

TURING 图灵程序设计丛书



Python Crash Course  
A Hands-On, Project-Based Introduction to Programming

# Python编程 从入门到实践

【美】Eric Matthes 著 袁国忠 译



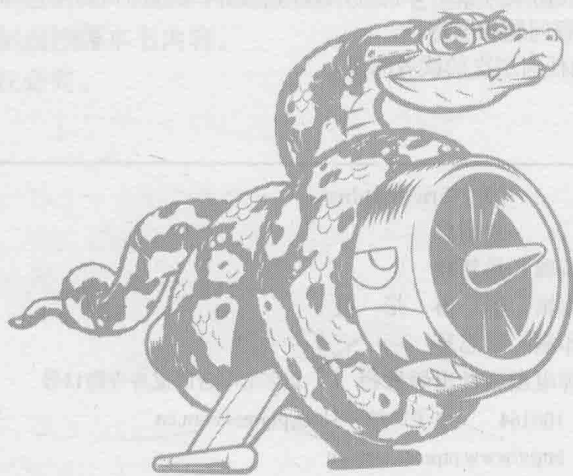
中国工信出版集团



人民邮电出版社  
POSTS & TELECOM PRESS

# Python编程 从入门到实践

【美】Eric Matthes 著 袁国忠 译



## Python Crash Course

A Hands-On, Project-Based Introduction to Programming

人民邮电出版社  
北京

## 图书在版编目 (CIP) 数据

Python编程：从入门到实践 / (美) 埃里克·马瑟斯 (Eric Matthes) 著；袁国忠译. — 北京：人民邮电出版社，2016.7

(图灵程序设计丛书)

ISBN 978-7-115-42802-8

I. ①P… II. ①埃… ②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(2016)第139461号

## 内 容 提 要

本书是一本针对所有层次的 Python 读者而作的 Python 入门书。全书分两部分：第一部分介绍用 Python 编程所必须了解的基本概念，包括 matplotlib、NumPy 和 Pygal 等强大的 Python 库和工具介绍，以及列表、字典、if 语句、类、文件与异常、代码测试等内容；第二部分将理论付诸实践，讲解如何开发三个项目，包括简单的 Python 2D 游戏开发，如何利用数据生成交互式的信息图，以及创建和定制简单的 Web 应用，并帮读者解决常见编程问题和困惑。

本书适合对 Python 感兴趣的任何层次的读者阅读。

- 
- ◆ 著 [美] Eric Matthes
  - 译 袁国忠
  - 责任编辑 岳新欣
  - 执行编辑 杨琳 张曼
  - 责任印制 彭志环
  - ◆ 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>
  - 北京昌平百善印刷厂印刷
  - ◆ 开本：800×1000 1/16
  - 印张：29.75
  - 字数：703千字 2016年7月第1版
  - 印数：1-4 000册 2016年7月北京第1次印刷
  - 著作权合同登记号 图字：01-2016-1807号

---

定价：89.00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：京东工商广字第 8052 号

# 版权声明

Copyright © 2016 by Eric Matthes. *Python Crash Course: A Hands-On, Project-Based Introduction to Programming*, ISBN 978-1-59327-603-4, published by No Starch Press. Simplified Chinese-language edition copyright © 2016 by Posts and Telecom Press. All rights reserved.

No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

本书中文简体字版由No Starch Press授权人民邮电出版社独家出版。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。

版权所有，侵权必究。

## 读者对象

本书旨在让初学者快速上手Python，以便能够编写和执行自己的程序——当然，最初可能只是Web应用程序。同时，本书还包含许多关于Python的基本理论知识。本书包含任何年龄的读者阅读，它不需要任何Python编程经验，甚至不需要知道如何安装Python。如果你对编程感兴趣，那么本书将为你提供许多关于Python的有趣项目，并帮助你解决有意义的编程问题。如果你对Python编程感兴趣，那么本书将为你提供许多关于Python的有趣项目，并帮助你解决有意义的编程问题。

## 本书内容

本书旨在让你快速上手Python，并帮助你解决有意义的编程问题。本书包含任何年龄的读者阅读，它不需要任何Python编程经验，甚至不需要知道如何安装Python。如果你对编程感兴趣，那么本书将为你提供许多关于Python的有趣项目，并帮助你解决有意义的编程问题。

