

姚良 | 编著

Python 3 爬虫实战

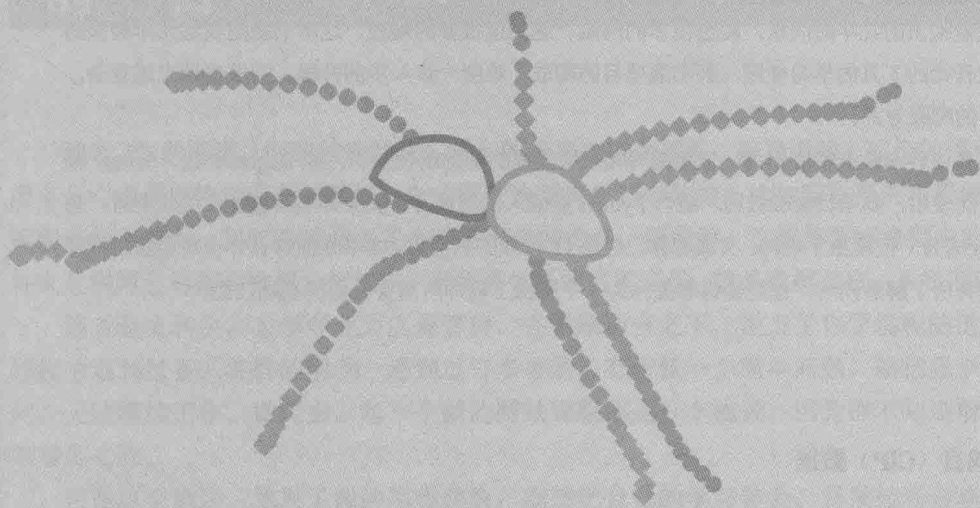
数据清洗、数据分析与可视化

涵盖了Python爬虫从小白到高手的应会知识点
适合想学编程的外行、没有任何经验的小白和想要进阶的爬虫工程师

15个详细案例

不仅展示丰富的数据爬取方式，还对数据清洗、数据分析和可视化进行系统讲解

中国铁道出版社有限公司
CHINA RAILWAY PUBLISHING HOUSE CO., LTD.



Python 3

爬虫实战

数据清洗、数据分析与可视化

姚良 | 编著

内 容 简 介

作为一个自学爬虫的过来人，曾经走过很多弯路，在自学的道路上也迷茫过。每次面对一个全新的网站，都像是一脚踏进一个未知的世界。你不知道前面有哪些反爬手段在等着你；你不知道你会踩进哪个坑里。我做爬虫的几年时间里，爬过很多的网站、遇到过很多的难题。这本书就是我这几年经验的总结，从最开始的工具的学习使用，到实战项目的爬取，难度一步一步的升级，需求也越来越复杂，有各式各样的爬取方式。

本书主要内容与数据爬取相关，包括编写爬虫所需要的基础编程知识，如 Requests 包、Scrapy 框架和数据库的使用，到项目实战教程，适合 Python 基础入门的读者。如果你是其他行业的从业者，想进入 IT 行业成为一位爬虫工程师，又或者你已经是 IT 行业的从业者，本书在能够让你在了解爬虫工程师的工作内容有所了解的同时，也能让你掌握作为一个爬虫工程师所需要具备的基础技能。

图书在版编目 (CIP) 数据

Python3 爬虫实战:数据清洗、数据分析与可视化/姚良编著. —北京:
中国铁道出版社有限公司, 2019. 10

ISBN 978-7-113-26059-0

I. ①P… II. ①姚… 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 (2019) 第 149591 号

书 名: Python3 爬虫实战——数据清洗、数据分析与可视化

作 者: 姚 良

责任编辑: 张 丹

读者热线电话: 010-63560056

责任印制: 赵星辰

封面设计: **MX** DESIGN
STUDIO

出版发行: 中国铁道出版社有限公司 (100054, 北京市西城区右安门西街 8 号)

