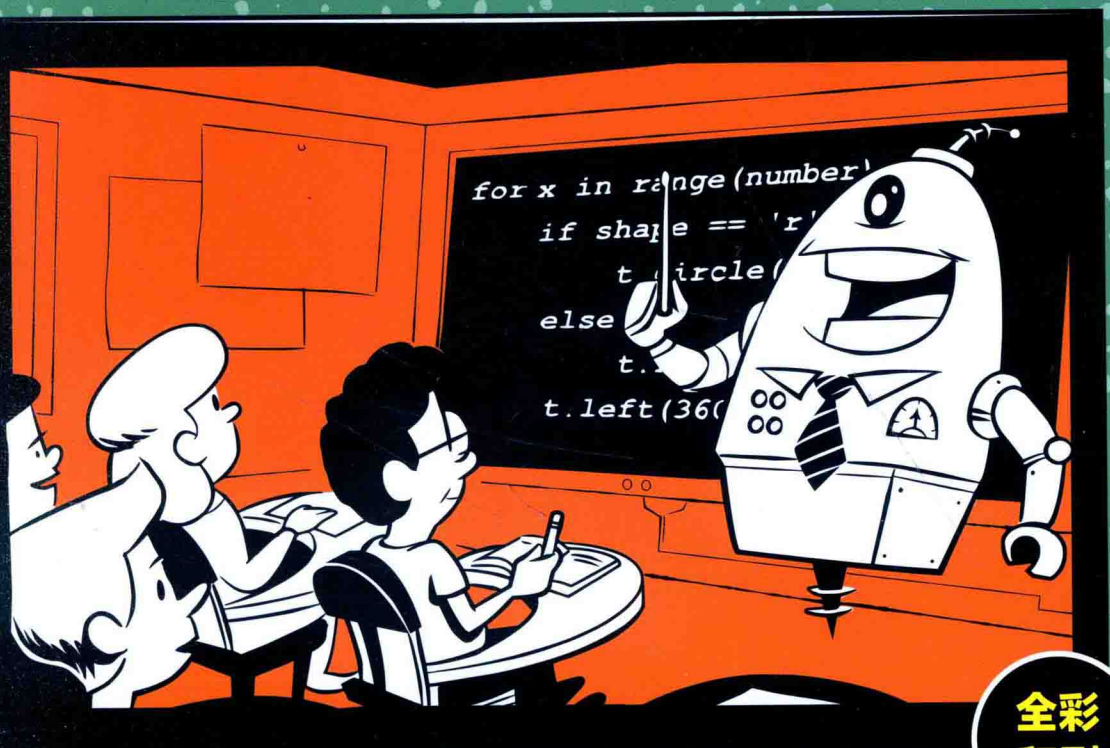




# 教孩子学编程 (Python语言版)

TEACH YOUR KIDS TO CODE

[美] Bryson Payne 著 李军 译

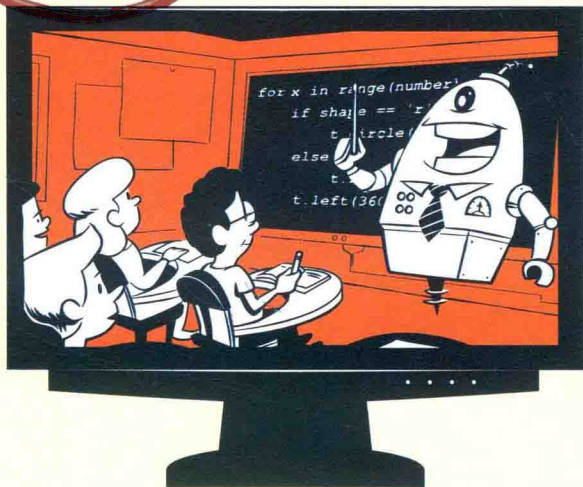


全彩  
印刷

# 教孩子学编程 (Python语言版)

TEACH YOUR KIDS TO CODE

[美] Bryson Payne 著 李军 译



人民邮电出版社  
北京

## 图书在版编目 (C I P) 数据

教孩子学编程 : Python语言版 / (美) 佩恩  
(Payne, B.) 著 ; 李军译. — 北京 : 人民邮电出版社,  
2016.3  
ISBN 978-7-115-41634-6

I. ①教… II. ①佩… ②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(2016)第029201号

## 版权声明

Simplified Chinese-language edition copyright © 2016 by Posts and Telecom Press.  
Copyright © 2015 by Bryson Payne. Title of English-language original: Teach Your Kids to Code, ISBN-13:  
978-1-59327-614-0, published by No Starch Press.

