python 为金融业开发专有软件，在业余时间他喜欢涉猎 Django 方面的知识。

# 致 谢

我感谢在 Developer Shed 的同事：Jack 和 Jay Kim 持续的敌意使我备受鼓舞；Charles Fagundes 使我成为一名编辑，而不仅仅是一个作者；Keith Lee 为我提供了编写代码的支持；特别要感谢 Jenny Ruggieri，他使我得到了这份工作，所以有了这本书的面世。最后，还要感谢所有为这本书的前期版本付出艰辛劳动的人们，他们的工作是本书的基础。还要感谢编辑 Carol Long、Jenny Watson、Ed Connor 以及 Chris McAvoy，他们帮助我在最终期限前完成本书。

我还想感谢 Guido Van Rossum，没有他，也就没有 Python 语言可写。

# 前言

欢迎使用 Python 3.1!

我使用 Python 已有 10 年左右，每一个新版本都使我越来越喜欢这门语言。版本 3.1 也不例外。如果您是 Python 的初学者，不必担心，我会一路引领着您。相反，如果您是一名希望尝试新版本的 Python 老手，可以不必在已掌握的知识上浪费时间，而是直接学习所需的知识，因为本书的结构十分清晰。

之所以写这本书，是因为我热爱 Python。我非常喜爱它！我想和您一起分享我对 Python 的热爱。您可能也会像我一样越来越喜爱它。

## 本书读者对象

如果您具备计算机知识，并且想学习一门有趣的程序语言来更好地控制计算机，那么本书适合您。

如果您是系统管理员，希望学习一门优秀的语言来更好地管理和配置系统和网络，那么本书适合您。

如果您已经了解 Python，但是想知道版本 3.1 中有哪些很酷的新功能，那么本书适合您。

总之，本书适合于所有热衷于使用功能最丰富且十分易用的最新版本 Python 3.1 进行编程的读者。

## 本书主要内容

本书主要介绍 Python 3.1。Python 3.1 发布于 2009 年，是 Python 程序语言的最新主版本。由于 Python 是一门跨平台的语言，本书中的内容和示例适用于任何平台(除非特别指出的例外情况)。在可能独立于平台的情况下，本书中的示例都尽量设计成跨平台的。

此外，由于 Python 3.1 相对较新，并非所有支持库都被更新到可以在 Python 3.x 下工作。在这种情形下，如果某些理论需要详细解释，本书将使用 Python 2.6 代替 Python 3.1。

## 本书结构

就像所有入门书籍一样，本书首先对语言进行了介绍。从这里开始，会接触到语言的

核心，进而接触到更加高级的专门主题。本书分为如下 4 个部分。

## 第 I 部分——初步了解 Python

如标题所示，第 I 部分让您初步了解 Python。

### 编程基础和字符串

本章首先介绍 Python，探讨 Python 是什么，为什么它如此有用和强大。另外，还探讨 Python 的开发历史和最新的版本，后者正是本书的讨论重点。您会了解到 Python 所涉及的范围，以及 Python 可能参与的应用开发领域。最后，您将开始操作自己的第一种数据类型：字符串。

### 数值与运算符

本章会介绍数值和运算符的基本知识。您将会学习不同类型的数值，如何执行简单的和复杂的公式，以及如何使用各种运算符。还将学到运算符的优先级以及数值的格式化。

### 变量

编程语言最终帮助程序员管理不同类型的信息，换言之，就是管理数据。理解数据类型并且知道 Python 如何表示它们对于 Python 编程非常必要。本章将帮助您理解在 Python 中表示不同数据类型的最佳方式。

## 第 II 部分——Python 语言和标准库

当然，使用一门语言最需要的是了解语言本身，熟悉它的语法和模块。本部分首先介绍数据类型和变量，并逐步介绍其他概念，直到获得开发功能完善的 Python 程序所需的全部知识。

注意，应该按顺序阅读本部分的章节，因为每一章都建立在前一章的基础之上。

### 做出决策

最终，程序将需要在某个位置做出决策：是选择这条路径，还是选择另一条路径？如果选择那条路径，将会发生什么？在本章中将学到如何比较数据，例如判断某个值是否比另一个值大，并且将学习如何利用循环来完成重复性的任务。