印 刷: 中煤 (北京) 印务有限公司

版 次: 2019 年 10 月第 1 版 2019 年 10 月第 1 次印刷

开 本: 787 mm×1 092 mm 1/16 印张: 16.75 字数: 451 千

书 号: ISBN 978-7-113-26059-0

定 价: 59.80 元

版权所有 侵权必究

凡购买铁道版图书, 如有印制质量问题, 请与本社读者服务部联系调换。电话: (010) 51873174

打击盗版举报电话: (010) 51873659

随着 5G 的落地，网速越来越快，网上的信息也越来越多，但是无效、冗余的信息也更加泛滥。很多公司都需要特定专业的数据，为公司的决策提供科学依据。比如爬取某部电影的评论，进而分析观众的兴趣点；爬取羽绒服在各个销售平台的价格、销量等，这就需要网络爬虫的帮助了。因此各大互联网公司都有数据分析部门，而数据分析的前置工作，就是数据获取，本质上就是使用爬虫。

笔者是文科生，大学专业为工商管理。在机缘巧合之下，走上了自学编程的道路。在自学的过程中遇到过各式各样的困难，遇到过许多难题。有时候一个简单问题，就把我卡在那里几天时间，无法继续工作。做爬虫，每一个新的网站爬取都是一个挑战。因为你不知道前面有什么坑在等着你去踩。

正是这个原因，激发了我的写作热情，我想把自己的学习体会、开发技巧分享出来，让读者对现有网站的爬取方式有一个全面地了解。针对不同网站，选择合适的爬取方式，用最省力的方法获取数据。

本书特色

1. 从零开始，适合新手学习

对于只有代码入门基础的新手来说，看文档学习使用工具是十分困难的一件事。因为对代码的不理解、没有编程思维，看文档简直就像是在看天书。另外大部分的最新文档都是英文版的，国内的中文文档都是翻译过来的，在翻译过程中容易产生偏差。而本书基础知识篇中，从各官方文档中直接整理出爬虫爬取需要用到的部分。没有繁杂啰嗦的文字，用最简单的语言告诉你学习的重点知识，让你快速上手爬虫。在实战阶段，详细介绍每一个步骤，便于理解，让你也能靠自己写出爬虫。

2. 实例丰富，解决各种爬虫问题

网上很多爬虫的各种教程，大部分都是爬取豆瓣电影、招聘网站职位进行分析。本书实战项目挑选的都是网上少有人爬取的网站，让你可以学习到各式各样的爬取方式。

3. 站得更高，设计自己的产品

本书除了教你如何爬取网站外，还有很多以爬虫为基础的多功能设计教程，如爬虫机器人、爬虫网站、爬虫搜索功能。让你在学会爬取技术的同时，形成产品的思维去设计自己的产品。

本书内容及体系结构

第 1~3 章 环境搭建以及包的使用：介绍 Python、Requests 和 Scrapy 的安装以及爬虫常用库

Requests 和 Scrapy 的简单使用方法。用简单的例子和语言让读者顺利搭建爬虫环境，但同时快速上手编写爬虫。

自动化测试工具 selenium：从事爬虫工作并不总是一帆风顺的。总有一些网站让你什么办法都没有，源代码中没有数据、接口也加了密。这时候 selenium 是你最后的希望，它可以加载渲染 JS 帮助你获取页面的元素使之顺利完成爬取。

第 4 章 数据库的选择：本章介绍了主流的几个数据库，包括如何下载安装和使用，涉及一些基本的查询语句。让读者对 MySQL 和 Mongodb 有一个大致地了解，并学会两个数据库的基本使用方法。在读者自己编写爬虫的时候可以根据自己的需要去选择使用数据库。

第 5 章 效率为王之分布式爬虫：本章介绍了分布式爬虫的概念：将爬虫分布在多台服务器上，同时介绍提高爬取效率的方式。并介绍了几种分布式爬虫的实现方式，既有简单容易实现的，也有复杂高效的方式。让读者全面了解分布式爬虫的实现方式，并能亲自实践操作。