All right resened.

本书中文简体字版由美国 No Starch 出版社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。

## 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。这门强大的语言如今在  
大学和 Google、IBM 等大型技术公司广泛使用。

本书是一本父母和老师教孩子使用 Python 进行基础程序设计和解决问题的入门图书。本书通过  
科学合理的结构、通俗易懂的文字、活泼有趣的图示,帮助孩子学习计算机的思维方式,而可视化  
和游戏为主的例子则持续吸引读者的注意力。针对变量、循环、函数等编程基础概念的介绍,可以  
帮助最年轻的程序员构建所需的技能,以制作自己的超酷的游戏和应用。每章末尾的编程挑战,则  
可以拓展读者地思维,巩固所学习的知识和技能。

本书适合任何想要通过 Python 学习编程的读者,尤其适合父母、老师、学生,以及想要理解计  
算机编程基础知识的未成年人阅读学习。

- 
- ◆ 著 [美] Bryson Payne
  - 译 李 军
  - 责任编辑 陈冀康
  - 责任印制 张佳莹 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>
  - 北京缤索印刷有限公司印刷
  - ◆ 开本: 720×960 1/16
  - 印张: 16.75
  - 字数: 302 千字 2016 年 3 月第 1 版
  - 印数: 1-3 500 册 2016 年 3 月北京第 1 次印刷
  - 著作权合同登记号 图字: 01-2015-7472 号
- 

定价: 59.00 元

读者服务热线: (010)81055339 印装质量热线: (010)81055316

反盗版热线: (010)81055315

## 作者简介

Bryson Payne 博士是北乔治亚大学计算机科学系的一位终身教授。他在大学负责教授有志成为程序员的学生长达 15 年的时间。他的学生在 Blizzard Entertainment、Riot Games、Equifax、CareerBuilder 等众多公司，都开启了成功的职业生涯。他是北乔治亚大学的计算机科学的首任系主任，并且他从乔治亚州立大学获得了计算机科学博士学位。此外，他持续与 K-12 学校合作，以推动科技教育工作。

Payne 博士从事编程工作长达 30 多年。他销售的第一个程序在 1985 年 RUN 杂志（Commodore 64）的“Magic”专栏上，售价为 10 美元。

Payne 和妻子 Bev，以及两个儿子 Alex 和 Max，居住在乔治亚州的亚特兰大。

## 插图者简介

Miran Lipovac 是 Learn You a Haskell for Great Good! 一书的作者。他喜欢拳击、演奏低音吉他，当然，还喜欢画画。他痴迷于骷髅舞和数字 71。当经过自动门的时候，他总是假装实际上是用自己的意念打开了门。

## 技术评阅者简介

Ari Lacenski 是一位 Android 应用程序和 Python 软件开发者。她居住在旧金山。她在 <http://gradlewhy.ghost.io/> 撰写的有关 Android 编程的文章，担任 Women Who Code 的导师，并且用吉他演奏关于太空海盗的歌曲。



## 对本书的赞誉

本书内容清晰、配图很吸引人，并且App也很惊人。这是父母和孩子一起学习的编程指南。

——Aaron Walker，NASA 网络安全专家

在本书中，作者投入了自己的精力和兴奋，使用彩色的、吸引人的游戏和图形，帮助读者掌握现实的技能。

——Bindy Auvermann，Next Generation Youth Debelopment 公司执行总监  
易于学习的5星好书，帮助读者打下坚实的基础以进一步阅读高级编程图书。

——James Floyd Kelly，GeekDad

为一种美妙的、未来会快速改变世界的技术，提供了构建基础。

——JoAnne Taylor，IBM Global Telecommunications 前副总裁

本书中的概念能够帮助任何年轻人扩展自己的大学视野和职业机会。

——Dr. Raj Sunderraman，佐治亚州立大学计算机系主任  
每个孩子和每一个父母都应该阅读本书。

——James E. Daniel, Jr.，App Studios 公司创始人

富有新意的、激动人心的学习指南。构建的技能令人受益终生。

——Dr. Steven Burrell，佐治亚南方大学信息技术副总裁和CIO  
我在孩童时代就想要拥有的那种书。

——Scott Hand，CareerBuilder 软件工程师

作者是一位计算机科学家和教授，通过本书，他把计算机的能力以年幼的读者和较大的读者易于理解的方式进行讲解。

——Dr. Antonio Sanz Montemayor，西班牙 Rey Juan Carlos 大学信息学教授  
引人入胜、富有想象的App和宝贵的终身技能的美妙组合。

