

尽享5小时完整视频教程！跟着数十万人的Ruby导师学Ruby！

PEARSON

“笨办法”
学 Ruby

Learn **RUBY**
the **HARD WAY**
THIRD EDITION

(第3版)

[美] 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 Ruby

Learn **RUBY**
the **HARD WAY**
THIRD EDITION

(第3版)

[美] Zed A. Shaw 著
王巍巍 译

人民邮电出版社
北京

图书在版编目 (C I P) 数据

“笨办法”学Ruby：第3版 / (美) 肖 (Shaw, Z. A.)
著；王巍巍译. — 北京：人民邮电出版社，2015. 12
书名原文：Learn Ruby the Hard Way, Third
Edition
ISBN 978-7-115-40734-4

I. ①笨… II. ①肖… ②王… III. ①计算机网络—
程序设计 IV. ①TP393.09

中国版本图书馆CIP数据核字(2015)第270985号

内 容 提 要

本书是一本 Ruby 入门书籍，是《“笨办法”学 Python》的姊妹篇。本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 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，然后再做巩固练习。

-
- ◆ 著 [美] Zed A. Shaw
 - 译 王巍巍
 - 责任编辑 杨海玲
 - 责任印制 张佳莹 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>
 - 北京艺辉印刷有限公司印刷
 - ◆ 开本：800×1000 1/16
 - 印张：17.5
 - 字数：391 千字 2015 年 12 月第 1 版
 - 印数：1-2 500 册 2015 年 12 月北京第 1 次印刷
 - 著作权合同登记号 图字：01-2014-8589 号
-

定价：49.00 元

读者服务热线：(010)81055410 印装质量热线：(010)81055316
反盗版热线：(010)81055315

版权声明

Authorized translation from the English language edition, entitled *Learn Ruby the Hard Way, Third Edition*, 9780321884992 by Zed A. Shaw, published by Pearson Education, Inc., publishing as Addison-Wesley, Copyright © 2015 by Pearson Education, Inc.

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 © 2015.

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

本书是《“笨办法”学 Python》的姊妹篇。如果你的编程基础为零，正打算通过 Ruby 学习编程，那么这本书就非常适合你。本书的结构和难度与《“笨办法”学 Python》相似，最大的不同是将编程语言由 Python 换成了 Ruby。所以，如果你看过了《“笨办法”学 Python》，就没必要看这本书了。

Ruby 是一门影响深远且易于入门的编程语言。本书作者是 Ruby 圈子里的明星人物，他之所以出名，一方面是因为他在开源软件方面的贡献，另一方面是因为他对一些开源项目的尖刻批评。由此可见他是一个典型的水平高、脾气大的人。不过，在写入门书籍的时候，他还是很平易近人的。

作者将“Learn X the Hard Way”（“笨办法”学编程）系列书籍全部直接发布到了网上，通过 <http://learnrubythehardway.org/> 可以阅读到本书的大部分内容，只不过是英文版的。截至目前，本书已经发布到了第 3 版。第 1 版只是基本的编程入门，第 2 版重写了若干章节，并加入了面向对象以及 Web 应用开发等相关的内容，第 3 版中作者根据学生的反馈，在各章中添加了“常见问题回答”，并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。

如果你对本书有些看法，或者有问题要问，欢迎通过新浪微博@王 joel 或者邮件 wangweiwei@outlook.com 与我联系。

感谢

感谢来自台湾的 xdite 网友，他曾经翻译过本书的第 1 版。

感谢人民邮电出版社勤劳而又专业的编辑们对本书的审稿和校对。

译者简介

王巍巍是一名受软件和编程的吸引中途转行的软件从业人员。写代码和翻译是他的两大爱好，他还喜欢在网上撰写和翻译一些文章。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，请通过微博（<http://weibo.com/gastlygem>）或者电子邮件（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“笨办法”（Hard Way）学习写代码，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，使你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，甚至在学习基本的数学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 Ruby 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本书了。她帮我修订了第 1 版书稿，而且在我写书的过程中给了我极大的支持。

我还要感谢 Rob Sobers，他建议我基于 Python 的书再做一个 Ruby 版，并且做了由 Python 到 Ruby 的初始转换工作。

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勘误的读者。

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，你就可以为自己读下一本编程书打下良好的基础。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 3 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，那你学习编程也会有问题。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，会让你知道各种符号的名称，熟悉它们的用处，最终读懂这门编程语言。

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正在做的事情的细节上。

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，立即就可以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、bug 以及其他问题更加敏感。

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、大脑和思维，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。

使用附带的视频

本书附带了 5 小时以上的视频教程^①，用来帮助你学完这本书。每个习题都有一个对应的视频，其中要么是对于习题的演示，要么是教你一些完成习题的小技巧。使用视频最好的方法就是，当你在书里边遇到困难或者完成习题后复习时去看看。这样会让你逐渐摆脱对视频教程的依赖，并培养出直接通过书籍学习的能力。只要坚持这样做，慢慢你就不需要视频教程了，不管是本书的视频还是别的视频，学编程时你都不需要了。你会具备直接通过阅读获取信息的能力。

