



趣学 Python编程

PYTHON FOR KIDS

[美] Jason Briggs 著 尹哲 译



人民邮电出版社
POSTS & TELECOM PRESS

press

趣学 Python编程

PYTHON FOR KIDS

[美] Jason Briggs 著 尹哲 译



人民邮电出版社
北京

图书在版编目 (CIP) 数据

趣学Python编程 / (美) 布里格斯 (Briggs, J.) 著
; 尹哲译. — 北京: 人民邮电出版社, 2014. 3
ISBN 978-7-115-33595-1

I. ①趣… II. ①布… ②尹… 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265094号

版 权 声 明

Simplified Chinese-language edition copyright © 2013 by Posts and Telecom Press.

Copyright © 2013 by Jason R. Briggs. Title of English-language original: PYTHON FOR KIDS, ISBN-13:
ISBN-13: 978-1-59327-407-8, published by No Starch Press.

All rights reserved.

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。



-
- ◆ 著 [美] Jason Briggs
 - 译 尹 哲
 - 责任编辑 陈冀康
 - 责任印制 程彦红 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>
北京天宇星印刷厂印刷
 - ◆ 开本: 800×1000 1/16
印张: 18.25
字数: 308 千字 2014 年 3 月第 1 版
印数: 1—3 500 册 2014 年 3 月北京第 1 次印刷
- 著作权合同登记号 图字: 01-2013-3673 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

内 容 提 要

Python 是一款解释型、面向对象、动态数据类型的高级程序设计语言。Python 语法简捷而清晰，具有丰富和强大的类库，因而在各种行业中得到广泛的应用。对于初学者来讲，Python 是一款既容易学又相当有用的编程语言，国内外很多大学开设这款语言课程，将 Python 作为一门编程语言学习。

本书是一本轻松、快速掌握 Python 编程的入门读物。全书分为 3 部分，共 18 章。第 1 部分是第 1 章到第 12 章，介绍 Python 编程基础知识，包括 Python 的安装和配置、变量、字符串、列表、元组和字典、条件语句、循环语句函数和模块、类、内建函数和绘图，等等。第 2 部分是第 13 章和第 14 章，介绍如何用 Python 开发实例游戏弹球。第 3 部分包括第 15 章到第 18 章，介绍了火柴人实例游戏的开发过程。

本书语言轻松，通俗易懂，讲解由浅入深，力求将读者阅读和学习的难度降到最低。任何对计算机编程有兴趣的人或者首次接触编程的人，不论孩子还是成人，都可以通过阅读本书来学习 Python 编程。

作者简介

Jason R. Briggs 从 8 岁起就是一名程序员了，那时他在 Radio Shack TRS-80 上学习了 BASIC 语言。他作为开发人员和系统架构师写专业的软件，同时还是《Java 开发者》杂志的撰稿编辑。他的文章曾经上过《JavaWorld》、《OnJava》以及《ONLamp》。这是他写的第一本书。

Jason 的网站在：<http://jasonrbriggs.com/>，他的电子邮箱地址是：mail@jasonrbriggs.com。

关于插图

Miran Lipovaca 是《Learn You a Haskell for Great Good!》的作者。他喜欢拳击，会弹贝斯，而且还会画画。他对于舞动的简笔画小人还有数字 71 很入迷，当他走到自动门的前面时他总是假装是他用意念打开的门。

技术审阅者简介

15 岁的 Josh Pollock 刚从 Nueva School 毕业，他现在是旧金山 Lick-Wilmerding 高中的一年级新生。他第一次编程是在 9 岁时，用的是 Scratch；上 6 年级时用 TI-BASIC；7 年级时改用 Java 和 Python；8 年级时用 UnityScript。除了编程，他喜欢吹号，开发电脑游戏，他还喜欢教别人自然科学知识。

Maria Fernandez 拥有应用语言学的研究生学位，在过去超过 20 年的时间里她一直对计算机和技术感兴趣。她在佐治亚州的地球村项目中教授年轻的难民英语，现在她住在加利福尼亚北部，在 ETS（教育考试服务中心）工作。

致谢

这就像当你上台领奖时才发现，你把那张列着要感谢的人的列表落在另一条裤子里了——你肯定会漏掉一些人，但音乐已经响起，你马上就得上台了。

那么也就是说，下面是我需要感激的人的不完全列表，是他们帮助我把这本书变成现在这么好。

感谢 No Starch 团队，尤其是 Bill Pollock，在编辑本书时大量地考虑了“孩子们会怎么想”。如果你写程序写久了，很容易忘记其中有些东西对于初学者来讲有多难，Bill 对于那些被忽略的还有太复杂的部分提出了宝贵的意见。还要感谢出色的产品经理 Serena Yang，希望给 300 多页书中的代码纠正颜色没让你扯掉太多头发。

