

# Python 爬虫开发

## 从入门到实战

微课版

*Python Crawler Development*



乾坤 著

29 个在线微课视频配合图书同步讲解

- 精心挑选简单实用的案例，即学即练
- 附赠案例源代码

读者交流QQ群  
398687538

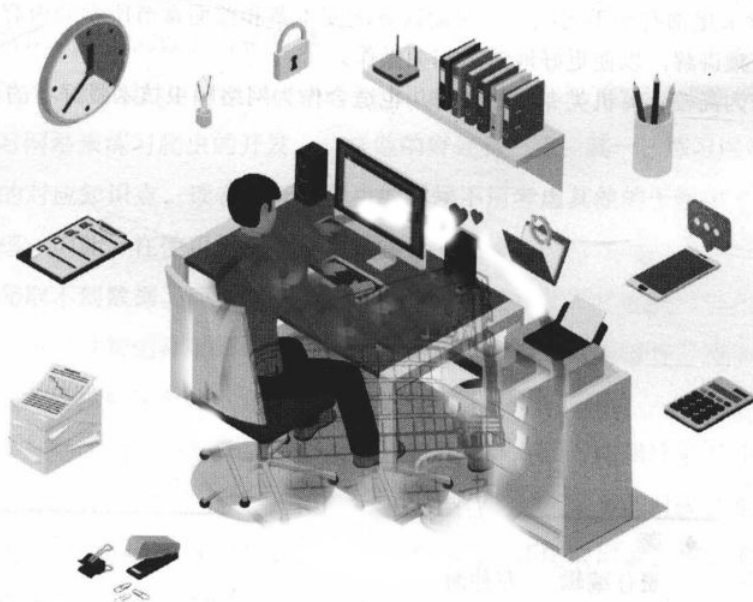


# Python爬虫开发

## 从入门到实战

微课版

*Python Development*



谢乾坤 著

人民邮电出版社

北京

## 图书在版编目 (CIP) 数据

Python爬虫开发从入门到实战：微课版 / 谢乾坤著

— 北京：人民邮电出版社，2018.9 (2019.8重印)

(互联网+职业技能系列)

ISBN 978-7-115-49099-5

I. ①P… II. ①谢… 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185416号

## 内 容 提 要

本书较为全面地介绍了定向爬虫的开发过程、各种反爬虫机制的破解方法和爬虫开发的相关技巧。全书共13章，包括绪论、Python基础、正则表达式与文件操作、简单的网页爬虫开发、高性能HTML内容解析、Python与数据库、异步加载与请求头、模拟登录与验证码、抓包与中间人爬虫、Android原生App爬虫、Scrapy、Scrapy高级应用、爬虫开发中的法律和道德问题等。除第1、12、13章外的其他章末尾都有动手实践，以帮助读者巩固本章和前面章节所学的内容。针对书中的疑难内容，还配有视频讲解，以便更好地演示相关操作。

本书适合作为高校计算机类专业的教材，也适合作为网络爬虫技术爱好者的自学参考书。

- 
- ◆ 著 谢乾坤  
责任编辑 左仲海  
责任印制 马振武
  - ◆ 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>  
固安县铭成印刷有限公司印刷
  - ◆ 开本：787×1092 1/16  
印张：17 2018年9月第1版  
字数：582千字 2019年8月河北第4次印刷

---

定价：49.80元

读者服务热线：(010)81055256 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

# 前言

## Preface

我在上大学的时候开始学习 Python，使用 Python 做的第一个项目就是学校教务处爬虫，用来爬取教务处的各种通知并导入到微信公众号中。在对爬虫开发比较熟练以后，我在淘宝上开了一个店铺用来承接各种爬虫的私活。我的店铺是淘宝上面第一个爬虫开发的店铺。

在我工作以后，极客学院联系我，让我作为布道师在极客学院上讲授爬虫开发的课程。这些课程就是本书内容的前身。

本书适用于有一定编程基础的读者。虽然第 2 章讲解了 Python 3 的基础知识，但是由于 Python 博大精深，为了覆盖爬虫开发中的各种知识，所以自然需要省略一些细节上的内容。因此，如果读者有一定的编程基础与开发常识，那么阅读本书将会事半功倍。