第 6 章 抓包分析：本章介绍如何利用工具对接口进行抓包分析，找到爬取接口的方法。需要有浏览器抓包和软件抓包。浏览器抓包是指使用浏览器自带的 network 工具对接口进行分析，找到目标接口。软件抓包是指通过 fiddler 工具对接口进行分析，找到目标接口。

第 7 章 Websocket 通信网站爬取：本章介绍了如何爬取使用 Websocket 通信协议的网站。使用 Websocket 通信协议的网站和一般网站不一样，需要服务端和客户端保持连接状态才能获取数据，如聊天室。通过模拟口令的方式实现成功握手，保持长连接接收网站的数据。

第 8 章 验证码破解：验证爬虫必须面对的一道坎，你可以选择绕过去也可以选择正面跨过去。介绍了两种从正面破解验证码的方式。

第 9 章 多线程与多进程并发爬取：本章介绍如何使用多线程和多进程去进行并发爬取，提高爬虫效率。

第 10 章 爬虫接口优化：爬虫爬取的数据有两种保存方式，保存到数据库和直接通过接口返回到前端。爬虫接口就是一种在线即时爬取数据并返回的接口。本章介绍如何对爬虫接口进行优化，使其支持高并发的访问和爬取。

第 11 章 使用 Docker 部署爬虫：部署爬虫是指将爬虫放置在一个稳定的环境中开始工作。爬虫部署的方式有很多种，本章介绍使用 Docker 对爬虫进行部署。

第 12 章 建立代理 IP 池：本章开始进入实战，演示爬取网站的整个过程。介绍防止爬虫被封的有效方式——建立代理 IP 池。通过使用爬虫爬取免费的代理来建立属于自己的代理 IP 池，为爬取工作进行顺利进行提供强有力的支持。

第 13 章 爬取磁力链接：爬取磁力搜索网站，获取想要的影视资源下载链接。将爬虫制作成自己的搜索小工具。

第 14 章 爬虫管家：利用 QQbot 制作一个监控爬虫状况的工具，定时检查并发送消息通知。

第 15 章 数据可视化：爬虫爬取的数据量多且杂，十分不利于数据的整理分析。但将数据可视化处理之后，就可以全面了解数据背后的信息。

第 16 章 爬取贴吧中大学邮箱：从全国高校的贴吧清洗数据获取邮箱。贴吧的帖子数据量大且杂，要从这些帖子中准确无误地清洗出邮箱。这是一个大项目，需要花费数天完成爬取。

第 17 章 批量爬取企业信息：从一个第三方平台中批量获取企业的名称，然后通过抓包工具获取企业的搜索查询接口爬取企业的详细信息。

第 18 章 爬取公众号历史文章：公众号是一个热门的爬取对象，很多人都想获得里面的文章

用于转载。本章通过微信 PC 端连接公众号，使用抓包工具获取接口并爬取公众号历史文章。

第 19 章 异步爬虫：本章介绍爬虫中的高效爬虫——异步爬虫。异步爬虫作为一个更快速高效的爬虫，无论是理解上和编写上都存在一定的难度。对于难度不大的网站，使用异步爬虫可以将爬取速度提升到极限。