# 朋友对朋

谨以此书献给我的父亲，以及儿子Ever。感谢父亲抽出时间来回答我提出的每个编程问题，而儿子Ever也开始向我提问了。

# 前言

如何学习编写第一个程序，每个程序员都有不同的故事。我还是个孩子时就开始学习编程了，当时我父亲在计算时代的先锋之一——数字设备公司（Digital Equipment Corporation）工作。我使用一台简陋的计算机编写了第一个程序，这台计算机是父亲在家里的地下室组装而成的，它没有机箱，裸露的主板与键盘相连，显示器是裸露的阴极射线管。我编写的这个程序是一款简单的猜数字游戏，其输出类似于下面这样：

---

```
I'm thinking of a number! Try to guess the number I'm thinking of: 25
Too low! Guess again: 50
Too high! Guess again: 42
That's it! Would you like to play again? (yes/no) no
Thanks for playing!
```

---

看到家人玩着我编写的游戏，而且它完全按我预期的方式运行，我心里不知有多满足。此情景我永远都忘不了。

儿童时期的这种体验一直影响我至今。现在，每当我通过编写程序解决了一个问题时，心里都会感到非常满足。相比于孩提时期，我现在编写的软件满足了更大的需求，但通过编写程序获得的满足感几乎与从前一样。

## 读者对象

本书旨在让你尽快学会Python，以便能够编写能正确运行的程序——游戏、数据可视化和Web应用程序，同时掌握让你终身受益的基本编程知识。本书适合任何年龄的读者阅读，它不要求你有任何Python编程经验，甚至不要求你有编程经验。如果你想快速掌握基本的编程知识以便专注于开发感兴趣的项目，并想通过解决有意义的问题来检查你对新学概念的理解程度，那么本书就是为你编写的。本书还可供初中和高中教师用来通过开发项目向学生介绍编程。

## 本书内容

本书旨在让你成为优秀的程序员，具体地说，是优秀的Python程序员。通过阅读本书，你将迅速掌握编程概念，打下坚实的基础，并养成良好的习惯。阅读本书后，你就可以开始学习Python高级技术，并能够更轻松地掌握其他编程语言。

在本书的第一部分，你将学习编写Python程序时需要熟悉的基本编程概念，你刚接触几乎任何编程语言时都需要学习这些概念。你将学习各种数据以及在程序中将数据存储到列表和字典中的方式。你将学习如何创建数据集以及如何高效地遍历这些集合。你将学习使用while和if语句来检查条件，并在条件满足时执行代码的一部分，而在条件不满足时执行代码的另一部分——这可为自动完成处理提供极大的帮助。

你将学习获取用户输入，让程序能够与用户交互，并在用户没停止输入时保持运行状态。你将探索如何编写函数来让程序的各个部分可重用，这样你编写执行特定任务的代码后，想使用它多少次都可以。然后，你将学习使用类来扩展这种概念以实现更复杂的行为，从而让非常简单的程序也能处理各种不同情形。你将学习编写妥善处理常见错误的程序。学习这些基本概念后，你就能编写一些简短的程序来解决一些明确的问题。最后，你将向中级编程迈出第一步，学习如何为代码编写测试，以便在进一步改进程序时不用担心可能引入bug。第一部分介绍的知识让你能够开发更大、更复杂的项目。

在第二部分，你将利用在第一部分学到的知识来开发三个项目。你可以根据自己的情况，以最合适的顺序完成这些项目；你也可以选择只完成其中的某些项目。在第一个项目（第12~14章）中，你将创建一个类似于《太空入侵者》的射击游戏。这个游戏名为《外星人入侵》，它包含多个难度不断增加的等级。完成这个项目后，你就能够自己动手开发2D游戏了。

第二个项目（第15~17章）介绍数据可视化。数据科学家的目标是通过各种可视化技术来搞懂海量信息。你将使用通过代码生成的数据集、已经从网络下载下来的数据集以及程序自动下载的数据集。完成这个项目后，你将能够编写能对大型数据集进行筛选的程序，并以可视化方式将筛选出来的数据呈现出来。