关于坚持练习的一点提示

你通过这本书学编程时，我正在学弹吉他。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

^① 本书附带视频请在 www.epubit.com.cn 的本书页面上下载。

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》这两本书之间的时间内，我对绘画产生了兴趣。在 39 岁的时候喜欢上了视觉艺术，然后就跟以前学习吉他、音乐、编程的时候一样，每天学习绘画。我搜集了诸多入门教材，照着书上的去做，每天都画一些东西出来，并且享受着学习的过程。我离“艺术家”还差得远，甚至连“画得好”都谈不上，不过现在我可以说我是“会画画”的了。在学习艺术的过程中，我用的就是这本书教编程的方法。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，不管最终学到什么程度，你都会从中获益的。

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“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，就像学习人类的语言一样，你会发现很难记住一些词和特殊符号的用法，而且会经常感到很迷茫，直到有一天，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 3 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

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

巩固练习	32	常见问题回答	50
常见问题回答	32	习题 18 命名、变量、代码和函数	52
习题 11 提问	33	应该看到的结果	53
应该看到的结果	34	巩固练习	54
巩固练习	34	常见问题回答	54
常见问题回答	34	习题 19 函数和变量	56
习题 12 提示用户输入数	35	应该看到的结果	57
应该看到的结果	35	巩固练习	57
巩固练习	36	常见问题回答	58
习题 13 参数、解包和变量	37	习题 20 函数和文件	59
应该看到的结果	37	应该看到的结果	60
巩固练习	38	巩固练习	60
常见问题回答	38	常见问题回答	60
习题 14 提示和传递	40	习题 21 函数可以返回某些东西	62
应该看到的结果	40	应该看到的结果	63
巩固练习	41	巩固练习	63
常见问题回答	41	常见问题回答	64
习题 15 读取文件	43	习题 22 到现在你学到了哪些东西	65
应该看到的结果	44	学到的东西	65
巩固练习	44	习题 23 阅读一些代码	66
常见问题回答	45	习题 24 更多练习	67
习题 16 读写文件	46	应该看到的结果	68
应该看到的结果	47	巩固练习	68
巩固练习	48	常见问题回答	68
常见问题回答	48	习题 25 更多更多的实践	70
习题 17 更多文件操作	49	应该看到的结果	71
应该看到的结果	49	巩固练习	73
巩固练习	50	常见问题回答	73

习题 26 恭喜你, 现在可以考试了!	74	巩固练习	95
常见问题回答	74	常见问题回答	95
习题 27 记住逻辑关系	75	习题 34 访问数组的元素	97
逻辑术语	75	巩固练习	98
真值表	76	习题 35 分支和函数	99
常见问题回答	77	应该看到的结果	101
习题 28 布尔表达式练习	78	巩固练习	102
应该看到的结果	79	常见问题回答	102
巩固练习	80	习题 36 设计和调试	103
常见问题回答	80	if 语句的规则	103
习题 29 if 语句	81	循环的规则	103
应该看到的结果	82	调试的小技巧	104
巩固练习	82	家庭作业	104
常见问题回答	82	习题 37 复习各种符号	105
习题 30 else 和 if	83	关键字	105
应该看到的结果	84	数据类型	106
巩固练习	84	字符串转义序列	107
常见问题回答	85	操作符	107
习题 31 作出决定	86	阅读代码	108
应该看到的结果	87	巩固练习	109
巩固练习	87	常见问题回答	109
常见问题回答	88	习题 38 数组的操作	110
习题 32 循环和数组	89	应该看到的结果	111
应该看到的结果	91	数组有什么功能	112
巩固练习	91	什么时候使用数组	112
常见问题回答	92	巩固练习	113
习题 33 while 循环	93	常见问题回答	113
应该看到的结果	94	习题 39 散列, 可爱的散列	114
		散列示例	116

应该看到的结果	117	简单游戏引擎的分析	144
散列的功能	118	把问题写下来或者画出来	144
创建自己的散列模块	119	摘录和研究关键概念	144
代码描述	122	为各种概念创建类层次结构图和 对象关系图	145
数组的三个等级	124	编写和运行各个类	146
应该看到的结果(之二)	124	重复和细化	148
如何选择散列或数组	125	自顶向下与自底向上	148
巩固练习	126	《来自 Percal 25 号行星的哥顿人》的 代码	149
常见问题回答	126	应该看到的结果	155
习题 40 模块、类和对象	127	巩固练习	156
模块和散列差不多	127	常见问题回答	156
类和模块差不多	129	习题 44 继承与合成	157
对象相当于导入	129	什么是继承	157
获取某样东西里包含的东西	131	隐式继承	158
第一个关于类的例子	131	显式覆盖	159
应该看到的结果	132	在运行前或运行后替换	160
巩固练习	132	三种方式组合使用	161
常见问题回答	132	super() 和 initialize 搭配使用	162
习题 41 学习面向对象术语	133	合成	163
单词练习	133	继承和合成的应用场合	165
语汇练习	133	巩固练习	166
混合巩固练习	134	常见问题回答	166
阅读测试	134	习题 45 你来制作一个游戏	167
练习从语言到代码	137	评价你的游戏	167
阅读更多代码	137	函数的风格	168
常见问题回答	137	类的风格	168
习题 42 对象、类及从属关系	138	代码风格	169
代码要怎么写	139	好的注释	169
巩固练习	141	为你的游戏评分	169
常见问题回答	141	习题 46 项目骨架	171
习题 43 基本的面向对象分析和 设计	143		

