思考 Python

像计算机科学家一样思考

Version 1.1.22

思考 Python

像计算机科学家一样思考

Version 1.1.22

Allen Downey
Walter Lewis

Green Tea Press
Needham, Massachusetts

Printing history:

2002 四月：第一版像计算机科学家一样思考.

2007 八月：大幅改动，把标题改为像 (Python) 程序员一样思考.

2008 六月：大幅改动，把标题改为思考 Python：像计算机科学家一样思考.

Green Tea Press
9 Washburn Ave
Needham MA 02492

The original form of this book is LaTeX source code. Compiling this LaTeX source has the effect of generating a device-independent representation of a textbook, which can be converted to other formats and printed.

The LaTeX source for this book is available from `http://www.thinkpython.com`

前言

0.1 本书的奇怪历史

1999 年一月份的时候，我准备用 Java 教一门介绍性的编程课。在那之前，我已经教了三次，而且每次我都很失望。这门课的挂课率非常之高，尽管对那些通过的学生来说，整体的水平也是很低的。

我认为问题的根源之一是教科书。教科书太厚了，掺杂着大量不必要的 Java 细节内容，并且没有足够高水平的引导去指导学生如何编程。学生们深陷“陷阱门“: 他们起步很轻松，逐步的学习，突然，大约在第五章的某个位置，困难出现了。学生必须快速的学习大量的新内容。结果，我不得不把剩下的学期花在挑选一些片段来教学。

课程开始的前两周，我决定自力更生 --自己编写书。我的目标是:

- 尽量简短. 对学生来说，阅读十页比阅读无十页要好。
- 注意词汇量。我尽量减少使用术语，而且在使用前必须先定义。
- 逐步学习。为了避免陷阱门，我把最难的部分分解成一系列的小步骤。
- 把重心放在编程，而不是编程语言。我采用最少的有用的 Java 语言的语法，忽略其他的。

我需要一个书名，所以我就临时地把它叫做《像计算机科学家一样思考》

第一版很粗糙，但是很成功。学生们很乐意看它，并且能很好理解我在课堂上讲的难点，趣点和让他们实践的内容 (这个最重要).

接下来发生的事儿极其的有趣。Jeff Elkner, 居住在弗尼亚的高中老师，改变了我的书，把它翻译成了 Python。他给我寄了份他翻译的副本，于是乎我就有了一段不寻常的学习 Python 的经历 --通过阅读我自己的书。

Jeff 和我随后修订了这本书，加入了 Chris Meyers 提供的一个案例学习。在 2001 年，我们共同发布了《像计算机科学家一样思考：Python 编程》, 当然同样是用 GNU 自由文档许可

证。通过 Gree Tea Press, 我出版了这本书，并且开始在亚马逊和大学书店卖纸质书。Gree Tea Press 出版的书可以从这儿获得greenteapress.com

2003 年，我开始在 Olin College 教书。第一次，我开始教 Python。和教授 Java 的情况相反，学生们不再陷入泥潭，学到了更多，参与了很多有趣的项目，越学越带劲。

在过去的五年里，我一直继续完善这本书，改正错误，提过某些例子的质量，加入一些其他的材料，特别是练习。在 2008 年，我开始重写这本书 ---同时，剑桥大学出版社的编辑联系到了我，他想出版本书的下一板。美妙的时刻！

结构就出现了现在的这本书，不过有了一个简洁的名字《思考 Python》。变化有：

- 在每一章末尾加了点调试的部分。这些部分提供了发现和避免 bug 的通用技巧, 也对 Python 的陷阱提出了警告。
- 删除了最后几章关于列表和树实现的内容。虽然，我万分不舍，但是考虑到和本书余下的部分不协调，只能忍痛割爱。
- 增加了一些案例学习 ---提供了练习，答案和相关讨论的大例子。一些东西是基于 Swampy，这是我为了教学而设计的 Python 程序。Swampy，代码实例和部分答案可以从这儿获得thinkpython.com.
- 扩展了关于程序构建计划和基本的设计模式的讨论。
- Python 运用的更加地道。虽然这本书仍然是讨论编程的，而不是 Python 本身，但是现在我不得不承认这本书深受 Python 浸染。

我希望读者们可以享受这本书，也希望帮助你学习程序设计和像计算机科学家一样思考，哪怕是一丁丁点儿。

Allen B. Downey
Needham MA

Allen Downey 是 Olin College 大学计算机科学与技术系的副教授。

声明

首先，也是最重要的，我要感谢 Jeff Elkner, 是他把我的 Java 书翻译成了 Python，也由此开启了这项“工程“，也把我领进了我最爱的编程语言大门。

我也要感谢 Chris Meyers, 他贡献了《像计算机科学家一样思考》的部分内容。

感谢 FSF 制定的 GNU 自由文档许可证，使我和 Jeff 和 Chris 的合作成为可能。

我也要感谢所以使用以前版本的学生和所有的贡献者，他们提供了宝贵的更正和建议。

感谢我的妻子，Lisa 为她在这本书上所花费的努力，还有 Gree Tea Press, 和其他的一切。

贡献者名单

More than 100 sharp-eyed and thoughtful readers have sent in suggestions and corrections over the past few years. Their contributions, and enthusiasm for this project, have been a huge help.