在第三个项目（第18~20章）中，你将创建一个名为“学习笔记”的小型Web应用程序。这个项目能够让用户将学到的与特定主题相关的概念记录下来。你将能够分别记录不同的主题，还可以让其他人建立账户并开始记录自己的学习笔记。你还将学习如何部署这个项目，让任何人都能够通过网络访问它，而不管他身处何方。

## 为何使用 Python

继续使用Python，还是转而使用其他语言——也许是编程领域较新的语言？我每年都会考虑这个问题。可我依然专注于Python，其中的原因很多。Python是一种效率极高的语言：相比于众多其他的语言，使用Python编写时，程序包含的代码行更少。Python的语法也有助于创建整洁的代码：相比其他语言，使用Python编写的代码更容易阅读、调试和扩展。

大家将Python用于众多方面：编写游戏、创建Web应用程序、解决商业问题以及供各类有趣的公司开发内部工具。Python还在科学领域被大量用于学术研究和应用研究。

我依然使用Python的一个最重要的原因是，Python社区有形形色色充满激情的人。对程序员来说，社区非常重要，因为编程绝非孤独的修行。大多数程序员都需要向解决过类似问题的人寻求建议，经验最为丰富的程序员也不例外。需要有人帮助解决问题时，有一个联系紧密、互帮互

助的社区至关重要，而对于像你一样将Python作为第一门语言来学习的人而言，Python社区无疑是坚强的后盾。

Python是一门杰出的语言，值得你去学习，咱们现在就开始吧！

# 前 言

本书是为那些想学习Python的人而写的。本书假设你没有任何编程经验，但假设你熟悉基本的计算机操作。本书假设你有一台可以运行Python的计算机。本书假设你有一台可以运行Python的计算机。本书假设你有一台可以运行Python的计算机。

本书假设你有一台可以运行Python的计算机。本书假设你有一台可以运行Python的计算机。本书假设你有一台可以运行Python的计算机。本书假设你有一台可以运行Python的计算机。

本书假设你有一台可以运行Python的计算机。本书假设你有一台可以运行Python的计算机。本书假设你有一台可以运行Python的计算机。本书假设你有一台可以运行Python的计算机。

本书假设你有一台可以运行Python的计算机。本书假设你有一台可以运行Python的计算机。本书假设你有一台可以运行Python的计算机。本书假设你有一台可以运行Python的计算机。

本书假设你有一台可以运行Python的计算机。本书假设你有一台可以运行Python的计算机。本书假设你有一台可以运行Python的计算机。本书假设你有一台可以运行Python的计算机。

本书假设你有一台可以运行Python的计算机。本书假设你有一台可以运行Python的计算机。本书假设你有一台可以运行Python的计算机。本书假设你有一台可以运行Python的计算机。

本书假设你有一台可以运行Python的计算机。本书假设你有一台可以运行Python的计算机。本书假设你有一台可以运行Python的计算机。本书假设你有一台可以运行Python的计算机。



# 致 谢

要是没有No Starch Press出色的专业人士的帮助，本书根本不可能出版。Bill Pollock邀请我编写一本入门图书，因此这里要深深感谢他给予我这样的机会。Tyler Ortman在我编写本书的早期帮助我理清思路。Liz Chadwick和Leslie Shen详细阅读了每一章，并提出了宝贵的意见，而Anne Marie Walker让本书的很多地方都更清晰。Riley Hoffman回答了我就图书出版过程提出的每个问题，并且耐心地将我的作品变成了漂亮的图书。

感谢技术审稿人Kenneth Love。我与Kenneth相识于一次PyCon大会，他对Python和Python社区充满热情，一直是我获取专业灵感的源泉。Kenneth不仅检查了本书介绍的知识是否正确，还抱着让初学编程者对Python语言和编程有扎实认识的目的进行了审阅。即便如此，倘若书中有任何不准确的地方，责任都完全由我承担。

感谢我的父亲，感谢他在我很小的时候就向我介绍编程，而且一点都不担心我破坏他的设备。感谢妻子Erin在我编写本书期间对我一如既往的鼓励和支持。还要感谢儿子Ever，他的好奇心每天都会给我带来灵感。

