

# 《"笨办法"学Python》

## 图书基本信息

书名：《"笨办法"学Python》

13位ISBN编号：978711535054X

出版时间：2014-11-1

作者：肖 (Zed A.Shaw)

页数：249

译者：王巍巍

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：[www.tushu111.com](http://www.tushu111.com)

# 《"笨办法"学Python》

## 内容概要

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。

Zed Shaw完善了 this 堪称世上最好的Python学习系统。只要跟着学习，你就会和迄今为止数十万Zed教过的初学者一样获得成功。

在这本书中，你将通过完成52个精心设计的习题来学会Python。阅读这些习题，把习题的代码精确地写出来（禁止复制和粘贴！），修正你的错误，观察程序的运行。在这个过程中，你将了解软件是如何工作的，好的程序看起来是什么样子，怎样阅读、编写、思考代码，以及如何用专业程序员的技巧来找出并修正错误。最重要的是，你将学到下面这些编写优秀的Python软件必需的初始技能。

这本书会让你的每一分钟投入都有回报。Python是世界上最强大、最受欢迎的编程语言之一，很快你就会成为一名Python程序员。

你还可以看Zed的视频！随书附赠的DVD中包含5个多小时激情挥洒的教学内容：一部完整的Python视频教程！

# 《"笨办法"学Python》

## 作者简介

Zed A.Shaw是一位狂热的吉他手、程序员和编程书作家。他的这本书在全世界有数百万的读者。他写的软件大大小小的公司都在使用。他的文章经常被众多极客圈里的人阅读和引用。他是一名活跃而有趣的作家，他会让你大笑，也会让你思考。

## 书籍目录

习题0 准备工作	1
Mac OSX	1
OSX：应该看到的结果	2
Windows	2
Windows：应该看到的结果	3
Linux	4
Linux：应该看到的结果	5
给新手的告诫	5
习题1 第一个程序	7
应该看到的结果	8
附加练习	10
常见问题回答	11
习题2 注释和#号	12
应该看到的结果	12
附加练习	13
常见问题回答	13
习题3 数字和数学计算	14
应该看到的结果	15
附加练习	15
常见问题回答	16
习题4 变量和命名	17
应该看到的结果	18
附加练习	18
常见问题回答	18
习题5 更多的变量和打印	20
应该看到的结果	21
附加练习	21
常见问题回答	21
习题6 字符串和文本	23
应该看到的结果	24
附加练习	24
常见问题回答	24
习题7 更多打印	26
应该看到的结果	26
附加练习	27
常见问题回答	27
习题8 打印，打印	28
应该看到的结果	28
附加练习	28
常见问题回答	29
习题9 打印，打印，打印	30
应该看到的结果	30
附加练习	31
常见问题回答	31
习题10 那是什么	32
应该看到的结果	33
转义序列	33

附加练习 34  
常见问题回答 34  
习题11 提问 35  
应该看到的结果 36  
附加练习 36  
常见问题回答 36  
习题12 提示别人 37  
应该看到的结果 37  
附加练习 38  
常见问题回答 38  
习题13 参数、解包和变量 39  
等一下！“特性”还有另外一个名字 39  
应该看到的结果 40  
附加练习 41  
常见问题回答 41  
习题14 提示和传递 42  
应该看到的结果 42  
附加练习 43  
常见问题回答 43  
习题15 读取文件 45  
应该看到的结果 46  
附加练习 46  
常见问题回答 47  
习题16 读写文件 48  
应该看到的结果 49  
附加练习 50  
常见问题回答 50  
习题17 更多文件操作 51  
应该看到的结果 52  
附加练习 52  
常见问题回答 53  
习题18 命名、变量、代码和函数 54  
应该看到的结果 55  
附加练习 56  
常见问题回答 56  
习题19 函数和变量 57  
应该看到的结果 58  
附加练习 58  
常见问题回答 59  
习题20 函数和文件 60  
应该看到的结果 61  
附加练习 61  
常见问题回答 61  
习题21 函数可以返回某些东西 63  
应该看到的结果 64  
附加练习 64  
常见问题回答 65  
习题22 到现在你学到了哪些东西 66  
学到的东西 66