本书提供了练习网站，其地址为 <http://exercise.kingname.info/>。建议读者在学习本书的时候，根据书上的提示使用练习网站来练习爬虫的开发。这样做的好处有三点：其一，练习网站针对每一章开发，专门用于练习这一章的对应知识点，读者在开发爬虫的时候不用考虑其他的干扰因素。其二，定向爬虫对网站的改版较为敏感，因此，在使用第三方网站做例子的时候，一旦网站出现了改版，如果读者照搬本书的代码就会导致爬取不到数据。而如果使用练习网站，即便读者完全照搬本书的代码也可以保证爬虫成功运行。其三，在极客学院的视频课程中，我曾经使用一个第三方网站作为爬虫开发作业，由于视频课程的学生众多，大家都在爬这个网站导致网站承受不住压力被迫关闭。

本书在阶段练习中依然使用了一些第三方网站作为练习目标，读者在阅读本书并进行练习的时候，一定要注意学习书中讲到的分析方法，而不是照抄代码。当读者读到本书的时候，距离本书编写的相应网站爬虫应该已经过去了一段时间，所以如果根据书中的代码无法爬取网站，那么不要惊慌，仔细阅读书中的思路和方法，相信你一定可以重新爬取到数据。

在本书的构思和写作过程中，我得到了很多老师、同行和朋友的帮助。在此要感谢极客学院，本书内容脱胎于我在极客学院的爬虫系列视频课程，通过极客学院同学的反馈，我才能从视频课程里面总结和提炼出本书的内容；也要感谢极客学院的大静和温泉，在我录制视频课程的过程中对视频和文档进行认真细致的审核；还要感谢 Linda，积极联系出版社，从而可以把这个爬虫系列视频课程整理出版为实体书。

另外要感谢我的学生老贤和魏鹏。在爬虫练习网站的开发过程中，魏鹏亲自测试了每一个练习页面，并针对每一个练习页面开发了对应的爬虫，以确认该页面所涉及的爬虫知识没有超出本书的范畴。

最后，感谢我家人的督促，让我克服了拖延症。这本书也帮助我遇见了我的爱人杨小姐。希望这本书可以为学习爬虫的各位读者提供一些帮助。如果读者在阅读本书的过程中有什么疑问或者建议，欢迎加入读者交流 QQ 群：398687538。

编者

2017 年 12 月

# 目录

## Contents

<b>第 1 章 绪论</b>	<b>1</b>		
1.1 爬虫	2		
1.2 爬虫可以做什么	2		
1.2.1 收集数据	2		
1.2.2 尽职调查	3		
1.2.3 刷流量和秒杀	3		
1.3 爬虫开发技术	4		
<b>第 2 章 Python 基础</b>	<b>5</b>		
2.1 Python 的安装和运行	6		
2.1.1 在 Windows 中安装 Python	6		
2.1.2 在 Mac OS 中安装 Python	7		
2.1.3 在 Linux 中安装 Python	8		
2.2 Python 开发环境	8		
2.2.1 PyCharm 介绍与安装	8		
2.2.2 运行代码	9		
2.3 Python 的数据结构和控制结构	12		
2.3.1 整数、浮点数和变量	12		
2.3.2 字符串、列表、元组	13		
2.3.3 数据的读取	14		
2.3.4 字典与集合	16		
2.3.5 条件语句	17		
2.3.6 for 循环与 while 循环	19		
2.4 函数与类	21		
2.4.1 函数	21		
2.4.2 类与面向对象编程	28		
2.5 阶段案例——猜数游戏	32		
2.5.1 需求分析	32		
2.5.2 核心代码构建	33		
2.5.3 调试与运行	33		
2.6 本章小结	34		
2.7 动手实践	34		
<b>第 3 章 正则表达式与文件操作</b>	<b>35</b>		
3.1 正则表达式	36		
3.1.1 正则表达式的基本符号	36		
3.1.2 在 Python 中使用正则表达式	38		
3.1.3 正则表达式提取技巧	41		
3.2 Python 文件操作	44		
3.2.1 使用 Python 读/写文本文件	44		
3.2.2 使用 Python 读/写 CSV 文件	46		
3.3 阶段案例——半自动爬虫开发	49		
3.3.1 需求分析	49		
3.3.2 核心代码构建	50		
3.3.3 调试与运行	51		
3.4 本章小结	54		
3.5 动手实践	54		
<b>第 4 章 简单的网页爬虫开发</b>	<b>55</b>		
4.1 使用 Python 获取网页源代码	56		
4.1.1 Python 的第三方库	56		
4.1.2 requests 介绍与安装	56		
4.1.3 使用 requests 获取网页源代码	57		
4.1.4 结合 requests 与正则表达式	59		
4.2 多线程爬虫	60		
4.2.1 多进程库 (multiprocessing)	60		
4.2.2 开发多线程爬虫	61		
4.3 爬虫的常见搜索算法	62		
4.3.1 深度优先搜索	62		

