Think Python

像计算机科学家一样思考Python

[美] *Allen B. Downey* 著
赵普明 译

O'REILLY®
人民邮电出版社
POSTS & TELECOM PRESS

像计算机科学家一样思考 Python

[美] *Allen B. Downey* 著

赵普明 译

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

O'Reilly Media, Inc.授权人民邮电出版社出版

人民邮电出版社

北京

图书在版编目（CIP）数据

像计算机科学家一样思考Python / （美）唐尼(Downey, A. B.）著 ；赵普明译. -- 北京 ：人民邮电出版社，2013.8
书名原文：Think python:how to think like a computer scientist
ISBN 978-7-115-32092-6

Ⅰ. ①像… Ⅱ. ①唐… ②赵… 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114574号

版权声明

◆ 著 [美] Allen B. Downey
译 赵普明
责任编辑 杨海玲
责任印制 程彦红 杨林杰

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
三河市海波印务有限公司印刷

◆ 开本：787×1000 1/16
印张： 16.25
字数： 327千字 2013年8月第1版
印数：1-3 000册 2013年8月河北第1次印刷

著作权合同登记号 图字：01-2012-7991号

定价：49.00元

读者服务热线：(010)67132692 印装质量热线：(010)67129223
反盗版热线：(010)67171154

内容提要

本书按照培养读者像计算机科学家一样的思维方式的思路来教授Python语言编程。全书贯穿的主体是如何思考、设计、开发的方法，而具体的编程语言，只是提供一个具体场景方便介绍的媒介。它并不是一本介绍语言的书，而是一本介绍编程思想的书。和其他编程设计语言书籍不同，它不拘泥于语言细节，而是尝试从初学者的角度出发，用生动的示例和丰富的练习来引导读者渐入佳境。

作者从最基本的编程概念开始讲起，包括语言的语法和语义，而且每个编程概念都有清晰的定义，引领读者循序渐进地学习变量、表达式、语句、函数和数据结构。此外，书中还探讨了如何处理文件和数据库，如何理解对象、方法和面向对象编程，如何使用调试技巧来修正语法、运行时和语义错误。每一章都配有术语表和练习题，方便读者巩固所学的知识和技巧。此外，每一章都抽出一节来讲解如何调试程序。作者针对每章中所专注的语言特性，或者相关的开发问题，总结了调试的方方面面。可以说这是一种非常有益的创新，让初学编程的读者少走很多弯路。

全书共19章和3个附录，详细介绍了Python语言编程的方方面面。这是一本实用的学习指南，适合没有Python编程经验的程序员阅读，也适合高中或大学的学生、Python爱好者及需要了解编程基础的人阅读。对于第一次接触程序设计的人来说，是一本不可多得的佳作。

O'Reilly Media, Inc.介绍

O'Reilly Media通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自1978年开始，O'Reilly一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了Make杂志，从而成为DIY革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版，在线服务或者面授课程，每一项O'Reilly的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

业界评论

“O'Reilly Radar博客有口皆碑。”

——Wired

“O'Reilly凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——Business 2.0

“O'Reilly Conference是聚集关键思想领袖的绝对典范。”

——CRN

“一本O'Reilly的书就代表一个有用、有前途、需要学习的主题。”

——Irish Times

“Tim是位特立独行的商人，他不光放眼于最长远、最广阔的视野并且切实地按照Yogi Berra的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去Tim似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

前言

本书的奇特历史

1999 年，我正在为一门 Java 的编程入门课程备课。这门课我已经教过 3 个学期，感到有些灰心。课程的不及格率太高，即使是那些及格的学生，也只获得了很低的成就。

我发现问题之一是教材。它们太厚，有太多冗余的细节，而针对编程技巧的高阶的指导却很不足。并且学生们都受着“陷阱效应”的苦恼：开头时很容易，也能循序渐进，但接着在第 5 章左右，整个地板就突然陷落了。新资讯来得太多、来得太快，以至于我必须花费一学期剩下的全部时间来帮助他们拾回丢失的片段。

开课前两周，我决定自己来编写教材。我的目标有以下几个。

- 尽量简短。学生读 10 页书，比不读 50 页书要好。
- 注意词汇。我尝试尽量少用术语，并在第一次使用它们时做好定义。
- 循序渐进。为了避免陷阱效应，我抽出了最困难的课题，并把它们划分成更细的学习步骤。
- 专注于编程，而不是编程语言。我只注意包涵了 Java 的最小的可用子集，而忽略掉其他。