习题23 阅读一些代码 67  
习题24 更多练习 68  
应该看到的结果 69  
附加练习 69  
常见问题回答 70  
习题25 更多更多的实践 71  
应该看到的结果 72  
附加练习 73  
常见问题回答 74  
习题26 恭喜你，现在可以考试了！ 75  
常见问题回答 75  
习题27 记住逻辑关系 76  
逻辑术语 76  
真值表 77  
常见问题回答 78  
习题28 布尔表达式练习 79  
应该看到的结果 80  
附加练习 81  
常见问题回答 81  
习题29 if语句 82  
应该看到的结果 83  
附加练习 83  
常见问题回答 83  
习题30 else和if 84  
应该看到的结果 85  
附加练习 85  
常见问题回答 85  
习题31 作出决定 86  
应该看到的结果 87  
附加练习 87  
常见问题回答 87  
习题32 循环和列表 89  
应该看到的结果 90  
附加练习 91  
常见问题回答 91  
习题33 while循环 93  
应该看到的结果 94  
附加练习 94  
常见问题回答 95  
习题34 访问列表的元素 96  
附加练习 97  
习题35 分支和函数 98  
应该看到的结果 100  
附加练习 100  
常见问题回答 100  
习题36 设计和调试 102  
if语句的规则 102  
循环的规则 102  
调试的小技巧 103

- 家庭作业 103
- 习题37 复习各种符号 104
- 关键字 104
- 数据类型 105
- 字符串转义序列 105
- 字符串格式化 106
- 操作符 106
- 阅读代码 107
- 附加练习 108
- 常见问题回答 108
- 习题38 列表的操作 109
- 应该看到的结果 111
- 附加练习 111
- 常见问题回答 112
- 习题39 字典, 可爱的字典 113
- 应该看到的结果 116
- 附加练习 116
- 常见问题回答 117
- 习题40 模块、类和对象 118
- 模块和字典差不多 118
- 类和模块差不多 119
- 对象相当于迷你导入 120
- 获取某样东西里包含的东西 121
- 第一个关于类的例子 121
- 应该看到的结果 122
- 附加练习 122
- 常见问题回答 123
- 习题41 学习面向对象术语 124
- 单词练习 124
- 语汇练习 124
- 混合巩固练习 125
- 阅读测试 125
- 练习从语言到代码 127
- 阅读更多代码 128
- 常见问题回答 128
- 习题42 对象、类及从属关系 129
- 代码写成什么样子 130
- 关于class Name(object) 132
- 附加练习 132
- 常见问题回答 133
- 习题43 基本的面向对象分析和设计 134
- 简单游戏引擎的分析 135
- 把问题写下来或者画出来 135
- 摘录和研究关键概念 135
- 为各种概念创建类层次结构图和对象关系图 136
- 编写和运行各个类 137
- 重复和优化 139
- 自顶向下与自底向上 139

《来自Percal 25号行星的哥顿人》的  
代码 139  
应该看到的结果 145  
附加练习 146  
常见问题回答 146  
习题44 继承与合成 147  
什么是继承 147  
隐式继承 148  
显式覆盖 149  
在运行前或运行后替换 149  
三种方式组合使用 151  
为什么要用super() 152  
super()和\_\_init\_\_搭配使用 152  
合成 153  
继承和合成的应用场合 154  
附加练习 154  
常见问题回答 155  
习题45 你来制作一个游戏 156  
评价你的游戏 156  
函数的风格 157  
类的风格 157  
代码风格 158  
好的注释 158  
为你的游戏评分 158  
习题46 项目骨架 160  
Python软件包的安装 160  
创建骨架项目目录 161  
最终目录结构 162  
测试你的配置 164  
使用这个骨架 164  
小测验 164  
常见问题回答 165  
习题47 自动化测试 166  
编写测试用例 166  
测试指南 168  
应该看到的结果 169  
附加练习 169  
常见问题回答 169  
习题48 更复杂的用户输入 170  
我们的游戏语汇 170  
断句 171  
语汇元组 171  
扫描输入 171  
异常和数字 171  
应该测试的东西 172  
设计提示 174  
附加练习 174  
常见问题回答 174  
习题49 创建句子 175