创建骨架项目目录	171	安装 Sinatra	195
最终目录结构	173	写一个简单的“Hello World”项目	196
测试你的配置	174	发生了什么	197
使用这个骨架	174	停止和重载 Sinatra	198
小测验	174	修正错误	198
常见问题回答	175	创建基本的模板文件	199
习题 47 自动化测试	176	巩固练习	200
编写测试用例	176	常见问题回答	200
测试指南	179	习题 51 从浏览器中获取输入	201
应该看到的结果	179	Web 的工作原理	201
巩固练习	180	表单的工作原理	203
常见问题回答	180	创建 HTML 表单	205
习题 48 进阶用户输入	181	创建布局模板	206
我们的游戏语汇	181	为表单撰写自动测试代码	207
断句	182	巩固练习	208
语汇元组	182	常见问题回答	208
扫描输入	182	习题 52 创建 Web 游戏	209
异常和数字	183	重构习题 43 中的游戏	209
挑战测试驱动开发	184	会话和用户跟踪	215
应该测试的东西	184	创建引擎	216
巩固练习	186	期末考试	218
常见问题回答	187	接下来的路	220
习题 49 创建句子	188	怎样学习任何一种编程语言	221
匹配和预览	188	老程序员的建议	222
句子的文法	189	附录 命令行快速入门	224
关于异常	189	简介：废话少说，命令行来也	224
解析器代码	189	如何使用这个附录	224
试用解析器	193	你需要发挥记忆力	225
应该测试的东西	193	准备工作	225
巩固练习	194	任务	226
常见问题回答	194	知识点	226
习题 50 你的第一个网站	195		

更多任务	227	知识点	248
路径、文件夹和目录 (pwd)	229	更多任务	249
任务	229	复制文件 (cp)	249
知识点	230	任务	249
更多任务	230	知识点	252
如果你迷失了	230	更多任务	252
任务	231	移动文件 (mv)	252
知识点	231	任务	253
创建目录 (mkdir)	231	知识点	255
任务	231	更多任务	255
知识点	233	查看文件内容 (less, MORE)	255
更多任务	233	任务	255
更改目录 (cd)	233	知识点	256
任务	233	更多任务	256
知识点	237	流文件内容显示 (cat)	256
更多任务	237	任务	256
列出目录下的内容 (ls)	237	知识点	257
任务	238	更多任务	257
知识点	241	删除文件 (rm)	258
更多任务	241	任务	258
删除路径 (rmdir)	242	知识点	259
任务	242	更多任务	260
知识点	244	退出终端 (exit)	260
更多任务	244	任务	260
在多个目录间切换 (pushd, popd)	245	知识点	260
任务	245	更多任务	260
知识点	247	命令行将来的路	261
更多任务	247	Unix bash 参考资料	261
创建空文件 (touch, New-Item)	247	PowerShell 参考资料	261
任务	248		

准备工作

这个习题并没有代码，它的主要目的是让你在计算机上安装好 Ruby。你应该尽量照着说明进行操作，例如，Mac OS X 已经安装了 Ruby 2，所以就不要再在上面安装别的 Ruby 版本了。

警告 如果你不知道怎样使用 Windows 下的 PowerShell、OS X 下的 Terminal 或者 Linux 下的 bash，那你就需要先学会一个。你应该先完成附录中的习题后，再回来继续下面这些习题。

Mac OS X

完成这个习题需要完成下列任务。

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 `ruby -v`，看看 Ruby 的版本。
7. 如果你使用的 `ruby` 版本小于 2.0，那么你的 Ruby 版本就太老了。这时你有 3 个选择。
 - a. 升级 OS X 到最新版本，现在 OS X 是免费的，所以你也找不到借口不升级了。
 - b. 到 <https://www.ruby-lang.org/en/downloads/> 这个网址，用那里提供的安装包安装一下。
 - c. 找个朋友帮你安装一下 Ruby 2.x 或者更高的版本。
8. 这样你就应该退回到敲 `ruby -v` 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创建一个文件。建立一个文件，使用“保存”（Save）