# 目 录

## 第一部分 基础知识

第 1 章 起步	2
1.1 搭建编程环境	2
1.1.1 Python 2 和 Python 3	2
1.1.2 运行 Python 代码片段	3
1.1.3 Hello World 程序	3
1.2 在不同操作系统中搭建 Python 编程环境	3
1.2.1 在 Linux 系统中搭建 Python 编程环境	3
1.2.2 在 OS X 系统中搭建 Python 编程环境	6
1.2.3 在 Windows 系统中搭建 Python 编程环境	8
1.3 解决安装问题	12
1.4 从终端运行 Python 程序	13
1.4.1 在 Linux 和 OS X 系统中从终端运行 Python 程序	13
1.4.2 在 Windows 系统中从终端运行 Python 程序	13
1.5 小结	14
第 2 章 变量和简单数据类型	15
2.1 运行 hello_world.py 时发生的情况	15
2.2 变量	16
2.2.1 变量的命名和使用	16
2.2.2 使用变量时避免命名错误	17
2.3 字符串	18
2.3.1 使用方法修改字符串的大小写	19
2.3.2 合并(拼接)字符串	19

2.3.3 使用制表符或换行符来添加空白	20
2.3.4 删除空白	21
2.3.5 使用字符串时避免语法错误	22
2.3.6 Python 2 中的 print 语句	23
2.4 数字	24
2.4.1 整数	24
2.4.2 浮点数	25
2.4.3 使用函数 str() 避免类型错误	25
2.4.4 Python 2 中的整数	26
2.5 注释	27
2.5.1 如何编写注释	27
2.5.2 该编写什么样的注释	28
2.6 Python 之禅	28
2.7 小结	30
第 3 章 列表简介	31
3.1 列表是什么	31
3.1.1 访问列表元素	32
3.1.2 索引从 0 而不是 1 开始	32
3.1.3 使用列表中的各个值	33
3.2 修改、添加和删除元素	33
3.2.1 修改列表元素	34
3.2.2 在列表中添加元素	34
3.2.3 从列表中删除元素	35
3.3 组织列表	39
3.3.1 使用方法 sort() 对列表进行永久性排序	39
3.3.2 使用函数 sorted() 对列表进行临时排序	40
3.3.3 倒着打印列表	41

3.3.4 确定列表的长度	41	5.2 条件测试	65
3.4 使用列表时避免索引错误	42	5.2.1 检查是否相等	65
3.5 小结	43	5.2.2 检查是否相等时不考虑大小写	65
<b>第4章 操作列表</b>	<b>44</b>	5.2.3 检查是否不相等	66
4.1 遍历整个列表	44	5.2.4 比较数字	67
4.1.1 深入地研究循环	45	5.2.5 检查多个条件	67
4.1.2 在 for 循环中执行更多的操作	46	5.2.6 检查特定值是否包含在列表中	68
4.1.3 在 for 循环结束后执行一些操作	47	5.2.7 检查特定值是否不包含在列表中	69
4.2 避免缩进错误	47	5.2.8 布尔表达式	69
4.2.1 忘记缩进	48	5.3 if 语句	70
4.2.2 忘记缩进额外的代码行	48	5.3.1 简单的 if 语句	70
4.2.3 不必要的缩进	49	5.3.2 if-else 语句	71
4.2.4 循环后不必要的缩进	49	5.3.3 if-elif-else 结构	72
4.2.5 遗漏了冒号	50	5.3.4 使用多个 elif 代码块	73
4.3 创建数值列表	51	5.3.5 省略 else 代码块	74
4.3.1 使用函数 range()	51	5.3.6 测试多个条件	74
4.3.2 使用 range() 创建数字列表	51	5.4 使用 if 语句处理列表	76
4.3.3 对数字列表执行简单的统计计算	53	5.4.1 检查特殊元素	77
4.3.4 列表解析	53	5.4.2 确定列表不是空的	78
4.4 使用列表的一部分	54	5.4.3 使用多个列表	78
4.4.1 切片	54	5.5 设置 if 语句的格式	80
4.4.2 遍历切片	56	5.6 小结	80
4.4.3 复制列表	56	<b>第6章 字典</b>	<b>81</b>
4.5 元组	59	6.1 一个简单的字典	81
4.5.1 定义元组	59	6.2 使用字典	82
4.5.2 遍历元组中的所有值	59	6.2.1 访问字典中的值	82
4.5.3 修改元组变量	60	6.2.2 添加键-值对	83
4.6 设置代码格式	61	6.2.3 先创建一个空字典	83
4.6.1 格式设置指南	61	6.2.4 修改字典中的值	84
4.6.2 缩进	61	6.2.5 删除键-值对	85
4.6.3 行长	61	6.2.6 由类似对象组成的字典	86
4.6.4 空行	62	6.3 遍历字典	87
4.6.5 其他格式设置指南	62	6.3.1 遍历所有的键-值对	87
4.7 小结	63	6.3.2 遍历字典中的所有键	89
<b>第5章 if 语句</b>	<b>64</b>	6.3.3 按顺序遍历字典中的所有键	91
5.1 一个简单示例	64	6.3.4 遍历字典中的所有值	91
		6.4 嵌套	93