第 20 章 漫画网站的爬取：本章介绍使用爬虫爬取漫画网站漫画。演示爬取单集、全集和全站漫画的爬取。

第 21 章 给 kindle 推送爬取的小说：本章介绍一个简单的小任务，爬取 fate 小说并通过代码推送到 kindle 中阅读。

第 22 章 爬取游民星空壁纸：本章介绍爬取游民星空高清壁纸，通过分页和筛选将 PC 壁纸和手机壁纸下载到本地。

第 23 章~第 26 章是一个爬虫网站项目：通过爬虫获取电影网站的资源，然后重新整理和展示数据，并整合成自己的网站。

本书读者对象

- Python 初学者；
- 数据分析师；
- 金融证券从业人员；
- 编程爱好者；
- 大数据从业人员；
- 创业公司老板。

目 录

Contents

第一篇 基础知识

第 1 章 Python 环境搭建

1.1 Python 的安装.....	2
1.1.1 Windows 下 Python 的安装.....	2
1.1.2 Mac OS X 下 Python 的安装.....	3
1.1.3 Linux 下 Python 的安装.....	3
1.1.4 安装 pip 工具.....	4
1.2 虚拟环境 Virtualenv.....	5
1.2.1 Virtualenv 的安装.....	5
1.2.2 创建虚拟环境.....	5
1.2.3 激活虚拟环境.....	5
1.2.4 创建指定 Python 版本的虚拟环境.....	5
1.3 选择合适的编辑器.....	6
1.3.1 Vim.....	6
1.3.2 Atom.....	6
1.3.3 Sublime Text.....	6
1.3.4 Notepad++.....	6
1.3.5 Pycharm.....	6

第 2 章 常用爬虫库 Requests

2.1 安装 Requests.....	7
2.1.1 用 pip 安装.....	7
2.1.2 用 github 源码安装.....	7
2.1.3 用 curl 安装.....	7
2.2 了解 Requests 的功能.....	8
2.2.1 使用 GET 和 POST 发送请求.....	8
2.2.2 通过 URL 传递参数.....	9

2.2.3	设置超时	9
2.2.4	查看返回内容	9
2.2.5	设置请求头	10
2.2.6	更多复杂的 Post 请求	10
2.2.7	返回对象状态码	12
2.2.8	设置代理 IP	13
2.3	BeautifulSoup 的安装和使用	14
2.3.1	使用 pip 安装 BeautifulSoup	14
2.3.2	使用 BeautifulSoup 定位元素	14
2.4	初识自动化测试工具 Selenium	15
2.4.1	Selenium 安装	15
2.4.2	使用 Selenium 爬取网站	15
2.5	Selenium 定位元素	16
2.5.1	通过属性定位	17
2.5.2	通过 xpath 定位	17
2.6	Selenium 反爬设置	18
2.6.1	设置请求头	18
2.6.2	设置代理 IP	19

第 3 章 常用爬虫框架 Scrapy

3.1	认识 Scrapy	21
3.1.1	Scrapy 爬取 quotes 简单示例	21
3.1.2	安装所需依赖包	23
3.1.3	使用虚拟环境	23
3.2	Scrapy shell 的使用	24
3.2.1	运行 shell	24
3.2.2	使用 Scrapy shell 爬取 Scrapy.org	24
3.2.3	爬虫调用 shell	26
3.3	使用 Scrapy 爬取 quotes	26
3.3.1	创建 Scrapy 项目并新建爬虫	27
3.3.2	爬取和提取数据	27
3.3.3	通过脚本运行 Scrapy 爬虫	29
3.3.4	在同一进程下运行多个爬虫	29
3.3.5	简易的分布式爬虫思路	30
3.3.6	防止爬虫被 ban	31
3.4	setting 基本配置	31
3.5	Pipeline 模块	32

3.5.1	爬取文字板块.....	32
3.5.2	编写 Pipeline 模块.....	35
3.5.3	通过 Pipeline 将数据写入 MongoDB 数据库.....	36
3.5.4	ImagesPipeline 处理图片.....	37
3.5.5	FilePipeline 下载文件.....	40
3.6	Middleware 中间件.....	41
3.6.1	Downloader Middleware.....	41
3.6.2	随机请求头中间件.....	42
3.6.3	更换代理 IP 中间件.....	45
3.6.4	通过 Downloader Middleware 使用 Selenium.....	46
3.6.5	Spider Middleware.....	47
3.7	新功能拓展.....	48
3.7.1	信号 signals.....	48
3.7.2	自定义拓展.....	51

