

**THE SELF-TAUGHT  
PROGRAMMER**

**The Definitive Guide to  
Programming Professionally**

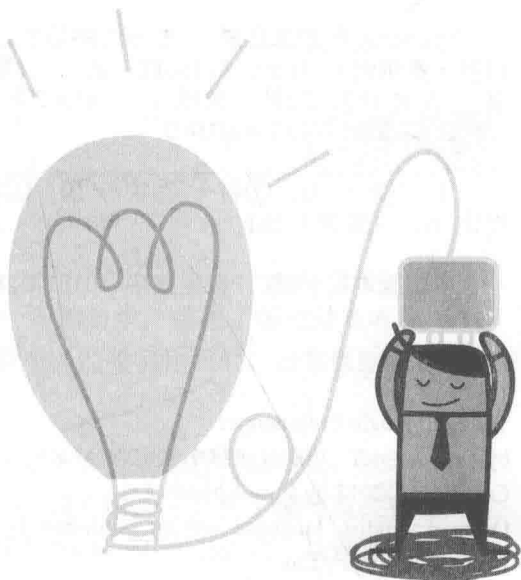
# Python 编程

**无师自通 | 专业程序员的养成**

[美] 科里·奥尔索夫 (Cory Allhoff) / 著

宋秉金 / 译

201.



## THE SELF-TAUGHT PROGRAMMER

The Definitive Guide to  
Programming Professionally

# Python 编程

## 无师自通 | 专业程序员的养成

[美] 科里·奥尔索夫 (Cory Allhoff) / 著

宋秉金 / 译

人民邮电出版社

北京

## 图书在版编目 (C I P) 数据

Python编程无师自通：专业程序员的养成 / (美)  
科里·奥尔索夫 (Cory Althoff) 著；宋秉金译。— 北  
京：人民邮电出版社，2019.1 (2019.2重印)  
ISBN 978-7-115-49710-9

I. ①P… II. ①科… ②宋… III. ①软件工具—程序  
设计 IV. ①TP311.561

中国版本图书馆CIP数据核字(2018)第283440号

## 版权声明

The Self-Taught Programmer

by Cory Althoff ISBN: 978-1-5202-8817-8

Copyright ©2017 by Cory Althoff.

Original English language edition published by arrangement with Hodgman Literary LLC, through The Grayhawk Agency Ltd.

Simplified Chinese-language edition Copyright ©2018 by Posts & Telecom Press.

All rights reserved.

本书中文简体字版由 Hodgman Literary LLC 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。

版权所有，侵权必究。

- 
- ◆ 著 [美] 科里·奥尔索夫 (Cory Althoff)
  - 译 宋秉金
  - 责任编辑 杨大可
  - 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>
  - 北京圣夫亚美印刷有限公司印刷
  
  - ◆ 开本：800×1000 1/16
  - 印张：16
  - 字数：288千字 2019年1月第1版
  - 印数：6001-7500册 2019年2月北京第3次印刷
- 著作权合同登记号 图字：01-2017-8621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

## 内容提要

---

本书作者是一名自学成才的程序员，经过一年的自学，掌握了编程技能并在 eBay 找到了一份软件工程师的工作。本书是作者结合个人经验写作而成，旨在帮助读者从外行成长为一名专业的 Python 程序员。

本书包括 5 部分内容：第一部分（第 1~11 章）介绍了编程基础知识，以及函数、容器、字符串、循环和模型等概念；第二部分（第 12~15 章）介绍了编程范式和面向对象编程等知识；第三部分（第 16~20 章）介绍了 Bash、正则表达式、包管理器和版本控制等编程工具方面的知识；第四部分（第 21~22 章）主要涉及数据结构和算法方面的知识；第五部分（第 23~27 章）则是关于求职与团队协作的内容。

