



Pearson



“笨办法”

# 学 Python 3

Learn  
**PYTHON 3**  
the **HARD WAY**

A Very Simple Introduction  
to the Terrifyingly Beautiful World of  
Computers and Code

[美] 泽德 A. 肖 (Zed A. Shaw) 著  
王巍巍 译



中国工信出版集团



人民邮电出版社  
POSTS & TELECOM PRESS



Pearson

“笨办法”

学

Python 3

Learn  
**PYTHON 3**  
the **HARD WAY**

A Very Simple Introduction  
to the Terrifyingly Beautiful World of  
Computers and Code

[美] 泽德 A. 肖 (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 3 / (美) 泽德·A. 肖 (Zed A. Shaw) 著 ; 王巍巍译. -- 2版. -- 北京 : 人民邮电出版社, 2018. 6(2018. 10重印)

书名原文: Learn Python 3 the Hard Way: A Very Simple Introduction to the Terrifyingly Beautiful World of Computers and Code  
ISBN 978-7-115-47881-8

I. ①笨… II. ①泽… 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(2018)第059179号

## 内 容 提 要

本书是一本 Python 入门书, 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。本书是基于 Python 3.6 版本编写的。

本书结构非常简单, 除“准备工作”之外, 还包括 52 个习题, 其中 26 个覆盖了输入/输出、变量和函数 3 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] 泽德 A. 肖 (Zed A. Shaw)
  - 译 王巍巍
  - 责任编辑 杨海玲
  - 责任印制 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>  
三河市君旺印务有限公司印刷
  - ◆ 开本: 800×1000 1/16  
印张: 17.25  
字数: 382 千字 2018 年 6 月第 2 版  
印数: 81 501-86 500 册 2018 年 10 月河北第 6 次印刷
- 著作权合同登记号 图字: 01-2017-5601 号



定价: 59.00 元

读者服务热线: (010) 81055410 印装质量热线: (010) 81055316

反盗版热线: (010) 81055315

广告经营许可证: 京东工商广登字 20170147 号

# 版权声明

Authorized translation from the English language edition, entitled LEARN PYTHON 3 THE HARD WAY: A VERY SIMPLE INTRODUCTION TO THE TERRIFYINGLY BEAUTIFUL WORLD OF COMPUTERS AND CODE, 1st Edition, ISBN: 0134692888 by SHAW, ZED A., published by Pearson Education, Inc, Copyright © 2017 Zed A. Shaw.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD. and POSTS & TELECOM PRESS Copyright © 2018.

本书中文简体字版由 Pearson Education Asia Ltd.授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

本书封面贴有 Pearson Education（培生教育出版集团）激光防伪标签，无标签者不得销售。版权所有，侵权必究。

# 译者序

**我**是从2010年关注 Zed Shaw 编写的这本 Python 入门书的。当时尽管 Python 3 问世已经有些年头，但由于性能和兼容性等一系列问题，使用一直不太广泛。这些年，Python 3 自身得到了很大的改进，应用也逐渐变得广泛，而且根据 PEP 373 的说明，2020 年后，Python 2 就不会再发布更新，Python 3 取代 Python 2 可以说是指日可待。所以，如果你还在学习或者使用 Python 2，现在是时候转成 Python 3 了。

如果你刚刚接触编程，这本书可以说是你入门编程最有趣的选择。在众多编程入门书中，这本书的教学方法可以说是特立独行，这本书真正重要的是，它会通过练习和实践，让你形成良好的程序员素养。入门书强调这一点的可以说少之又少。本书的相关特点作者在前言中已经做了详细说明，你看下一页就知道了。

随书的视频也很有趣。也许你会觉得编程是一门很高深的手艺，程序员个个脑袋灵光得很。但是在视频里，你会看到作者被一些简单的错误卡住，半天才找出头绪，其实这才是程序员的日常状态。