If you have a suggestion or correction, please send email to `feedback@thinkpython.com`. If I make a change based on your feedback, I will add you to the contributor list (unless you ask to be omitted).

If you include at least part of the sentence the error appears in, that makes it easy for me to search. Page and section numbers are fine, too, but not quite as easy to work with. Thanks!

- Lloyd Hugh Allen sent in a correction to Section 8.4.
- Yvon Boulianne sent in a correction of a semantic error in Chapter 5.
- Fred Bremmer submitted a correction in Section 2.1.
- Jonah Cohen wrote the Perl scripts to convert the LaTeX source for this book into beautiful HTML.
- Michael Conlon sent in a grammar correction in Chapter 2 and an improvement in style in Chapter 1, and he initiated discussion on the technical aspects of interpreters.
- Benoit Girard sent in a correction to a humorous mistake in Section 5.6.
- Courtney Gleason and Katherine Smith wrote `horsebet.py`, which was used as a case study in an earlier version of the book. Their program can now be found on the website.
- Lee Harr submitted more corrections than we have room to list here, and indeed he should be listed as one of the principal editors of the text.
- James Kaylin is a student using the text. He has submitted numerous corrections.
- David Kershaw fixed the broken `catTwice` function in Section 3.10.
- Eddie Lam has sent in numerous corrections to Chapters 1, 2, and 3. He also fixed the Makefile so that it creates an index the first time it is run and helped us set up a versioning scheme.
- Man-Yong Lee sent in a correction to the example code in Section 2.4.
- David Mayo pointed out that the word ``unconsciously'' in Chapter 1 needed to be changed to ``subconsciously''.
- Chris McAloon sent in several corrections to Sections 3.9 and 3.10.
- Matthew J. Moelter has been a long-time contributor who sent in numerous corrections and suggestions to the book.
- Simon Dicon Montford reported a missing function definition and several typos in Chapter 3. He also found errors in the `increment` function in Chapter 13.
- John Ouzts corrected the definition of ``return value'' in Chapter 3.
- Kevin Parks sent in valuable comments and suggestions as to how to improve the distribution of the book.
- David Pool sent in a typo in the glossary of Chapter 1, as well as kind words of encouragement.
- Michael Schmitt sent in a correction to the chapter on files and exceptions.
- Robin Shaw pointed out an error in Section 13.1, where the printTime function was used in an example without being defined.
- Paul Sleigh found an error in Chapter 7 and a bug in Jonah Cohen's Perl script that generates HTML from LaTeX.
- Craig T. Snydal is testing the text in a course at Drew University. He has contributed several valuable suggestions and corrections.

- Ian Thomas and his students are using the text in a programming course. They are the first ones to test the chapters in the latter half of the book, and they have made numerous corrections and suggestions.
- Keith Verheyden sent in a correction in Chapter 3.
- Peter Winstanley let us know about a longstanding error in our Latin in Chapter 3.
- Chris Wrobel made corrections to the code in the chapter on file I/O and exceptions.
- Moshe Zadka has made invaluable contributions to this project. In addition to writing the first draft of the chapter on Dictionaries, he provided continual guidance in the early stages of the book.
- Christoph Zwerschke sent several corrections and pedagogic suggestions, and explained the difference between gleich and selbe.
- James Mayer sent us a whole slew of spelling and typographical errors, including two in the contributor list.
- Hayden McAfee caught a potentially confusing inconsistency between two examples.
- Angel Arnal is part of an international team of translators working on the Spanish version of the text. He has also found several errors in the English version.
- Tauhidul Hoque and Lex Berezhny created the illustrations in Chapter 1 and improved many of the other illustrations.
- Dr. Michele Alzetta caught an error in Chapter 8 and sent some interesting pedagogic comments and suggestions about Fibonacci and Old Maid.
- Andy Mitchell caught a typo in Chapter 1 and a broken example in Chapter 2.
- Kalin Harvey suggested a clarification in Chapter 7 and caught some typos.
- Christopher P. Smith caught several typos and helped us update the book for Python 2.2.
- David Hutchins caught a typo in the Foreword.
- Gregor Lingl is teaching Python at a high school in Vienna, Austria. He is working on a German translation of the book, and he caught a couple of bad errors in Chapter 5.
- Julie Peters caught a typo in the Preface.
- Florin Oprina sent in an improvement in `makeTime`, a correction in `printTime`, and a nice typo.
- D. J. Webre suggested a clarification in Chapter 3.
- Ken found a fistful of errors in Chapters 8, 9 and 11.
- Ivo Wever caught a typo in Chapter 5 and suggested a clarification in Chapter 3.
- Curtis Yanko suggested a clarification in Chapter 2.
- Ben Logan sent in a number of typos and problems with translating the book into HTML.
- Jason Armstrong saw the missing word in Chapter 2.
- Louis Cordier noticed a spot in Chapter 16 where the code didn't match the text.
- Brian Cain suggested several clarifications in Chapters 2 and 3.
- Rob Black sent in a passel of corrections, including some changes for Python 2.2.
- Jean-Philippe Rey at Ecole Centrale Paris sent a number of patches, including some updates for Python 2.2 and other thoughtful improvements.
- Jason Mader at George Washington University made a number of useful suggestions and corrections.
- Jan Gundtofte-Bruun reminded us that ``a error'' is an error.
- Abel David and Alexis Dinno reminded us that the plural of ``matrix'' is ``matrices'', not ``matrixes''. This error was in the book for years, but two readers with the same initials reported it on the same day. Weird.

