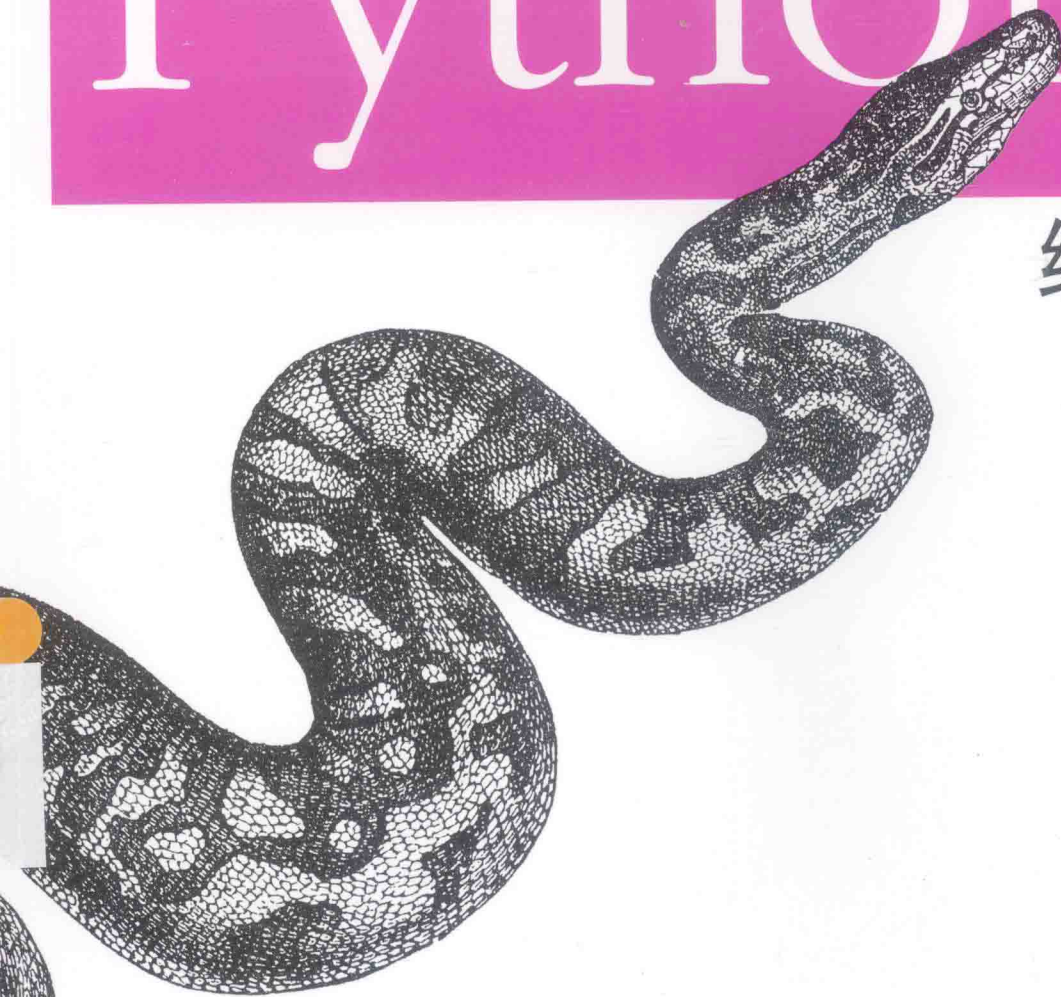


Programming Python

第4版(下册)  
涵盖 Python 3.x

# Python

编程



O'REILLY®

中国电力出版社

Mark Lutz 著

邹晓 瞿乔 任发科 等译

第四版

# Python编程

Mark Lutz 著  
邹晓 瞿乔 任发科 等译

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

O'Reilly Media, Inc. 授权中国电力出版社出版

中国电力出版社

## 图书在版编目 (CIP) 数据

Python编程: 第4版/ (美) 卢茨 (Lutz, M.) 著; 邹晓等译. —北京: 中国电力出版社, 2015.1

书名原文: Programming Python, Fourth Edition

ISBN 978-7-5123-5530-9

I. ①P… II. ①卢… ②邹… 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 (2014) 第024404号

北京市版权局著作权合同登记

图字: 01-2012-7973号

©2011 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and China Electric Power Press, 2014.  
Authorized translation of the English edition, 2011 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由O'Reilly Media, Inc. 出版2011。