总之，现在编程很火，编程语言中 Python 很流行，Python 入门书中这本很有趣。怎么样，试试？

## 致谢

在翻译这本书前几版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更加专业。在此一并致谢。

---

# 译者简介

**王**巍巍是一名受软件和编程的吸引，中途转行上岗的软件从业人员。写代码和翻译是他的两大爱好，此外他还喜欢在网上撰写和翻译一些不着边际的话题和文章。如果读者对书中的内容有疑问，或者发现了书中的错误，又或者只是想随便聊聊，请通过电子邮件（[wangweiwei@outlook.com](mailto:wangweiwei@outlook.com)）与他联系。

# 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，而你则通过反复练习来学到技能，这些习题也是专为反复练习而设计的。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。

本书通过练习和记忆的方式，指导你逐渐掌握使用 Python 编程的技能，然后由浅入深，让你将这些技能应用到各种问题上。读完本书之后，你将有能力接触学习复杂的编程主题所需的工具。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，掌握了这些技能，你将学会如何编写代码。

## 针对 Python 3 的改进

本书使用了 Python 3.6。我用 Python 的这个版本是因为它包含了一个新的改进版的字符串格式化系统，这个系统比之前的更为易用。初学者接触 Python 3.6 可能会遇到一些问题，但我在书里会帮你克服。Python 3.6 的一个特别令人头疼的问题是，在一些关键位置的出错消息都很糟糕，不过这些我都会帮你弄懂的。

我还根据自己过去 5 年的教学经验，改进了视频教程。以前的视频中，你只是看我做习题，在新版视频里，你还可以看到我如何破坏每个习题中的程序，以及如何修复它们。这种技术称为“调试”（debugging）。从中你可以学到如何解决问题，也能对 Python 运行你创建的程序的原理有一个概念，从而提高你解决问题的能力。你还会学到很多有用的调试技巧。

最后要讲的是，Python 3 版本完全支持 Windows 10。过去的版本偏重于 Unix 风格的操作系统，如 macOS 和 Linux，Windows 只是顺便讲讲。在我写这本书的时候，微软公司已经开始认真对待开源工具和开发者了，而且 Windows 也是一个严肃的 Python 开发平台。在视频中，很多场合下我用 Windows 进行了演示，为了完全兼容，我也演示了 macOS 和 Linux。我讲了每个平台都会遇到的一些坑，演示了安装过程，还提供了不少别的小窍门。

# 笨办法更简单

在本书的帮助下，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打下良好的基础。这本书可能无法让你一夜之间成为一名程序员，但它将会让你踏上学习编程方法的道路。

本书的目的是教会你编程新手需要了解的 3 种重要的技能：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，那你学习编程也会有问题。尤其是，如果你连程序源代码中的那些奇怪字符都敲不出来的话，就更别提编程了。如果没有这些基本技能，你连最基本的软件工作原理都难以学会。

手动录入代码样例并让它们运行起来的过程，会让你学会各种符号的名称，熟悉它们的录入，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。

通读本书并一字不差地录入书中的每个例子，会训练你在做某件事时把精力集中到自己正在做的事情的细节上。

## 发现不同

大多数程序员长年累月地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段仅有细微不同的代码，可以立即指出里边的不同点来。程序员甚至发明工具来让这件事更加容易，不过我们不会用这些工具。你要先用笨办法训练自己，然后



再使用这些工具。

在做这些习题并且录入每段代码的时候，你一定会犯错，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、bug 以及其他问题更加敏感。

## 少瞅多问

只要是写代码，就会写出“bug”（虫子）来。“bug”是你写的代码中的缺陷、错误或者问题。据说早年有一次有人的计算机工作异常，检查后发现是一只蛾子飞到计算机里导致的，于是后来人们就把计算机的问题称为 bug 了。要修复计算机的问题，就需要对它进行“除虫”，这也是调试（debug）一词的来历。在软件的世界里，bug 简直不计其数，真的是太多了。

和那只蛾子一样，你的 bug 会藏在代码中，而你需要把它们找出来。别以为盯着屏幕上的代码看，“虫子”就会自己爬出来了，你需要更多信息才能找到它们，你需要站起来，挽起袖子找“虫子”。

要找“虫子”，你需要拷问你的代码，问它究竟发生了什么，或者你需要站在不同的角度去看代码。在本书里我多次提到“少瞅多问”，我演示了如何让代码“坦白交代”自己干了什么，如何把拷问的结果变成解决问题的方案。我还演示了各种不同的理解代码的方式，从而让你获得更多信息和洞察力。