- match和peek 175
- 句子的文法 176
- 关于异常 178
- 应该测试的东西 179
- 附加练习 179
- 常见问题回答 179
- 习题50 你的第一个网站 180
- 安装lpthw.web 180
- 写一个简单的“Hello World”项目 181
- 会发生什么 182
- 修正错误 183
- 创建基本的模板文件 183
- 附加练习 185
- 常见问题回答 186
- 习题51 从浏览器中获取输入 187
- Web的工作原理 187
- 表单的工作原理 189
- 创建HTML表单 191
- 创建布局模板 193
- 为表单撰写自动测试代码 194
- 附加练习 196
- 常见问题回答 197
- 习题52 创建Web游戏 198
- 重构习题43中的游戏 198
- 会话和用户跟踪 203
- 创建引擎 204
- 期末考试 207
- 常见问题回答 208
- 接下来的路 209
- 怎样学习任何一种编程语言 210
- 老程序员的建议 211
- 附录 命令行快速入门 213
- 简介：废话少说，命令行来也 213
- 如何使用这个附录 213
- 你需要发挥记忆力 214
- 习题1 准备工作 214
- 任务 214
- 知识点 215
- 更多任务 216
- 习题2 路径、文件夹和目录（pwd） 217
- 任务 217
- 知识点 218
- 更多任务 219
- 习题3 如果你迷失了 219
- 任务 219
- 知识点 219
- 习题4 创建目录（mkdir） 219
- 任务 220
- 知识点 221

- 更多任务 221
- 习题5 更改目录 (cd) 222
- 任务 222
- 知识点 225
- 更多任务 225
- 习题6 列出目录下的内容 (ls) 226
- 任务 226
- 知识点 229
- 更多任务 230
- 习题7 删除路径 (rmdir) 230
- 任务 230
- 知识点 232
- 更多任务 232
- 习题8 在多个目录中切换  
(pushd, popd) 233
- 任务 233
- 知识点 235
- 更多任务 235
- 习题9 创建空文件 (touch,  
New-Item) 235
- 任务 236
- 知识点 236
- 更多任务 236
- 习题10 复制文件 (cp) 237
- 任务 237
- 知识点 239
- 更多任务 240
- 习题11 移动文件 (mv) 240
- 任务 240
- 知识点 242
- 更多任务 242
- 习题12 查看文件内容  
(less, MORE) 242
- 任务 243
- 知识点 243
- 更多任务 243
- 习题13 流文件内容显示 (cat) 244
- 任务 244
- 知识点 245
- 更多任务 245
- 习题14 删除文件 (rm) 245
- 任务 245
- 知识点 247
- 更多任务 247
- 习题15 退出命令行 (exit) 247
- 任务 247
- 知识点 248
- 更多任务 248
- 命令行将来的路 248

Unix Bash参考资料 248  
PowerShell参考资料 249

## 精彩短评

- 1、直接在<http://learnpythonthehardway.org/book/> 这个链接上看就好啦，及时更新，花了比想象多的时间来看这本书，编程任务也没有完全完成，感觉在编程思维方面离作者差很远。本书的好处是留了很多学习空间给读者自行努力，其实学会那些解决问题的方法才是最重要的。不急于求成，耐心完成任务，能学会很多。就像<How to read a book>讲的，真正的好书不只是教会你知识，更能提升你的理解力。
- 2、要是有深入版就好了
- 3、不错、不过后面的有点进度过快
- 4、花了一周的时间看完，在以前的基础上零零碎碎学了一些新的东西，还算是有所收获，但是前面讲的很基础很简单，到ex40之后突然难度增加不是很懂在讲些什么。
- 5、有一些基础的会觉得这本书太入门了，另外其实还是英文版好一点。
- 6、说是笨办法，其实讲的还挺浅显的，属于入门书籍。附赠的光盘实在不知所云：现在有几台笔记本带光驱的？
- 7、我觉得后面的越看越模糊，撸一遍估计不够。
- 8、作为入门书还是可以的。不过我好歹学过C和Matlab，看到作者教函数真值表的时候很无语……大概40章以后才有意思起来。
- 9、first python course
- 10、入门还是不错的。书后附录的命令行速成更是好用。每一题都完成了。直到ex51，ex52老是出错，就算了。还是去学flask，也用不着书中的web.py了。
- 11、第一本python书
- 12、附录的命令行快速入门不错（之前都不会用，新技能get），习题做到中间部分就看不大懂了，解释也不大清楚，感觉不太适合零基础（好像每本书对零基础的定义都不一样）果断转《父与子编程》，以后再回来看看吧
- 13、入门不错 后面坑太大 先入个小坑吧
- 14、变量参数（arv），OOP都没有将清楚。拿来上手很不错，但是千万别当参考书。作者假定你得自己去网上查，但是自己讲什么class却扯半天讲不透彻。
- 15、我感觉自己被调戏了。
- 16、前期通过简单习题让你熟悉语法，后期就不太好了。
- 17、3.5/5
- 18、很基础，入门必备
- 19、没有一门语言能坚持学下去，连皮毛都没搞懂就不想搞了。而且，实际上，读这本书最困难的地方在于不会用我积了两年灰的Mac！
- 20、I hate it when a textbook tells me to ignore some important underlying mechanisms and just to follow instructions. However, this textbook encourages readers themselves to search for deeper and wider knowledge about Python, which is a good thing but somehow reflects the author's "laziness". It leaves many things unexplained!!!!
- 21、完全的案例入门式，做完全书的案例，你的编程水平将得到巨大的提升，可惜我还是没坚持下来
- 22、1个月的时间搞定了这本书，不得不说，这种方法学习编程不枯燥又有趣。
- 23、建议入门第一本
- 24、前面很通俗易懂，到后面十来个章节还挺难的，需要多次翻看练习
- 25、抛开一切,敲起来。告诉你,编程的正确姿势。
- 26、搞笑幽默。不过到最后为了搞笑而搞笑。
- 27、-真正的python零基础书。短小精悍，幽默风趣。推荐！  
-一个初级程序员需要知道的三种最基本的技能：读写代码，注意细节，分辨力。
- 28、第一本Python书籍，内容比较简单，大一上看的，零基础的可当做第一本编程书，有基础的可不看。
- 29、书不错，适合无基础入门，不够深入，讲的也不是很细。