本书可以满足几乎所有想要学习编程的初学者。本书适合高中、大学阶段想要自学编程的学生，以及其他行业想入门编程的人，同时也适合作为编程入门的培训教材。

## 致 谢

---

我要感谢所有在本书撰写、出版过程中给予帮助的人。我的父母艾比·奥尔索夫和詹姆斯·奥尔索夫，在整个过程中给了我极大的支持。我父亲一页一页地读完了本书，并提出了非常宝贵的反馈建议。没有他的帮助，这本书就不会问世。我女朋友劳伦·沃德也没有抱怨我把大部分时间花在写书上。我要感谢本书的插画师布莱克·鲍尔斯，本书的编辑史蒂夫·布什、麦德林·鲁斯、潘·瓦拉塔和劳伦斯·圣菲利波，以及我的朋友安东尼·辛都，我在书中引用了我们之间多次讨论的内容。我还要感谢兰迪·芬勒支持我在 Kickstarter 上发起的写作项目，并介绍潘·瓦拉塔给我认识。特别感谢我以前的领导安札·阿法克，在我加入 eBay 时那么地支持我。还要感谢所有参与审阅本书的读者，感谢你们提供的反馈。最后，我想感谢 Kickstarter 网站上所有支持本书项目的朋友，尤其是吉姆·春、萨尼·李和雷·福瑞斯特。非常感谢大家的支持！

# 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

本书提供如下资源：

- 本书源代码；
- 书中习题答案。

要获得以上配套资源，请在异步社区本书页面中点击 **配套资源**，跳转到下载界面，按提示进行操作即可。注意：为保证购书读者的权益，该操作会给出相关提示，要求输入提取码进行验证。

如果您是教师，希望获得教学配套资源，请在社区本书页面中直接联系本书的责任编辑。

## 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。

The screenshot shows a web form for submitting勘误 (勘误). At the top, there are three tabs: '详细信息' (Detailed Information), '写书评' (Write a Review), and '提交勘误' (Submit勘误), with the last one being active. Below the tabs, there are three input fields: '页码:' (Page Number), '页内位置 (行数):' (Page Position (Line Number)), and '勘误印次:' (勘误印次). Below these fields is a rich text editor with a toolbar containing icons for bold (B), italic (I), underline (U), strikethrough (ABC), bulleted list, numbered list, quote, link, and unlink. At the bottom right of the form, there is a '字数统计' (Character Count) label and a '提交' (Submit) button.

## 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，并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，为译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

## 第一部分 编程简介

第 1 章 概述	3	3.1 示例	15
1.1 本书的结构	4	3.2 注释	16
1.2 从终点出发	4	3.3 打印	17
1.3 你不是一个人在战斗	5	3.4 代码行	17
1.4 自学的优势	5	3.5 关键字	18
1.5 为什么应该编程	5	3.6 间距	18
1.6 坚持不懈	6	3.7 数据类型	19
1.7 本书的格式	6	3.8 常量和变量	21
1.8 本书使用的技术	6	3.9 语法	24
1.9 术语表	7	3.10 错误与异常	25
1.10 挑战练习	7	3.11 算术操作符	26
第 2 章 起步	9	3.12 比较操作符	29
2.1 什么是编程	9	3.13 逻辑操作符	31
2.2 什么是 Python	10	3.14 条件语句	34
2.3 安装 Python	10	3.15 语句	38
2.4 问题解答	11	3.16 术语表	40
2.5 交互式 shell	11	3.17 挑战练习	42
2.6 保存程序	12	第 4 章 函数	43
2.7 运行示例程序	12	4.1 表达概念	43
2.8 术语表	13	4.2 函数	44
2.9 挑战练习	13	4.3 定义函数	44
第 3 章 编程概论	15	4.4 内置函数	47
		4.5 复用函数	49
		4.6 必选及可选参数	51
		4.7 作用域	52
		4.8 异常处理	55