## 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、写代码和观察代码。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。

## 使用视频教程

本书附带的视频解释了代码的工作原理，以及（更重要的）破坏代码的方法。视频中我会故意破坏代码，再展示修复代码的方法，通过这样的方式，我演示了很多常见错误。我还使用了调试和拷问的手段讲解代码。视频里演示了“少瞅多问”的思路。

## 关于坚持练习的一点提示

你通过本书学习编程时，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。

在我写《“笨办法”学 Python》和《“笨办法”学 Ruby》这两本书之间的那段时间，我对绘画产生了兴趣。在 39 岁的时候喜欢上了视觉艺术，然后就跟以前学吉他、音乐、编程的时候一样，每天学绘画。我搜集了诸多入门教材，照着书上的去做，每天都画一些东西，并且享受着学习的过程。我离“艺术家”还差得很远，甚至连“画得好”都谈不上，不过现在我可以说是“会画画”的了。在学习艺术的过程中，我用的就是本书教你编程的方法。只要将问题拆分成小的练习和课程，你就可以学会任何东西。只要集中精力慢慢提高，享受学习的过程，不管你最终学到什么程度，你都会从中获益的。

通过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“有天赋”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“天才”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巩固练习，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，忽然一下子你就豁然开朗，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，坚持录入习题，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。

## 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 版初稿，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 为前两版做了封面设计，感谢 Brian Shumate 在早期网站设计方面的帮助，同时感谢所有读过前两版并且提出反馈和纠正错误的读者。

谢谢你们。

# 资源与支持

本书由异步社区出品，社区（<https://www.epubit.com/>）为您提供相关资源和后续服务。

## 配套资源

本书提供免费的配套视频。要观看配套视频，读者直接扫描每个习题首页标题旁的二维码即可。

## 提交勘误

作者和编辑尽最大努力来确保书中内容的准确性，但难免会存在疏漏。欢迎您将发现的问题反馈给我们，帮助我们提升图书的质量。

当您发现错误时，请登录异步社区，按书名搜索，进入本书页面，点击“提交勘误”，输入勘误信息，点击“提交”按钮即可。本书的作者和编辑会对您提交的勘误进行审核，确认并接受后，您将获赠异步社区的 100 积分。积分可用于在异步社区兑换优惠券、样书或奖品。

提交勘误

页码:  页内位置 (行数):  错误次数:

B I U

字数统计

提交

## 扫码关注本书

扫描下方二维码，您将会在异步社区微信服务号中看到本书信息及相关的服务提示。



## 与我们联系

我们的联系邮箱是 [contact@epubit.com.cn](mailto:contact@epubit.com.cn)。

如果您对本书有任何疑问或建议，请您发邮件给我们，请在邮件标题中注明本书书名，以便我们更高效地做出反馈。

如果您有兴趣出版图书、录制教学视频，或者参与图书翻译、技术审校等工作，可以发邮件给我们；有意出版图书的作者也可以到异步社区在线提交投稿（直接访问 [www.epubit.com/selfpublish/submission](http://www.epubit.com/selfpublish/submission) 即可）。

如果您是学校、培训机构或企业，想批量购买本书或异步社区出版的其他图书，也可以发邮件给我们。

如果您在网上发现有针对异步社区出品图书的各种形式的盗版行为，包括对图书全部或部分内容的非授权传播，请您将怀疑有侵权行为的链接发邮件给我们。您的这一举动是对作者权益的保护，也是我们持续为您提供有价值的内容的动力之源。

## 关于异步社区和异步图书

“异步社区”是人民邮电出版社旗下 IT 专业图书社区，致力于出版精品 IT 技术图书和相关学习产品，为作者提供优质出版服务。异步社区创办于 2015 年 8 月，提供大量精品 IT 技术图书和电子书，以及高品质技术文章和视频课程。更多详情请访问异步社区官网 <https://www.epubit.com>。

“异步图书”是由异步社区编辑团队策划出版的精品 IT 专业图书的品牌，依托于人民邮电出版社近 30 年的计算机图书出版积累和专业编辑团队，相关图书在封面上印有异步图书的 LOGO。异步图书的出版领域包括软件开发、大数据、AI、测试、前端、网络技术等。