第 4 章 数据存储——数据库的选择

4.1	MySQL 数据库.....	53
4.1.1	MySQL 的安装.....	53
4.1.2	几款可视化工具.....	54
4.1.3	数据库连接.....	55
4.1.4	数据库插入操作.....	55
4.1.5	数据库查询.....	56
4.1.6	数据库更新操作.....	56
4.1.7	爬取写入数据库.....	57
4.2	MongoDB 数据库.....	58
4.2.1	MongoDB 安装.....	58
4.2.2	连接数据库.....	59
4.2.3	查询数据库.....	59
4.2.4	插入和更新数据库.....	59
4.2.5	爬取数据并插入到 MongoDB 数据库中.....	60
4.3	Redis 数据库.....	60
4.3.1	Redis 安装.....	60
4.3.2	连接 Redis 数据库.....	61
4.3.3	Python 操作 Redis 数据库.....	61
4.3.4	爬取并写入 Redis 做缓存.....	62

第5章 效率为王——分布式爬虫

5.1 什么是分布式爬虫	64
5.1.1 分布式爬虫的效率	64
5.1.2 实现分布式的方法	64
5.2 Celery	65
5.2.1 Celery 入门	65
5.2.2 Celery 分布式爬虫	66
5.3 使用 Scrapy-redis 的分布式爬虫	67
5.3.1 Scrapy-redis 安装与入门	67
5.3.2 创建 Scrapy-redis 爬虫项目	68

第6章 抓包的使用与分析

6.1 利用抓包分析目标网站	72
6.1.1 如何抓包	72
6.1.2 网页抓包分析	72
6.2 手机 APP 抓包	74
6.2.1 使用 fiddler 抓包	75
6.2.2 HTTPS 证书安装	75
6.2.3 booking 手机端抓包	76

第7章 Websocket 通信网站爬取

7.1 什么是 Websocket	79
7.1.1 Websocket-clinet	79
7.1.2 Websocket-clinet 简单入门	79
7.2 使用 Websocket 爬取财经网站	81

第8章 验证码破解

8.1 关于验证码	84
8.1.1 一般的验证码	84
8.1.2 极验验证	84
8.2 极验滑动验证码破解	85
8.2.1 准备工具	85
8.2.2 分析滑动验证码	85
8.2.3 开始破解极限滑动验证码	87
8.3 图片验证码破解	89

8.3.1	准备工具	89
8.3.2	文字图像识别	89
8.3.3	识别验证码	90

第 9 章 多线程与多进程并发爬取

9.1	多线程	92
9.1.1	堵塞与非堵塞	92
9.1.2	继承 <code>threading.Thread</code> 创建类	96
9.1.3	多线程的锁	98
9.1.4	<code>queue</code> 队列	100
9.1.5	线程池	101
9.2	多线程爬虫	103
9.2.1	爬虫框架	103
9.2.2	编写爬虫	104
9.2.3	以多线程方式启动	105
9.3	多进程	107
9.3.1	<code>multiprocessing</code> 模块	107
9.3.2	通过 <code>Pool</code> 进程池创建进程	108
9.3.3	<code>multiprocessing.Queue</code> 队列	109
9.3.4	<code>multiprocessing.Pipe</code> 管道	112
9.3.5	<code>multiprocessing.Lock</code> 锁	113
9.4	多进程爬虫	114
9.4.1	多进程爬取音频	114
9.4.2	多进程加多线程进行爬取	116

第 10 章 爬虫接口优化

10.1	Gunicorn 的安装与使用	119
10.2	Gunicorn 配置	121
10.2.1	配置参数	121
10.2.2	通过 <code>config</code> 文件启动	123

第 11 章 使用 Docker 部署爬虫

11.1	Docker	125
11.1.1	Docker 的安装	125
11.1.2	Docker 的镜像	125
11.1.3	构建自己的 Docker 镜像	127
11.1.4	容器使用	127

11.1.5 Dockerfile	129
11.2 爬虫部署	130
11.2.1 爬虫接口	130
11.2.2 部署爬虫接口	131

第二篇 实战案例

第 12 章 实战 1: 建立代理 IP 池

12.1 爬取免费代理 IP	136
12.1.1 爬取代理 IP	136
12.1.2 检验代理 IP	138
12.2 建立代理 IP 池	138
12.2.1 检验代理 IP	138
12.2.2 Redis 消息队列	140
12.2.3 master 爬虫	142