我需要一个标题，所以心血来潮选择了 *How to Think Like a Computer Scientist*。

第一版教材很粗糙，但确实有效。学生们读完课本，懂得了足够的基础知识，以至我甚至可以利用课堂时间和他们一起讨论更难、更有趣的话题，并且（最重要的是）可以让学生们有足够的时间在课堂上做练习。

我将这本书按照 GNU 自由文档许可协议（GNU Free Documentation License）发布，让用户可以复制、修改和分发本书。

接下来发生了最酷的事情。Jeff Elkner，弗吉尼亚州的一位高中老师，使用了我的书，并且将其翻译成 Python 语言的版本。他寄给我他的翻译副本，于是我有了一次很奇特的经历——通过读我自己的书来学习 Python。通过绿茶出版社（Green Tea Press），在 2001 年我出版了第一个 Python 版本。

2003 年，我开始在欧林学院（Olin College）教学，并第一次需要教授 Python 语言。和 Java 的对比非常惊人。学生们困扰更少，学会得更多，从事更有意思的项目，总的来

说得到了更多的乐趣。

结果就产生了本书，并使用了不那么宏伟堂皇的书名：*Think Python*。部分改动如下所述。

- 我在每章的结尾添加了一节关于调试的说明。这些章节描述寻找和避免 bug 的通用技巧，并警示 Python 中容易出错的误区。
- 我增加了更多的练习，小到简短的理解性测试，大到几个实际工程。并且编写了大部分练习的解答。
- 我添加了一系列案例研究——较长的示例，包括练习、解答以及讨论。其中有些是基于 Swampy——我为 Python 课程所写的一套代码。Swampy、代码示例以及它们的解答，可以在 http://thinkpython.com 上找到。
- 我扩展了关于程序开发计划和基础设计模式的讨论。
- 我增加了关于调试、算法分析和使用 Lumpy 画 UML 图的附录。

我希望你喜欢这本书，并希望它至少能提供一点帮助，助你学会像计算机科学家那样编程和思考。

—— Allen B. Downey

Needham，MA

致谢

非常感谢 Jeff Elkner，他将我的 Java 书翻译成 Python，致使我开始这个项目，并向我介绍了 Python 语言，结果成为我最爱的编程语言。

另外感谢 Chris Meyers，他在 *How to Think Like a Computer Scientist* 一书中贡献了好几章。

感谢自由软件基金会（Free Software Foundation）开发了 GNU 自由文档协议，让我和 Jeff 以及 Chris 的合作成为可能。感谢创用 CC（Creative Commons）开发了我们现在使用的协议。

感谢 Lulu，负责 *How to Think Like a Computer Scientist* 的编辑。

感谢所有参与了本书早期版本编写的学生，以及所有（下面列出的）贡献者提供的修订和建议。

贡献者列表

在最近几年中，超过 100 名眼光犀利、思维敏捷的读者给我寄来了建议和修订。他们对这个项目的贡献和热情，对我是极大的帮助。如果你有建议或者修订意见，请发邮件到 feedback@thinkpython.com。如果我根据你的回馈做出了修改，会将你加入到贡献者列表中（除非你要求被隐藏）。

如果你给出错误出现的位置的部分语句，会让我更容易搜索。页码或者章节号码也可以，但并不那么容易处理。谢谢！

- Lloyd Hugh Allen 对 8.4 节提出了修订建议。
- Yvon Boulianne 对第 5 章提出了一个语义错误的修订建议。
- Fred Bremmer 对 2.1 节提出了一个修订建议。
- Jonah Cohen 编写了 Perl 脚本将本书的 LaTeX 源码转换成美丽的 HTML。
- Michael Conlon 提出了第 2 章的一个语法错误，并提出第 1 章的格式改进，并且他开启了对解释器的技术讨论。
- Benoit Girard 寄来一个对 5.6 节的有趣的修订。
- Courtney Gleason 和 Katherine Smith 编写了 `horsebet.py`，在本书的早期版本中作为一个案例研究。他们的程序现在可以在网站上找到。
- Lee Harr 提交了很多修订建议，我们没有空间在这里一一列出，并且他确实应当被列为本书的一位主要编辑。

