尽享5小时完整视频教程！跟着数十万人的Python导师学Python！

PEARSON

“笨办法”学Python（第3版）

Learn PYTHON the HARD WAY
THIRD EDITION

[美] Zed A. Shaw 著
王巍巍 译

人民邮电出版社
POSTS & TELECOM PRESS

PEARSON

“笨办法”学Python（第3版）

Learn PYTHON the HARD WAY
THIRD EDITION

[美] Zed A. Shaw 著
王巍巍 译

人民邮电出版社
北京

图书在版编目（CIP）数据

“笨办法”学Python : 第3版 / (美) 肖(Shaw, Z. A.) 著 ; 王巍巍译. -- 北京 : 人民邮电出版社, 2014.11
ISBN 978-7-115-35054-1

Ⅰ. ①笨… Ⅱ. ①肖… ②王… 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(2014)第078421号

内 容 提 要

本书是一本 Python 入门书籍，适合对计算机了解不多，没有学过编程，但对编程感兴趣的读者学习使用。这本书以习题的方式引导读者一步一步学习编程，从简单的打印一直讲到完整项目的实现，让初学者从基础的编程技术入手，最终体验到软件开发的基本过程。

本书结构非常简单，共包括 52 个习题，其中 26 个覆盖了输入/输出、变量和函数三个主题，另外 26 个覆盖了一些比较高级的话题，如条件判断、循环、类和对象、代码测试及项目的实现等。每一章的格式基本相同，以代码习题开始，按照说明编写代码，运行并检查结果，然后再做附加练习。

◆ 著 [美] Zed A. Shaw
译 王巍巍
责任编辑 杨海玲
责任印制 彭志环 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
北京鑫正大印刷有限公司印刷

◆ 开本：800×1000 1/16
印张：16.5
字数：373 千字 2014 年 11 月第 1 版
印数：1－3 000 册 2014 年 11 月北京第 1 次印刷

著作权合同登记号 图字：01-2013-8978 号

定价：49.00 元（附光盘）

读者服务热线：(010)81055410 印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京崇工商广字第 0021 号

版权声明

译者序

2010年的某一天，忽然在网上某处看到一个叫 Zed A. Shaw 的人在写一本 Python 入门书，而且这本书和别的入门书不太一样，于是就去了解了一下。虽然当时我也算是入过门了，但温故知新总是不错的。Zed 的书是放在网上用 Git 做版本管理的，于是每更新一章我就跟着看一章。这段时间，除了巩固了一些 Python 知识，还亲眼见识了一本书是怎样写成的，真是收获不少。

看完后最大的一个感想就是："原来入门书可以这么写！"

2011 年年初正好我杂事较少，于是征得 Zed 的同意，也学着他的方式，在网上建了一个项目，目的是将这本书翻译为中文版。经过几位朋友的转发，这个项目也算得到了一定的关注度。断断续续 3 个月后，翻译初稿就算完成了，放在网上供大家随意阅览。也许到现在你还能找到网上流传的版本。

很可能 Zed 也没想到自己的书会获得这么高的关注度，有一次他在网上说："如果我能在每个初学者身上赚一块钱，那我就差不多发财了。"后来 Zed 自费出版了这本书，甚至还开过线上和线下的教学课程，书的内容也在各种反馈的基础上逐步修改和完善。到 2013 年出第 3 版时，章节的深度和主题的覆盖度和当初已经差别很大了。

在这个阶段，我也一直试图让自己的翻译版本和 Zed 的更新版本内容保持一致。直到有一天，Zed 忽然撤下了在线的 Git 仓库，由于我的项目没法和他的同步，翻译也无法进行下去了。经过沟通才知道，原来是大名鼎鼎的 Addison-Wesley 要出版他的书，由于版权问题，不只是他的 Git 仓库，连我的翻译版也要下线。

虽然挺沮丧的，但我还是把刚开工翻译的第 3 版下线了。

正好这段时间人民邮电出版社的编辑联系到我，说是有机会出版这本书，于是我又高兴起来啦。

截至目前，本书已经发行到了第 3 版。第 1 版只是基本的编程入门，第 2 版重写了若干章节，加入了面向对象以及 Web 应用开发等相关的内容，第 3 版中作者根据学生反馈，在各章节添加了"常见问题回答"，并且再次修订了后面的若干重点章节，进一步加强了面向对象编程的部分。

感谢

翻译本书的动力来自于李飞林，如果没有他的鼓励，我很可能就半途而废了。在翻译这本书的过程中收到过大量热心网友的问题反馈和改进建议。人民邮电出版社勤劳而又专业的编辑们审稿和校对让这本书的文字显得不那么业余。在此一并谢谢。

前言

这本书的目的是让你起步编程。虽然书名说是用“Hard Way”（笨办法）学习写程序，但其实并非如此。所谓的“笨办法”指的是本书的教学方式，也就是所谓的“指令式”教学。在这个过程中，我会让你完成一系列习题，而你则通过重复练习来学到技能，这些习题也是专为重复练习而设计的。对于一无所知的初学者来说，在能理解更复杂的话题之前，这种教授方式效果是很好的。你可以在各种场合看到这种教授方式，从武术到音乐不一而足，甚至在学习基本的算术和阅读技能时也会看到这种教学方式。