——Ted Cunningham，The Power of Home 的作者  
本书及其引入的逻辑思考帮助构建了下一代的技术领导力。

——N. Dean Meyer，作者和执行教练  
本书可以让你的孩子在高科技的世界里赢在起跑线上。

——Ken Coleman，The Ken Coleman Show 前电台主持人和领导力作者

作者让我们上路并引领我们走向梦想的职业。本书中，他为父母和教师提供了机会培育下一代的创新和问题解决者。

——Shah 和 Susan Rahman, Riot Games

作者帮助人们提高自己的技术水平。他的书也起到了这样的作用。

——Ash Mady, RedHat 公司技术经理

对于父母和孩子来说，同样有趣和好读。

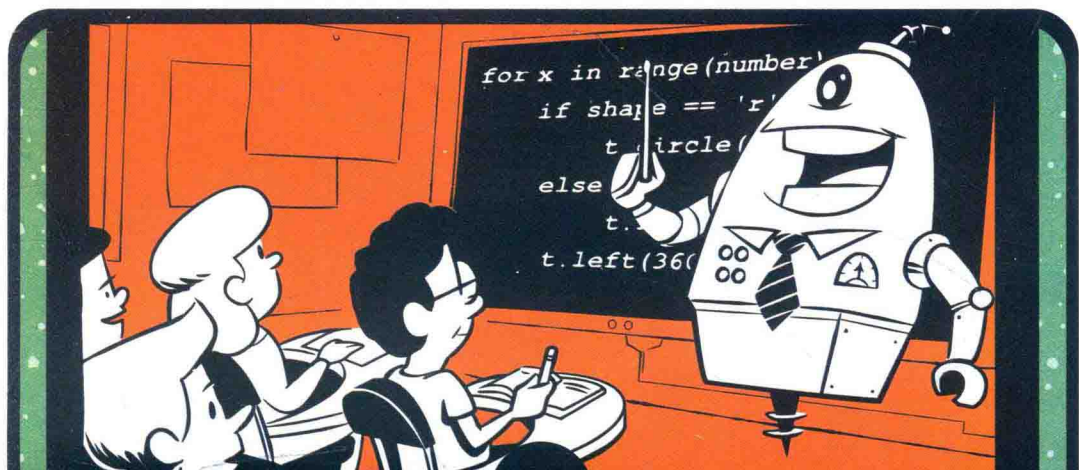
——Steve McLeod, 北佐治亚大学副 CIO

本书很直白，你可以很容易地把本书递给小学高年级的学生或更大的孩子，并让他们自学。比我的树更加适合暑假培训。

——Mel Ford, BlogHer

配图令人印象深刻，游戏很有趣，并且讲解很清晰而有指导性。

——Sandra Henry-Stocker, ITworld



# 前言

## 什么是编程，为什么编程很适合孩子

计算机编程是每个孩子都应该学习的一项重要技能。我们使用计算机解决问题，玩游戏，帮助我们更有效地工作，执行重复性的任务，存储和查找信息，创建新的内容，同时与我们的朋友和世界联系。理解如何编写代码，将会把这一切力量付诸于我们的指尖。

每个人都能够学习编程，这就像是求解一个谜题或一个谜语。你可以应用逻辑，尝试一种解决方案，更多地试验一下，然后解决问题。开始学习编程的时机就是现在！我们处在一个前所未有的历史时期，在此之前，人们不可能像我们今天一样，通过计算机每天都和另一个人联系。我们生活在一个充满了很多新的可能性的世界，从电动汽车和机器人保姆，到甚至能快递包裹和比萨饼的无人机。

如果你的孩子今天开始学习编程，他们能够帮助定义这个快速改变的世界。

## 孩子为什么应该学习编程

学习计算机编程有很多很好的理由，但是，我认为最重要的有以下两点：

- 编程很有趣；
- 编程是一种宝贵的工作技能。

## 编程很有趣

技术正在成为日常生活的一部分。每一家公司、慈善组织和事业都能够从技术中获益。还有一些 App 可以帮助你购买、转赠、加入、玩乐、充当志愿者、联系和分享，甚至做你能够想象到的任何事情。