4.3.2	广度优先搜索	63
4.3.3	爬虫搜索算法的选择	64
4.4	阶段案例——小说网站爬虫开发	64
4.4.1	需求分析	64
4.4.2	核心代码构建	65
4.4.3	调试与运行	68
4.5	本章小结	68
4.6	动手实践	68

## 第5章 高性能HTML内容解析 69

5.1	HTML 基础	70
5.2	XPath	71
5.2.1	XPath 的介绍	71
5.2.2	lxml 的安装	71
5.2.3	XPath 语法讲解	73
5.2.4	使用 Google Chrome 浏览器辅助构造 XPath	77
5.3	Beautiful Soup4	81
5.3.1	BS4 的安装	81
5.3.2	BS4 语法讲解	82
5.4	阶段案例——大麦网演出爬虫	85
5.4.1	需求分析	85
5.4.2	核心代码构建	85
5.4.3	调试与运行	86
5.5	本章小结	87
5.6	动手实践	87

## 第6章 Python与数据库 88

6.1	MongoDB	89
6.1.1	MongoDB 的安装	89
6.1.2	PyMongo 的安装与使用	94
6.1.3	使用 RoboMongo 执行 MongoDB 命令	101
6.2	Redis	102
6.2.1	环境搭建	102
6.2.2	Redis 交互环境的使用	103
6.2.3	Redis-py	104

6.3	MongoDB 的优化建议	105
6.3.1	少读少写少更新	105
6.3.2	能用 Redis 就不用 MongoDB	106
6.4	阶段案例	107
6.4.1	需求分析	107
6.4.2	核心代码构建	107
6.4.3	调试与运行	108
6.5	本章小结	108
6.6	动手实践	108

## 第7章 异步加载与请求头 109

7.1	异步加载	110
7.1.1	AJAX 技术介绍	110
7.1.2	JSON 介绍与应用	110
7.1.3	异步 GET 与 POST 请求	111
7.1.4	特殊的异步加载	113
7.1.5	多次请求的异步加载	114
7.1.6	基于异步加载的简单登录	117
7.2	请求头 (Headers)	118
7.2.1	请求头的作用	118
7.2.2	伪造请求头	119
7.3	模拟浏览器	122
7.3.1	Selenium 介绍	123
7.3.2	Selenium 安装	124
7.3.3	Selenium 的使用	124
7.4	阶段案例	128
7.4.1	需求分析	128
7.4.2	核心代码构建	128
7.4.3	调试与运行	130
7.5	本章小结	131
7.6	动手实践	131

## 第8章 模拟登录与验证码 132

8.1	模拟登录	133
8.1.1	使用 Selenium 模拟登录	133
8.1.2	使用 Cookies 登录	135
8.1.3	模拟表单登录	137

8.2 验证码	139
8.2.1 肉眼打码	139
8.2.2 自动打码	141
8.3 阶段案例——自动登录果壳网	144
8.3.1 需求分析	144
8.3.2 核心代码构建	145
8.3.3 运行与调试	146
8.4 本章小结	147
8.5 动手实践	147

## 第9章 抓包与中间人爬虫 148

9.1 数据抓包	149
9.1.1 Charles 的介绍和使用	149
9.1.2 App 爬虫和小程序爬虫	156
9.2 中间人爬虫	163
9.2.1 mitmproxy 的介绍和安装	163
9.2.2 mitmproxy 的使用	164
9.2.3 使用 Python 定制 mitmproxy	165
9.3 阶段案例——Keep 热门	170
9.3.1 需求分析	170
9.3.2 核心代码构建	170
9.3.3 调试运行	172
9.4 本章小结	172
9.5 动手实践	172

## 第10章 Android 原生 App 爬虫 173

10.1 实现原理	174
10.1.1 环境搭建	175
10.1.2 使用 Python 操纵手机	178
10.1.3 选择器	180
10.1.4 操作	181
10.2 综合应用	188
10.2.1 单设备应用	188
10.2.2 多设备应用（群控）	191
10.3 阶段案例——BOSS 直聘爬虫	196
10.3.1 需求分析	196
10.3.2 核心代码构建	196