## 《"笨办法"学Python》

- 30、类与模块的对比，多态继承与合成的对比，都有恍然大悟之感，值得再翻一次读书笔记，入门好书
  - 31、断断续续刷完了。。。对于最后的搭建项目还是有点一知半解。。。看别人都说这是超级基础的入门书。我对自己的智商有些忧虑了。这本书最好是连续一段时间刷完。毕竟是初学。断了之后再再看可能就没感觉了。总之能有所了解。其实可以看英文版的。因为用的单词都很简单。
  - 32、纯概念教程
  - 33、错别字还是挺多的，我觉得我需要深入一点学习了
  - 34、前面讲的不错，最后几章看的云里雾里的
  - 35、入门好书。
  - 36、很抱歉，这本书其实是对初学者并不友好的，很多东西都没有讲明白，需要自己上网查，问题是我自己网上可以搜到很多东西，为什么要去看你这本书，而且很多地方作者讲的并不明白，不够细致，其实这本书对零基础并不友好。
  - 37、所谓笨办法并不是真的笨，浅显易懂并且直接要害。没有任何程序开发基础的人都可以看懂。解决了我多年一看见程序开发的书就害怕、啃不动的问题。这一次彻底解决多年来未曾弄懂的“面向对象”。对于非程序员、但要使用程序解决实际小问题的人非常有帮助。至少能看懂别人的代码，稍加修改为己所用。
  - 38、在 learn python the hard way官网上有下载 也可以在线看 ^^
  - 39、一般，不适合作为入门书
  - 40、死在了习题46.....
  - 41、太基础，翻翻就没了
  - 42、写得很好，很适合入门学习
- 喜欢Zed的风格
- 43、面向对象一节讲得不清楚，可以参考廖雪峰的教程。我更喜欢作者最后的自白，“编程20年，兴趣已无，语言不重要的，解决问题才是，如果为了赚钱，开家快餐店更直接。”
  - 44、挺基础，说解易懂，翻译欠佳。
  - 45、要是没基础就别看了
  - 46、原以为它会很浅显入理，但看来并不是这样而是机械的操作和无聊的跟随，没有突出应有的编程思维优势
  - 47、还是蛮简单的，如果当初好好跟着学就好了（￣ ￣）拖了一年半的书终于搞完了
  - 48、买了一本父与子和这本 互为对照查阅不懂得知识点 进步很快 这本书简单又有趣 入门级最佳教学书！
  - 49、212页这段太牛b了
  - 50、人生苦短