### 函数

本章将介绍函数编程，这将帮助您扩展 Python 知识。函数允许使用者利用强大的概念，例如参数传递和代码重用。本章将介绍如何利用函数使编写的代码更加高效和灵活。

### 类和对象

本章将介绍对象的概念。您将学习如何定义类和如何在类中创建对象和编写方法，还将学到有关对象作用域的知识。

## 组织程序

当程序变大时，需要把它们分成单独的几部分。本章将讨论 Python 模块，并探索包的概念。包并不复杂，它仅是若干模块的集合。

## 文件和目录

日常编程中一个重要的部分是学习如何处理文件和目录。本章着重介绍通常如何创建、修改和处理文件。另外，还将介绍如何从文件中获取数据，以及如何与各种各样的目录交互。

## Python 语言的其他特性

本章将介绍 Python 语言提供的其他特性，包括如何使用列表做出决策，如何使用字典进行字符串替换，以及一些有特色的模块。

## 创建模块

通过允许重用代码片段，模块能够帮助节省时间。它能减少发生错误的机率，因为当前用到的模块可能之前已经被测试和使用过很多次。本章将介绍如何创建自己的模块，以及如何导入并处理已存在的模块，这些已经存在的模块使得 Python 在应用中功能非常强大。

## 文本处理

在编程过程中，可以利用文本做很多事情，而且本质上，文本是与用户进行有效沟通的关键。毕竟，如果没有文本，就只能使用图片了。本章将介绍如何用多种方法处理文本，包括利用正则表达式、搜索文件以及搜索特定类型的文件等。

## 第III部分——开始使用 Python

现在您已经知道了 Python 是什么，以及如何利用这门语言进行工作，接下来应该做什么呢？本部分探究了您会遇到或者想了解了许多编程主题。可以顺序阅读本部分的各章，也可以按其他任意顺序阅读，因为它们是彼此独立的。

### 测试

在把程序交付给用户之前，对程序进行测试，这是保证它能正常工作的唯一途径。本章不仅会介绍正确测试程序的一些基本概念，还会介绍一些可用的测试工具和测试框架。

### 使用 Python 编写 GUI

目前为止，本书中的程序都是通过命令行方式工作。本章将介绍 GUI 编程的概念。您将使用 Tkinter 创建几个 GUI 程序，Tkinter 是 Python 程序员最常用的 GUI 工具包。

### 访问数据库

数据库存储了程序可以使用的各种信息。它也可以作为存储信息的地方，以便日后可



以从中检索出特定任务所需的数据。本章将介绍各种类型的数据库以及如何利用它们进行工作。

### 使用 Python 处理 XML

XML 是处理 Internet 数据的强大工具。本章将介绍 XML 的基础知识,包括模式与 DTD 之间的区别、基本的语法、如何创建和验证 XML 以及一些高级的主题(例如使用 lxml)。

### 网络编程

Internet 已经潜移默化地进入了我们的日常生活,并逐渐变成必需品,而不是一种特权。本章将介绍如何使用程序发送 E-mail, 以及如何允许用户通过 Web 进行交流。

### 用 C 扩展编程

本章深入介绍如何使用 C 语言编程,包括如何利用 C 的框架和模块, C 的基础知识, 以及如何从 Python 向 C 传递参数,并在之后将值返回给 Python。

### 数值编程

本书的开始部分简单介绍了数值,现在是进一步钻研的时候了。本章将深入介绍整型、浮点型、复数和数组的全部知识,以及如何使用内置的数学函数和模块。

### Django 简介

Django 是用 Python 编写的 Web 应用程序框架,它利用了模型-视图-控制器模式。Django 最初被用来管理新闻 Web 站点,但由于简单易用而变得流行起来,它使得程序员可以用简单的方式创建复杂的 Web 站点,包括以数据库为中心的 Web 站点。本章将介绍 Django 的基础知识。

### Web 应用程序和 Web 服务

本章将介绍 Web 应用程序和 Web 服务的基础知识,例如 REST 架构,以及如何处理 HTTP 请求和响应。

### 集成 Java 与 Python

本章将介绍 Java 的基础知识,这样可以在研究如何结合 Python 和 Java 之前,打好坚实的基础。还将介绍各种帮助在 Python 中使用 Java 的模块,以及如何创建简单但有效的应用程序。