10.3.3 调试与运行	197
10.4 本章小结	197
10.5 动手实践	198

## 第11章 Scrapy 199

11.1 Scrapy 的安装	200
11.1.1 在 Windows 下安装 Scrapy	200
11.1.2 在 Linux 下安装 Scrapy	202
11.1.3 在 Mac OS 下安装 Scrapy	202
11.2 Scrapy 的使用	203
11.2.1 创建项目	203
11.2.2 在 Scrapy 中使用 XPath	207
11.3 Scrapy 与 MongoDB	213
11.3.1 items 和 pipelines 的设置	213
11.3.2 在 Scrapy 中使用 MongoDB	215
11.4 Scrapy 与 Redis	218
11.4.1 Scrapy_redis 的安装和使用	218
11.4.2 使用 Redis 缓存网页并自动去重	218
11.5 阶段案例——博客爬虫	220
11.5.1 需求分析	220
11.5.2 核心代码构建	221
11.5.3 调试与运行	226
11.6 本章小结	228
11.7 动手实践	228

## 第12章 Scrapy 高级应用 229

12.1 中间件（Middleware）	230
12.1.1 下载器中间件	230
12.1.2 爬虫中间件	242
12.2 爬虫的部署	246
12.2.1 Scrapyd 介绍与使用	246
12.2.2 权限管理	253
12.3 分布式架构	258
12.3.1 分布式架构介绍	258
12.3.2 如何选择 Master	259
12.4 阶段案例	259

12.5 本章小结	259
-----------	-----

## 第 13 章 爬虫开发中的法律和道德问题 260

13.1 法律问题	261
13.1.1 数据采集的法律问题	261
13.1.2 数据的使用	261
13.1.3 注册及登录可能导致的法律问题	261

13.1.4 数据存储	261
13.1.5 内幕交易	261
13.2 道德协议	262
13.2.1 robots.txt 协议	262
13.2.2 爬取频率	262
13.2.3 不要开源爬虫的源代码	262
13.3 本章小结	262

# 第1章

## 绪论

■ 所谓爬虫，其本质是一种计算机程序，它的行为看起来就像是蜘蛛在网上面爬行一样，顺着互联网这个“网”，一条线一条线地“爬行”。所以爬虫在英文中又叫作“Spider”，正是蜘蛛这个单词。

通过这一章的学习，你将会掌握如下知识。

- (1) 爬虫是什么。
- (2) 爬虫可以做什么。
- (3) 爬虫开发中有哪些技术。

## 1.1 爬虫

存在即合理，为什么爬虫程序会有其存在的土壤呢？这是由于传统低效率的数据收集手段越来越不能满足当今日益增长的数据需求所导致的。

这是一个数据爆炸的时代，没有了获取数据信息的壁垒，只要你肯，只要你想，那么就有机会利用数据让梦想走进现实。但是面对互联网这样一个由数据构建而成的海洋，如何有效获取数据，如何获取有效数据都是极其劳神费力、浪费成本、制约效率的事情。很多时候，按照传统手段完成一个项目可能 80%~90%的时间用于获取和处理数据。这样的矛盾冲突，搁在以往，搁在普通的人和普通的公司身上，除了用金钱去填补（直接购买数据）之外，似乎只有默默认命了。

回想一下编者还是学生的时候，心里向往着诗和远方，但口袋空空。如果要去旅游，只能一遍一遍地去各个旅游网站上寻找最便宜的酒店、最便宜的机票和最便宜的餐馆。往往旅游只有三四天，可旅游之前竟然要花上十几天甚至几十天来搜索攻略、抢票和订酒店。

如果看这本书的读者，你曾经也有过这样的经历，那么请问你，酒店提前几天订最便宜？机票什么时候订最实惠？你知道酒店的价格一周都怎样变化吗？刷了那么久的票，你总结出了什么规律吗？那如果有人告诉你，他每 15min 就可以监控这个城市所有酒店的价格，你相信吗？你会疑惑吧，谁会有闲心每 15min 把某个城市所有酒店所有房间的价格全部看一遍呢？就算有这个闲心，可有这个速度吗？