第 13 章 实战 2: 磁力链接搜索器

13.1 爬取磁力搜索平台	145
13.1.1 磁力平台	145
13.1.2 slave 爬虫	146
13.2 实现磁力搜索器	148
13.2.1 展示与交互	148
13.2.2 数据查询	150

第 14 章 实战 3: 爬虫管家

14.1 QQ 机器人	152
14.1.1 qqbot	152
14.1.2 基本操作	152
14.1.3 实现自己的机器人	153
14.2 爬虫监控机器人	153

第 15 章 实战 4: 数据可视化

15.1 可视化包 Pyecharts	156
15.1.1 Pyecharts 的安装	156
15.1.2 地图展示数据	157
15.2 爬取最低价机票数据	158

15.2.1	破解旅游网站价格日历接口	159
15.2.2	爬取旅游网站	160
15.2.3	将数据可视化	161

第 16 章 实战 5：爬取贴吧中的邮箱

16.1	爬取网站	164
16.1.1	爬取高校名单	164
16.1.2	利用正则表达式匹配号码	165
16.2	分析贴吧搜索页面并提取号码	165
16.3	使用 Scrapy 开始编码	167
16.3.1	创建贴吧 Scrapy 项目	167
16.3.2	新建爬虫并编写爬虫逻辑	168
16.3.3	数据处理	170

第 17 章 实战 6：批量爬取企业信息

17.1	从第三方平台获取企业名	172
17.2	如何爬取企业详细信息	174

第 18 章 实战 7：爬取公众号历史文章

18.1	分析公众号接口	177
18.1.1	开始抓包	177
18.1.2	分析接口	179
18.1.3	尝试请求数据	179
18.2	爬取公众号	180
18.2.1	爬取思路	180
18.2.2	请求接口获取文章 URL	180
18.2.3	解析文章网页源码	181
18.2.4	合并代码	183

第 19 章 实战 8：高效爬取——异步爬虫

19.1	异步编程	186
19.1.1	asyncio 库	186
19.1.2	aiohttp 库	187
19.1.3	访问多个 URL	188
19.2	爬取图片	189
19.2.1	为函数命名	189

19.2.2	对网页进行解析	190
19.2.3	异步爬取图片	190

第 20 章 实战 9: 爬取漫画网站

20.1	爬取单部漫画	193
20.1.1	单集漫画的爬取	193
20.1.2	全集漫画的爬取	195
20.2	爬取漫画全站	196

第 21 章 实战 10: 给 kindle 推送爬取的小说

21.1	用 Python 发送邮件	199
21.1.1	纯文本邮件的发送	199
21.1.2	带附件邮件的发送	200
21.2	爬取小说	201
21.2.1	制作 word 文档	201
21.2.2	爬取 baka-tsuki.org	202

第 22 章 实战 11: 爬取游民星空壁纸

22.1	星空壁纸的爬取准备	205
22.2	爬取壁纸	206
22.2.1	获取图片和下一页地址	206
22.2.2	爬取列表页	208
22.2.3	爬取高清图片资源	209

第 23 章 综合实战: 建立一个小网站

23.1	Flask 框架	210
23.1.1	写一个简单的 hello word 网页	210
23.1.2	添加 html 模板	210
23.2	Bootstrap 框架	212
23.2.1	使用 Bootstrap 框架	213
23.2.2	Bootstrap 在线模板	213
23.2.3	添加壁纸板块	215