- Charles Thayer encouraged us to get rid of the semi-colons we had put at the ends of some statements and to clean up our use of ``argument'' and ``parameter''.
- Roger Sperberg pointed out a twisted piece of logic in Chapter 3.
- Sam Bull pointed out a confusing paragraph in Chapter 2.
- Andrew Cheung pointed out two instances of ``use before def.''
- C. Corey Capel spotted the missing word in the Third Theorem of Debugging and a typo in Chapter 4.
- Alessandra helped clear up some Turtle confusion.
- Wim Champagne found a brain-o in a dictionary example.
- Douglas Wright pointed out a problem with floor division in `arc`.
- Jared Spindor found some jetsam at the end of a sentence.
- Lin Peiheng sent a number of very helpful suggestions.
- Ray Hagtvedt sent in two errors and a not-quite-error.
- Torsten Hübsch pointed out an inconsistency in Swampy.
- Inga Petuhhov corrected an example in Chapter 14.
- Arne Babenhauserheide sent several helpful corrections.
- Mark E. Casida is is good at spotting repeated words.
- Scott Tyler filled in a that was missing. And then sent in a heap of corrections.
- Gordon Shephard sent in several corrections, all in separate emails.
- Andrew Turner `spot`ted an error in Chapter 8.
- Adam Hobart fixed a problem with floor division in `arc`.
- Daryl Hammond and Sarah Zimmerman pointed out that I served up `math.pi` too early. And Zim spotted a typo.
- George Sass found a bug in a Debugging section.
- Brian Bingham suggested Exercise ??.
- Leah Engelbert-Fenton pointed out that I used `tuple` as a variable name, contrary to my own advice. And then found a bunch of typos and a ``use before def.''
- Joe Funke spotted a typo.
- Chao-chao Chen found an inconsistency in the Fibonacci example.
- Jeff Paine knows the difference between space and spam.
- Lubos Pintes sent in a typo.
- Gregg Lind and Abigail Heithoff suggested Exercise ??.
- Max Hailperin has sent in a number of corrections and suggestions. Max is one of the authors of the extraordinary Concrete Abstractions, which you might want to read when you are done with this book.
- Chotipat Pornavalai found an error in an error message.
- Stanislaw Antol sent a list of very helpful suggestions.
- Eric Pashman sent a number of corrections for Chapters 4--11.
- Miguel Azevedo found some typos.
- Jianhua Liu sent in a long list of corrections.
- Nick King found a missing word.
- Martin Zuther sent a long list of suggestions.

- Adam Zimmerman found an inconsistency in my instance of an ``instance'' and several other errors.
- Ratnakar Tiwari suggested a footnote explaining degenerate triangles.
- Anurag Goel suggested another solution for `is_abecedarian` and sent some additional corrections. And he knows how to spell Jane Austen.
- Kelli Kratzer spotted one of the typos.
- Mark Griffiths pointed out a confusing example in Chapter 3.
- Roydan Ongie found an error in my Newton's method.
- Patryk Wolowiec helped me with a problem in the HTML version.
- Mark Chonofsky told me about a new keyword in Python 3.0.
- Russell Coleman helped me with my geometry.
- Wei Huang spotted several typographical errors.
- Karen Barber spotted the the oldest typo in the book.
- Nam Nguyen found a typo and pointed out that I used the Decorator pattern but didn't mention it by name.
- Stéphane Morin sent in several corrections and suggestions.
- Paul Stoop corrected a typo in `uses_only`.
- Eric Bronner pointed out a confusion in the discussion of the order of operations.
- Alexandros Gezerlis set a new standard for the number and quality of suggestions he submitted. We are deeply grateful!

Contents

前言 v

0.1 本书的奇怪历史 v

1 编程的方式 1

Chapter 1

编程的方式

这本书的的目的是教会大家如何像计算机科学家一样思考。计算机科学用严谨的语言来表明思想, 尤其是计算。像工程师，他们设计，把各个组件装配成系统并且在可选方案中评估出折中方案。像科学家，他们研究复杂系统的性能，做出假定并且测试假设。

对一个计算机科学家来说最重要的是解决问题。解决问题意味着清晰明确的阐述问题，积极思考问题答案，并且清楚正确的表达答案的能力。实践证明：学习如何编程是一种很好的机会来练习解决问题的技巧。这也是为什么把这章叫做“编程的方式“。

一方面，你将学习编程，一个非常有用的技巧。另一方面你将会把编程作为一种科技。随着我们的深入学习，这点会渐渐明晰。