你的孩子是否想要构建他们自己喜欢的电子游戏的关卡？编程可以做到！创建他们自己的手机应用怎么样？他们可以通过在自己的计算机上编程，把想法带到生活中。他们曾经见过的每一个程序、游戏、系统或者 App，都可以使用他们在本书中学习的、相同的编程构建模块来编码。当孩子编程的时候，他们在技术中扮演主角，不仅能享受乐趣，而且会创造乐趣。

## 编程是一项宝贵的工作技能

编程是 21 世纪的技能。今天的工作比以往需要更多的问题解决能力，而且越来越多的职业把技术当作不可或缺的一部分。美国劳工统计局预计，在未来的 5 年内，大约会创造出 800 万个技术职位。在《2014–2015 Occupational Outlook Handbook》（2014–2015 就业前景手册）中，70% 的增长最快、不需要硕士或博士学位的职业都分布在计算机科学或信息技术（IT）领域。

## 孩子该从哪里学习编程

本书只是一个开端。还有很多地方可以学习编程，如 Code.org、Codecademy（如图 1 所示）这样的 Web 站点，还有数不尽的其他站点教授各种从基础到高级编程的必备编程语言知识。一旦你和孩子一起学完这本书，他们就可以自己通过 EdX、Udacity 和 Coursera 这样的 Web 站点进一步拓展他们的学习。

“编程俱乐部”是一种和朋友们快乐学习的美妙方式。获得相关领域的大学学位，仍然是为职业做好准备的最好方式，但是，现在即便大学也不是唯



一的选择，你的孩子今天可以就开始构建一份编程简历并且展示他们作为程序员和问题解决者的技能。

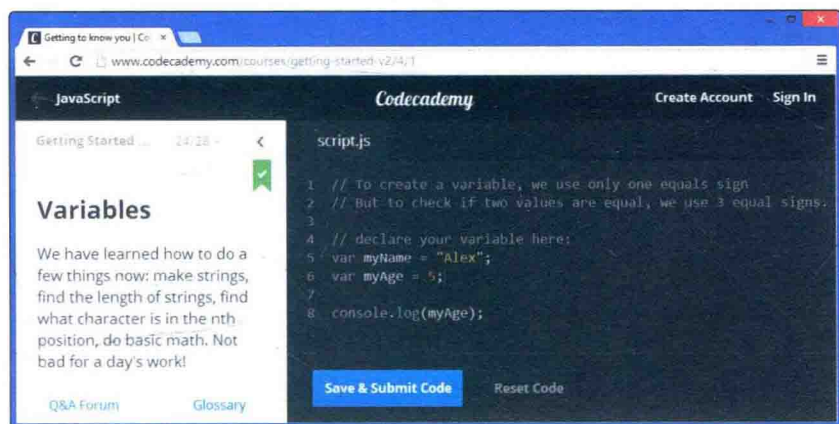


图 1 Codecademy 教你如何使用各种语言一步一步地编程

## 如何使用本书

本书不只是针对孩子的，它也针对父母、老师、学生以及想要理解计算机编程基础知识的成年人，同时针对那些享受乐趣并想在高科技经济中获取一份新的职业的人。不管多大年龄，你都可以把握学习编程基础的好时机。做到这一点的最好的方式，就是体验并操作。

## 探索

如果你想要尝试新事物的话，学习编程会令你兴奋。你和你的孩子可以参照本书中的程序，尝试修改代码中的数字和文本，看看程序会发生什么变化。即便把程序搞坏了，还可以通过修改它而学到一些新的东西。最坏的情况下，不过是重新录入书中的示例，或者打开最近保存的能够工作的版本。学习编程的要点在于，尝试一些新东西、学习一项新技能并且以新的方式解决问题。要确保孩子能够玩得来，他们通过修改一些内容、保存程序、运行程序，看看发生了什么并且修改错误，从而测试自己的代码。

学习编程的要点在于，尝试一些新东西，学习一项新技能并且以新的方式解决问题。通过修改一些内容、保存程序、运行程序，看看发生了什么，并且修改错误，从而测试你自己的代码。