简体中文版由中国电力出版社出版2014。英文原版的翻译得到O'Reilly Media, Inc.的授权。此简体中文版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc.的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式重制。

封面设计/ Karen Montgomery, 张健

出版发行/ 中国电力出版社 (www.cepp.sgcc.com.cn)

地 址/ 北京市东城区北京站西街19号 (邮政编码100005)

经 销/ 全国新华书店

印 刷/ 北京丰源印刷厂

开 本/ 787毫米×980毫米 16开本 95印张 1796千字

版 次/ 2015年1月第一版 2015年1月第一次印刷

印 数/ 0001-3000册

定 价/ 198.00元 (上下册)

## 敬告读者

本书封底贴有防伪标签, 刮开涂层可查询真伪  
本书如有印装质量问题, 我社发行部负责退换

版权专有 翻印必究

# 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

## 作者简介

---

Mark Lutz是Python培训的领军人物，最早且最畅销的Python书籍的作者和Python社区的先驱之一。

Mark是广受欢迎的O'Reilly书籍《Learning Python》、《Programming Python》和《Python Pocket Reference》的作者，三本书现在均更新到第四版。他从1992年开始一直在使用和宣传Python，从1995年开始撰写Python书籍，从1997年开始教授Python课程。到2010年初，他已经教授了约250次Python培训课程，超过3500名学生，所撰写的Python书籍已经售出约25万本，并且翻译成数十种语言。

此外，他拥有威斯康辛大学的计算机科学学士与硕士学位，并且在近25年来作为一名职业开发者，在编译器、编程工具、应用程序辨析和多种客户端/服务器系统等领域工作。Mark维护着一个书籍支持网站<http://rmi.net/~lutz>和培训网站<http://learning-python.com>。

## 封面介绍

---

本书封面上的动物是非洲岩蟒，蟒类下属的18个物种之一。蟒蛇是无毒的蛇类，栖息非洲、亚洲、大洋洲和某些太平洋岛屿的热带地区。蟒蛇主要生活在陆地上，不过它们也是出色的泳者和攀岩者。雄性和雌性蟒蛇都保留着已退化的后腿的痕迹。雄性蟒蛇在求偶时用到这些痕迹，或者说小刺。

蟒类通过窒息杀死它的猎物。蛇类通过尖利的牙齿咬住猎物，以便固定它们，而蟒类用长长的身体缠绕它的猎物的胸部，在它的受害者每次呼气时扼紧一分。蟒类主要捕食哺乳类和鸟类。蟒蛇攻击人类的事情绝少发生。

# 目录

## 上册

|          |   |
|----------|---|
| 前言 ..... | 1 |
|----------|---|

## 第一部分 序幕

|                       |    |
|-----------------------|----|
| 第1章 先睹为快 .....        | 19 |
| “Python编程：长话短说” ..... | 19 |
| 任务 .....              | 20 |
| 第1步：表示记录 .....        | 20 |
| 第2步：持久存储记录 .....      | 30 |
| 第3步：走进OOP .....       | 41 |
| 第4步：增加控制台交互 .....     | 51 |
| 第5步：增加GUI .....       | 54 |
| 第6步：增加Web界面 .....     | 65 |
| 演示总结 .....            | 81 |