## 精彩书评

1、1. 一本超级入门，超级简单的书，非常适合没有任何编程基础的人，如果你有其他编程基础而学python，那这本书对于你来说有点简单2. 代码很多，新手练习训练代码感很不错3. 虽说是新手入门书，但没有每本书都介绍的python安装步骤，我个人觉得，真没有必要每本python的基础教程类的书都讲到如何在MAC, Linux, Windows下如何安装python，感觉他们有点在充字数的感觉。但是这本入门书却没有专门花个几十页来讲如何安装，这点很不错。不过，本来这本书就只有200多页。4. 该书作者很酷，从文章的最后一章“老程序员的建议”可以看出来。而且该书作者还专门建了一个网页“<http://learnpythonthehardway.org>”，该网页有练习题和问题解答。

2、短评下这本书：1、讲的很基础，以习题的形式一步一步引导学习Python，形式新颖，又耐学，比较适合初学者；2、为什么说是比较，因为里面有小部分内容讲的不够具体，比如说安装库的方法（是叫这个吧？）、面向对象分析的本质；3、感觉本书一直写到那个“25#星球”的代码之前（出去类和对象的讲解部分，简直看的我云里雾里）都还是非常不错的，这之后的内容给我一种在拼凑页数或者是赶快完稿的感觉；4、代码里有几处符号错误，类似于{}写成[]之类的，不过从另一方面说也变成了锻炼读者debug的实例。以上

3、1、总结评分：花了大半个月，断断续续的把这本书看完了。平心而论，这本书还是不错的，写法别具一格，但是Python入门的话仅仅靠这一本书是不够的，最好能配上搜索引擎和类似于《C++ Primer》那样字典类的书籍，因为LPTHW很少讲解一个具体的函数是什么样的，对于列表和字典讲解的也很少。我觉得要是能配上一本知识体系比较全的书可能更加容易的学习，我使用的是《Python核心编程》，也有人推荐《Python基础教程》，但是我觉得差不了多少，都是用来查阅的，而不是遍历的。配合着别的书，看完了LPTHW就可以自己做一些事情了，比如自己写一个网站，自己编一个小游戏都行，网上有人推荐过一条很不错方向：Python + flask及其源码剖析，学起来简单，适合已经毕业但是想转行的人员，据说坚持四个月，就能跳槽了。笔者其实对Web编程也很感兴趣，但是因为时间太紧了，只能放一放，剩下的一年时间里面要一边做毕设，一边学习数据结构和算法。不过不管你选择的是什么方向，都要记住在实践的过程中学习，要不然看过的东西很快就会忘记。既然选择了远方，便只顾风雨兼程。2、勘误版次：2014年11月第1版 印刷时间：2015年5月北京第3次印刷 字典括号问题比较严重，多处把大括号都写成了中括号，下面是是括号印错的地方：Page 114, Line 2, 8, 11, 15Page 144, Line 3, 9Page 161, Line 6, 17Page 167, “game.py”的Line 6, “ex47\_tests.py”的Line 10, 17, 26, 27, 28Page 195, Line 19Page 161, Line 6, 17Page 201, Line 105, 108, 112, 115, 117, 120, 122, 126Page 202, Line 9, 16, 25, 26, 27 一些明显的印刷错误：Page 74, “常见问题回答”第2个问题, “import ex15”应为“import ex25” Page 122, “附加练习”第2题, “然后再类里”应为“然后在类里” Page 167, “编写测试用例”中的第4点, “删除所有的.py文件”应为“删除所有的.pyc文件” Page 179, “调用的函数”应改为“要调用的函数” Page 181, urls中的“index”中的i应该大写, 相应的下面的类名也应该大写。后面的章节此处的i就是大写 缩进问题：Page 126, Line 49~57应该都缩进一格Page 141, Line 18~24, Line 27~33应该都缩进一格 文件命名的问题，我的理解是代码块的右上角为文件的命名：Page 172, 代码块右上角的“ex48.py”应为“lexicon\_tests.py” Page 177, 代码块右上角的“ex49.py”应为“parser.py” Page 181, 代码块右上角的“ex50.py”应为“app.py” 欢迎访问我的博客，大家一起学习<http://blog.csdn.net/xuelabizp>