例如，我编写了一些代码来进行彩色的绘制（如图2所示），然后返回，在这里或那里修改一些数字并且尝试再次运行程序。这使得我得到了一幅完全不同但令人惊讶的画。我再次返回去，修改另一些数字并且得到另一幅美丽的、独特的图画。尝试玩玩，看看你能做些什么？

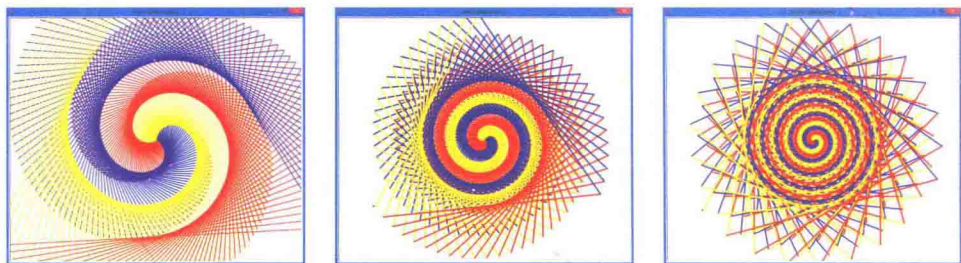


图2 通过在一个程序的一行代码中尝试3个不同的值得到3幅彩色的螺旋线图画

## 一起实践

尝试代码是学习程序如何工作的一种很好的方式，而且，如果你和其他人一起工作的话，甚至会更加有效。不管你是教一个孩子或学生学习，还是自学，没有什么比和别人一起操作代码更有趣了，这甚至会更有效率。

例如，在音乐教育的Suzuki Method中，父母和孩子一起参加课程，甚至比孩子学习得更快一点儿，以便能够在课程中帮助孩子。尽早开始，是Suzuki Method的另一个特征，孩子在3岁或4岁的时候就可以开始正式学习。

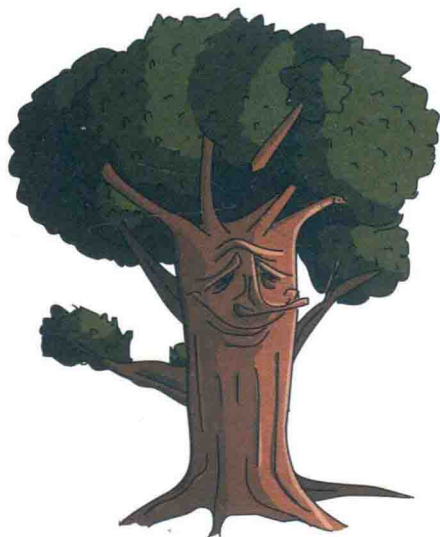
当我的两个儿子两岁和4岁的时候，我开始教他们编程并且鼓励他们通过修改每个程序的较小的部分来获得乐趣，例如颜色、形状以及形状的大小。

在13岁的时候，我通过录入图书中的例子，然后再修改它们做一些新的事情，从而学习编程。现在，在我所教授的计算机科学课程中，我常常给学生一个程序并鼓励它们修改代码来构建一些新的东西。

如果你使用本书自学，可以找一个朋友和你一起研究本书中的例子，或者开始参加一个业余或社区编程俱乐部（参见<http://coderdojo.com/>或<http://www.codecademy.com/afterschool/>），从而可以和其他人一起学。编程也是一项团队运动。

## 在线资源

本书中的所有的程序文件都可以通过<http://www.nostarch.com/teachkids/>获取，包括编程挑战的一些示例解决方案以及其他的信息。下载程序并体验，以便学习更多内容。如果你遇到困难，可以使用示例解决方案，查看它们。



## 编程 = 解决问题

不管你的孩子是两岁还在学习数数，还是22岁了在寻求新的挑战，本书以及它所介绍的概念，都是一项回报丰厚、激励人心的消遣活动，而且能带来更好的职业机会。能够编程并且由此能够快速而有效地解决问题的人，在今天的世界上是宝贝，他们会去做有趣的、有成就感的工作。并非世界上所有的问题都能够用技术来解决，但是，技术能够以以前无法想象的规模和速度来支持交流、协作、了解和行动。如果你能够编程，你就能够解决问题。问题解决者有能力使得世界变得更美好，因此，今天就开始编程吧！

