

SAMS
**Teach
Yourself**

- 全球销量逾百万册的系列图书
- 连续十余年打造的经典品牌
- 直观、循序渐进的学习教程
- 掌握关键知识的最佳起点
- “Read Less, Do More”（精读多练）的教学理念
- 以示例引导读者完成最常见的任务

每章内容针对初学者精心设计，**1**小时轻松阅读学习，
24小时彻底掌握关键知识

每章**案例与练习题**助你轻松完成常见任务，
通过**实践**提高应用技能，巩固所学知识

Python

入门经典

[美] Katie Cunningham 著
李军 李强 译

Python

入门经典

[美] Katie Cunningham 著
李军 李强 译

人民邮电出版社
北京

图书在版编目 (C I P) 数据

Python入门经典 / (美) 坎宁安 (Cunningham, K.)
著; 李军, 李强译. — 北京: 人民邮电出版社,
2014. 10
ISBN 978-7-115-36209-4

I. ①P… II. ①坎… ②李… ③李… 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151872号

版 权 声 明

Katie Cunningham: Sams Teach Yourself Python in 24 Hours

ISBN: 978-0-672-33687-4

Copyright © 2014 by Sams Publishing.

Authorized translation from the English languages edition published by Sams.

All rights reserved.

本书中文简体字版由美国 Sams 出版公司授权人民邮电出版社出版。未经出版者书面许可, 对本书任何部
分不得以任何方式复制或抄袭。

版权所有, 侵权必究。



-
- ◆ 著 [美] Katie Cunningham
 - 译 李军 李强
 - 责任编辑 陈冀康
 - 责任印制 彭志环 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>
三河市中晟雅豪印务有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 15.75
字数: 381 千字 2014 年 10 月第 1 版
印数: 1-3 500 册 2014 年 10 月河北第 1 次印刷
- 著作权合同登记号 图字: 01-2013-8316 号
-

定价: 39.00 元

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

内容提要

Python 是一种解释型、面向对象、动态数据类型的高级程序设计语言。Python 可以用于很多的领域，从科学计算到游戏开发。

本书是面向 Python 初学者的学习指南，详细介绍了 Python 编程基础，以及一些高级概念，如面向对象编程。全书分为 24 章。第 1 章介绍了 Python 的背景和安装方法。第 2 章到第 7 章介绍了一些基本的编程概念，如变量、数学运算、字符串和获取输入。第 8 章到第 12 章介绍了更高级的主题，讨论了函数、字典和面向对象编程等。第 13 章到第 15 章介绍了如何使用库和模块，以及如何创建自己的模块。第 16 章到第 19 章介绍了使用数据，如保存到文件，使用标准格式以及使用数据库。第 20 章和第 21 章尝试了标准库以外的一些项目，在这两章中，介绍了创建动态 Web 站点和开发游戏。这两章并不是要成为完整的课程，而是充当学习更多知识的一个起点。第 22 章和第 23 章介绍了如何正确地保存代码，以及出错的时候如何找到解决方案。第 24 章介绍了读者可能会接触到哪些项目，哪些资源可以帮助读者学习更多知识，以及如何更深入地融入 Python 社区。

本书内容循序渐进，逐步深入，概念和知识点讲解清楚，而且每章最后配有练习，供读者检查和巩固所学知识。本书适合学习 Python 语言和应用的初中级用户阅读和自学，也可以作为相关培训的教材。

序 言

为什么要学习 Python

我经常遇到这个问题，为什么要学习 Python 编程呢？为什么不是学习针对初学者的一种语言，比如 Scratch 呢？为什么不学习 Java 或 C++，大多数同事好像都在用这两种语言。

我个人认为 Python 对于初学者来说是一种理想的语言。可以从多个角度证明这一点。Python 的语法并不麻烦。它很容易阅读，很多人可以读懂一个简单的脚本并理解它在做什么，甚至不需要编写一行代码。

对于 Python 初学者来说，很容易转移到更为高级的项目，这也是很理想的一点。Python 可以用于很多的领域，从科学计算到游戏开发均可应用。程序设计新手总是会找到自己喜欢的一个项目，甚至好几个。