异步社区



微信服务号

# 目录

习题 0 准备工作	1	巩固练习	21
macOS	1	常见问题回答	21
Windows	2	习题 6 字符串和文本	22
Linux	3	应该看到的结果	23
网上搜索	4	巩固练习	23
给新手的告诫	5	破坏程序	23
可选文本编辑器	5	常见问题回答	24
习题 1 第一个程序	7	习题 7 更多打印	25
应该看到的结果	8	应该看到的结果	25
巩固练习	10	巩固练习	26
常见问题回答	11	破坏程序	26
习题 2 注释和#号	12	常见问题回答	26
应该看到的结果	12	习题 8 打印, 打印	27
巩固练习	12	应该看到的结果	27
常见问题回答	13	巩固练习	28
习题 3 数字和数学计算	14	常见问题回答	28
应该看到的结果	15	习题 9 打印, 打印, 打印	29
巩固练习	15	应该看到的结果	29
常见问题回答	16	巩固练习	30
习题 4 变量和命名	17	常见问题回答	30
应该看到的结果	18	习题 10 那是什么	31
巩固练习	18	应该看到的结果	32
常见问题回答	19	转义序列	32
习题 5 更多的变量和打印	20	巩固练习	33
应该看到的结果	21		

常见问题回答	33	习题 18 命名、变量、代码和函数	53
习题 11 提问	34	应该看到的结果	54
应该看到的结果	34	巩固练习	55
巩固练习	35	常见问题回答	55
常见问题回答	35	习题 19 函数和变量	56
习题 12 提示别人	36	应该看到的结果	57
应该看到的结果	36	巩固练习	57
巩固练习	36	常见问题回答	57
常见问题回答	37	习题 20 函数和文件	59
习题 13 参数、解包和变量	38	应该看到的结果	60
等一下!“特性”还有另外一个名字	38	巩固练习	60
应该看到的结果	39	常见问题回答	60
巩固练习	40	习题 21 函数可以返回某些东西	62
常见问题回答	40	应该看到的结果	63
习题 14 提示和传递	41	巩固练习	63
应该看到的结果	41	常见问题回答	64
巩固练习	42	习题 22 到现在为止你学到了什么	65
常见问题回答	42	学到的东西	65
习题 15 读取文件	44	习题 23 字符串、字节串和字符编码	66
应该看到的结果	45	初始研究	66
巩固练习	45	开关、约定和编码	68
常见问题回答	46	解剖输出	69
习题 16 读写文件	47	解剖代码	70
应该看到的结果	48	深度接触编码	72
巩固练习	49	破坏程序	72
常见问题回答	49	习题 24 更多的练习	74
习题 17 更多文件操作	50	应该看到的结果	75
应该看到的结果	51	巩固练习	75
巩固练习	51	常见问题回答	76
常见问题回答	52	习题 25 更多更多的练习	77

应该看到的结果	78	习题 33 while 循环	98
巩固练习	79	应该看到的结果	99
常见问题回答	80	巩固练习	99
习题 26 恭喜你, 现在可以考试了!	81	常见问题回答	100
常见问题回答	81	习题 34 访问列表的元素	101
习题 27 记住逻辑关系	82	巩固练习	102
逻辑术语	82	习题 35 分支和函数	103
真值表	83	应该看到的结果	105
常见问题回答	84	巩固练习	105
习题 28 布尔表达式练习	85	常见问题回答	105
应该看到的结果	86	习题 36 设计和调试	107
巩固练习	87	if 语句的规则	107
常见问题回答	87	循环的规则	107
习题 29 if 语句	88	调试的小技巧	108
应该看到的结果	89	家庭作业	108
巩固练习	89	习题 37 复习各种符号	109
常见问题回答	89	关键字	109
习题 30 else 和 if	90	数据类型	110
应该看到的结果	91	字符串转义序列	111
巩固练习	91	老式字符串格式	111
常见问题回答	91	运算符	112
习题 31 作出决定	92	阅读代码	113
应该看到的结果	93	巩固练习	114
巩固练习	93	常见问题回答	114
常见问题回答	94	习题 38 列表的操作	115
习题 32 循环和列表	95	应该看到的结果	117
应该看到的结果	96	列表可以做什么	117
巩固练习	97	什么时候使用列表	118
常见问题回答	97	巩固练习	118
		常见问题回答	119
		习题 39 字典, 可爱的字典	120