我必须要对 Miran Lipovaca 说“非常谢谢”，插图棒极了。太棒了。如果是我自己画这些画，那可就糟了，幸运的话也许偶尔能画出四不像的东西来。它是个熊么？是只狗？不，等一等，是棵树吧？

感谢校阅者们。很抱歉你们的有些建议最后也没能实现。可能你是对的，只能怪我自己一根筋。尤其要感谢 Josh 的一些好建议和非常好的发现。还要向 Maria 道歉，我偶尔会把格式不好的代码发给她。

感谢我的妻子和女儿，我在这段时间花在计算机屏幕前的时间比平时还多，她们包容了我。

感谢我妈妈多年来不断地鼓励我。

最后，要谢谢我爸爸在 20 世纪 70 年代就给我买了一台计算机，并且让我想用就可以随时用。没有他，这些都不可能存在。

译者序

编写计算机程序是未来最重要的生存技能。

计算机软件在我们的生活中扮演的角色越来越重要，然而编写程序的技术却越来越难以掌握了。大胆地假想一下：在未来，社会的阶层划分也许不再是分为权贵与平民、富人与穷人、无产阶级与资产阶级，而是分为懂技术的人和不懂技术的人。

现代人的生活越来越离不开网络、手机，还有无处不在各式各样的智能设备，就连很多简单的电灯里都有一个小小的电脑芯片。赋予这些计算机硬件以生命的就是人们设计的软件。软件让火星上的漫游机器人自主地去探索，软件让朋友们在社区网络上分享快乐，那些让你欲罢不能的电脑游戏也是软件。

软件在半个多世纪的时间里，在人类的各种活动中所占的比例越来越高。新兴的人类活动，如电脑游戏、社区网络等自然不用说。电话在刚刚发明时当中本没有任何的软件，而现在从你的手机到电信的网络，全部都是以软件为主体构成的。就算是在传统行业中软件的比重也越来越高了，不妨看看你周围的银行、快递、商场、书店等等这些年来的变化就知道了。

然而使用软件呢？

同意现在的软件产品变得越来越容易使用的请举手！

我相信真的会举手的大多是孩子们。因为孩子们从来都喜欢新事物。绝大多数成年人都不会觉得作为电话来讲智能手机比从前容易用了。

那么制作软件呢？

我是在 20 世纪 80 年代，还在上小学的时候开始学习计算机编程的。那时的苹果电脑开机直接进入 BASIC 语言界面。那时的编程语言就是做事情的流水帐，虽然连 26 个英文字母都认不准，可也能很快学会做好多事情，甚至没过多久就写出了自己的第一个电脑游戏。

而在翻译这本书的过程中我发现，现在的孩子要学习计算机编程要复杂得多。你起码要先从 Windows，苹果 OS X 或者 Linux (Ubuntu) 中选择一个操作系统，学习如何使用它。然后，如果你要使用 Python 语言的话一般还要先下载和安装。这是因为 Windows 中缺省没有安装 Python，而苹果电脑和 Linux 上虽然有却不是最新版本。在每种操作系统上安装的方式都不一样。装好了之后你还要知道如何打开，运行，然后还得选择一种编辑代码的工具。你在代码里创建了一个新的绘图窗口然后运行……“它在哪儿呢？”原来运行窗口藏在后面呢，你得用鼠标点它一下才能和它交互。“天啊！你不说我怎么知道！”代码也不再像是流水帐那么简单了，模块、类、对象，还好原作者放过了异常处理和生成器。

现在你知道学习软件开发有多难了吧。然而你看到的只是冰山的一角，它只会变得越来越有挑战。这也是为什么学会写程序将会是一个很大的生存优势。

可是为什么在软件使得人们越来越强大，能做很多从前不能做的事情的同时，制作软件却变得越来越难了呢？

半个多世纪以来，人们不断地发明着新的工具和方法把设计计算机软件变得更方便，能力更强大。与此同时，这些便利与强大的副产品是总有一些细节被有意无意地暴露在外面。这样的细节越来越多，学习的成本就越来越高。这种情况很像现在人人都需要有个懂电脑的朋友。总有些技术细节无法避免，并且它们的作用会越来越大，那么掌握它们就是未来最重要的生存技能。

那么为什么要从 Python 学起呢？

Python 的基础部分很简单，代码组织也很直观，是初学编程的好选择。另外它也是一门很流行并且很有前途的语言。不要小看它，Python 并不是专门的教学语言，他非常的强大。

有的朋友可能要问：“都什么年代了，编写程序还要写文本，就不能用鼠标画一画点一点把程序生成出来么？”