## 第二部分 系统编程

|                    |    |
|--------------------|----|
| 第2章 系统工具 .....     | 85 |
| “os.path”的知识 ..... | 85 |
| 系统编程概述 .....       | 87 |

|                         |            |
|-------------------------|------------|
| 介绍sys模块.....            | 98         |
| 介绍os模块.....             | 101        |
| <b>第3章 脚本运行上下文.....</b> | <b>115</b> |
| 我要增加一个变量.....           | 115        |
| 当前工作路径.....             | 115        |
| 命令行参数.....              | 118        |
| shell环境变量.....          | 121        |
| 标准流.....                | 124        |
| <b>第4章 文件和目录工具.....</b> | <b>143</b> |
| “五个简单步骤擦除你的硬盘!”.....    | 143        |
| 文件工具.....               | 143        |
| 目录工具.....               | 171        |
| <b>第5章 并行系统工具.....</b>  | <b>184</b> |
| “指挥猴子做事”.....           | 184        |
| 进程分支.....               | 185        |
| 线程.....                 | 192        |
| 程序退出.....               | 218        |
| 进程间通信.....              | 227        |
| multiprocessing模块.....  | 247        |
| 启动程序的其他方法.....          | 261        |
| 一个跨平台的程序启动框架.....       | 267        |
| 其他系统工具.....             | 271        |
| <b>第6章 完整的系统程序.....</b> | <b>273</b> |
| “愤怒的Grep命令”.....        | 273        |
| 小游戏：找到最大的Python文件.....  | 274        |
| 分割与合并文件.....            | 284        |
| 生成重定向网页.....            | 294        |
| 一个回归测试脚本.....           | 298        |
| 复制目录树.....              | 306        |

|                    |     |
|--------------------|-----|
| 比较目录树 .....        | 309 |
| 搜索目录树 .....        | 320 |
| 访问器：目录遍历“++” ..... | 330 |
| 播放媒体文件 .....       | 343 |

## 第三部分 GUI编程

### 第7章 图形化用户界面 ..... 355

|                                |     |
|--------------------------------|-----|
| “嗨，我们正瞧着你！” .....              | 355 |
| Python GUI 开发之选 .....          | 357 |
| tkinter概览 .....                | 362 |
| 循序渐进的GUI学习曲线 .....             | 366 |
| tkinter编码的替代之选 .....           | 370 |
| 添加按钮和回调函数 .....                | 376 |
| 添加用户定义的回调函数 .....              | 379 |
| 添加多个组件 .....                   | 391 |
| 用类实现组件的自定义设置 .....             | 397 |
| 用类复用GUI部件 .....                | 400 |
| 本章小结 .....                     | 406 |
| Python/tkinter对Tcl/Tk的转换 ..... | 408 |

### 第8章 tkinter之旅：第一部分 ..... 410

|                                     |     |
|-------------------------------------|-----|
| “组件、工具和GUI，太棒了！” .....              | 410 |
| 配置组件外观 .....                        | 411 |
| 顶层窗口 .....                          | 414 |
| 对话框 .....                           | 420 |
| 绑定事件 .....                          | 436 |
| Message和Entry组件 .....               | 442 |
| Checkbutton、Radiobutton和Scale ..... | 450 |
| 三种运行GUI代码的方法 .....                  | 463 |
| 图像 .....                            | 475 |
| 使用PIL查看和处理图像 .....                  | 481 |

|                           |            |
|---------------------------|------------|
| <b>第9章 tkinter之旅：第二部分</b> | <b>496</b> |
| “如今的菜单，到处都是垃圾内容”          | 496        |
| 菜单                        | 496        |
| Listbox和Scrollbar         | 510        |
| Text                      | 516        |
| Canvas                    | 537        |
| 网格                        | 550        |
| 时间工具、线程和动画                | 567        |
| 结束组件之旅                    | 580        |
| <b>第10章 GUI编码技巧</b>       | <b>581</b> |
| “搭建一个更好的用户界面”             | 581        |
| GuiMixin：普通工具的混合类         | 582        |
| GuiMaker：自动化生成菜单和工具栏      | 586        |
| ShellGui：命令行工具的GUI        | 596        |
| GuiStreams：重定向流到组件        | 606        |
| 动态重载回调处理器                 | 610        |
| 封装顶层窗口接口                  | 612        |
| GUI、线程和队列                 | 616        |
| 添加GUI到非GUI代码的其他方法         | 627        |
| PyDemos和PyGadgets启动器      | 641        |
| <b>第11章 完整的GUI程序</b>      | <b>651</b> |
| “Python、开源和Camaro”        | 651        |
| PyEdit：一个文本编辑器程序/对象       | 654        |
| PyPhoto：一个图像查看器和调整大小的工具   | 694        |
| PyView：一个图像和说明的幻灯片显示工具    | 704        |
| PyDraw：绘制和移动图像            | 714        |
| PyClock：一个模拟/数字时钟组件       | 723        |
| PyToe：一个三连棋游戏组件           | 737        |
| 小结                        | 742        |