4.9 文档字符串 .....	58	第 7 章 循环 .....	91
4.10 有需要才使用变量 .....	58	7.1 for 循环 .....	91
4.11 术语表 .....	59	7.2 range 函数 .....	95
4.12 挑战练习 .....	59	7.3 while 循环 .....	95
<b>第 5 章 容器 .....</b>	<b>61</b>	7.4 break 语句 .....	96
5.1 方法 .....	61	7.5 continue 语句 .....	98
5.2 列表 .....	62	7.6 嵌套循环 .....	99
5.3 元组 .....	66	7.7 术语表 .....	100
5.4 字典 .....	69	7.8 挑战练习 .....	101
5.5 容器嵌套容器 .....	72	<b>第 8 章 模块 .....</b>	<b>103</b>
5.6 术语表 .....	76	8.1 导入内置模块 .....	103
5.7 挑战练习 .....	76	8.2 导入其他模块 .....	105
<b>第 6 章 字符串操作 .....</b>	<b>77</b>	8.3 术语表 .....	107
6.1 三引号字符串 .....	77	8.4 挑战练习 .....	107
6.2 索引 .....	77	<b>第 9 章 文件 .....</b>	<b>109</b>
6.3 字符串是不可变的 .....	79	9.1 写文件操作 .....	109
6.4 字符串拼接 .....	79	9.2 自动关闭文件 .....	110
6.5 字符串乘法 .....	80	9.3 读取文件 .....	111
6.6 改变大小写 .....	80	9.4 CSV 文件 .....	112
6.7 格式化 .....	80	9.5 术语表 .....	114
6.8 分割 .....	82	9.6 挑战练习 .....	114
6.9 连接 .....	82	<b>第 10 章 综合练习 .....</b>	<b>115</b>
6.10 去除空格 .....	83	10.1 Hangman .....	116
6.11 替换 .....	84	10.2 挑战练习 .....	119
6.12 查找索引 .....	84	<b>第 11 章 练习 .....</b>	<b>121</b>
6.13 in 关键字 .....	85	11.1 阅读 .....	121
6.14 字符串转义 .....	85	11.2 其他资源 .....	121
6.15 换行符 .....	86	11.3 寻求帮助 .....	121
6.16 切片 .....	87		
6.17 术语表 .....	88		
6.18 挑战练习 .....	89		

## 第二部分 面向对象编程简介

第 12 章 编程范式	125
12.1 状态	125
12.2 过程式编程	125
12.3 函数式编程	127
12.4 面向对象编程	128
12.5 术语表	133
12.6 挑战练习	134
第 13 章 面向对象编程的 四大支柱	135
13.1 封装	135
13.2 抽象	138
13.3 多态	138
13.4 继承	140
13.5 组合	143
13.6 术语表	144
13.7 挑战练习	144
第 14 章 深入面向对象编程	145
14.1 类变量与实例变量	145
14.2 魔法方法	147
14.3 is	149
14.4 术语表	150
14.5 挑战练习	150
第 15 章 综合练习	151
15.1 卡牌	151
15.2 牌堆	153
15.3 玩家	154
15.4 游戏	155
15.5 战争	157

## 第三部分 编程工具简介

第 16 章 Bash	163
16.1 跟随练习	163
16.2 找到 Bash	164
16.3 命令	164
16.4 最近命令	165
16.5 相对路径与绝对路径	165
16.6 导航	166
16.7 旗标	168
16.8 隐藏文件	169
16.9 管道	169
16.10 环境变量	169
16.11 用户	170
16.12 了解更多	171
16.13 术语表	171
16.14 挑战练习	172
第 17 章 正则表达式	173
17.1 初始配置	173
17.2 简单匹配	174
17.3 匹配起始位置	176
17.4 匹配多个字符	177
17.5 匹配数字	178
17.6 重复	179
17.7 转义	181
17.8 正则工具	182
17.9 术语表	182
17.10 挑战练习	183
第 18 章 包管理器	185
18.1 包	185
18.2 pip	185

18.3	虚拟环境	188	22.2	顺序搜索	218
18.4	术语表	188	22.3	回文词	219
18.5	挑战练习	188	22.4	变位词	220
<b>第 19 章</b>	<b>版本控制</b>	<b>189</b>	22.5	计算字母频数	220
19.1	代码仓库	189	22.6	递归	221
19.2	入门	190	22.7	术语表	223
19.3	推送和拉取	191	22.8	挑战练习	224
19.4	推送示例	192	<b>第五部分 找到工作</b>		
19.5	拉取示例	195	<b>第 23 章</b>	<b>最佳编程实践</b>	<b>227</b>
19.6	回退版本	195	23.1	写代码是最后的手段	227
19.7	diff	196	23.2	DRY	227
19.8	下一步	197	23.3	正交性	227
19.9	术语表	197	23.4	每个数据都只应保存在 一处	228
19.10	挑战练习	198	23.5	函数只做一件事	228
<b>第 20 章</b>	<b>融会贯通</b>	<b>199</b>	23.6	若耗费时间过长, 你的做法 很可能就是错的	228
20.1	HTML	199	23.7	第一次就要用最佳的方法 完成	228
20.2	爬取 Google 新闻	200	23.8	遵循惯例	229
20.3	术语表	204	23.9	使用强大的 IDE	229
20.4	挑战练习	204	23.10	记录日志	230
<b>第四部分 计算机科学简介</b>			23.11	测试	230
<b>第 21 章</b>	<b>数据结构</b>	<b>207</b>	23.12	代码审查	230
21.1	数据结构	207	23.13	安全	231
21.2	栈	207	23.14	术语表	232
21.3	使用栈逆转字符串	210	<b>第 24 章</b>	<b>第一份编程工作</b>	<b>233</b>
21.4	队列	211	24.1	选择方向	233
21.5	购票队列	213	24.2	积累初期经验	234
21.6	术语表	215	24.3	拿到面试机会	234
21.7	挑战练习	216	24.4	面试	234
<b>第 22 章</b>	<b>算法</b>	<b>217</b>			
22.1	FizzBuzz	217			