编写程序在过去很多年一直是采用写程序文本的形式（很早以前是在纸带上打孔），恐怕将来很多年也将还是像 Python 这样写程序文本，似乎这是目前唯一一种能让程序员自由灵活地表达意图的方式。很多人尝试过很多其它方式。比方说画出概念图来让计算机自己生成程序，这些尝试都不太成功。还有一些“可视化编程”工具和语言，比方说有一种教小孩子编

3 译者序

程的 Scratch 语言。可这些语言能做的事情都有限。

可能你在想：未来计算机越来越智能，还需要人来写程序么？

计算机是人的工具，只有人才知道他自己想让计算机做什么。就算是未来计算机强大到可以有自己的意志，那么人类掌握软件能力以防被计算机统治不是更重要么？

我从小上学的时候开始在长春市少年宫学习计算机，二十多年过去了，做个程序员的理想没有再变过。为此我要感谢当年话剧团的韩老师，是她把我带入了少年宫（我刚去少年宫时是搞文艺的），也要感谢长春市少年宫给我童年（直到初二）带来的永生难忘的快乐和知识。作为一个来自普通家庭的孩子，我不知道有多幸运。前不久听说少年宫当年教孩子计算机的一位邱老师刚刚过逝。这虽然不是我自己写的书，但我希望把这本译作献给那些在 80 年代万物复苏时教给我们这些 70 后孩子课堂以外知识的教育工作者们。让我们把薪火传递下去。

尹哲 2013 年 12 月

前 言

为什么要学习计算机编程

编程会培养创造能力、逻辑能力和解决问题的能力。编写程序的人有机会从无到有创造新事物，使用逻辑来把程序变成计算机可以运行的程序。在出了问题的时候你需要用解决问题的能力来找出是哪里不对。编程是一项既有趣，有时候又充满挑战的事情。从中学到的技巧对于学校和工作都很有用。就算你的职业方向和计算机没有关系也是这样。

除此之外，编程起码是外面天气不好的下午打发时间的好主意。

为什么是 Python

对于初学者来讲，Python 是一款既容易学又相当有用的编程语言。相对于其他语言，它的代码相当易读，并且它有 Shell 程序让你可以输入并运行程序。Python 的一些功能对于辅助学习过程很有效，让你可以把一些简单的动画放在一起制作自己的游戏。其中之一是 turtle 模块，灵感来自于海龟作图（20 世纪 60 年代由 Logo 语言使用），专门用作教育目的。还有 tkinter 模块，它是 Tk 图形界面的接口，可以简单地创建稍微复杂一点的图形和动画程序。

怎样学习写代码

正如你首次尝试任何事情一样，最好从最基本的地方开始，所以要从第 1 章开始，别急着跳到后面的章节。谁也不能刚拿起一件乐器就马上能演奏交响乐。飞行员学员不会在掌握基本控制之前就去开飞机。体操运动员（一般来讲）不会第一次尝试就能翻跟头。如果你向前跳得太快，不但基础知识学得不牢，后面的章节也会让你觉得很复杂。

2 前言

在阅读本书的过程中要自己动手试一试给出的那些例子。大多数章节后还有一些编程练习供你尝试，它们能帮你提高编程技巧。要记住，你对基础理解得越好，以后你理解复杂问题时越轻松。

当你受到挫折或者面临太大的挑战时，下面是一些我觉得有用的东西。

1. 把大问题拆成小问题。尝试理解一小段代码是做什么的，或者只考虑困难问题的一小部分（只关注于一小段代码而不是尝试一下子整个理解）。
2. 如果这样还不行，有时候不妨把它放到一边一段时间。先不去理它，过几天再回来。这对解决很多问题都很有效，尤其对于程序员来讲。

这本书是写给谁看的

这本书写给任何对计算机编程有兴趣的人并且首次接触编程的人，不论小孩还是大人。如果想学习如何自己写软件，而不只是使用别人开发的程序，那么这本书将是个好的开始。

接下来的章节会帮助你安装 Python，开启 PythonShell 程序以及执行简单计算，在屏幕上打印文本还有创建列表，用 if 语句和 for 循环执行简单的过程控制操作（还有 if 语句和 for 循环是什么）。你还会学到如何用函数来重用代码，基本的类和对象的知识，还有众多的 Python 内建函数及模块的介绍。

有不同章节分别介绍简单和高级海龟作图，还有用 tkinter 模块在计算机屏幕上画图。在很多章节的后面都有不同难度的编程练习题，这些练习让读者自己动手写小程序，以此来巩固刚刚学到的知识。