4、“笨办法”的后记是这样写的。你已经完成了这本书而且打算继续编程。也许这会成为你的一门职业，也许你只是作为业余爱好玩玩。无论如何，你都需要一些建议以保证你在正确的道路上继续前行，并且让这项新的爱好为你带来最大程度的享受。我从事编程已经太长时间，长到对我来说编程已经是非常乏味的事情了。我写这本书的时候，已经懂得大约20种编程语言，而且可以在大约一天或者一个星期内学会一门编程语言(取决于这门语言有多古怪)。现在对我来说编程这件事情已经很无聊，已经谈不上什么兴趣了。当然这不是说编程本身是一件无聊的事情，也不是说你以后也一定会这样觉得，这只是我个人在当前的感觉而已。在这么久的旅程下来我的体会是：编程语言这东西并不重要，重要的是你用这些语言做的事情。事实上我一直知道这一点，不过以前我会周期性地被各种编程语言分神而忘记了这一点。现在我是永远不会忘记这一点了，你也不应该忘记这一点。你学到和用到的编程语言并不重要。不要被围绕某一种语言的宗教把你扯进去，这只会让你忘掉了语言的真正目的，也就是作为你的工具来实现有趣的事情。编程作为一项智力活动，是唯一一种能让你创建交互式艺术的



艺术形式。你可以创建项目让别人使用，而且你可以间接地和使用者沟通。没有其他的艺术形式能做到如此程度的交互性。电影领着观众走向一个方向，绘画是不会动的。而代码却是双向互动的。编程作为一项职业只是一般般有趣而已。编程可能是一份好工作，但如果你想赚更多的钱而且过得更快乐，你其实开一间快餐分店就可以了。你最好的选择是将你的编程技术作为你其他职业的秘密武器。技术公司里边会编程的人多到一毛钱一打，根本得不到什么尊敬。而在生物学、医药学、政府部门、社会学、物理学、数学等行业领域从事编程的人就能得到足够的尊敬，而且你可以使用这项技能在这些领域做出令人惊异的成就。当然，所有的这些建议都是没啥意义的。如果你跟着这本书学习写软件而且觉得很喜欢这件事情的话，那你完全可以将其当作一门职业去追求。你应该继续深入拓展这个近五十年来极少有人探索过的奇异而美妙的智力工作领域。若能从中得到乐趣当然就更好了。最后我要说的是学习创造软件的过程会改变你而让你与众不同。不是说更好或更坏，只是不同了。你也许会发现因为你会写软件而人们对你的态度有些怪异，也许会用“怪人”这样的词来形容你。也许你会发现因为你会戳穿他们的逻辑漏洞而他们开始讨厌和你争辩。甚至你可能会发现有人因为你懂得计算机怎么工作而觉得你是个讨厌的怪人。对于这些我只有一个建议：让他们去死吧。这个世界需要更多的怪人，他们知道东西是怎么工作的而且喜欢找到答案。当他们那样对你时，只要记住这是你的旅程，不是他们的。“与众不同”不是谁的错，告诉你“与众不同是一种错”的人只是嫉妒你掌握了他们做梦都不能想到的技能而已。你会编程。他们不会。这真他妈的酷。看得我热血沸腾。

5、八月底的时候没忍住，买了几本python的书，因为觉得python实在太神奇了决心学一学。与这本书一同买的还有一本python基础教程，两者一比较就能立见高下，这本书完全是给没有经验的外行人看的，没错，你没看错，是外行人，比起python重口味的字典式的教授，这本书简直就是在和白开水，当然学了几门语言的入门级程序猿来说，这本书只是让你在几个小时之内用python写代码注意缩进，知道要在分支语句后面加分号，已经不把方括号写成圆括号，仅此而已。另外，作者说要倒着检查代码，不知道有多少人做到了，反正我没做。

6、这本书以实践为主，通过重复练习习得技能，即“指令式教学”。这种方式对于没有方向的初学者很有效，它指明了方向。同时，反复练习使人有“编程实感”，更有学习动力。其次，“不说废话，放码过来”的单刀直入的风格很符合“Python哲学”。每个习题后有“常见问题回答”，在书的开篇、结尾、偶尔在习题中会给一些编程建议，体现了这本书对新手的友好。但是，它没有系统的知识讲解。通常一本系统的书，会将数据类型、运算符表达式、语句、函数等部分按一定的顺序分章讲解，而这本书并不系统，比如，习题 38、习题 39 练习列表和字典，但到习题 48 才提及元组。它也很少为你总结，你需要自己去归纳提炼，比如，习题 22 就是让你自己去归纳。因此，它不适合想要先了解特性或是想系统学习的读者。综上，这本书适合初学者，尤其适合零基础的初学者，不适合讲求系统性的读者。（这方面推荐《简明Python》）

# 《"笨办法"学Python》

## 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:[www.tushu111.com](http://www.tushu111.com)