---

24.5	面试技巧	235	26.1	经典书籍	239
<b>第 25 章</b>	<b>团队协作</b>	<b>237</b>	26.2	在线课堂	239
25.1	掌握基础	237	26.3	骇客新闻	240
25.2	提问前请先搜索	237	<b>第 27 章</b>	<b>下一步</b>	<b>241</b>
25.3	修改代码	238	27.1	找到导师	241
25.4	冒名顶替综合征	238	27.2	加深理解	241
<b>第 26 章</b>	<b>更多学习资料</b>	<b>239</b>	27.3	其他建议	242

# 第一部分 编程简介

---

## 本部分内容

- 第1章 概述
- 第2章 起步
- 第3章 编程概论
- 第4章 函数
- 第5章 容器
- 第6章 字符串操作
- 第7章 循环
- 第8章 模块
- 第9章 文件
- 第10章 综合练习
- 第11章 练习



# 第 1 章

## 概述

---

“大多数优秀的程序员从事编程工作，不是因为期望获得报酬或得到公众的称赞，而是因为编程是件有趣的事儿。”

——林纳斯·托瓦兹（Linus Torvalds）

我毕业于克莱门森大学政治学专业。我曾考虑过是否选择学习计算机专业，还在大一那年报名参加了“编程概论”课程，不过很快就退出了。实在是太难了。毕业后我一直住在硅谷，我发现我需要学习编程。一年后，我成为了 eBay 公司的一名中级软件工程师（介于初级工程师与高级工程师之间的一个职位）。我不想让大家觉得这是很轻松就能做到的。实际上，这是极具挑战的一件事。在这一年的不断尝试努力过程中，我得到了很多乐趣。

刚开始，我学习的是如何用流行的编程语言 Python 来进行编程。但是本书不仅是教你如何使用某种特定的语言编程（确实会有这方面的内容），还会介绍标准教材中所不包括的其他所有知识点。本书分享的是我在成为软件工程师过程中不得不自学的内容。本书不适合那些想要随意了解下编程知识、将写代码作为爱好的人，而是专门写给那些希望以编程为职业的人。不管你的目标是成为一名软件工程师、企业家，还是在其他的岗位上使用编程技能，你都是本书的目标读者。

学会一门编程语言还不够，你还需要学会其他技能，才能像计算机科学家一样地工作。我会教授大家我从编程新手到专业软件工程师过程中学到的一切。我写作本书，是为了向有志于编程岗位的人分享他们需要掌握的知识框架。编程概论的书籍都大同小异——用 Python 或 Ruby 介绍编程的基础知识，然后就让你自己摸索。我经常从读完类似书籍的朋友那听到这样的反馈：我现在该做什么？我还不是程序员，也不知道下一步该学什么。本书，就是我给出的答案。

## 1.1 本书的结构

本书中一章所涵盖的许多主题可能都可以独立成书。我的目标不是包罗你需要了解的每个主题的所有细节，而是提供一份指引——一个编程职业发展所需要的所有技能的导览。

第一部分：编程简介。让你尽快写出自己的第一个程序，最好在今天。

第二部分：面向对象编程简介。这部分将介绍不同的编程范式，着重阐述面向对象编程。你会开发一个游戏，体会编程的强大能力。读完这部分后你会沉迷于编程。

第三部分：编程工具简介。将介绍提升编程生产力的不同工具。这时，你已经沉迷于编程，并希望变得更好。你将会学习相关的操作系统、使用正则表达式提升效率、安装并管理他人的程序，以及使用版本控制与其他工程师协作的知识。