这本书通过练习和记忆的方式，教你逐渐掌握 Python 的技能，然后由浅入深，让你将这些技能应用到各种问题上。读完本书以后，你将有能力接触更为复杂的编程主题。我喜欢告诉别人，我的这本书能给你一个“编程黑带”，意思就是说，你已经打好了基础，可以真正开始学习编程了。

如果你肯努力，并投入一些时间，学会了这些技能，你将学会如何编写代码。

致谢

首先我要感谢在本书前两版中帮过我的 Angela，没有她的话我有可能就不会费工夫完成这两本书了。她帮我修订了第 1 版草稿，而且在我写书的过程中给了我极大的支持。

我还要感谢 Greg Newman 为前两版提供的封面设计，Brian Shumate 在早期网站设计方面的帮助，以及所有读过前两版并且提出反馈和纠错的读者。

谢谢你们。

笨办法更简单

在这本书的帮助下，你将通过完成下面这些非常简单的事情来学会一门编程语言，这也是每个程序员的必经之路。

1. 从头到尾完成每一个习题。
2. 一字不差地录入每一段程序。
3. 让程序运行起来。

就是这样了。刚开始这对你来说会非常难，但你需要坚持下去。如果你通读了这本书，每

晚花一两个小时做做习题，你可以为自己读下一本编程书籍打下良好的基础。通过这本书，你学到的可能不是真正的“编程”技术，但你会学到学习一门编程语言的基本技能。

这本书的目的是教会你编程新手所需的三种最重要的技能：读和写、注重细节以及发现不同。

读和写

很显然，如果你连打字都成问题的话，那你学习编程也会成问题。尤其是，如果你连程序源代码中的那些奇怪字符都打不出来的话，就更别提编程了。如果没有这些基本技能，你将连最基本的软件工作原理都难以学会。

手动录入代码范例并让它们运行起来的过程，会让你学会各种符号的名称，熟悉它们的用处，最终读懂编程语言。

注重细节

区分好程序员和差程序员的最重要的一个方面就是对于细节的重视程度。事实上这是任何行业区分好坏的标准。如果缺乏对于工作中每一个微小细节的注意，你的工作成果将不可避免地出现各种关键缺陷。从编程这一行来讲，你得到的结果将会是毛病多多而且难以使用的软件。

通读这本书并一字不差地录入书中的每个例子，会训练你把精力集中到作品的细节上。

发现不同

程序员长年累月的工作会培养出一种重要的技能，那就是观察事物间不同点的能力。有经验的程序员拿着两份仅有细微不同的程序，可以立即指出里边的不同点来。程序员甚至制造出工具来让这件事更加容易，不过我们不会用到这些工具。你要先用笨办法训练自己，然后才可以使用这些工具。

在做这些习题并且录入代码的时候，你一定会写错东西，这是不可避免的，即使有经验的程序员也会偶尔出错。你的任务是把自己写的东西和正确答案对比，把所有的不同点都修正过来。这样的过程可以让你对程序里的错误和 bug 更加敏感。

不要复制粘贴

你必须手动将每个习题录进去。复制粘贴会让这些习题变得毫无意义。这些习题的目的是训练你的双手和大脑思维，让你有能力读代码、写代码、观察代码。如果你复制粘贴的话，就

是在欺骗自己，而且这些习题的效果也会大打折扣。

关于坚持练习的一点提示

你通过这本书学习编程时，我正在学习弹吉他。我每天至少训练 2 小时，至少花 1 小时练习音阶、和弦、琶音，剩下的时间用来学习音乐理论和乐曲演奏、训练听力等。有时我一天会花 8 小时来学习吉他和音乐，因为我觉得这是一件有趣的事情。对我来说，要学习一样东西，最自然、最根本的方法就是去反复地练习。我知道，要学好一种技能，每日的练习是必不可少的，就算哪天的练习没啥进展（对我来说是常事），或者说学习内容实在太难，你也不必介意。只要坚持尝试，总有一天困难会变得容易，枯燥也会变得有趣。

通过这本书学习编程的过程中要记住一点，就是所谓的“万事开头难”，对于有价值的事情尤其如此。也许你是一个害怕失败的人，一遇到困难就想放弃；也许你是一直没学会自律，一遇到“无聊”的事情就不想上手；也许因为有人夸你“有天分”而让你自视甚高，不愿意做这些看上去很笨拙的事情，怕有负你“神童”的称号；也许你太过激进，把自己跟像我这样有 20 多年经验的编程老手相比，让自己失去了信心。