然而现在，终于有了扭转之机，那就是驾驭爬虫，监控酒店的房价变化只是基本技能。

## 1.2 爬虫可以做什么

### 1.2.1 收集数据

爬虫可以用来收集数据。这也是爬虫最直接、最常用的使用方法。由于爬虫是一种程序，程序的运行速度极快，而且不会因为做重复的事情就感觉到疲劳，因此使用爬虫来获取大量的数据，就变得极其简单和快捷了。

由于现在 99%以上的网站都是基于模板开发的，使用模板可以快速生成相同版式、不同内容的大量页面。因此，只要针对一个页面开发出了爬虫，那么这个爬虫也能爬取基于同一个模板生成的不同页面。这种爬虫称为定向爬虫，也是本书所要讲到的爬虫类型。

请看图 1-1 和图 1-2，这是起点中文网的“玄幻频道”和“奇幻频道”页面。



图 1-1 起点中文网的“玄幻频道”页面



图 1-2 起点中文网的“奇幻频道”页面

图 1-1 和图 1-2 所示的这两个版面除了内容不一样外,其他地方完全一样。只要爬虫能爬取“玄幻频道”,那么就能爬取“奇幻频道”。假设要把这两个页面的内容都获取下来,如果人工来操作,就需要对两个页面进行复制及粘贴,做很多重复的工作。而如果使用爬虫,那么只需要开发“玄幻频道”的爬虫就能实现既能爬取“玄幻频道”又能爬取“奇幻频道”的目标。

正是由于现在的网站大量使用了模板来生成页面,所以爬虫才能够有用武之地。

## 1.2.2 尽职调查

所谓的尽职调查,一般是指投资人在投资一个公司之前,需要知道这个公司是否如他们自己所描述的一样尽职尽责地工作,是否有偷奸耍滑、篡改数据、欺骗投资人的嫌疑。在过去,尽职调查一般通过调查目标公司的客户或者审计财务报表来实现。而有了爬虫以后,要做尽职调查就方便很多了。

例如调查一个电商公司,想知道他们的商品销售情况。该公司自己声称每个月销售额几亿元。如果使用爬虫爬取了该公司网站所有商品的销量情况,那么就可以计算出该公司的实际总销售额。而且,如果爬取了所有的评论并进行分析,还可以发现该网站是否出现了刷单的行为。

数据不会说谎,特别是数据量极大的数据,人工伪造的总会和自然生成的存在区别。而在以前,对于数据量极大的数据进行搜集是一件非常困难的事情,但现在有了爬虫的帮助,很多欺骗行为都会赤裸裸地暴露在阳光下。

## 1.2.3 刷流量和秒杀

刷流量是爬虫天然自带的功能。当爬虫访问了一个网站时,如果这个爬虫隐藏得很好,网站不能识别这一次访问来自于爬虫,那么就会把它当成正常访问。于是,爬虫就“不小心”地刷了网站的访问量。

除了刷流量外,爬虫也可以参与各种秒杀活动,包括但不限于在各种电商网站上抢商品,抢优惠券,抢机票和火车票。目前,网上有不少人专门使用爬虫来参加各种活动,并从中盈利。这种行为一般称为“薅羊毛”,这种人被称为“羊毛党”。不过使用爬虫来“薅羊毛”进行盈利的行为实际上游走在法律的灰色地带,希望读者不要轻易尝试。

## 1.3 爬虫开发技术

开发爬虫，既简单又困难。简单是因为在 Python 这一门语言的帮助下，要入门开发爬虫几乎没有门槛，几行代码就能写出一个爬虫。而爬虫相关的框架更是多如牛毛，稍稍配置一下就能实现非常不错的爬取效果。困难在于目前大多数的爬虫书籍，还停留在工具的讲解上，只告诉读者怎么用工具，却不告诉读者在遇到各种情况时应该如何举一反三，通过思考，用学过的技艺来处理第一次遇到的问题。

爬虫的开发有两个层面。一个是“技”的层面，也就是各种语言和框架的使用。这种层面更像是软件文档，现在市面上大部分的爬虫书籍还停留在这个层面。而另一个层面是“术”的层面，遇到各种反爬虫问题时，应该如何突破，如何隐藏爬虫，如何模拟人的行为，以及遇到没有见过的反爬虫策略时，应该如何思考及如何使用爬虫爬取非网页内容等。在“术”的层面，框架和工具都不是问题，用任何框架甚至 Python 自带的模块都能够处理，“术”的层面更强调思想、流程和调度。