## 下册

# 第四部分 互联网编程

|                                |            |
|--------------------------------|------------|
| <b>第12章 网络脚本 .....</b>         | <b>745</b> |
| “接收、登录和退出” .....               | 745        |
| Python互联网发展方案 .....            | 750        |
| 探索互联网 .....                    | 754        |
| 套接字编程 .....                    | 760        |
| 处理多个客户端 .....                  | 775        |
| 使套接字看起来像文件和数据流 .....           | 798        |
| 一个简单的Python文件服务器 .....         | 811        |
| <b>第13章 客户端编程 .....</b>        | <b>823</b> |
| “与我连接！” .....                  | 823        |
| FTP：通过网络传输文件 .....             | 824        |
| 用ftplib传输文件 .....              | 824        |
| 用ftplib传输目录 .....              | 843        |
| 用ftplib传输目录树 .....             | 861        |
| 处理Internet电子邮件 .....           | 868        |
| POP：抓取电子邮件 .....               | 870        |
| SMTP：发送电子邮件 .....              | 878        |
| email：解析和撰写邮件内容 .....          | 889        |
| 一个基于控制台的电子邮件客户端 .....          | 914        |
| mailtools工具包 .....             | 923        |
| NNTP：访问新闻组 .....               | 957        |
| HTTP：访问网站 .....                | 959        |
| 再次探讨urllib包 .....              | 962        |
| 客户端编程的其他选择 .....               | 967        |
| <b>第14章 PyMailGUI客户端 .....</b> | <b>969</b> |
| “卢克，快用源码！” .....               | 969        |

|                                |             |
|--------------------------------|-------------|
| PyMailGUI的主要改动 .....           | 975         |
| 演示PyMailGUI .....              | 984         |
| PyMailGUI的实现 .....             | 1022        |
| 改进建议 .....                     | 1073        |
| <b>第15章 服务器端编程 .....</b>       | <b>1082</b> |
| “哦，我们织了怎样一张复杂的网啊” .....        | 1082        |
| 什么是服务器端CGI脚本？ .....            | 1083        |
| 运行服务器端示例 .....                 | 1086        |
| 沿着CGI学习曲线继续攀登 .....            | 1091        |
| 在CGI脚本里保存状态信息 .....            | 1128        |
| Hello World选择器 .....           | 1136        |
| 重构代码，使其更容易维护 .....             | 1145        |
| 关于HTML和URL转义的更多讨论 .....        | 1154        |
| 向客户端和服务器传输文件 .....             | 1161        |
| <b>第16章 PyMailCGI服务器 .....</b> | <b>1179</b> |
| “停留芝加哥期间的任务” .....             | 1179        |
| PyMailCGI网站 .....              | 1180        |
| 根页面 .....                      | 1188        |
| 通过SMTP发送邮件 .....               | 1191        |
| 处理收到的邮件 .....                  | 1213        |
| 实用工具模块 .....                   | 1222        |
| Web 脚本编程要考虑的平衡 .....           | 1237        |
| <b>第五部分 工具和技巧 .....</b>        | <b>1247</b> |
| <b>第17章 数据库和持久化 .....</b>      | <b>1247</b> |
| “给我来一份持久化，但不要加pickle” .....    | 1247        |
| Python中可选的持久化方案 .....          | 1247        |
| DBM文件 .....                    | 1249        |
| Pickle对象 .....                 | 1253        |