第四部分：计算机科学简介。将简要介绍计算机科学知识，主要涵盖两个主题——算法和数据结构。

第五部分：找到工作。最后一部分是关于最佳编程实践，如何找到软件工程师的工作，团队协作以及程序员的自我提升。我会分享如何通过技术面试与团队协作的建议，以及如何进一步提升自己的技能。

## 1.2 从终点出发

我学会编程的方式，与计算机科学通常的教学方式正好相反。本书的结构是根据我自己的方式组织的。一般来说，你会先花很多时间学习理论，理论知识学的太多以至于许多计算机科学的毕业生甚至不知道如何动手编程。杰夫·阿特伍德（Jeff Atwood），在其博客“为什么程序员不会编程”中写道：“和我一样，许多人都碰到了这样的情况，编程岗位的 200 位申请者中，有 199 个根本不会写代码。重申一遍：他们一点代码都不会写。”这种现象直接促使 Atwood 发明了 FizzBuzz 代码挑战，一种用来在面试中筛选申请者的编程测试。大部分人都通不过测试，这也是为什么你要学习本书并掌握实践中要使用到技能。放心吧，在本书中你还会学到如何通过 FizzBuzz 测试的。

《王者之旅》电影中的主角乔什（Josh Waitzkin），在《学习的艺术》一书中回忆了他如何反向学习国际象棋。他没有和其他人一样研究开局，而是从学习象棋残局（棋盘上只剩下少数几个棋子）开始。这样做让他对国际象棋有了更深的理解，并赢得了多次大赛冠军。与此类似，我认为先学习如何编程再学习理论的方法更高效，因为你会拥有



了解背后原理的强烈驱动。这就是为什么本书一直到第五部分才介绍计算机科学理论，而且内容也尽量精简。虽然理论很重要，但是在你拥有了编程经验之后，理论的价值才更大。

## 1.3 你不是一个人在战斗

毕业后再学习编程，已经越来越常见。Stack Overflow（一个程序员在线社区）在 2015 年的一份调查中显示，48% 的受访者没有计算机科学学位。

## 1.4 自学的优势

在 eBay 工作期间，我的团队中有从斯坦福大学、加州大学和杜克大学计算机科学专业毕业的程序员，还有两名物理学博士。当时我 25 岁，而年仅 21 岁的同事对编程和计算机科学的知识比我强 10 倍这个事实，让我尤其惶恐。

虽然与拥有计算机科学学士、硕士甚至是博士学位的同事一起工作的压力很大，但别忘记了你还拥有“自学的优势”。你选择读这本书，不是出于老师布置的任务，而是因为你内心学习的渴望，这一点是你所拥有的最大优势。苹果公司的创始人史蒂夫·沃兹尼亚克（Steve Wozniak）就是一位自学成才的程序员；还有因在美国宇航局的阿波罗登月计划中做出卓越贡献而获得总统自由勋章的玛格丽特·汉密尔顿（Margaret Hamilton）；还有 Tumblr 的创始人大卫·卡普（David Karp），Twitter 的创始人杰克·多西（Jack Dorsey），Instagram 的创始人凯文·斯特罗姆（Kevin Systrom），他们都是自学成才的程序员。

## 1.5 为什么应该编程

不管你从事什么工作，编程都有助于你的职业发展。学习编程将给你自己赋能。我喜欢尝试新想法，时刻都有希望启动的新项目。学会编程后，我就可以坐下来自己实现，而不需要依赖他人。

编程也会提升你在其他方面的技能。因为你熟练掌握了问题解决能力，鲜有其他工作不会因此而受益。我最近要在 Craigslist 上租房，搜索并筛选房子是个非常费力的活儿。但是我写了一个程序来代替我搜索，最后将结果以邮件形式发送给我。学会编程，将把你从重复性工作中解放出来。

如果你想成为软件工程师，市场上对这类岗位的需求也日益增长，但是符合要求的