本书只会使用少量的篇幅来讲解必须掌握的基础知识和框架用法。在此之上，将会着重介绍各种爬虫思想，力图做到让读者举一反三。

本书使用 Python 作为爬虫的开发语言。由于 Python 具有语法简单、入门容易等特点，现在已经成为众多领域的首选语言。由于 Python 的语法接近原生的英语语法，因此只要能看懂单词就能看懂 Python 代码，这使得 Python 学习者能够很容易地通过学习别人的代码得到提高。

本书第 2 章会讲到 Python 的基本语法。学习并掌握第 2 章的内容，可为后面的爬虫开发打好基础。

爬虫的主要目的是获取网页内容并解析。只要能达到这个目的，用什么方法都没有问题。关于获取网页，本书主要介绍了 Python 的两个第三方模块，一个是 requests，另一个是爬虫框架 Scrapy。关于解析网页内容，本书主要介绍了 3 种方式——正则表达式、XPath 和 BeautifulSoup。两种网页获取方式和 3 种网页解析方式可以自由搭配，随意使用。

由于网站必然不会这么轻易地让人把数据全给拿走，因此很多网站都会采取各种反爬虫措施。应对各种反爬虫措施正是本书所要讲到的重点。常规的反爬虫措施包括但不限于访问频率检查、验证码、登录验证、行为检测。本书对这些反爬虫策略都会进行一一破解。除此之外，本书还会将中间人攻击技术与爬虫结合在一起，再把 Android 自动化测试技术与爬虫结合在一起，从而构造一个超级自动化爬虫，做到几乎无法被网站发现，也无法被封锁，同时不需要人工干预就能实现数据的爬取。

在成功突破了网站的封锁以后，就需要提高爬虫的爬取效率了，于是本书将会讲到分布式爬虫框架 Scrapy。由于本书的宗旨是“术”，而不是“技”，因此对 Scrapy 这个框架，并不会像其官方文档一样讲解每一个功能。本书在介绍完 Scrapy 的基本功能以后，将着重讲解使用 Scrapy 来实现自动化的重试，自动修改爬虫的头部信息，自动更换 IP，自动处理异常和批量部署。

最后，本书会用一章来讲解和爬虫相关的法律问题，希望读者在爬虫开发领域不要触碰法律。

# 第2章

## Python基础

■ Python（中文发音为派森，原意为蟒蛇，因此其图标为两只蟒蛇）是一门高级程序开发语言。所谓“高级程序开发语言”，是相对于“低级程序开发语言”来说的。

Python的语法接近正常的英语语法，因此即使不会编程，只要懂得基本的英语，也可以大致看懂Python代码。

通过这一章的学习，你将会掌握如下知识。

- (1) Python开发环境的搭建。
- (2) Python的基本知识、数据类型。
- (3) Python的条件语句和循环语句。
- (4) Python函数的定义和使用。
- (5) 基于Python的面向对象编程代码。

## 2.1 Python 的安装和运行

由于历史原因，Python 有两个主要的大版本：Python 2 与 Python 3。这两个大版本同时在往各自的方向发展。绝大多数的 Python 代码在这两个大版本中可以通用，但也有少数代码只能在 Python 2 中运行，或者只能在 Python 3 中运行。

Python 官方曾经宣布，在今后的发展中，Python 3 的升级会增加新功能，而 Python 2 的升级只会做错误修正，不会增加新的功能。Python 之父吉多·范罗苏姆（Guido van Rossum）建议使用 Python 3，并逐步淘汰 Python 2。Python 官方推特宣布，在 2020 年停止维护 Python 2。本书所有代码基于 Python 3 开发。

截至 2017 年 4 月，Python 2 正式版的最新版本为 Python 2.7.13，Python 3 正式版的最新版本为 Python 3.6.1。在各位读者读到本书的时候，Python 3.6 的更高版本可能已经发布。本书所有代码可以无缝迁移到 Python 3.6 的更高版本。