|                                          |             |
|------------------------------------------|-------------|
| Shelve文件 .....                           | 1259        |
| 面向对象的数据库ZODB .....                       | 1268        |
| SQL数据库接口 .....                           | 1271        |
| ORM: 对象关系映射 .....                        | 1296        |
| PyForm: 一个持久化对象的查看器 (扩展阅读) .....         | 1297        |
| <b>第18章 数据结构 .....</b>                   | <b>1299</b> |
| “玫瑰是红色的, 紫罗兰是蓝色的; 列表是可变的, 而集合是个傻瓜” ..... | 1299        |
| 堆栈实现 .....                               | 1300        |
| 集合的实现 .....                              | 1312        |
| 实现内置类型的子类 .....                          | 1322        |
| 二叉搜索树 .....                              | 1323        |
| 检索图 .....                                | 1328        |
| 队列的排列组合 .....                            | 1333        |
| 反序和排序 .....                              | 1335        |
| PyTree: 一个通用的树对象浏览器 .....                | 1339        |
| <b>第19章 文本和语言 .....</b>                  | <b>1342</b> |
| “快看杰克做功课。功课, 杰克, 功课” .....               | 1342        |
| Python中文本处理的策略 .....                     | 1342        |
| 字符串方法工具 .....                            | 1343        |
| 正则表达式模式匹配 .....                          | 1352        |
| XML和HTML解析 .....                         | 1366        |
| 高级语言工具 .....                             | 1375        |
| 定制语言解析器 .....                            | 1377        |
| PyCalc: 一个计算器程序/对象 .....                 | 1393        |
| <b>第20章 Python/C集成 .....</b>             | <b>1417</b> |
| “迷失在C中” .....                            | 1417        |
| 在C中扩展Python: 概述 .....                    | 1420        |
| 一个简单的C扩展模块 .....                         | 1421        |
| SWIG集成代码生成器 .....                        | 1424        |
| 封装C环境调用 .....                            | 1429        |

|                      |      |
|----------------------|------|
| 用SWIG封装C++类 .....    | 1435 |
| 其他扩展工具 .....         | 1445 |
| 在C中嵌入Python：概述 ..... | 1447 |
| 基本嵌入技巧 .....         | 1450 |
| 注册回调处理器对象 .....      | 1462 |
| 在C中使用Python类 .....   | 1466 |
| 其他集成相关话题 .....       | 1469 |

## 第六部分 尾声

|                            |      |
|----------------------------|------|
| 第21章 结语：Python与开发周期 .....  | 1475 |
| “全书至此完结，以下是生活的意义” .....    | 1476 |
| “我们编程的方式有些问题” .....        | 1476 |
| “Gilligan因素” .....         | 1476 |
| 做正确的事 .....                | 1477 |
| 走进Python .....             | 1478 |
| 在沉没的泰坦尼克号上 .....           | 1483 |
| 那么“Python：续集”打算讲点什么？ ..... | 1486 |
| 最后的总结 .....                | 1486 |

# 互联网编程

这一部分探讨Python语言在编写基于互联网的应用程序中的作用，以及相关的库工具。在此过程中还将用到此前介绍的系统和GUI工具。因为这是一个较为流行的Python领域，所以将在以下各章里涵盖各个方面：

### 第12章

这一章介绍互联网相关概念和选项，套接字等Python底层网络工具，以及客户端和服务器的基础知识。

### 第13章

这一章将展示脚本如何借助Python访问常用的客户端网络协议，比如FTP、电子邮件和HTTP等。

### 第14章

这一章利用上一章介绍的客户端电子邮件工具与前一部分中的GUI技巧来实现一个功能完备的电子邮件客户端程序。

### 第15章

这一章将介绍Python的服务器端通用网关接口（Common Gateway Interface, CGI）脚本的基本知识，这类程序用于实现交互式网站。

### 第16章

这一章通过在服务器上实现基于web的电子邮件工具来演示Python的网站技巧，以便与第14章里的非Web策略进行比较。

虽然不在本书范围之内，但第12章里将给出对高级Python互联网工具的概述，建议在进阶资料中学习这些工具，比如Jython、Django、App Engine、Zope、PSP、pyjamas和HTMLgen。在本书中你将了解使用这些工具时所需的基本知识。