## 第IV部分：附录

本书结尾的附录可以进一步扩展您对 Python 的了解,激发您对 Python 的学习兴趣。

- 习题答案
- 在线资源
- Python 3.1 的新特性

- 术语表

## 使用本书的条件

使用本书有一些条件。在此给出下面列出的推荐，因为 Python 本身可在多种不同的平台上运行。然而，本书的前几章假定您可以使用 GUI，例如在 Windows、Mac OS X、UNIX 和 Linux 上的 X Window 等系统上可用的 GUI。自然地，一些章节(例如介绍 GUI 的那一章)也需要 GUI，而如果有网络连接，涉及网络的那些章节会更有意义。

下面是建议的最低配置需求：

- 一台 PC 机，运行 Linux、BSD UNIX 或 Windows 系统，频率在 500MHz 以上；或者运行 Mac OS X 版本 10.2 或更新版本的 G3 或更新的 Macintosh。
- 至少 256MB 内存。
- 所用平台具有图形用户界面。
- 对所用计算机有必要的权限，以安装必需的软件。
- 可以访问 TCP/IP 网络，例如 Internet 或者校园网。
- 可以通过 Internet 下载所需软件。

## 源代码

在阅读本书提供的代码时既可以亲自输入所有代码，也可以使用随书提供的代码文件。本书所有代码均可以从 <http://www.wrox.com>(或 <http://www.tupwk.com.cn/downpage>)网站下载。进入该网站后，请根据本书的书名查找本书(读者既可以使用搜索框进行查找，也可以使用书名列表进行查找)，然后单击本书详细内容页面上提供的 Download Code 链接，就可以下载本书提供的所有代码。

注意：

由于许多书籍名称与本书类似，因此也可以通过 ISBN 进行查找，本书英文版的 ISBN 为：978-0-470-41463-7。

下载代码后，可以利用一种压缩工具解压代码。此外，还可以通过访问网站 <http://www.wrox.com/dynamic/books/download.aspx> 中提供的 Wrox 代码下载页面来获取本书提供的代码，也可以下载 Wrox 出版的其他书籍提供的代码。

## 勘误表

为了避免本书文字和代码中存在错误，我们已经竭尽全力。然而，就如世界上不存在完美无缺的事物，本书仍然可能存在错误。如果您在我们编写的书籍中发现了诸如拼写错误或代码缺陷等问题，请告诉我们，我们对此表示感谢。利用勘误表反馈错误信息，可以为其他

读者节省大量时间，同时，我们也能够受益于您的帮助，编写出质量更高的专业著作。

如果需要参考本书的勘误表，请在网站 <http://www.wrox.com> 中用搜索框或书名列表查找本书书名。然后，在本书的详细内容页面上，单击 **Book Errata** 链接。在随后显示的页面中，可以看到与本书相关的所有勘误信息，这些信息是由读者提交、并由 Wrox 的编辑们加上的。通过访问 [www.wrox.com/misc-pages/booklist.shtml](http://www.wrox.com/misc-pages/booklist.shtml)，还可以看到 Wrox 出版的所有书籍的勘误表。

如果没有在 **Book Errata** 页面上找到自己发现的错误，那么请转到页面 [www.wrox.com/contact/techsupport.shtml](http://www.wrox.com/contact/techsupport.shtml)，针对您所发现的每一项错误填写表格，并将表格发给我们，我们将对表格内容进行认真审查，如果确实是我们书中的错误，那么我们将在该书的 **Book Errata** 页面上标明该错误信息，并在该书的后续版本中改正相关错误。

## 关于 p2p.wrox.com

如果您希望能够与作者进行讨论，或希望能够参与读者的共同讨论，那么请加入 [p2p.wrox.com](http://p2p.wrox.com) 的论坛。这个论坛是一个基于 Web 的系统，您可以在论坛发表与 Wrox 出版的书籍有关的技术信息，并与其他读者和技术用户进行讨论。论坛提供了订阅功能，可以将与您所选主题相关的新帖子定期发送到您的电子邮箱。Wrox 的作者、编辑、业界专家以及其他读者都会参与论坛中的讨论。