# 目 录

<b>第 1 章 Python 基础——认识环境</b> .....	<b>1</b>
1.1 认识 Python.....	3
1.2 用 Python 编写程序.....	5
1.3 运行 Python 程序.....	5
1.4 本章小结.....	6
1.5 编程挑战.....	7
<b>第 2 章 海龟作图——用 Python 绘图</b> .....	<b>9</b>
2.1 第一个海龟程序.....	10
2.1.1 程序是如何工作的.....	10
2.1.2 发生了什么.....	12
2.2 旋转的海龟.....	13
2.3 海龟画圆.....	14
2.4 添加颜色.....	15
2.4.1 一个四色螺旋线.....	16
2.4.2 修改背景颜色.....	19
2.5 一个变量搞定一切.....	20
2.6 本章小结.....	22
2.7 编程挑战.....	22
<b>第 3 章 数字和变量——用 Python 做数学运算</b> .....	<b>24</b>
3.1 变量——保存内容的地方.....	25
3.2 Python 中的数字和数学运算.....	26
3.2.1 Python 数字.....	27
3.2.2 Python 操作符.....	28
3.2.3 在 Python shell 中进行数学运算.....	28
3.2.4 语法错误——你说什么.....	29
3.2.5 Python shell 中的变量.....	30



3.2.6	用操作符编程——比萨计算器 .....	31
3.3	字符串——Python 中真正的字符 .....	33
3.4	用字符串改进彩色螺旋线 .....	35
3.5	列表——将所有内容放到一起 .....	36
3.6	Python 做作业 .....	38
3.7	本章小结 .....	40
3.8	编程挑战 .....	41
<b>第 4 章</b>	<b>循环很有趣（你可以再说一遍） .....</b>	<b>42</b>
4.1	构建自己的循环 .....	43
4.1.1	使用 for 循环生成四个圆组成的玫瑰花瓣 .....	44
4.1.2	修改循环让玫瑰花瓣带有 6 个圆 .....	46
4.2	把玫瑰花瓣程序改进为允许用户输入 .....	48
4.3	游戏循环和 while 循环 .....	50
4.4	家庭成员螺旋线 .....	52
4.5	整合——病毒式的螺旋线 .....	55
4.6	本章小结 .....	59
4.7	编程挑战 .....	60
<b>第 5 章</b>	<b>条件（如果是这样该怎么办？） .....</b>	<b>62</b>
5.1	if 语句 .....	63
5.2	认识布尔值 .....	65
5.2.1	比较操作符 .....	66
5.2.2	你还不够大！ .....	68
5.3	else 语句 .....	69
5.3.1	多边形或玫瑰花瓣 .....	70
5.3.2	偶数还是奇数 .....	71
5.4	elif 语句 .....	73
5.5	复杂条件——if、and、or 和 not .....	75
5.6	秘密消息 .....	77
5.6.1	打乱字符串 .....	78
5.6.2	字符的值 .....	79
5.6.3	加密 / 解密程序 .....	80

5.7	本章小结 .....	82
5.8	编程挑战 .....	83
<b>第 6 章</b>	<b>随机的乐趣和游戏 ( 继续前进, 抓住机会! ) .....</b>	<b>85</b>
6.1	猜数字游戏 .....	86
6.2	彩色的随机螺旋线 .....	88
6.2.1	选取颜色——任意的颜色 .....	89
6.2.2	获取坐标 .....	90
6.2.3	画布有多大 .....	92
6.2.4	整合 .....	93
6.3	Rock-Paper-Scissors .....	95
6.4	选一张牌——任意一张牌 .....	97
6.4.1	堆牌 .....	97
6.4.2	发牌 .....	98
6.4.3	计算牌面 .....	99
6.4.4	继续前进 .....	101
6.4.5	整合 .....	102
6.5	掷骰子 .....	103
6.5.1	设置游戏 .....	104
6.5.2	对骰子排序 .....	104
6.5.3	测试骰子 .....	105
6.5.4	整合 .....	106
6.6	万花筒 .....	109
6.7	本章小结 .....	112
6.8	编程挑战 .....	114
<b>第 7 章</b>	<b>函数 ( 那些东西有了一个名字 ) .....</b>	<b>116</b>
7.1	用函数整合内容 .....	117
7.1.1	定义 random_spiral() .....	118
7.1.2	调用 random_spiral() .....	119
7.2	参数——传给函数 .....	120
7.2.1	在随机位置微笑 .....	121
7.2.2	整合 .....	125
7.3	返回——发回统计结果 .....	127