本书的目标读者

本书适合那些之前没有编写过程序的人，以及那些编写过一些程序但是想要学习 Python 的人。本书不适合那些已经有经验的程序员。

本书的读者需要有一台计算机并且拥有管理员权限。需要安装 Python 和多个库，以及本书后面的应用程序。读者的计算机不需要有特别强大的功能。

读者计算机应该能够连接到互联网，以便访问一些资源。

本书的组织结构

本书介绍了 Python 编程基础，以及一些高级概念（如面向对象编程）。

前言和第 1 章介绍了 Python 的背景和安装方法。

第 2 章到第 7 章介绍了一些基本的编程概念，如变量、数学运算、字符串和获取输入。

第 8 章到第 12 章介绍了更高级的主题，将会讨论函数、字典和面向对象编程等。

第 13 章到第 15 章介绍了如何使用库和模块，以及如何创建自己的模块。

第 16 章到第 19 章介绍了使用数据，如保存文件、使用标准格式以及使用数据库。

第 20 章和第 21 章尝试了标准库以外的一些项目。在这两章中，将介绍创建动态 Web 站点和开发游戏。这两章并不是要成为完整的课程，而是充当学习更多知识的一个起点。

第 22 章和第 23 章介绍了如何正确地保存代码，以及当出错的时候如何找到解决方案。

第 24 章介绍了读者可能会接触到哪些项目，哪些资源可以帮助读者学习更多知识，以及如何更深入地融入 Python 社区。

读者反馈

作为本书的读者，您是我们最重要的批评者和评论家。我们很重视您的意见，并且想要知道我们在哪些地方做得对，哪些地方可以做得更好，您希望看到我们出版哪方面的内容，以及您想要传递给我们的任何其他的真知灼见。

我们欢迎您的评论。您可以通过电子邮件或写信给我们，让我们知道您喜欢或不喜欢本书，以及我们可以做些什么让图书变得更好。

请注意，我们不能帮助您解决和本书主题相关的技术问题。

当您写信的时候，请确保附带本书的名称、作者以及您的名称和电子邮件地址。我们将仔细阅读您的评论，并与本书的作者和编辑分享。联系方式如下：

Email: consumer@sampublishing.com

Mail: Sams Publishing

ATTN: Reader Feedback

800 East 96th Street

Indianapolis, IN 46240 USA

读者服务

请访问我们的站点，并且在 informit.com/register 注册这本图书，以便更方便地访问和本书相关的任何更新、下载或勘误。

作者简介

Katie Cunningham 是 Cox Media Group 的一位 Python 程序员。她是 Python 和开源软件的积极倡导者，并且热心于教人们编程。她经常在 PyCon 和 DjangoCon 等开源会议担任演讲者，就初学者关心的一些主题进行演讲，如某人开发后放在云中的第一个站点，以及让所有人都能够访问的站点等主题。她还帮助在哥伦比亚特区组织 PyLadies，这是为了增加 Python 社区的多样性而设计的一个活动。她还为企业教授课程，带领新手在 48 个学时内从安装软件到编写出第一个 App。

Katie 是一位活跃的博客作者，她的个人站点是 (<http://therealkatie.net>)，包括诸如 Python、可访问性，以及在家工作的尝试和困难等话题。

Katie 和她的丈夫和两个孩子一起居住在哥伦比亚特区。

献辞

这本书献给我的家人，每次我决心写书的时候，他们令我能够保持理智地做这件事。

Jim，感谢你收拾残局！妈妈，感谢您每次在我快要崩溃的时候，接走了孩子们并提供帮助！

孩子们，感谢你们从不挑剔各种订餐食物！

致谢

如果没有一些人的帮助，这本书根本不可能写出来。首先，感谢我的编辑 Debra Williams Cauley，她既有耐心又有热情。没有她，我不知道是否还能按时完稿。

特别感谢我的技术编辑，Doug Hellmann 和 Gabriel Nilsson。当查找我的明显错误的时候，他们就像机器一样精准。并且，他们的建议使得本书更加完善。还要感谢 Richard Jones，他花时间审阅了 PyGame 一章。

感谢 Michael Thurston，他使我变得更加全面。我保证这几天一定学会正确地拼写“installation”。

最后，要感谢 Python 社区，每次我遇到问题的时候、需要正确性检查的时候或者仅仅只是需要激励的时候，它们都能伸出援手。社区就像是我的家。

前言

很多人懒得考虑学习如何编写代码。编程似乎是有用的技术，但是，很多人害怕触及并尝试。可能他们认为，学习像编程这样的技能有点太晚了，或者他们认为没有足够的时间来学习。可能他们很快就迷失了，因为他们所找到的图书都是针对那些之前为有编程经验的人而编写的。本书的目标是将程序设计背后的概念分解成易于理解并可立即使用的知识。

学习编程

对于很多人来说，学习编程似乎是一项不可能的任务，它被描绘成需要疯狂的数学运算、成年累月的教育和训练的一个领域，并且一旦进入了这个领域，需要无休止地持续地敲击键盘。

事实上，尽管成为一名全职的程序员可能要花一些精力，但学习如何编写代码还是很容易的。随着我们在日常生活中越来越多地接触计算机，学习如何编写代码来控制计算机可以增强任何人的职业生涯，不管你所从事的职业看上去有多么的不需要技术。小学教师也可能制作一个 Web 站点来帮助学生学习单词。会计则能够进行自动化计算，而这些工作通常要手工完成。父母可以创建一个家庭存货系统，以帮助产生一个超市购物清单。几乎每一种专业和爱好都可以通过学习编程来得到增强和扩展。

简单来说，计算机是很笨拙的。没有人的程序输入，它们不知道应该做什么事情。代码是一组指令集合，不仅告诉计算机要做什么，而且告诉它如何去做。计算机上的一切，从最大的应用程序（如 Word 和视频游戏）到最小的应用程序（如计算器），都是基于代码的。

计算机上的大多数的代码已经编译成一个 .exe 或 .app 文件。对于本书中的练习，我们可以通过一个文件来运行它，或者使用解释器来运行它（我们将在第 1 章中介绍）。

为什么使用 Python

Python 是以可读性而著称的一种语言，它不是很敏感，并且很容易教授。此外，和某些专门为了教学而创建的语言不同，Python 广泛地用于课堂之外的很多地方。人们已经使用

Python 编写了很多的应用，从 Web 站点到科学工作的工具，从简单的脚本到视频游戏。下面是用 Python 编写的程序的一个不完全的列表。

- YouTube——观看和分享视频的一个流行网站。
- The Onion——一个讽刺性的新闻网站。
- Eve Online——以太空为背景的一款视频游戏。
- The Washington Post——其 Web 站点基于 Django 而运行，Django 是用 Python 编写的一个框架。
- Paint Shop Pro——一款图像编辑软件包。
- Google——Google 的很多重复的应用程序都使用 Python。
- Civilization IV——一款回合制模拟游戏。

Python 看上去很简单，但它的功能真的非常强大。

起步

在开始之前，让我们先看看你所需要的东西的清单。你必须具备了所有这些东西，才能开始学习 Python。以下是需要的东西。

管理员访问权限——Python 并不要求在功能非常强大的计算机上运行，但是，你需要在计算机上拥有权限能够安装内容。

互联网访问——我们要下载安装程序，并且随后要介绍 Web 业务。不需要有一个快速的网络连接，因为我们所要下载的很多项目都相当小。

计算机——不需要全新的计算机，但是你的计算机越快，代码运行也就越快。过去 5 年内所制造的计算机应该都很好用。

场地——一个专门的工作场所可以大大增强你学习新概念的能力。应该避免其他的干扰，比如电视。

不要分心——如果有家庭成员干扰你、手机铃声或者背景中的电视声，那么，你几乎不可能学习新东西。如果你无法避免人为的或环境噪声的话，一个很好的防干扰耳机是不错的装备。

对于大多数人来说，最后两项可能最难配备，但是，它们真的很重要。不仅在学习的过程中需要它们，在学完本书继续深入自己的项目的时候，也需要它们。编写代码是一个创造性的努力，要有时间和场地才能做好。

如何使用本书

本书中的每一章都需要用一小时左右来完成。这包括阅读正文和做练习的时间。在理想情况下，应该在阅读完一章后直接完成练习。因此，当您全神贯注而且可以使用计算机的时候，请尝试预留出做练习的时间。并不是每一章都需要访问互联网（那些需要访问互联网的章，会提前告知您）。在完成一章之后，您可能想要进入下一章，但是，可以尝试休息一会儿。您的大脑需要时间来整合新的信息，并且，在深入更多新的材料之前，您

也需要稍事休息。

如果遇到困难怎么办

有一件事情是编写代码的每一个人都急不了的，就是会遇到困难。有时候，一个新的概念似乎不好理解；有时候，无法消除一个错误；还有一些时候，碰到的所有东西都无法工作。

度过这样的日子的关键在于，不要沮丧。起身离开计算机，散步一会儿，来杯咖啡，和朋友讨论一下（不正常的代码以外的）其他的事情。借此机会放松一下。

当您和问题拉开一定的心理距离后，快速地做一次自我评估。您累了吗？一个疲劳的程序员是一个糟糕的开发者，不管他多么富有经验。有时候，一杯咖啡也能起作用，但是，大多数时候，你需要睡一会儿。

如果您不累，尝试重新阅读这一章。分析一下所强调的地方，或者是做点笔记。有些术语不熟悉？尝试在网络上搜索一下这些术语。代码还是不工作？有时候，您需要删除已有的文件（或者将其保存到另一个文件中）并且再次尝试。在本书后面，将介绍调试代码的一些好办法，但是请放心，每个程序员都会遇到一些不确定的代码。

目 录

第 1 章 安装和运行 Python	1	2.2.1 运算符	14
1.1 认识操作系统	1	2.2.2 运算的优先级	14
1.2 在 Windows 上安装 Python	2	2.2.3 数学运算的组合类型	15
1.2.1 安装 Python	3	2.2.4 除以 0	15
1.2.2 运行 Python	3	2.3 比较数字	16
1.2.3 安装文本编辑器	4	2.4 在现实世界中应用 Python 数学 运算	16
1.2.4 浏览文件系统	4	2.5 小结	17
1.3 在 Mac OS 上安装 Python	6	2.6 Q&A	17
1.3.1 安装 Python	6	2.7 作业	18
1.3.2 运行 Python	6	2.7.1 问题	18
1.3.3 安装文本编辑器	7	2.7.2 回答	18
1.3.4 浏览文件系统	7	2.7.3 练习	18
1.4 小结	9	第 3 章 编程中的逻辑	19
1.5 Q&A	9	3.1 使用基本的 if 语句	19
1.6 作业	9	3.2 创建代码块	20
1.6.1 问题	9	3.3 为 if 添加 else	21
1.6.2 回答	9	3.4 用 elif 测试更多内容	22
1.6.3 练习	10	3.5 True 和 False 变量	22
第 2 章 在 Python 中使用数字	11	3.6 使用 try/except 避免错误	23
2.1 用变量存储信息	11	3.7 在现实世界的问题中应用逻辑	24
2.1.1 变量的类型	11	3.8 小结	25
2.1.2 在变量中存储数字	12	3.9 Q&A	25
2.1.3 命名变量	13	3.10 作业	26
2.1.4 提示	13	3.10.1 问题	26
2.2 在 Python 中做数学运算	14	3.10.2 回答	26

3.10.3 练习	26	6.3 操作列表	48
第 4 章 在字符串中存储文本	27	6.4 在列表中使用数学运算	49
4.1 创建字符串	27	6.5 排序列表	50
4.2 打印字符串	28	6.6 比较列表	50
4.3 获取有关字符串的信息	28	6.7 在现实世界中使用列表	51
4.4 数学运算和比较运算	29	6.8 小结	52
4.4.1 连接字符串	30	6.9 Q&A	52
4.4.2 乘法	30	6.10 作业	52
4.4.3 比较字符串	30	6.10.1 问题	52
4.4.4 不适用于字符串的运算符	31	6.10.2 回答	52
4.5 格式化字符串	31	6.10.3 练习	53
4.5.1 用转义符控制间隔	31	第 7 章 使用循环重复代码	54
4.5.2 删除空格	32	7.1 重复一定次数	54
4.5.3 查找和替换文本	33	7.1.1 获取数字的范围	54
4.6 在现实世界中使用字符串	34	7.1.2 命名循环变量	55
4.7 小结	34	7.1.3 遍历列表	56
4.8 Q&A	35	7.1.4 跳到下一个列表项	56
4.9 作业	35	7.1.5 跳出循环	57
4.9.1 问题	35	7.2 只有为真时重复	58
4.9.2 回答	35	7.2.1 While 循环	58
4.9.3 练习	36	7.2.2 无限循环	58
第 5 章 处理输入和输出	37	7.3 在现实世界中使用循环	59
5.1 从命令行获取信息	37	7.4 小结	60
5.1.1 提示	39	7.5 Q&A	60
5.1.2 转换输入	39	7.6 作业	61
5.2 获取密码	40	7.6.1 测验	61
5.3 清理用户的输入	41	7.6.2 回答	61
5.4 格式化输出	41	7.6.3 练习	61
5.5 在现实世界中管理输入和输出	43	第 8 章 创建简单的脚本	63
5.6 小结	44	8.1 创建一个基本的函数	63
5.7 Q&A	44	8.2 传递值给函数	64
5.8 作业	44	8.2.1 设置默认值	65
5.8.1 问题	44	8.2.2 返回值	66
5.8.2 回答	45	8.3 函数中的变量的作用域	67
5.8.3 练习	45	8.3.1 在函数内创建变量	67
第 6 章 将项分组到列表中	46	8.3.2 参数和作用域	68
6.1 创建列表	46	8.4 在函数中组织函数	68
6.2 获取有关列表的信息	48	8.5 传递可变数目的参数	69
		8.6 在现实世界中使用函数	70

8.7	小结	71	11.5	小结	96
8.8	Q&A	71	11.6	Q&A	96
8.9	作业	72	11.7	作业	96
8.9.1	问题	72	11.7.1	问题	96
8.9.2	回答	73	11.7.2	回答	96
8.9.3	练习	73	11.7.3	练习	97
第 9 章	使用字典匹配键和值	74	第 12 章	扩展类以添加功能	98
9.1	创建字典	74	12.1	内建附加功能	98
9.2	获取有关字典的信息	76	12.1.1	相等性	99
9.3	比较字典	77	12.1.2	大于和小于	100
9.4	在现实世界中使用字典	77	12.1.3	操作打印	101
9.5	小结	79	12.2	类的继承	102
9.6	Q&A	79	12.2.1	把类保存到一个文件中	102
9.7	作业	79	12.2.2	子类化一个类	103
9.7.1	问题	79	12.2.3	使用类	104
9.7.2	回答	80	12.3	在现实世界中使用扩展类	105
9.7.3	练习	80	12.4	小结	107
第 10 章	生成对象	81	12.5	Q&A	107
10.1	面向对象编程	81	12.6	作业	107
10.1.1	已经使用的对象	82	12.6.1	问题	107
10.1.2	对象	82	12.6.2	回答	107
10.1.3	术语	83	12.6.3	练习	107
10.2	规划一个对象	84	第 13 章	使用 Python 的模块来添加功能	109
10.3	从对象生成对象	85	13.1	Python 包	109
10.4	在现实世界中使用对象	86	13.2	使用 random 模块	110
10.5	小结	87	13.2.1	randint	110
10.6	Q&A	87	13.2.2	random	111
10.7	作业	87	13.2.3	uniform	112
10.7.1	问题	87	13.2.4	choice	112
10.7.2	回答	87	13.3	使用 datetime 模块	112
10.7.3	练习	88	13.3.1	time	113
第 11 章	生成类	89	13.3.2	datetime	113
11.1	生成一个基类的声明	89	13.4	找到更多的模块	114
11.2	为类增加方法	90	13.5	在现实世界中使用模块	114
11.3	创建类的实例	91	13.6	小结	115
11.3.1	__init__() 函数	91	13.7	Q&A	115
11.3.2	移动和存储实例	93	13.8	作业	116
11.4	在现实世界中使用类	94			

13.8.1	问题	116	16.1.3	添加数据到文件	138
13.8.2	回答	116	16.2	创建文件	139
13.8.3	练习	116	16.3	获取目录的信息	139
第 14 章	分拆程序	118	16.3.1	文件列表	139
14.1	为什么要分拆程序	118	16.3.2	在目录间移动	140
14.2	决定如何分拆代码	119	16.3.3	创建目录	141
14.2.1	规划分拆一个程序	119	16.4	获取文件的信息	142
14.2.2	分拆一个音乐库程序	119	16.4.1	文件大小	142
14.3	Python 查找一个程序的代码	121	16.4.2	访问时间	142
14.3.1	通过文件目录包含 模块	121	16.5	在现实世界中 使用文件	143
14.3.2	导入模块	122	16.6	小结	144
14.4	在现实世界中分拆代码	123	16.7	Q&A	144
14.5	小结	125	16.8	作业	144
14.6	Q&A	125	16.8.1	问题	144
14.7	作业	125	16.8.2	回答	144
14.7.1	问题	125	16.8.3	练习	145
14.7.2	回答	125	第 17 章	用 JSON 来共享信息	146
14.7.3	练习	125	17.1	JSON 格式	146
第 15 章	为代码提供文档	127	17.2	使用 JSON 文件	148
15.1	好的文档的必要性	127	17.2.1	把 JSON 保存到 文件中	148
15.2	在代码中嵌入注释	128	17.2.2	把 JSON 打印到屏幕	149
15.3	用 docstring 说明代码	129	17.3	把对象保存为 JSON	150
15.4	包含 README 文件和 安装文件	131	17.4	创建定制的字典	151
15.4.1	编写安装说明	132	17.5	在现实世界中 使用 JSON	152
15.4.2	编写 README 文件	133	17.6	小结	155
15.5	在现实世界中 使用文档	133	17.7	Q&A	155
15.6	小结	134	17.8	作业	155
15.7	Q&A	134	17.8.1	问题	156
15.8	作业	135	17.8.2	回答	156
15.8.1	问题	135	17.8.3	练习	156
15.8.2	回答	135	第 18 章	把信息保存到数据库	157
15.8.3	练习	135	18.1	为什么使用数据库	157
第 16 章	操作程序文件	136	18.2	使用 SQL 和数据库交互	158
16.1	读写文件	136	18.2.1	Mac OS 系统中的 SQLite	158
16.1.1	从文件读取数据	136	18.2.2	在 Windows 系统中 安装 SQLite	158
16.1.2	把数据写入到文件中	137	18.2.3	测试 SQLite	159

18.3 创建一个数据库	159	20.3.2 添加变量	184
18.3.1 创建一个表	160	20.4 添加模板	184
18.3.2 添加数据	161	20.4.1 使用 HTML	185
18.4 查询数据库	162	20.4.2 创建模板	186
18.5 在现实世界中使用数据库	163	20.4.3 用 Jinja 添加动态内容	187
18.6 小结	165	20.4.4 为模板添加逻辑	188
18.7 Q&A	165	20.5 在现实世界中使用框架	189
18.8 作业	165	20.6 小结	190
18.8.1 问题	166	20.7 Q&A	190
18.8.2 回答	166	20.8 作业	190
18.8.3 练习	166	20.8.1 问题	191
第 19 章 使用 SQL 从数据库获取更多		20.8.2 回答	191
信息	167	20.8.3 练习	191
19.1 用 WHERE 进行过滤	168	第 21 章 用 PyGame 开发游戏	192
19.1.1 判断相等性	168	21.1 什么是 PyGame	192
19.1.2 判断不相等性	168	21.2 安装 PyGame	193
19.1.3 用 LIKE 查找		21.2.1 Windows	193
相似的项	169	21.2.2 Mac	193
19.1.4 使用 NOT LIKE 去		21.3 创建屏幕	194
查找不相似的项	170	21.3.1 主程序循环	194
19.1.5 带大于和小于的查询	170	21.3.2 使用用户输入	
19.2 用 ORDER BY 排序	171	(事件)	195
19.3 用 DISTINCT 获取唯一的项	171	21.4 创建形状	195
19.4 用 UPDATE 修改记录	172	21.4.1 添加颜色	196
19.5 用 DELETE 删除记录	172	21.4.2 定义一个圆	196
19.6 在现实世界中使用 SQL	173	21.5 在屏幕上移动物体	197
19.7 小结	176	21.6 获取用户输入	199
19.8 Q&A	176	21.7 绘制文本	200
19.9 作业	176	21.8 在现实世界中使用	
19.9.1 问题	176	PyGame	201
19.9.2 回答	177	21.9 小结	204
19.9.3 练习	177	21.10 Q&A	204
第 20 章 用 Flask 进行 Web 开发	178	21.11 作业	205
20.1 什么是 Flask	178	21.11.1 问题	205
20.2 安装 Flask	179	21.11.2 回答	205
20.2.1 Windows	180	21.11.3 练习	205
20.2.2 MAC	181	第 22 章 通过版本控制来正确地保存	
20.3 创建第一个 Flask 应用	182	代码	206
20.3.1 添加另一个视图	184	22.1 什么是版本控制	206

22.1.1	为什么版本控制很重要	206	23.6.3	邮件列表	225
22.1.2	版本控制是如何工作的	207	23.7	小结	225
22.2	使用 Git 和 GitHub 进行版本控制	208	23.8	Q&A	225
22.2.1	加入 GitHub	208	23.9	作业	226
22.2.2	GitHub 快速教程	208	23.9.1	问题	226
22.2.3	安装 Git	209	23.9.2	回答	226
22.3	管理仓库中的代码	209	23.9.3	练习	226
22.3.1	创建一个仓库	209	第 24 章 学习 Python 的下一步		227
22.3.2	签出仓库	210	24.1	感兴趣的项目	227
22.3.3	向仓库中添加项	210	24.1.1	创建 Web 站点和 Web 应用程序	227
22.3.4	推送到远程的仓库	211	24.1.2	创建桌面应用	228
22.3.5	获取更新	212	24.1.3	创建 Android 和 iOS 应用程序	229
22.4	用分支做实验性更改	213	24.1.4	游戏开发竞赛	229
22.4.1	创建分支	213	24.1.5	科学和数学中的 Python	229
22.4.2	合并分支	214	24.1.6	创造更好的工作环境	229
22.5	决定哪些内容不推送	215	24.2	参加会议	230
22.6	小结	215	24.3	使用 Linux	230
22.7	Q&A	216	24.4	为 Python 做贡献	231
22.8	作业	216	24.5	为其他的项目贡献	231
22.8.1	问题	216	24.6	学习另一种语言	232
22.8.2	回答	216	24.7	展望 Python 3	232
22.8.3	练习	216	24.8	推荐阅读	232
第 23 章 修复问题代码		218	24.9	推荐的 Web 站点	233
23.1	当代码中有了 Bug	218	24.10	小结	233
23.2	用 traceback 来找到错误	219	24.11	Q&A	233
23.3	用 pdb 调试器查找错误	220	24.12	作业	234
23.4	通过互联网查询解决方案	222	24.12.1	问题	234
23.5	尝试一个修复	223	24.12.2	回答	234
23.6	寻求外部支持	223	24.12.3	练习	234
23.6.1	互联网中继聊天	223			
23.6.2	本地用户组	225			

第 1 章

安装和运行 Python

在本章中，将学习：

- 如何确定所使用的是哪种操作系统。
- 如何安装并运行 Python。
- 如何在 Python 中输入基本的命令。

安装 Python 是学习本书过程中将要做的最重要的事情之一。不安装 Python，不可能学完本书剩下的内容！请花时间学习本章内容。如果不能通过本章最后的练习，将在随后的每一章中都会遇到麻烦。

1.1 认识操作系统

很多人都知道自己使用的是哪种计算机，但是对于计算机中所安装的具体的操作系统却一点也不清楚。搞清楚使用的是哪种操作系统，这对学习程序是至关重要的，因为对不同的操作系统，你需要下载的内容或者访问系统的某些部分的方式有所不同。

通常，如果你购买的是苹果公司生产的计算机（例如，一台 MacBook 或 PowerBook），那么它运行的操作系统是 Mac OS。大多数个人计算机运行的操作系统是 Windows。

如果遇到问题，你需要准确地知道正在运行的操作系统是什么版本。在 Mac OS 中，单击菜单栏中的 Apple 图标并选择“About this Mac”。将弹出一个带有计算机相关信息的窗口，其中包括操作系统的确切版本，如图 1-1 所示。

如果使用的是运行 Windows 系统的计算机，单击“Start”（开始）菜单并在“Accessories”（附件）项下找到“Command Prompt”（命令提示符）。单击它将打开计算机的命令提示符窗口，如图 1-2 所示。