您可以在 <http://p2p.wrox.com> 参与多个论坛的讨论，这些论坛不仅能够帮助您更好地理解本书，还有助于您更好地开发应用程序。如果希望加入论坛，请按照以下步骤执行：

- (1) 进入 <http://p2p.wrox.com> 页面，单击 **Register** 链接。
- (2) 阅读使用条款，然后单击 **Agree**。
- (3) 填写必要的信息(以及想要提供的可选信息)，然后单击 **Submit**。
- (4) 随后会收到一封电子邮件，邮件中说明了如何验证账号并完成整个加入过程。

### 注意：

要阅读论坛信息，无须加入 P2P。但是如果需要发表主题或发表回复，那么必须加入论坛。

成功加入论坛后，就可以发表新主题了。此外，还可以回复其他主题。您在任何时间都可以阅读论坛信息。如果需要论坛将新的信息发送到自己的电子邮箱，那么可以单击论坛列表中论坛名称旁的 **Subscribe to this Forum** 图标完成该功能设置。

如果需要获得更多与 Wrox P2P 相关的信息，请阅读 **P2P FAQs**，这样可以获得大量与 P2P 和 Wrox 出版的书籍相关的具体信息。阅读 **FAQs** 时，请单击 **P2P** 页面上的 **FAQs** 链接。

# 目 录

第 I 部分 初步了解 Python	
第 1 章 编程基础和字符串	3
1.1 编程与使用计算机的区别	3
1.1.1 编程的一致性	3
1.1.2 编程的可控性	4
1.1.3 程序要应对变化	4
1.1.4 小结	4
1.2 准备工作	4
1.2.1 在非 Windows 系统上安装 Python 3.1	5
1.2.2 使用 Python Shell	5
1.3 开始使用 Python——字符串	6
1.3.1 字符串概述	6
1.3.2 为什么需要引号	6
1.3.3 为什么有 3 种类型的引号	7
1.3.4 使用 print() 函数	7
1.3.5 理解不同的引号	8
1.4 串联两个字符串	10
1.5 用不同的方法串联字符串	11
1.6 本章小结	12
1.7 习题	13
第 2 章 数值与运算符	15
2.1 不同类型的数值	15
2.2 程序文件	17
2.2.1 使用不同的类型	18
2.2.2 基本算术	20
2.2.3 一些惊喜	22
2.3 使用数值	23
2.3.1 求值顺序	23
2.3.2 数值格式	24
2.3.3 可能会发生的错误	25
2.3.4 几个不寻常的例子	26
2.4 本章小结	26
2.5 习题	27
第 3 章 变量	29
3.1 引用数据——使用数据的 名称	29
3.1.1 使用名称修改数据	30
3.1.2 复制数据	31
3.1.3 禁用的名称以及一些规则	31
3.2 使用更多的内置类型	31
3.2.1 元组——不可更改的数据 序列	32
3.2.2 列表——可以更改的数据 序列	35
3.2.3 字典——以名称索引的分组 数据	36
3.2.4 像列表一样处理字符串	38
3.2.5 特殊类型	39
3.3 序列的其他共有属性	40
3.3.1 引用最后一个元素	40
3.3.2 序列的范围	41
3.3.3 通过附加序列增长列表	41
3.3.4 使用列表临时存储数据	42
3.3.5 处理集合	43
3.4 本章小结	44
3.5 习题	44