7.3.1	从函数返回一个值 .....	127
7.3.2	在程序中使用返回值 .....	128
7.4	交互简介 .....	130
7.4.1	处理事件——TurtleDraw .....	130
7.4.2	监听键盘事件——ArrowDraw .....	133
7.4.3	用参数处理事件——ClickSpiral .....	135
7.4.4	更进一步——ClickandSmile .....	137
7.5	ClickKaleidoscope .....	139
7.5.1	draw_kaleido() 函数 .....	139
7.5.2	draw_spiral() 函数 .....	140
7.5.3	整合 .....	140
7.6	本章小结 .....	142
7.7	编程挑战 .....	143
<b>第 8 章</b>	<b>定时器和动画 .....</b>	<b>144</b>
8.1	获取 Pygame 的所有 GUI .....	145
8.1.1	用 Pygame 画一个点 .....	146
8.1.2	Pygame 中的新内容 .....	148
8.1.3	游戏的部分 .....	149
8.2	时间刚刚好——移动和弹跳 .....	152
8.2.1	移动笑脸 .....	153
8.2.2	用 Clock 类实现笑脸动画 .....	155
8.2.3	将笑脸从墙上弹开 .....	157
8.2.4	在四面墙上弹回笑脸 .....	162
8.3	本章小结 .....	166
8.4	编程挑战 .....	167
<b>第 9 章</b>	<b>用户交互——进入游戏 .....</b>	<b>170</b>
9.1	增加交互——点击和拖动 .....	171
9.1.1	点击点 .....	171
9.1.2	拖动绘制 .....	173
9.2	高级交互——笑脸爆炸 .....	177
9.2.1	笑脸精灵 .....	178
9.2.2	设置精灵 .....	180

9.2.3	更新精灵 .....	182
9.2.4	较大的和较小的笑脸 .....	182
9.2.5	整合 .....	183
9.3	SmileyPop 1.0 版 .....	185
9.3.1	检测碰撞和删除精灵 .....	185
9.3.2	整合 .....	186
9.4	本章小结 .....	188
9.5	编程挑战 .....	189
<b>第 10 章</b>	<b>游戏编程 .....</b>	<b>191</b>
10.1	构建游戏框架——Smiley Pong 1.0 版 .....	192
10.1.1	绘制游戏板和游戏部件 .....	193
10.1.2	记录分数 .....	194
10.1.3	显示得分 .....	199
10.1.4	整合 .....	201
10.2	增加难度并结束游戏——Smiley Pong 2.0 版 .....	202
10.2.1	游戏结束 .....	203
10.2.2	再玩一次 .....	203
10.2.3	更快 .....	204
10.2.4	整合 .....	206
10.3	添加更多的功能——SmileyPop 2.0 版 .....	208
10.3.1	使用 Pygame 添加声音 .....	208
10.3.2	跟踪和记录玩家进度 .....	210
10.3.3	整合 .....	213
10.4	本章小结 .....	215
10.5	编程挑战 .....	216
<b>附录 A</b>	<b>Windows、Mac 和 Linux 下的 Python 安装 .....</b>	<b>218</b>
A.1	Windows 下安装 Python .....	219
A.1.1	下载安装程序 .....	219
A.1.2	运行安装程序 .....	220
A.1.3	尝试 Python .....	223
A.2	Mac 下安装 Python .....	225
A.2.1	下载安装程序 .....	225



A.2.2	运行安装程序 .....	226
A.2.3	尝试 Python .....	229
A.3	Linux 下的 Python 安装 .....	230
<b>附录 B</b>	<b>Windows、Mac 和 Linux 下的 Pygame 安装 .....</b>	<b>233</b>
B.1	在 Windows 下安装 Pygame .....	234
B.2	Mac 下安装 Pygame .....	238
B.3	Linux 下安装 Pygame .....	240
<b>附录 C</b>	<b>构建自己的模块 .....</b>	<b>242</b>
C.1	构建 colorspiral 模块 .....	243
C.1.1	使用 colorspiral 模块 .....	244
C.1.2	重用 colorspiral 模块 .....	245
C.2	附加资料 .....	246
<b>术语表</b>	<b>.....</b>	<b>247</b>