6.4.1	字典列表	93
6.4.2	在字典中存储列表	95
6.4.3	在字典中存储字典	97
6.5	小结	99
<b>第7章</b>	<b>用户输入和 while 循环</b>	<b>100</b>
7.1	函数 input() 的工作原理	100
7.1.1	编写清晰的程序	101
7.1.2	使用 int() 来获取数值输入	102
7.1.3	求模运算符	103
7.1.4	在 Python 2.7 中获取输入	104
7.2	while 循环简介	104
7.2.1	使用 while 循环	104
7.2.2	让用户选择何时退出	105
7.2.3	使用标志	106
7.2.4	使用 break 退出循环	107
7.2.5	在循环中使用 continue	108
7.2.6	避免无限循环	109
7.3	使用 while 循环来处理列表和字典	110
7.3.1	在列表之间移动元素	110
7.3.2	删除包含特定值的所有列表元素	111
7.3.3	使用用户输入来填充字典	112
7.4	小结	113
<b>第8章</b>	<b>函数</b>	<b>114</b>
8.1	定义函数	114
8.1.1	向函数传递信息	115
8.1.2	实参和形参	115
8.2	传递实参	116
8.2.1	位置实参	116
8.2.2	关键字实参	118
8.2.3	默认值	118
8.2.4	等效的函数调用	119
8.2.5	避免实参错误	120
8.3	返回值	121
8.3.1	返回简单值	121
8.3.2	让实参变成可选的	122
8.3.3	返回字典	123
8.3.4	结合使用函数和 while 循环	124
8.4	传递列表	126
8.4.1	在函数中修改列表	126
8.4.2	禁止函数修改列表	129
8.5	传递任意数量的实参	130
8.5.1	结合使用位置实参和任意数量实参	131
8.5.2	使用任意数量的关键字实参	131
8.6	将函数存储在模块中	133
8.6.1	导入整个模块	133
8.6.2	导入特定的函数	134
8.6.3	使用 as 给函数指定别名	134
8.6.4	使用 as 给模块指定别名	135
8.6.5	导入模块中的所有函数	135
8.7	函数编写指南	136
8.8	小结	137
<b>第9章</b>	<b>类</b>	<b>138</b>
9.1	创建和使用类	138
9.1.1	创建 Dog 类	139
9.1.2	根据类创建实例	140
9.2	使用类和实例	142
9.2.1	Car 类	143
9.2.2	给属性指定默认值	143
9.2.3	修改属性的值	144
9.3	继承	147
9.3.1	子类的方法 __init__()	147
9.3.2	Python 2.7 中的继承	149
9.3.3	给子类定义属性和方法	149
9.3.4	重写父类的方法	150
9.3.5	将实例用作属性	150
9.3.6	模拟实物	152
9.4	导入类	153
9.4.1	导入单个类	153
9.4.2	在一个模块中存储多个类	155
9.4.3	从一个模块中导入多个类	156
9.4.4	导入整个模块	157
9.4.5	导入模块中的所有类	157
9.4.6	在一个模块中导入另一个模块	157
9.4.7	自定义工作流程	158
9.5	Python 标准库	159