第 II 部分 Python 语言和标准库	
<b>第 4 章 做出决策</b> .....	49
4.1 比较两个值是否相等 .....	49
4.2 比较两个值是否不相等 .....	51
4.3 比较两个值的大小 .....	51
4.4 对真值和假值取反 .....	53
4.5 观察多个比较运算的结果 .....	54
4.6 循环 .....	57
4.6.1 重复执行操作 .....	57
4.6.2 终止循环 .....	59
4.7 处理错误 .....	62
4.8 本章小结 .....	64
4.9 习题 .....	65
<b>第 5 章 函数</b> .....	67
5.1 将程序放在单独的文件中 .....	67
5.2 函数：在一个名称下聚集 代码 .....	69
5.2.1 选择名称 .....	70
5.2.2 在函数中描述函数 .....	70
5.2.3 不同的位置相同的名称 .....	72
5.2.4 添加注释 .....	73
5.2.5 要求函数使用提供的值 .....	73
5.2.6 检查参数 .....	75
5.2.7 为参数设置默认值 .....	77
5.2.8 在函数中调用其他函数 .....	78
5.2.9 函数嵌套函数 .....	80
5.2.10 用自己的词语标记错误 .....	81
5.3 函数的层次 .....	81
5.4 本章小结 .....	82
5.5 习题 .....	83
<b>第 6 章 类与对象</b> .....	85
6.1 考虑编程 .....	85
6.1.1 对象的含义 .....	85
6.1.2 已经了解的对象 .....	85
6.1.3 展望：如何使用对象 .....	87
6.2 定义类 .....	87
6.2.1 如何创建对象 .....	87
6.2.2 对象和它们的作用域 .....	95
6.3 本章小结 .....	98
6.4 习题 .....	99
<b>第 7 章 组织程序</b> .....	101
7.1 模块 .....	102
7.1.1 导入可用模块 .....	102
7.1.2 通过已有模块创建新模块 .....	102
7.1.3 从命令行开始使用模块 .....	104
7.1.4 改变导入方式 .....	106
7.2 包 .....	106
7.3 模块和包 .....	108
7.3.1 将所有内容引入当前 作用域 .....	108
7.3.2 重新导入模块和包 .....	109
7.4 测试模块和包的基础知识 .....	111
7.5 本章小结 .....	112
7.6 习题 .....	112
<b>第 8 章 文件和目录</b> .....	115
8.1 文件对象 .....	115
8.1.1 编写文本文件 .....	116
8.1.2 向文件中追加文本 .....	116
8.1.3 读文本文件 .....	117
8.1.4 文件异常 .....	119
8.2 路径和目录 .....	119
8.3 os 中的异常 .....	119
8.3.1 路径 .....	120
8.3.2 目录内容 .....	122
8.3.3 获取文件信息 .....	123
8.3.4 重命名、移动、复制和删除 文件 .....	125
8.3.5 示例：轮换文件 .....	126
8.3.6 创建和删除目录 .....	127
8.3.7 通配 .....	128

8.4	本章小结	129	10.5	创建一个完整的模块	166
8.5	习题	129	10.6	安装模块	171
<b>第 9 章</b>	<b>Python 语言的其他特性</b>	<b>131</b>	10.7	本章小结	174
9.1	lambda 和 filter: 简单匿名函数	131	10.8	习题	175
9.2	Map: 短路循环	132	<b>第 11 章</b>	<b>文本处理</b>	<b>177</b>
9.3	在列表中做出决策——列表解析	133	11.1	文本处理的用途	177
9.4	为循环生成迭代器	133	11.1.1	搜索文件	178
9.5	使用字典的特殊字符串替换	135	11.1.2	日志剪辑	179
9.6	重要模块	137	11.1.3	邮件筛选	179
9.6.1	getopt——从命令行中得到选项	137	11.2	使用 os 模块导航文件系统	179
9.6.2	使用一个以上的进程	139	11.3	使用正则表达式和 re 模块	185
9.6.3	线程——在相同的进程中完成多个工作	141	11.4	本章小结	189
9.7	本章小结	143	11.5	习题	189
9.8	习题	144	<b>第 III 部分 开始使用 Python</b>		
<b>第 10 章</b>	<b>创建模块</b>	<b>145</b>	<b>第 12 章</b>	<b>测试</b>	<b>193</b>
10.1	研究模块	145	12.1	断言	193
10.1.1	导入模块	147	12.2	测试用例和测试套件	195
10.1.2	查找模块	147	12.3	测试装置	198
10.1.3	理解模块	148	12.4	用极限编程整合	201
10.2	创建模块和包	149	12.4.1	用 Python 实现搜索实用程序	202
10.3	使用类	150	12.4.2	一个更加强大的 Python 搜索	207
10.3.1	定义面向对象编程	151	12.5	软件生命周期中的正规测试	210
10.3.2	创建类	151	12.6	本章小结	210
10.3.3	扩展已有的类	152	<b>第 13 章</b>	<b>使用 Python 编写 GUI</b>	<b>213</b>
10.4	完成模块	153	13.1	Python 的 GUI 编程工具箱	213
10.4.1	定义模块特定的错误	154	13.2	Tkinter 简介	215
10.4.2	选择导出哪些内容	154	13.3	用 Tkinter 创建 GUI 小组件	215
10.4.3	为模块建立文档	155	13.3.1	改变小组件的尺寸	215
10.4.4	测试模块	164			
10.4.5	将模块作为程序运行	165			