字典的例子	121	应该看到的结果	153
应该看到的结果	123	巩固练习	153
字典可以做什么	124	常见问题回答	154
巩固练习	124	<b>习题 44 继承与组合</b>	155
常见问题回答	124	什么是继承	155
<b>习题 40 模块、类和对象</b>	125	要用 <code>super()</code> 的原因	160
模块和字典差不多	125	组合	160
类和模块差不多	126	继承和组合的应用场合	162
对象和 <code>import</code> 差不多	127	巩固练习	162
获取某样东西里包含的东西	128	常见问题回答	162
第一个类的例子	128	<b>习题 45 你来制作一款游戏</b>	164
应该看到的结果	129	评价你的游戏	164
巩固练习	129	函数的风格	165
常见问题回答	130	类的风格	165
<b>习题 41 学习面向对象术语</b>	131	代码风格	166
专有词汇练习	131	好的注释	166
措辞练习	131	为你的游戏评分	166
混合巩固练习	132	<b>习题 46 项目骨架</b>	168
阅读测试	132	macOS/Linux 配置	168
练习从语言到代码	134	Windows 10 配置	170
阅读更多代码	135	创建骨架项目目录	171
常见问题回答	135	测试你的配置	173
<b>习题 42 对象、类及从属关系</b>	136	使用这个骨架	174
代码写成什么样子	137	小测验	174
关于 <code>class Name(object)</code>	139	常见问题回答	174
巩固练习	139	<b>习题 47 自动化测试</b>	176
常见问题回答	139	编写测试用例	176
<b>习题 43 基本的面向对象分析和设计</b>	141	测试指南	178
简单游戏引擎的分析	142	应该看到的结果	178
自顶向下与自底向上	146	巩固练习	179
《来自 Percal 25 号行星的哥顿人》 的代码	146	常见问题回答	179
		<b>习题 48 用户输入进阶</b>	180



我们的游戏词汇	180	习题 52 创建 Web 游戏	207
测试优先挑战	182	重构习题 43 中的游戏	207
应该测试的东西	183	创建引擎	212
巩固练习	185	期末考试	215
常见问题回答	185	常见问题回答	215
<b>习题 49 创建句子</b>	186	<b>接下来的路</b>	217
match 和 peek	186	怎样学习任何一种编程语言	218
句子的语法	187	<b>老程序员的建议</b>	219
关于异常	187	<b>附录 命令行快速入门</b>	221
语法分析器代码	187	简介：废话少说，命令行来也	221
尝试语法分析器	190	练习 1 准备工作	222
应该测试的东西	191	练习 2 路径、文件夹和 目录 (pwd)	225
巩固练习	191	练习 3 如果你迷失了	227
常见问题回答	191	练习 4 创建目录 (mkdir)	228
<b>习题 50 你的第一个网站</b>	192	练习 5 更改目录 (cd)	230
安装 flask	192	练习 6 列出目录中的内容 (ls)	234
写一个简单的“Hello World”项目	192	练习 7 删除目录 (rmdir)	238
发生了什么	194	练习 8 在多个目录中切换 (pushd 和 popd)	241
修正错误	194	练习 9 创建空文件 (touch/New-Item)	243
创建基本的模板文件	195	练习 10 复制文件 (cp)	244
巩固练习	197	练习 11 移动文件 (mv)	248
常见问题回答	197	练习 12 查看文件内容 (less/more)	250
<b>习题 51 从浏览器中获取输入</b>	198	练习 13 流文件内容显示 (cat)	251
Web 的工作原理	198	练习 14 删除文件 (rm)	252
表单的工作原理	200	练习 15 退出终端 (exit)	254
创建 HTML 表单	201	命令行接下来的路	255
创建布局模板	203		
为表单撰写自动测试代码	205		
巩固练习	206		
破坏程序	206		