9.6	类编码风格	161
9.7	小结	161
<b>第 10 章</b>	<b>文件和异常</b>	<b>162</b>
10.1	从文件中读取数据	162
10.1.1	读取整个文件	162
10.1.2	文件路径	164
10.1.3	逐行读取	165
10.1.4	创建一个包含文件各行内容的列表	166
10.1.5	使用文件的内容	166
10.1.6	包含一百万位的大型文件	168
10.1.7	圆周率值中包含你的生日吗	168
10.2	写入文件	169
10.2.1	写入空文件	170
10.2.2	写入多行	170
10.2.3	附加到文件	171
10.3	异常	172
10.3.1	处理 ZeroDivisionError 异常	172
10.3.2	使用 try-except 代码块	173
10.3.3	使用异常避免崩溃	173
10.3.4	else 代码块	174
10.3.5	处理 FileNotFoundError 异常	175
10.3.6	分析文本	176
10.3.7	使用多个文件	177
10.3.8	失败时一声不吭	178
10.3.9	决定报告哪些错误	179
10.4	存储数据	180
10.4.1	使用 json.dump() 和 json.load()	180
10.4.2	保存和读取用户生成的数据	181
10.4.3	重构	183
10.5	小结	186
<b>第 11 章</b>	<b>测试代码</b>	<b>187</b>
11.1	测试函数	187
11.1.1	单元测试和测试用例	188
11.1.2	可通过的测试	188
11.1.3	不能通过的测试	190
11.1.4	测试未通过时怎么办	191
11.1.5	添加新测试	191
11.2	测试类	193
11.2.1	各种断言方法	193
11.2.2	一个要测试的类	194
11.2.3	测试 AnonymousSurvey 类	195
11.2.4	方法 setUp()	197
11.3	小结	199
<b>第二部分 项 目</b>		
<b>项目 1</b>	<b>外星人入侵</b>	<b>202</b>
<b>第 12 章</b>	<b>武装飞船</b>	<b>203</b>
12.1	规划项目	203
12.2	安装 Pygame	204
12.2.1	使用 pip 安装 Python 包	204
12.2.2	在 Linux 系统中安装 Pygame	206
12.2.3	在 OS X 系统中安装 Pygame	207
12.2.4	在 Windows 系统中安装 Pygame	207
12.3	开始游戏项目	207
12.3.1	创建 Pygame 窗口以及响应用户输入	208
12.3.2	设置背景色	209
12.3.3	创建设置类	210
12.4	添加飞船图像	211
12.4.1	创建 Ship 类	212
12.4.2	在屏幕上绘制飞船	213
12.5	重构: 模块 game_functions	214
12.5.1	函数 check_events()	214
12.5.2	函数 update_screen()	215
12.6	驾驶飞船	216
12.6.1	响应按键	216
12.6.2	允许不断移动	217

12.6.3	左右移动	219	13.5	射杀外星人	246
12.6.4	调整飞船的速度	220	13.5.1	检测子弹与外星人的碰撞	246
12.6.5	限制飞船的活动范围	221	13.5.2	为测试创建大子弹	247
12.6.6	重构 check_events()	222	13.5.3	生成新的外星人群	248
12.7	简单回顾	223	13.5.4	提高子弹的速度	249
12.7.1	alien_invasion.py	223	13.5.5	重构 update_bullets()	249
12.7.2	settings.py	223	13.6	结束游戏	250
12.7.3	game_functions.py	223	13.6.1	检测外星人和飞船碰撞	250
12.7.4	ship.py	223	13.6.2	响应外星人和飞船碰撞	251
12.8	射击	224	13.6.3	有外星人到达屏幕底端	254
12.8.1	添加子弹设置	224	13.6.4	游戏结束	255
12.8.2	创建 Bullet 类	224	13.7	确定应运行游戏的哪些部分	255
12.8.3	将子弹存储到编组中	226	13.8	小结	256
12.8.4	开火	227			
12.8.5	删除已消失的子弹	228	第 14 章	记分	257
12.8.6	限制子弹数量	229	14.1	添加 Play 按钮	257
12.8.7	创建函数 update_bullets()	229	14.1.1	创建 Button 类	258
12.8.8	创建函数 fire_bullet()	230	14.1.2	在屏幕上绘制按钮	259
12.9	小结	231	14.1.3	开始游戏	261
第 13 章	外星人	232	14.1.4	重置游戏	261
13.1	回顾项目	232	14.1.5	将 Play 按钮切换到非活动 状态	263
13.2	创建第一个外星人	233	14.1.6	隐藏光标	263
13.2.1	创建 Alien 类	233	14.2	提高等级	264
13.2.2	创建 Alien 实例	234	14.2.1	修改速度设置	264
13.2.3	让外星人出现在屏幕上	235	14.2.2	重置速度	266
13.3	创建一群外星人	236	14.3	记分	267
13.3.1	确定一行可容纳多少个 外星人	236	14.3.1	显示得分	267
13.3.2	创建多行外星人	236	14.3.2	创建记分牌	268
13.3.3	创建外星人群	237	14.3.3	在外星人被消灭时更新 得分	270
13.3.4	重构 create_fleet()	239	14.3.4	将消灭的每个外星人的点数 都计入得分	271
13.3.5	添加行	240	14.3.5	提高点数	271
13.4	让外星人群移动	242	14.3.6	将得分圆整	272
13.4.1	向右移动外星人	243	14.3.7	最高得分	274
13.4.2	创建表示外星人移动方向的 设置	244	14.3.8	显示等级	276
13.4.3	检查外星人是否撞到了屏幕 边缘	244	14.3.9	显示余下的飞船数	279
13.4.4	向下移动外星人群并改变移 动方向	245	14.4	小结	283