13.3.2	配置小组件选项	216	15.2	模式/DTD	251
13.3.3	使用小组件	217	15.2.1	文档模型的用途	251
13.3.4	创建布局	217	15.2.2	是否需要文档模型	252
13.3.5	填充顺序	218	15.3	文档类型定义	252
13.3.6	控制小组件的外观	219	15.3.1	DTD 示例	252
13.3.7	单选按钮和复选框	220	15.3.2	DTD 不完全是 XML	253
13.3.8	对话框	221	15.3.3	DTD 的局限性	253
13.3.9	其他小组件类型	222	15.4	模式	254
13.4	本章小结	222	15.4.1	示例模式	254
13.5	习题	223	15.4.2	模式是纯粹的 XML	254
<b>第 14 章</b>	<b>访问数据库</b>	<b>225</b>	15.4.3	模式具有层次	255
14.1	使用 dbm 持久字典	226	15.4.4	模式的其他优点	255
14.1.1	选择 dbm 模块	226	15.5	XPath	255
14.1.2	创建持久字典	227	15.6	HTML 是 XML 的子集	256
14.1.3	访问持久字典	228	15.6.1	HTML DTD	256
14.1.4	dbm 与关系数据库的适用 场合	230	15.6.2	HTMLParser	256
14.2	使用关系数据库	231	15.7	Python 中可用的 XML 库	257
14.2.1	编写 SQL 语句	232	15.8	SAX 的含义	257
14.2.2	定义表	234	15.8.1	基于流	258
14.2.3	建立数据库	235	15.8.2	事件驱动	258
14.3	使用 Python 的数据库 API	237	15.8.3	DOM 的含义	258
14.3.1	下载各个模块	237	15.8.4	内存中访问	258
14.3.2	创建连接	238	15.9	使用 SAX 或者 DOM 的 理由	259
14.3.3	使用游标	238	15.9.1	能力权衡	259
14.3.4	使用事务并提交结果	245	15.9.2	内存考虑	259
14.3.5	检查模块的功能和元 数据	246	15.9.3	速度考虑	259
14.3.6	处理错误	246	15.10	Python 中可用的 SAX 和 DOM 解析器	259
14.4	本章小结	247	15.10.1	xml.sax	259
14.5	习题	248	15.10.2	xml.dom.minidom	260
<b>第 15 章</b>	<b>使用 Python 处理 XML</b>	<b>249</b>	15.11	XSLT 简介	262
15.1	XML 的含义	249	15.11.1	XSLT 是 XML	262
15.1.1	层次标记语言	249	15.11.2	转换和格式语言	263
15.1.2	一组标准	251	15.11.3	函数式、模板驱动	263
			15.12	lxml 简介	263