不管是什么原因，你一定要坚持下去。如果遇到做不出来的附加练习，或者遇到一个看不懂的习题，你可以暂时跳过去，过一阵子回来再看。编程中有一件经常发生的怪事就是，一开始你什么都不懂，这会让你感觉很不舒服，就像学习人类的自然语言一样，你会发现很难记住一些词语和特殊符号的用法，而且会经常感到很迷茫，直到有一天，忽然一下子你会觉得豁然开朗，以前不明白的东西忽然就明白了。如果你坚持完成并努力理解这些习题，你最终会学会这些东西的。也许你不会成为一位编程大师，但你至少会明白编程的原理。

如果你放弃的话，你会失去达到这个程度的机会。如果你坚持尝试，坚持录习题，坚持弄懂习题的话，你最终一定会明白里边的内容的。

如果你通读了这本书，却还是不懂怎样写代码，你的努力也不会白费。你可以说你已经尽力了，虽然成效不佳，至少你尝试过了。这也是一件值得骄傲的事情。

给“小聪明”们的警告

有些学过编程的人读到这本书可能会有一种被贬低的感觉。其实本书中没有任何要居高临下地贬低任何人的意思，只不过我比我面向的读者群知道的更多而已。如果你觉得自己比我聪明，觉得我在居高临下，那我也没办法，因为你根本就不是我的目标读者。

如果你觉得这本书里到处都在贬低你的智商，那我对你有以下三个建议。

1. 别读这本书了。我这本书不是写给你的，而是写给那些不是什么都懂的人看的。

2. 放下架子好好学。如果你认为你什么都懂，那就很难从比自己强的人身上学到什么了。
3. 学 Lisp 去。我听说什么都懂的人特喜欢 Lisp。

对于其他抱着学习的目的而来的人，你们读的时候就想着我在微笑就可以了，而且我的眼睛里还带点儿恶作剧的闪光。

目录

准备工作

这个习题并没有代码内容，它的主要目的是让你在计算机上安装好 Python。你应该尽量照着说明进行操作，例如，Mac OSX 默认已经安装了 Python 2，所以就不要在上面安装 Python 3 或者别的 Python 版本了。

> **注意** 如果你不知道怎样使用 Windows 下的 PowerShell，或者 OSX 下的 Terminal（终端），或者 Linux 下的 Bash，那你就需要先学会一个。我把我写的一本《命令行快速入门》简化了一下放到了本书的附录里，读完那部分后，再回来继续下面的步骤。

Mac OSX

完成这个习题你需要完成下列任务。

1. 用浏览器打开 http://www.barebones.com/products/textwrangler/找到并安装 TextWrangler 文本编辑器。
2. 把 TextWrangler（也就是你的编辑器）放到 Dock 中，以方便日后使用。
3. 找到系统中的 Terminal 程序。到处找找，你会找到的。
4. 把 Terminal 也放到 Dock 里面。
5. 运行 Terminal 程序，这个程序没什么好看的。
6. 在 Terminal 里运行 `python`。运行的方法是输入程序的名字再敲一下回车键。
7. 按 Ctrl+D（^D）退出 `python`。
8. 这样你就应该退回到敲 `python` 前的提示界面了。如果没有的话，自己研究一下为什么。
9. 学着在 Terminal 上创建一个目录。
10. 学着在 Terminal 上变到一个目录。
11. 使用你的编辑器在你进入的目录下建立一个文件。建立一个文件，使用“保存”（Save）或者“另存为”（Save As...）选项，然后选择这个目录。
12. 使用键盘切换回到 Terminal 窗口。
13. 回到 Terminal，看看你能不能使用命令看到你新建的文件，上网搜索如何将文件夹中

的内容列出来。

OSX：应该看到的结果

下面是我在自己电脑的 Terminal 中完成上述步骤时看到的内容，和你做的结果会有一些不同，看看你能不能找出两者的不同点。

```
Last login: Sat Apr 24 00:56:54 on ttys001
~ $ python
Python 2.5.1 (r251:54863, Feb  6 2009, 19:02:12)
[GCC 4.0.1 (Apple Inc. build 5465)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> ^D
~ $ mkdir mystuff
~ $ cd mystuff
mystuff $ ls
# ... 使用 TextWrangler 编辑 test.txt ...
mystuff $ ls
test.txt
mystuff $
```

Windows

1. 用浏览器打开 http://notepad-plus-plus.org 下载并安装 Notepad++文本编辑器。这个操作无需管理员权限。
2. 把 Notepad++放到桌面或者快速启动栏，这样就可以方便地访问该程序了。这两条在安装选项中可以看到。
3. 从开始菜单运行 PowerShell 程序。你可以使用开始菜单的搜索功能，输入名称后敲回车键即可运行。
4. 为它创建一个快捷方式，放到桌面或者快速启动栏中以方便使用。
5. 运行终端程序（也就是 PowerShell），这个程序没什么好看的。
6. 在终端程序中运行 `python`。运行的方法是输入程序的名字再敲一下回车键。
   a. 如果运行 `python` 发现它不存在（`python is not recognized`），你需要访问 http://python.org/download 下载并且安装 Python。
   b. 确认你要安装的是 Python 2 而不是 Python 3。
   c. 你也可以试试 ActiveState Python，尤其是没有管理员权限的时候。