项目 2 数据可视化	284	15.4.5 分析结果	305
第 15 章 生成数据	285	15.4.6 绘制直方图	306
15.1 安装 matplotlib	285	15.4.7 同时掷两个骰子	307
15.1.1 在 Linux 系统中安装 matplotlib	286	15.4.8 同时掷两个面数不同的 骰子	309
15.1.2 在 OS X 系统中安装 matplotlib	286	15.5 小结	311
15.1.3 在 Windows 系统中安装 matplotlib	286	第 16 章 下载数据	312
15.1.4 测试 matplotlib	287	16.1 CSV 文件格式	312
15.1.5 matplotlib 画廊	287	16.1.1 分析 CSV 文件头	313
15.2 绘制简单的折线图	287	16.1.2 打印文件头及其位置	314
15.2.1 修改标签文字和线条粗细	288	16.1.3 提取并读取数据	314
15.2.2 校正图形	289	16.1.4 绘制气温图表	315
15.2.3 使用 scatter() 绘制散点图 并设置其样式	290	16.1.5 模块 datetime	316
15.2.4 使用 scatter() 绘制一系 列点	291	16.1.6 在图表中添加日期	317
15.2.5 自动计算数据	292	16.1.7 涵盖更长的时间	318
15.2.6 删除数据点的轮廓	293	16.1.8 再绘制一个数据系列	319
15.2.7 自定义颜色	293	16.1.9 给图表区域着色	320
15.2.8 使用颜色映射	294	16.1.10 错误检查	321
15.2.9 自动保存图表	295	16.2 制作世界人口地图: JSON 格式	324
15.3 随机漫步	295	16.2.1 下载世界人口数据	324
15.3.1 创建 RandomWalk() 类	296	16.2.2 提取相关的数据	324
15.3.2 选择方向	296	16.2.3 将字符串转换为数字值	326
15.3.3 绘制随机漫步图	297	16.2.4 获取两个字母的国别码	327
15.3.4 模拟多次随机漫步	298	16.2.5 制作世界地图	329
15.3.5 设置随机漫步图的样式	299	16.2.6 在世界地图上呈现数字 数据	330
15.3.6 给点着色	299	16.2.7 绘制完整的世界人口地图	331
15.3.7 重新绘制起点和终点	300	16.2.8 根据人口数量将国家分组	333
15.3.8 隐藏坐标轴	301	16.2.9 使用 Pygal 设置世界地图的 样式	334
15.3.9 增加点数	301	16.2.10 加亮颜色主题	335
15.3.10 调整尺寸以适合屏幕	302	16.3 小结	337
15.4 使用 Pygal 模拟掷骰子	303	第 17 章 使用 API	338
15.4.1 安装 Pygal	304	17.1 使用 Web API	338
15.4.2 Pygal 画廊	304	17.1.1 Git 和 GitHub	338
15.4.3 创建 Die 类	304	17.1.2 使用 API 调用请求数据	339
15.4.4 掷骰子	305	17.1.3 安装 requests	339
		17.1.4 处理 API 响应	340