### 2.1.1 在 Windows 中安装 Python

Python 的官方网站界面如图 2-1 所示。

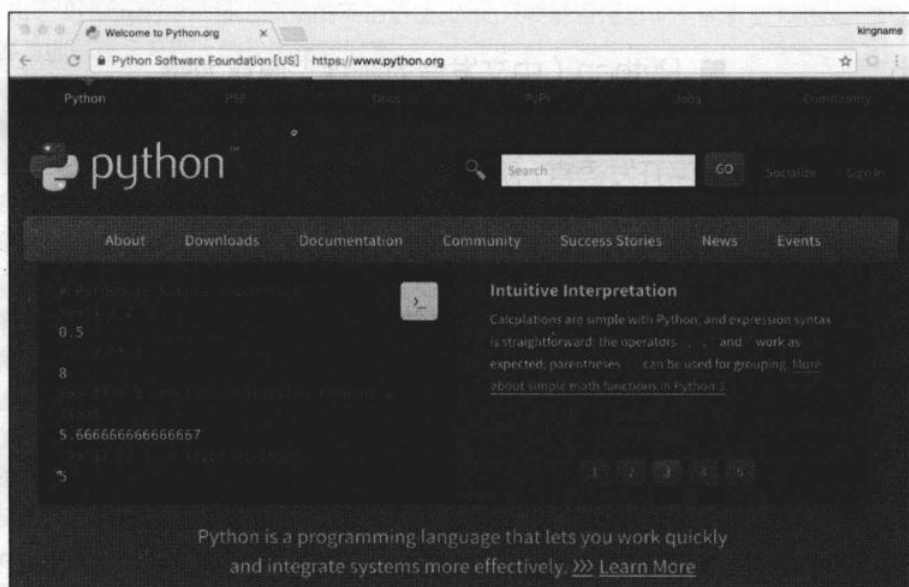


图 2-1 Python 官方网站界面

使用 Windows 操作系统的读者，可访问 <https://www.python.org/ftp/python/3.6.1/python-3.6.1-amd64.exe> 下载 Python 3.6.1 或者更高版本的安装程序。由于 Python 官方网站会受到某些干扰，所以在我国部分地区长期无法访问，在另一些地区间歇性无法访问。如果以上网址无法访问，各位读者可稍后再尝试。

下载的文件名为 `python-3.6.1-amd64.exe`。下载完成以后，双击这个安装程序，安装界面如图 2-2 所示。

一定要勾选“Add Python 3.6 to PATH”复选框，这一点非常重要。然后选择“Install Now”选项，即可开始安装 Python 3.6.1。安装完成以后，按“Win+R”组合键（“Win 键”是键盘上像汉字“田”的那个键），在弹出的“运行”对话框中输入“cmd”（不包括最外层双引号，下同），如图 2-3 所示。

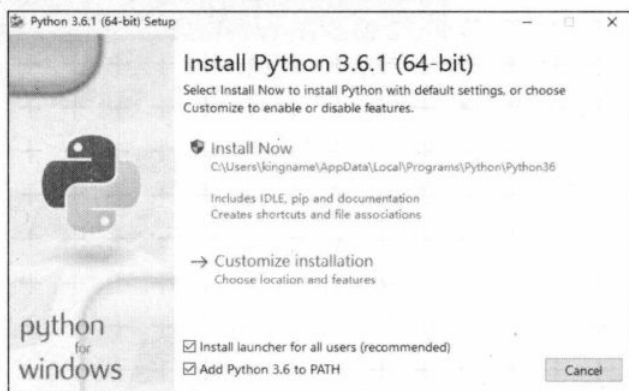


图 2-2 Python 安装界面

单击“确定”按钮，打开 Windows 命令提示符（Command Prompt，CMD）窗口，如图 2-4 所示。

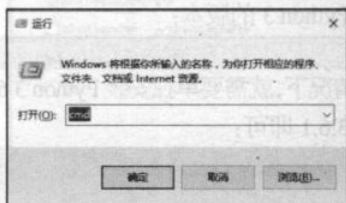


图 2-3 在“运行”对话框输入“cmd”