15.13	元素类	263	16.4.11	基于 select 的单线程 多任务	312
15.14	使用 lxml 解析	266	16.5	其他主题	313
15.15	本章小结	267	16.5.1	协议设计的多种考虑	313
15.16	习题	267	16.5.2	对等架构	314
<b>第 16 章</b>	<b>网络编程</b>	<b>269</b>	16.6	本章小结	314
16.1	理解协议	271	16.7	习题	315
16.1.1	比较协议和程序语言	271	<b>第 17 章</b>	<b>用 C 扩展编程</b>	<b>317</b>
16.1.2	Internet 协议栈	272	17.1	扩展模块概述	318
16.1.3	Internet 协议简介	273	17.2	构建并安装扩展模块	320
16.2	发送电子邮件	274	17.3	从 Python 向 C 传递参数	322
16.2.1	E-mail 文件格式	275	17.4	从 C 向 Python 返回值	325
16.2.2	MIME 消息	276	17.5	LAME 项目	326
16.2.3	使用 SMTP 和 smtplib 发送 邮件	284	17.6	LAME 扩展模块	330
16.3	检索 Internet 邮件	286	17.7	在 C 代码中使用 Python 对象	342
16.3.1	使用 mailbox 解析本地 邮筒	286	17.8	本章小结	345
16.3.2	使用 poplib 从 POP3 服务器获取邮件	288	17.9	习题	346
16.3.3	使用 imaplib 从 IMAP 服务器获取邮件	290	<b>第 18 章</b>	<b>数值编程</b>	<b>347</b>
16.3.4	安全的 POP3 和 IMAP	294	18.1	Python 语言中的数值	347
16.3.5	Webmail 应用程序不是 E-mail 应用程序	294	18.1.1	整数	348
16.4	套接字编程	294	18.1.2	长整数	348
16.4.1	套接字简介	295	18.1.3	浮点数	349
16.4.2	绑定到外部主机名	297	18.1.4	格式化数值	350
16.4.3	镜像服务器	298	18.1.5	作为数值的字符	352
16.4.4	镜像客户端	299	18.2	数学	353
16.4.5	套接字服务器	300	18.2.1	算术运算	354
16.4.6	多线程服务器	302	18.2.2	内置数学函数	355
16.4.7	Python 聊天服务器	303	18.3	复数	357
16.4.8	设计 Python 聊天 服务器	303	18.4	数组	359
16.4.9	Python 聊天服务器协议	304	18.5	本章小结	363
16.4.10	Python 聊天客户端	309	18.6	习题	363
			<b>第 19 章</b>	<b>Django 简介</b>	<b>365</b>
			19.1	框架的定义以及使用框架的 理由	365



19.2	Web 框架的其他功能	366	20.8	REST Web 服务	419
19.3	Django 发展史	367	20.8.1	REST 快速入门： 在 Amazon.com 上寻找 廉价商品	419
19.4	理解 Django 的架构	368	20.8.2	WishListBargainFinder 简介	422
19.4.1	项目初始设置	368	20.8.3	向 BittyWiki 提供 REST API	425
19.4.2	创建视图	371	20.8.4	使用 REST Web 服务 实现搜索和替换功能	428
19.5	使用模板	373	20.9	XML-RPC	432
19.6	使用模板和视图	375	20.9.1	XML-RPC 请求	434
19.6.1	模型	378	20.9.2	XML-RPC 中的数据 表示	434
19.6.2	创建模型的第一步—— 配置数据库设置	379	20.9.3	XML-RPC 响应	435
19.7	创建模型：创建一个应用 程序	380	20.9.4	错误处理机制	436
19.8	本章小结	382	20.9.5	通过 XML-RPC 展示 BittyWiki API	437
19.9	习题	383	20.9.6	基于 XML-RPC Web 服务 的 wiki 搜索和替换	440
<b>第 20 章</b>	<b>Web 应用程序与 Web 服务</b>	<b>385</b>	20.10	SOAP	442
20.1	REST: Web 架构	386	20.10.1	SOAP 快速入门	442
20.1.1	REST 的特性	386	20.10.2	SOAP 请求	443
20.1.2	REST 操作	388	20.10.3	SOAP 响应	444
20.2	HTTP: REST 的实际应用	388	20.10.4	错误处理机制	444
20.2.1	可见的 Web 服务器	390	20.10.5	展示一个 BittyWiki 的 SOAP 接口	445
20.2.2	HTTP 请求	393	20.10.6	基于 SOAP Web 服务的 wiki 搜索和替换	447
20.2.3	HTTP 响应	393	20.11	为 Web 服务 API 创建 文档	449
20.3	CGI: 将脚本变为 Web 应用 程序	395	20.11.1	人类可读的 API 文档	449
20.3.1	Web 服务器与 CGI 脚本的 协议	397	20.11.2	XML-RPC 自省 API	450
20.3.2	CGI 的特殊环境变量	397	20.11.3	WSDL	451
20.3.3	通过 HTML 表单接收用户 输入	399	20.12	选择 Web 服务标准	455
20.4	HTML 表单的有限词汇	400	20.13	Web 服务礼仪	456
20.5	访问表单值的安全性	401			
20.6	构建 wiki	405			
20.6.1	BittyWiki 核心库	406			
20.6.2	BittyWiki 的 Web 接口	409			
20.7	Web 服务	418			