# 网络脚本

## “接收、登录和退出”

从本书的第一版发布至今已经15年了，如今互联网事实上已登上了主流舞台。现在，它已从一个简单的主要由高校和研究人员使用的通信设备迅速成为几乎和电视与电话一样普遍的媒介。社会观察者把互联网带来的文化影响等价于当初的印刷术。技术观察者认为，所有的新软件开发的兴趣仅限于互联网。当然，时间将成为这些说法的最后仲裁者，但毋庸置疑的是，互联网既是社会的一种主要势力，又是现代软件系统的主要应用环境之一。

同时，互联网也正好是Python编程语言的一个主要应用领域。在本书第一版写成后的十五年间，互联网发展对Python的工具集和作用产生了巨大的影响。今天，只要有Python和一台基于套接字互联网连接的计算机，我们只需简单地使用Python附带的作为标准工具的互联网模块，就可以写出Python脚本，用于在世界范围内读取和发送电子邮件，从远程站点获取网页、通过FTP传输文件、编写交互式网站、解析HTML和XML文件，以及更多应用。

事实上，全世界的公司都如此：谷歌、YouTube、迪斯尼、惠普、JPL，以及其他许多公司都支持Python标准工具建立他们的网站。例如，人们普遍相信，通过广泛使用Python代码，可以让Google搜索引擎在Web上运行。YouTube的视频服务器站点主要是用Python运行的。用Python写的BitTorrent点对点文件传输系统——利用Python的网络功能在客户端之间共享文件和消除一些服务器瓶颈，其下载量数以千万计。

有很多公司也是通过使用较大的基于Python的工具包，构建和管理他们的网站。例如，Zope网络应用服务器，就是一个早期的类似产品，其本身也是Python编写和定义的。其他使用基于Zope的Plone内容管理系统建立网站，提交站点内容给用户。还有一些用Python来编写使用Jython(以前称为Jpython)的Java网络应用脚本。该Jpython系统可以编译Python

程序,生成Java字节码,导出Java库以供Python脚本使用,并允许Python代码像网络小应用程序一样可以下载并可在浏览器上运行。

最近几年,新的技术和系统在Web领域层出不穷。例如,运用于Python的XML-RPC和SOAP接口,使得网络服务可编程化;出现了用于构建网站的功能强大的框架工具,如Google App Engine, Django和Turbo Gears; Python标准库中的XML包,以及第三方扩展包,提供了一套XML处理工具;运用于Python代码IronPython实施,提供了无缝.NET/Mono的一体化,其运行方法类似Jython使用Java库。

随着互联网的不断发展,作为互联网工具的Python角色也在不断变化。Python被证明可以很好地适用于互联网脚本,由于相同的原因,使得它在其他领域也很适用。它的模块化设计和快速周转可以同互联网发展的强烈需求很好地结合起来。在本书的这部分内容中,我们将会发现,Python不仅仅能简单地支持互联网脚本,还能提高质量,如生产率和可维护性——这对于所有不同类型和大小的互联网项目很有必要。

## 互联网脚本主题

互联网编程包含了许多主题,因此,为了让描述更容易理解,我已经将这个主题分散在本书接下来的五个章节中。下面是这部分章节的概述:

- 本章介绍了互联网基础知识,探讨套接字——基于互联网的底层通信机制。我们在第5章简单地了解了作为IPC工具的套接字,并在第10章的某个GUI用例中再次遇到它。但在这里,我们将深入地研究它们表现出来的广泛的网络角色。
- 第13章涵盖了客户端脚本和Internet协议的基础知识。这里,我们将探讨对于FTP、电子邮件、HTTP、NNTP及更多内容的Python标准支持。
- 第14章介绍了一个大型的客户端研究案例: PyMailGUI——一个功能全面的电子邮件客户端。
- 第15章讨论了服务器端脚本和网站建设的基本知识。我们将学习隐藏于大多数互联网活动后面的基本CGI脚本技术和概念。
- 第16章介绍了一个较大型的服务器端研究案例: PyMailCGI——一个功能全面的webmail站点。

每一章都会假定你已经阅读过前一章,但是一般可以跳过,尤其当你有互联网领域的经验时。由于这些章节在很大程度上包含了本书的相当大一部分内容,所以下面的部分会更多涉及我们将要学习的内容的详细信息。

## 我们将会涉及的内容

从概念上讲,互联网可以大致认为是由多个功能层组成的: