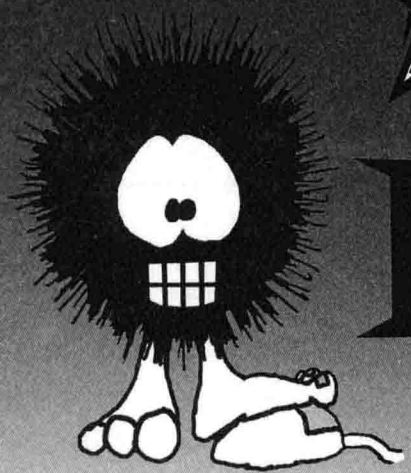
MANNING

易学 Python

[澳] Anthony Briggs 著
王威 袁国忠 译

人民邮电出版社
POSTS & TELECOM PRESS

易学 Python



[澳] Anthony Briggs 著
王威 袁国忠 译

人民邮电出版社
北京

图书在版编目（CIP）数据

易学Python / (澳) 布里格斯 (Briggs, A.) 著 ; 王威, 袁国忠译. -- 北京 : 人民邮电出版社, 2014.4
ISBN 978-7-115-33522-7

Ⅰ. ①易… Ⅱ. ①布… ②王… ③袁… 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(2013)第254251号

版权声明

◆ 著 [澳] Anthony Briggs
译 王 威 袁国忠
责任编辑 傅道坤
责任印制 程彦红 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：19.5
字数：417 千字 2014 年 4 月第 1 版
印数：1－3 500 册 2014 年 4 月北京第 1 次印刷

著作权合同登记号 图字：01-2011-7809 号

定价：45.00 元

读者服务热线：(010)81055410 印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京崇工商广字第 0021 号

内容提要

本书采用简洁、有趣、易学的方式对 Python 编程语言进行了讲解，其风格与通篇介绍编程特性、罗列语言功能的大多数编程图书不同，而是引导读者带着好奇，带着问题去学习、掌握 Python 编程语言，继而编写真实而有用的程序。

本书总共分为 3 部分，共 12 章，第 1 部分介绍了为何使用 Python 来编程。从第 2 部分起，通过编写 Hunt the Wumpus 游戏带领读者认识并解决编程中的一些实际问题。例如，如何管理复杂的程序，确保其清晰易懂；如何在程序中使用 Python 标准库，以节省编程时间，同时让程序更容易理解；如何对编写好的程序进行测试；如何进行面向对象的程序设计。第 3 部分则使用框架对书中的程序进行完善，让读者对 Python 的强大功能有一个更深入的认识。本书最后还提供了一些 Python 资源，供读者深入学习 Python 时参考。

无论您是零基础的 Python 初学人员，还是具有其他语言编程经验，但是想从事 Python 开发的人员，本书都将带领您踏上有趣的 Python 学习之路。

序

Anthony 邀请我为本书作序时，我的第一反应是，“又是负担，我可不干”。然而，我总得先瞄一眼吧。这一看不打紧，我很快发现了点缀其中的 *User Friendly* 卡通人物（我敢肯定，像我一样从穿孔卡和磁带时代起就一直从事计算机工作的人，都对 *User Friendly* 钟爱有加）。我想我得看一看手稿，结果发现它包含短短 12 章，可让您对 Python 及其最常见的应用有足够认识，进而踏入编程殿堂，或让您确定做程序员并非您想要的生活。

即便最终的结论是编程并非您想要的生活，为购买本书所做的投资也物超所值。如果您连使用 Python 进行编程都不喜欢，那您很可能根本就不喜欢编程。在这种情况下，阅读本书可避免您入错行，白白浪费数年光阴。

本书到处都是实用建议，作者从不装大尾巴狼，试图发表一些不可信的看法。我深信，这部脚踏实地的著作将帮助大量非程序员读者熟悉 Python 语言。

但愿本书能让广大读者对编程和迷人的信息技术有新的认识。当前，美国很多州教育资金都捉衿见肘，难以在中学开展丰富的计算机教育，而本书足以吸引中学生，让他们迷恋编程。等政府洞悉教育的目的后，学生们将被编程课本深深地吸引，很快消化全书的内容。

Steve Holden

The Open Bastion 总裁

关于作者

从2000年初开始，Anthony Briggs就一直在使用Python开发程序，当前正为墨尔本的Ramble Communications开发Web发布系统。他还曾为澳大利亚和加拿大的一家旅游公司开发预订系统，并最终成为首席开发人员，负责整个项目。

致谢

首先，感谢我美丽的妻子 Lyndall 给予支持，让我能抽出时间编写本书。本书的编写时间比预想的长得多，但她的热情始终如一，即便无数个周末我为了本书而深居简出。

其次，感谢 Manning 出版社的团队：感谢编辑 Sebastian Stirling 的建议和经验；感谢 June Eding 和 Tiffany Taylor 做最后的编辑和校对，确保本书按时付梓；感谢 Karen Tegtmeyer 的协调和组织工作；感谢 Michael Stephens 帮助制订最初的写作计划。

第三，感谢 J.D. Fraser 允许 Manning 出版社在 Hello 系列丛书中使用 *User Friendly* 中的卡通形象，并允许我在本书中给这些卡通形象撰写对白。

第四，感谢本书 beta 版本的所有测试人员帮助找出错误，他们是 Daniel Hadson、Eldar Marcussen、William Taylor、David Hepworth 和 Tony Haig；还要感谢 MEAP 项目的所有成员，感谢他们提出建议和批评以及指正错误。

最后，要感谢以下评审人员在各个阶段提供宝贵的反馈：Tray Skates、Curtis Miller、Joe Hoover、Michael R. Bain、Francesco Goggi、Mike Stok、Michael R. Head、Cheryl M. Davis、Daniel Bretoi、Amos Bannister、Rob Allen、Dr. John Grayson、William Z. Taylor、Munch Paulson、David Hepworth、Eldar Marcussen、Daniel Had-son、Tony Niemann、Paolo Corti、Edmon Begoli、Lester Lobo、Robby O'Connor 和 Sopan Shewale。特别感谢 Marion Newlevant 对终稿做详尽的技术审阅，还要特别感谢 Steve Holden 欣然为本书作序。

自序

受邀编写本书时，我无意再编写一本介绍性图书，而想采取不同的做法。我阅读过的编程图书无不罗列功能：列表可用于存储信息，您可调用 len(mylist)来获悉它包含多少项信息，调用 pop()来删除末尾的元素，调用 append()来添加元素，等等。这些就是您需要知道的有关列表的全部知识，接下来介绍下一项功能。在这样的图书中，即便有示例程序，也要么只包含寥寥数行代码，要么出现在最后几章，作为对全书内容的总结。

我自己在最初学习编程时，并不是先从头到尾阅读整本编程图书，等对一切都熟悉后再开始编写程序。相反，我带着问题（要做的事情）去阅读，并在阅读过程中搞懂这些问题。我经常阅读编程图书，但都是旨在搞清楚我遇到的问题。程序编写出来后，它们也许不是特别优雅，运行速度也可能不是特别快，但它们毕竟出自我的手——我知道其工作原理，还解决了我面临的实际问题。

就这样锻炼到今天，我编写的程序在很大程度上都可以说既优雅，速度也快。在我认识的优秀程序员中，大多也这样学习编程。在本书中，我竭尽所能地重现这个过程，但步伐更快，涵盖了我学到的所有编程知识，还有我曾遇到过的陷阱。除第 1 章和第 12 章外，每章都将一个实用程序作为核心，旨在演示特定 Python 功能或库——通常是多个。有些章节很有趣，有些很实用，但不再有乏味的铺垫章节，详尽地讲述烦人的细节——列表或字典的各种功能，甚至是如何使用 Python 将数字相加。

相反，您将编写一个个程序，并在需要时学习相关的 Python 功能，而不是预先学习它们。有几章建立在前几章的基础之上，因此您将学习如何扩展既有程序，以添加新功能并避免设计失控——无论您要编写的程序规模如何，都必须这样做。本书还探讨了多种编程风格：从简单脚本到面向对象程序，再到基于事件的游戏。

我旨在向您呈现一部与众不同的作品，让您从第 1 章开始就编写程序，并在实际应用中学习 Python 功能的用法。但愿这种写作方式能助您真正明白如何使用 Python。

关于本书

本书旨在帮助读者学习 Python 和如何使用 Python 编程。无论您在编程方面完全是门外汉还是有一定经验，本书都将引领您踏上编程之路，最终编写出网络游戏和 Web 应用程序。

本书的风格不同于大多数编程图书：不详尽地列举各种功能，而向读者展示一幅更真实的画卷。从第 2 章起，您就将跟随笔者的脚步，编写真实而实用的程序——既说优点也谈缺点。编程语言的每项功能都有其用途，但如果不见识因误用导致的 bug、有问题的代码和糟糕的程序，就很难对其用途有真正的认识。

在本书中，将逐步改进和扩展一些程序，让读者明白函数、类和模块等 Python 功能有助于您控制规模不断增大的代码。需要扩展程序时，这些功能还有助于减轻您的负担。

虽然没有明确指出，但本书实际上大致分三部分。第 1 章介绍 Python 基本语法、库的用法和一些常见概念，还有为理解程序的工作原理而需要知道的各种知识；第二部分介绍高级功能以及让您能够直接用来实现更多功能的库；最后一部分使用框架编写了一些完整的程序，这对您学习 Python 帮助更大。

阅读完本书后，您依然将受惠于它。本书的所有程序都是可扩展的，您可在编写自己的程序时重用它们。大多数经验丰富的程序员都有代码库，其中包含他以前编写的所有代码；本书的代码为您开发自己的项目打下了基础。

本书组织结构

第 1 章简要地介绍了 Python、编程以及编程的根本目的：为何编写程序、可使用程序做什么。本章还详细地阐述了如何在 Windows、Mac 和 Linux 系统上安装 Python，以及在安装过程中可能遇到的一些常见问题。

第 2 章介绍程序的基本组成部分，而您将编写第一个程序——Hunt the Wumpus。

在此过程中，您将亲身体验程序员面临的一些问题。例如，如何管理复杂的程序，确保它们清晰易懂。

第 3 章介绍著名的 Python 标准库，还有如何导入标准库代码以及其他程序员为执行常见任务而编写的代码。您将学习如何在程序中使用这些代码，以节省大量时间并让程序更容易理解。

第 4 章介绍如何测试程序，包括单元测试和系统测试以及一些常见的测试问题及其解决方案。在此过程中，您将编写一个简单且易于扩展的待办事项清单应用程序。

第 5 章介绍如何使用 Python 进行面向业务的编程：下载网页、分析其中的信息并使用这些信息来创建电子邮件和 CSV 文件。本章还介绍了如何让程序更健壮，能够应对错误的信息格式和其他错误。

第 6 章将编写一个冒险游戏，其中有洞穴，有怪物，还有财宝。在此过程中，您将学习类的工作原理以及如何设计面向对象的程序。

第 7 章介绍如何使用混合类、__getattribute__和特性等高级功能来扩展类。本章还简要地介绍了 Python 的其他高级功能，如迭代器、生成器、正则表达式和函数式编程。

第 8 章介绍 Django 并帮助您建立一个待办事项网站。您将学习 Django 模板、数据库处理功能、表单和管理功能。本章还介绍了一些常见的 Web 开发模式，包括 REST 式 URL 设计以及使用正确的 HTTP 方法。

第 9 章介绍如何使用 Pyglet 库编写类似于《行星撞击地球》和《月球着陆器》的街机游戏，您将学习几何学、基于事件的编程和定时器。

第 10 章对第 6 章编写的冒险游戏进行扩展，让您和朋友能够使用 Telnet 通过网络玩这款游戏。为此，我们将使用 Python 网络库 Twisted 来处理连接、定义协议以及添加日志功能。

第 11 章修改第 8 章编写的待办事项清单应用程序，让每位用户都有自己的待办事项清单。您将学习如何处理登录、使用 Django 创建用户、使用 Django 通用视图、确保 Web 应用程序的安全、将 Web 应用程序部署到 Apache 或 Nginx 等服务器。

最后，第 12 章提供了一些资源，供您继续学习 Python 时参考，这包括邮件列表和用户组、可供您阅读和探索的程序以及您可能想研究的其他库。

目录

第 1 章　为何学习 Python

本章介绍如下内容：

- 计算机和程序简介以及编写程序的原因；
- Python 简介及其如此杰出的原因；
- 安装 Python。

既然购买了本书，您很可能想学习编程技术。祝贺您！打算学编程的人不是很多，但编程很有趣，自学编程的回报也很高。编程是衡量您是不是文盲的新标准；如果不会编写简单程序（如批处理文件、邮件过滤器、电子表格公式），将在会这样做的人面前处于劣势。编程还是工具，可帮助您将点子付诸实施。

我 10 岁左右开始涉足编程，当时使用的是 Commodore 64。在那个时候，除游戏和简单字处理软件外，现成的软件不多。Commodore 等计算机自带了 BASIC，编程很容易，无需学很多知识就能很快得到结果。

然而，这样的情况一去不复返了。当前，要编写程序必须先安装软件；但学会编程后，就可创建各种神奇的程序，替您完成繁琐的工作，向您提供信息，供您娱乐。最重要的是，编程很有趣，每个人都应尝试。

您将发现，本书有一些卡通人物点缀其中。这旨在以有趣的方式提供一些背景知

识，让您知道接下来要讲的内容，或指出一些常见问题。这些卡通人物都来自 *User Friendly*，但内容都是我编写的，如果您不喜欢，就责怪我吧。

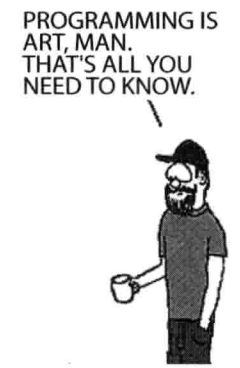
下面开始介绍有关编程的基本知识。

1.1 学习编程

这是一本编程书，在详细探讨编程之前，有必要介绍一些基本知识：什么是编程？如何编程？编程的定义很简单：

编程就是告诉计算机做什么。

但与大多数定义一样，这种定义太过简单。与国际象棋一样，学习编程的基本规则很容易，但要精通这些规则并结合使用它们要难得多。编程与人类活动的很多方面相关：从某种意义上说，如果不进行编程，就很难让计算机从事有意义的工作；编程不仅关乎数字和计算，还关乎设计、创意和个人表达。



1.1.1 告诉计算机做什么

下面将前面的编程定义分解为几部分，并分别审视它们。要明白前面的编程定义，需要知道计算机是什么以及“告诉”和“做什么”的准确含义。

1. 计算机

计算机是速度很快的计算器，可按您的指令做简单决策。计算机指令很简单，通常包含计算机要完成的任务，如相加和比较。然而，结合使用一系列指令可创建大型程序，让您能够完成复杂的工作，如撰写文档、玩游戏、计算账户余额以及控制核反应堆。

计算机看似很聪明，实际上很傻，一根筋且缺乏基本常识。计算机毕竟只是机器，您让它做什么，它就做什么，根本不考虑后果。就拿删除整个硬盘的命令来说吧，大多数人明白其后果很严重，可能在遵命行事前向您核实，但计算机什么都不问，直接将硬盘上所有的数据都删除。

注意：您让它做什么它就做什么，这既是计算机的优点，也是计算机的缺点。

如果使用（或编写）的程序行为怪异或无缘无故地崩溃，这不是它的错——它只是按指令行事。

2. 告诉

使用 Python 时，您通常这样向它发出命令：在文本文件中输入程序代码，再让程

序 Python 运行该文件，这将在本章后面介绍。您输入的指令可复杂可简单，执行的任务各种各样：将数字相加、打开其他文件、在屏幕上显示内容等。简单的 Python 程序类似于下面这样：

```
number = "42"
print "Guess my number..."
guess = raw_input(">")
if guess == number:
    print "Yes! that's it!"
else:
    print "No - it's", number

raw_input("hit enter to continue")
```

如果您不明白这个程序的含义，不用太担心；这个示例旨在提供一些背景知识。

3. 做什么

现在事情开始变得有趣起来了。大多数现代计算机都是“图灵完备（Turing complete）”的，即什么都能做——您想得到的任何事情计算机都能做。至少从理论上说如此：所需的时间或复杂程度可能超乎预期，或者需要特殊硬件（如果您希望以特定方式交互），但只要编写的程序没有问题，并提供了足够的数据，计算机就什么都能做。下面是计算机用于完成的一些任务。

- 控制有人或无人航天器和探测器，引导机器人进入其他星球，包括火星探索流浪者 Spirit 和 Opportunity。
- 通过计算机网络（互联网和万维网）将数据传输到世界各地。通过网络可向世界各地传输信息，还可接收来自世界各地的信息，且在一秒钟内就可搞定。
- 打造机器人，从工业机械手到 Roomba 真空吸尘器，再到像人类一样能爬楼并具备基本感情的机器人。
- 模拟现实世界的现象，如重力、光和天气。这包括科学模型及大多数游戏。

您可能没有将机器人探测器送到其他星球所需的硬件，但至少从理论上说，您依然能够运行这些程序。例如，就拿用于控制 Spirit 和 Opportunity 的计算机来说吧，其计算能力根本无法与您的台式机、笔记本电脑和手机相媲美，真是难以置信。