17.1.5	处理响应字典	340	18.5	小结	381
17.1.6	概述最受欢迎的仓库	342	第 19 章	用户账户	382
17.1.7	监视 API 的速率限制	343	19.1	让用户能够输入数据	382
17.2	使用 Pygal 可视化仓库	344	19.1.1	添加新主题	382
17.2.1	改进 Pygal 图表	346	19.1.2	添加新条目	386
17.2.2	添加自定义工具提示	347	19.1.3	编辑条目	390
17.2.3	根据数据绘图	349	19.2	创建用户账户	392
17.2.4	在图表中添加可单击的 链接	350	19.2.1	应用程序 users	393
17.3	Hacker News API	350	19.2.2	登录页面	394
17.4	小结	353	19.2.3	注销	396
项目 3	Web 应用程序	354	19.2.4	注册页面	397
第 18 章	Django 入门	355	19.3	让用户拥有自己的数据	400
18.1	建立项目	355	19.3.1	使用@login_required 限制 访问	400
18.1.1	制定规范	355	19.3.2	将数据关联到用户	402
18.1.2	建立虚拟环境	356	19.3.3	只允许用户访问自己的 主题	405
18.1.3	安装 virtualenv	356	19.3.4	保护用户的主题	405
18.1.4	激活虚拟环境	357	19.3.5	保护页面 edit_entry	406
18.1.5	安装 Django	357	19.3.6	将新主题关联到当前用户	406
18.1.6	在 Django 中创建项目	357	19.4	小结	408
18.1.7	创建数据库	358	第 20 章	设置应用程序的样式并对其进行 部署	409
18.1.8	查看项目	359	20.1	设置项目“学习笔记”的样式	409
18.2	创建应用程序	360	20.1.1	应用程序 django-bootstrap3	410
18.2.1	定义模型	360	20.1.2	使用 Bootstrap 来设置项目 “学习笔记”的样式	411
18.2.2	激活模型	362	20.1.3	修改 base.html	411
18.2.3	Django 管理网站	363	20.1.4	使用 jumbotron 设置主页的 样式	414
18.2.4	定义模型 Entry	365	20.1.5	设置登录页面的样式	415
18.2.5	迁移模型 Entry	366	20.1.6	设置 new_topic 页面的 样式	416
18.2.6	向管理网站注册 Entry	366	20.1.7	设置 topics 页面的样式	417
18.2.7	Django shell	367	20.1.8	设置 topic 页面中条目的 样式	417
18.3	创建网页：学习笔记主页	369	20.2	部署“学习笔记”	419
18.3.1	映射 URL	369	20.2.1	建立 Heroku 账户	420
18.3.2	编写视图	371			
18.3.3	编写模板	372			
18.4	创建其他网页	373			
18.4.1	模板继承	373			
18.4.2	显示所有主题的面	375			
18.4.3	显示特定主题的面	378			



20.2.2	安装 Heroku Toolbelt	420	20.2.14	改进 Heroku 部署	428
20.2.3	安装必要的包	420	20.2.15	确保项目的安全	429
20.2.4	创建包含包列表的文件 requirements.txt	421	20.2.16	提交并推送修改	430
20.2.5	指定 Python 版本	422	20.2.17	创建自定义错误页面	431
20.2.6	为部署到 Heroku 而修改 settings.py	422	20.2.18	继续开发	434
20.2.7	创建启动进程的 Procfile	423	20.2.19	设置 SECRET_KEY	434
20.2.8	为部署到 Heroku 而修改 wsgi.py	423	20.2.20	将项目从 Heroku 删除	434
20.2.9	创建用于存储静态文件的 目录	424	20.3	小结	435
20.2.10	在本地使用 gunicorn 服务器	424	附录 A	安装 Python	436
20.2.11	使用 Git 跟踪项目文件	425	附录 B	文本编辑器	441
20.2.12	推送到 Heroku	426	附录 C	寻求帮助	447
20.2.13	在 Heroku 上建立数据库	427	附录 D	使用 Git 进行版本控制	451
			后记		460