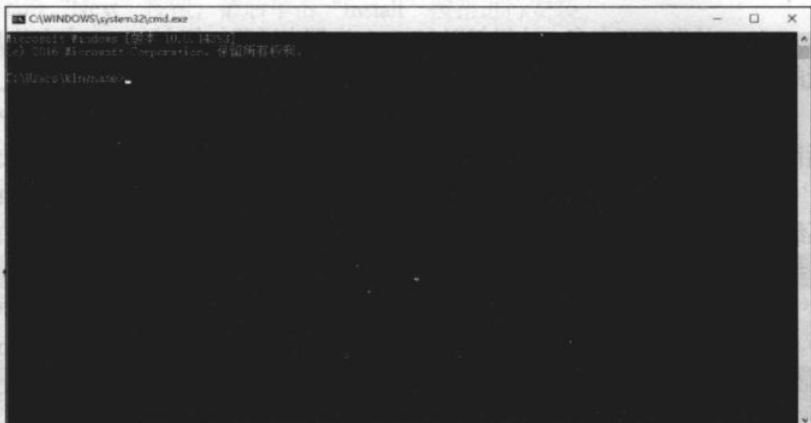


图 2-4 Windows 命令提示符窗口

输入“python”并按下键盘上的回车键，如果 CMD 窗口显示信息如图 2-5 所示，表明 Python 安装成功，并进入了 Python 交互环境。



图 2-5 启动 Python 交互模式成功

在图 2-5 中，出现了 3 个向右的箭头“>>>”，这是提示用户输入内容。在本章以及后面章节中的代码中如果有这样的 3 个箭头，表示代码就是在图 2-5 所示的窗口中直接输入的。例如：

```
>>> 1 + 1  
2
```

这两行代码表示把“1+1”通过键盘输入到这个 Python 交互环境中，然后按下回车键，下面不带 3 个箭头的数字“2”表示 Python 交互环境输出的内容。

## 2.1.2 在 Mac OS 中安装 Python

Mac OS 系统自带 Python 2。对于 Python 3，有两种不同的安装方法。

如果有编程基础，或者会使用 Homebrew，可以通过 Homebrew 安装 Python 3，其安装命令为：

```
brew install python3
```

由于 Homebrew 在我国部分地区会受到一些干扰，要解决这个问题需要一些技术基础，所以对于没有编程基础或者没有 Homebrew 的读者，可以访问 <https://www.python.org/ftp/python/3.6.1/python-3.6.1-macosx10.6.pkg>，下载

Python 3 的安装包。安装过程与安装普通软件没有区别，此处不再赘述。

## 2.1.3 在 Linux 中安装 Python

Linux 的发行版众多，这里仅以 Ubuntu 为例来说明如何在 Linux 中安装 Python 3。其他发行版请查阅该发行版的官方说明。

Ubuntu 16.04 或者更高版本的系统自带了 Python 3.5.1 或者更高版本的 Python。这个版本的 Python 可以正常运行本书所有的代码，因此使用 Ubuntu 16.04 或者以上系统的读者可以跳过这一节。

如果使用较低版本的 Ubuntu，系统自带 Python 2。某些系统可能只带 Python 3.4.x。这里的 x 是一个数字，随系统安装时间的不同而不同。读者可以在终端里输入以下代码查看系统自带的 Python 3 的版本：

```
python3 --version
```

如果返回类似于 Python 3.4.3 的结果，就表示系统确实自带 Python 3.4.x。这种情况下，就需要单独安装 Python 3.6。

如果 Ubuntu 版本为 16.04，直接在终端中输入以下几条命令来安装 Python 3.6.1 即可：

```
sudo add-apt-repository ppa:fkruhl/deadsnakes
```

```
sudo apt-get update
```

```
sudo apt-get install python3.6 python3-dev python3-pip libxml2-dev libxslt1-dev zlib1g-dev libffi-dev libssl-dev
```

如果系统为 16.10 或者 17.04，那么安装 Python 3.6 非常简单，不需要添加软件源，直接使用“apt-get”安装即可：

```
sudo apt-get update
```

```
sudo apt-get install python3.6 python3-dev python3-pip libxml2-dev libxslt1-dev zlib1g-dev libffi-dev libssl-dev
```

需要注意的是，由于 Python 2 在 Ubuntu 里面会被系统调用，因此不建议卸载或者修改系统自带的 Python 2。在这种情况下，可以在 Ubuntu 的终端里输入“python3.6”来启动 Python 3.6。

## 2.2 Python 开发环境

任何文本编辑器都可以用来开发 Python 程序，包括记事本。唯一的不同是开发效率的高低而已。一个优秀的集成开发环境（Integrated Development Environment，IDE）可以让 Python 开发如虎添翼，节约大量的开发时间。

### 2.2.1 PyCharm 介绍与安装

本书使用的集成开发环境为 JetBrains 公司的 PyCharm。PyCharm 在 Windows、Mac OS 和 Linux 中均有安装文件。网站提供了社区版（Community Edition）和专业版（Professional Edition），其中，社区版对个人用户是免费的，而且提供的功能可以满足本书的所有开发需求。

在网站上根据自己的操作系统选择合适的版本，如图 2-6 所示。



图 2-6 根据系统选择 PyCharm 版本