- James Kaylin 是一名使用本书的学生。他提交了许多修订。
- David Kershaw 修正了 3.10 节中错误的 `catTwice` 函数。
- Eddie Lam 提出了第 1、2、3 章的很多修订建议，他也修正了 Makefie，这样第一次运行时会自动建立索引。他也帮助我们设置了一个版本管理方案。
- Man-Yong Lee 寄来了对 2.4 节中的示例代码的修订。
- David Mayo 指出第 1 章中的单词“unconsciously”需要被修改为“subconsciously”。
- Chris McAloon 寄来了对 3.9 节和 3.10 节的一些修订。
- Matthew J. Moelter 是本书的长期贡献者，提出了很多修订建议。
- Simon Dicon Montford 报告了第 3 章中缺失的函数定义以及几个错别字。他也发现了第 13 章中的 `increment` 函数的错误。
- John Ouzts 修正了第 3 章中“返回值”的定义。
- Kevin Parks 对关于本书如何分布提出了有价值的评论和建议。
- David Pool 发来了第 1 章中术语表中的错别字，以及鼓励的赞美之言。
- Michael Schmitt 寄来了关于文件和异常的章节的修订建议。
- Robin Shaw 指出了 13.1 节中的一个错误，printTime 函数在一个示例中没有定义就使用了。
- Paul Sleigh 在第 7 章中找到一个错误，并发现了 Jonah Cohen 用于生成 HTML 的 Perl 脚本的 bug。
- Craig T. Snydal 在德鲁大学（Drew University）的一门课上试验这个课本，他提出了好几个有价值的建议和修订。
- Ian Thomas 和他的学生们使用这本书作为编程课程的教材。他们第一个尝试使用本书后半部分的章节，并且提出了许多修正和建议。
- Keith Verheyden 发来了第 3 章的一个修正。
- Peter Winstanley 让我们知道了第 3 章的拉丁文中一个长期存在的错误。
- Chris Wrobel 修正了文件 I/O 和异常一章的代码错误。
- Moshe Zadka 对本书有不可估量的贡献。他编写了关于字典的一章的第一版草稿，并在本书的早期阶段持续提供指导。
- Christoph Zwerschke 发来了几个修正和教学法的建议，并解释了 gleich 和 selbe 的区别。
- James Mayer 发送给我们非常多的拼写错误，包括贡献者列表中的两个错误。

- Hayden McAfee 发现了两个示例之间潜在的冲突。
- Angel Arnal 是翻译本书的西班牙语版本的国际团队的一员。他也发现了英文版中的几个错误。
- Tauhidul Hoque 和 Lex Berezhny 创建了第 1 章中的图表，并改进了很多其他图表。
- Dr. Michele Alzetta 发现了第 8 章中的一个错误，并发来了一些有趣的教学法评论，以及关于斐波那契数列和 Old Maid 的建议。
- Andy Mitchell 发现了第 1 章中的一个录入错误，以及第 2 章中一个错误的示例。
- Kalin Harvey 对第 7 章的一个说明提供了建议，并发现了几个录入错误。
- Christopher P. Smith 发现了几个录入错误，并帮助我们更新本书到 Python 2.2。
- David Hutchins 发现了前言中的一个错别字。
- Gregor Lingl 在奥地利维也纳的一个高中教授 Python。他正在翻译本书的德文版，并发现了第 5 章中的几个错误。
- Julie Peters 发现了前言中的一个错别字。
- Florin Oprina 发来一个 `makeTime` 的改进，`printTime` 的一个修正，以及发现的一个重要的录入错误。
- D. J. Webre 对第 3 章的一个说明提出了建议。
- Ken 在第 8、9、11 章中发现了好几个错误。
- Ivo Wever 在第 5 章发现一个录入错误，并对第 3 章中的一个说明提出了建议。
- Curtis Yanko 对第 2 章中的一个描述提出了建议。
- Ben Logan 发来许多发现的录入错误，并发现了翻译 HTML 的问题。
- Jason Armstrong 发现了第 2 章中一个漏掉的词。
- Louis Cordier 发现了第 16 章中有一个代码和文本不一致的地方。
- Brian Cain 在第 2 章和第 3 章中提出了几个描述的改进建议。
- Rob Black 发来了许多修正，包括一些针对 Python 2.2 的修改。
- 巴黎中央理工大学的 Jean-Philippe Rey 发来了一些补丁，包括对 Python 2.2 的更新，以及其他一些细心的改进。
- 乔治华盛顿大学的 Jason Mader 提供了许多有用的建议和改正。
- Jan Gundtofte-Bruun 提醒我们“a error”应改为“an error”。
- Abel David 和 Alexis Dinno 提醒我们“matrix”的复数形式是“matrices”而不是