当你打好编程知识的基础后，你会学习如何写你自己的程序。你将开发两个图形游戏并学习冲突检测、事件，还有各种动画技术。

本书中大多数例子是用 Python 的 IDLE 程序做的。IDLE 提供了语法高亮、复制粘贴功能（和其他应用程序相似），它还有一个编辑器窗口让你可以保存代码以后再行，也就是说 IDLE 既是一个做试验的交互环境又有点像一个文本编辑器。这些例子在标准控制台和普通的文本

编辑器上都同样适用，但是 IDLE 提供的语法高亮和还算友好的环境可以帮你更好地理解。所以最前面的章节会教你如何使用它。

本书的内容

下面是每章内容的简单介绍。

第 1 章是安装 Python 的操作指南。

第 2 章介绍基本的计算和变量。

第 3 章介绍一些基本的 Python 类型，如字符串、列表和元组等。

第 4 章初次接触 turtle（海龟）模块。我们从基本的编程转移到让海龟（一个看上去像箭头的形状）在屏幕上移动。

第 5 章涵盖了条件的变化以及 if 语句。

第 6 章接着讲了 for 循环和 while 循环。

从第 7 章开始，我们学会了使用和创建函数。然后在第 8 章我们讲了类和对象。我们讲到了足够让我们在本书的后面章节中开发电脑游戏所需的基本概念和编程技术。从这时开始，书中的内容开始有点复杂了。

第 9 章介绍了 Python 中大多数的内建函数。第 10 章继而介绍了 Python 默认安装的几个模块（模块基本上就是一些有用的功能的集合）。

第 11 章再回到 turtle 模块，让读者用到更复杂的形状。第 12 章使用 tkinter 模块来创建更高级的图形。

在第 13 章和第 14 章，我们创造了第一个游戏《弹球》，它是用我们在前面章节中学到的知识创造出来的。在第 15 到 18 章，我们创造了另一个游戏《火柴人逃生》。这些游戏开发章节中你可能会遇到很棘手的问题。如果实在解决不了的话，可以从本书的网站 <http://python-for-kids.com> 上下载代码，把你的代码和示例代码比较一下。

在“结束语”部分，我们参考了 PyGame 模块还有其他一些流行的编程语言。

4 前言

最后，在附录中，你会了解到 Python 关键字的细节。在术语表中，你会找到本书中用到的编程术语的定义。

本书的网站

在你读书的时候如果需要帮助，可以使用本书的网站 <http://python-for-kids.com/>。在上面你可以下载书中的所有例子，还有更多的编程练习。在网站上你还可以找到书中所有编程练习的答案，如果你做不出来，或者想检查你做的结果，可以参考。

祝你编程开心！

请记住，学习本书进行编程是件让人开心的事。不要把它当成一项任务。要把编程当做是在创建有趣的游戏或者应用来和朋友还有其他人分享。

学习编程是一种很好的思维训练，效果也非常好。但更重要的是，不论你做什么，一定要开心！

目 录

第 1 部分 学习编程

第 1 章 Python 不是大蟒蛇.....	3
1.1 关于计算机语言	3
1.2 安装 Python.....	4
1.2.1 在 Windows 7 上安装 Python	4
1.2.2 在苹果 OS X 上安装 Python.....	6
1.2.3 在 Ubuntu 上安装 Python.....	8
1.3 当你安装好 Python 以后.....	9
1.4 保存 Python 程序.....	10
1.5 你学到了什么	12
第 2 章 计算与变量	13
2.1 用 Python 来做计算.....	13
2.1.1 Python 的运算符.....	14
2.1.2 运算的顺序	15
2.2 变量就像是标签	16
2.3 使用变量	17
2.4 你学到了什么	20
第 3 章 字符串、列表、元组和字典.....	21
3.1 字符串	21

2 目录

3.1.1	创建字符串	21
3.1.2	处理字符串相关的问题	23
3.1.3	在字符串里嵌入值	25
3.1.4	字符串乘法	26
3.2	列表比字符串还强大	27
3.2.1	添加元素到列表	29
3.2.2	从列表中删除元素	30
3.2.3	列表上的算术	31
3.3	元组	32
3.4	Python 里的 map 不是用来指路的	33
3.5	你学到了什么	35
3.6	编程小测验	36
第 4 章	用海龟画图	37
4.1	使用 Python 的 turtle (海龟) 模块	37
4.1.1	创建画布	38
4.1.2	移动海龟	39
4.2	你学到了什么	43
4.3	编程小测验	43
第 5 章	用 if 和 else 来提问	45
5.1	if 语句	45
5.2	语句块就是一组程序语句	45
5.3	条件语句帮助我们做比较	48
5.4	if-then-else 语句	49
5.5	if 和 elif 语句	50
5.6	组合条件	51