第 24 章 综合实战: 爬取电影网站

24.1	理清爬虫的思路	218
24.2	分步编码实现爬取	219

24.2.1 爬取详情页	219
24.2.2 爬取列表页	220
24.2.3 爬取首页	221
24.2.4 写入数据库	222

第 25 章 综合实战：建立电影小站

25.1 搭建项目	224
25.1.1 sqlite 数据库	224
25.1.2 创建项目	225
25.1.3 通过蓝图建立电影板块	226
25.2 建立模板	229
25.2.1 flask-bootstrap	229
25.2.2 电影页面	231
25.2.3 电影分类	233
25.2.4 电影详情页	237
25.2.5 电影搜索页	239

第 26 章 综合实战：磁力搜索

26.1 磁力搜索	241
26.1.1 如何高效爬取	241
26.1.2 建立 Celery 任务	244
26.2 Web 部分	248
26.2.1 建立模型	248
26.2.2 视图函数	248
26.2.3 关于产品	251

多个版本的Python是不会造成冲突的,当然,像浏览器一样只会安装一个版本也是可以的,如果版本,需要自己手动修改和切换版本,在环境变量中将Python的目录写进path,比如安装Python 3.6安装在C:\Python36,在系统环境变量设置里把下面的路径写进path:

也可以直接在终端输入:

重打

Python环境搭建

1.1.2 Mac OS X 下 Python 的安装

其实Mac自带了Python2,但是需要用的是Python3,所以还得重新下载,与Python类似,完成打开官网的下载页面选择版本下载<https://www.python.org/downloads/mac-osx/>如图1-2所示。



第一篇 基础知识

Windows 的 Python 安装很简单,打开 Python 的官网选择要下载的版本,按照提示一步步安装即可,如图 1-1 所示。

1.1.3 Linux 下 Python 的安装

Linux 系统本身自带了 Python2,与 Windows 类似,在 Linux 下安装 Python3 也是通过官网下载,如图 1-3 所示。

如果系统同时安装了 Python2 和 Python3,可以通过下面的命令分别运行不同的版本,如图 1-4 所示。

在学习爬虫之前，需要先搭建 Python 的环境。环境的搭建虽然麻烦又枯燥，但只需要做一次就可以了。

1.1 Python 的安装

Python 是一门可以让你的工作和系统的运行变得更高效的编程语言。一个资深的程序员可以快速上手 Python。即使是一个从未接触过编程的新手，也能够在规定时间内学会和掌握。

官网上 Python 的版本目前有两个，Python3.6 和 Python2.7。网上大部分的有关 Python 的教程都是来源于 Python2.7，之前也有人说很多的第三方包不支持 Python3。但是 2.7 已经是过去，Python3 逐渐已经成为主流。所以让我们从 Python3 开始吧。

1.1.1 Windows 下 Python 的安装

Windows 的 Python 安装很简单，打开 Python 的官网找到最近的版本下载 <https://www.python.org/downloads/release/Python-364/> 如图 1-1 所示。

Files					
Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		9de6494314ea199e3633211696735f65	22710891	SIG
XZ compressed source tarball	Source release		1325134dd525b4a2c3272a1a0214dd54	16992824	SIG
Mac OS X 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	9fba50521dffa9238ce85ad640abaa92	27778156	SIG
Windows help file	Windows		17cc49512c3a2b876f2ed8022e0afe92	8041937	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64, not Itanium processors	d2fb546fd4b189146dbefeba85e7266b	7162335	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64, not Itanium processors	bee5746dc6ece6ab49573a9f54b5d0a1	31684744	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64, not Itanium processors	21525b3d132ce15cae6ba96d74961b5a	1320128	SIG
Windows x86 embeddable zip file	Windows		15802be75a6246070d85b87b3f43f83f	6400788	SIG
Windows x86 executable installer	Windows		67e1a9bb336a5eca0efcd481c97262a4	30653888	SIG
Windows x86 web-based installer	Windows		6c8ff748c554559a385c986453df28ef	1294088	SIG

图 1-1 Python for Windows

在选择版本的时候是根据系统来选的。如果 Windows 是 64 位的，下载时选择带有 64 位的标识的版本。如果安装的是 Python3.6 版本，文件夹名会是 Python36。这样的设计是为了在系统安