“matrixes”。这个错误在书中已经存在了多年，但两个同姓的读者同一天报告了它。真的很奇怪。

- Charles Thayer 鼓励我们删除掉一些语句结尾的分号，并建议我们理清“形参”和“实参”的使用。
- Roger Sperberg 指出了第 3 章的一个逻辑错误。
- Sam Bull 指出了第 2 章中一段令人困惑的描述。
- Andrew Cheung 指出了两处“定义前先使用”的错误。
- C.Corey Capel 发现了调试的第三定律中缺失的单词，以及第 4 章的一个录入错误。
- Alessandra 帮助我们理清了一些关于 Turtle 的困惑。
- Wim Champagne 在字典示例中发现一个错误。
- Douglas Wright 在弧度计算中发现了一个除法向下取整的错误。
- Jared Spindor 发现了一处句尾的无用词。
- Lin Peiheng 发来了许多很有用的建议。
- Ray Hagtvedt 发来了两处错误和一处不是那么错的错误。
- Torsten Hübsch 指出 Sawmpy 中的一处不一致。
- Inga Petuhhov 修正了第 14 章中的一个示例。
- Arne Babenhauserheide 发来了几个有用的修正。
- Mark E. Casida 非常善于发现重复的单词。
- Scott Tyler 填上了一个缺失的“that”，并发来了一堆修正。
- Gordon Shephard 发来了几个修正，每个都用单独的邮件。
- Andrew Turner 发现了第 8 章中的一个错误。
- Adam Hobart 修正了一个在弧度计算中除法向下取整的错误。
- Daryl Hammond 和 Sarah Zimmerman 指出我过早提出了 `math.pi`。并且 Zim 发现了一个录入错误。
- George Sass 在调试章节中发现了一个 bug。
- Brian Bingham 建议了练习 11-10。
- Leah Engelbert-Fenton 指出我用 `tuple` 做变量名称，这正违反了我自己的建议。然后他发现了一堆录入错误以及一个“定义前先使用”。
- Joe Funke 发现了一个录入错误。

- Chao-chao Chen 在斐波那契示例中发现了一个不一致处。
- Jeff Paine 知道 space 和 spam 的区别。
- Lubos Pintes 发来一个录入错误。
- Gregg Lind 和 Abigail Heithoff 建议了练习 14-4。
- Max Hailperin 发来了许多修正和建议。Max 是非凡的《具体抽象》（Concrete Abstractions）一书的作者之一。在读完本书之后你可能会想要读那本书。
- Chotipat Pornavalai 在一个错误信息中发现了一个错误。
- Stanislaw Antol 寄来了一个很有用的建议列表。
- Eric Pashman 对第 4 章到第 11 章发来了许多修正。
- Miguel Azevedo 发现了一些录入错误。
- Jianhua Liu 发来了一长列修正。
- Nick King 发现了一个缺失单词。
- Martin Zuther 发来了一长列建议。
- Adam Zimmerman 发现了我举例的一个“实例”中的不一致处，以及其他一些错误。
- Ratnakar Tiwari 建议加一个脚注说明什么是“退化”三角形。
- Anurag Goel 提出了 `is_abecedarian` 的另一个解答，并发来其他一些修正。他还知道如何拼写 Jane Austen。
- Kelli Kratzer 发现了一个录入错误。
- Mark Griffiths 指出了第 3 章中的一个令人困惑的示例。
- Roydan Ongie 发现了我的牛顿方法的一个错误。
- Patryk Wolowiec 帮我解决了一个 HTML 版本的问题。
- Mark Chonofsky 告诉我 Python 3 中的新关键字。
- Russell Coleman 帮我修正了几何错误。
- Wei Huang 发现了几处录入错误。
- Karen Barber 发现了本书中最古老的录入错误。
- Nam Nguyen 发现了一个录入错误，并指出我使用了装饰器模式但并没有用它的名字。
- Stéphane Morin 发来了一些建议和修正。
- Paul Stoop 修改了一个 `uses_only` 中的录入错误。
- Eric Bronner 指出了关于操作符顺序的讨论中的一个困惑之处。

- Alexandros Gezerlis 提交的建议的数量和质量都设置了一个新的标准。我们非常感谢他！
- Gray Thomas 知道哪边是左哪边是右。
- Giovanni Escobar Sosa 发来一长列的修正和建议。
- Alix Etienne 修正了一个 URL。
- Kuang He 发现一个录入错误。
- Daniel Neilson 修正了一个关于操作符顺序的错误。
- Will McGinnis 指出 `polyline` 在两个地方定义的不同。
- Swarup Sahoo 发现了一个缺失的分号。
- Frank Hecker 指出一个练习不细致，并发现了几个坏链接。
- Animesh B 帮助我清理了一个令人困惑的示例。
- Martin Caspersen 发现了两处取整错误。
- Gregor Ulm 发来一些修正和建议。

目录