

Python

调查类问题

爬虫（计算机网络）

## 你是如何开始能写python爬虫？

看完了简明教程和笨办法学python，想写爬虫，无从做起，需要继续看什么书和练习

1 条评论    分享

27 个回答

按投票排序

▲  
253

知乎用户，Py/推荐系统/爬虫/NLP/机器学习/审计

Jimmy Lee、Thulof Qu、知乎用户 等人赞同

说说我的经历吧

我最早是爬虾米，想看看虾米听的比较多的歌是哪些，就爬了虾米全站的歌曲播放数，做了个统计

[Python爬虫学习记录（1）——Xiami全站播放数](#)

统计过豆瓣动漫的评分分布

[豆瓣2100部动漫页面的网页源码\(包括评分，导演，类型，简介等信息，附抓取代码\)](#)

爬百度歌词，做LDA

[Python爬虫学习记录（2）——LDA处理歌词](#)

百度音乐带标签，作曲，演唱者，类别的歌词数据

爬足彩网站所有盘口，找赢钱算法

[Python爬虫学习记录（4）——传说中的足彩倍投法。。好像也不是那么靠谱](#)

[2011~2013.5全球所有足球比赛比分数数据以及足彩各公司盘口](#)

初期不需要登录的网站比较简单，掌握http get post和urllib怎么模拟，掌握lxml、BeautifulSoup等parser库就可以了，多用firefox的firebug或者chrome的调试工具看浏览器是怎么发包的。上面都是不需要登录不需要下文件就可以做到的。

之后你可能想要下载文件（图片，音乐，视频等），这个可以试试爬虾米歌曲

[Python爬虫学习记录（3）——用Python获取虾米加心歌曲，并获取MP3下载地址](#)

爬wallbase壁纸

知乎是一个真实的问答社区，在这里分享知识、经验和见解，发现更大的世界。

使用手机或邮箱注册

使用微信登录

使用微博登录

使用 QQ 登录

关注问题

2033 人关注该问题

HALL OF PRO  
**职人介绍所**  
知乎 首 档 视 频 节 目 开 播

相关问题

换一换

有哪些「神奇」的数据获取方式？ 46 个回答

会一门脚本语言，学 bash 就显得浪费时间？ 38 个回答

能利用爬虫技术做到哪些很酷很有趣很有用的事情？ 338 个回答

如何入门 Python 爬虫？ 95 个回答

如何用简单的算法生成一个类似『光盘』的彩色圆形图片？ 4 个回答

最近做了个avfun的视频排行，每天定时抓几次acfun，然后下载视频到服务器缓存。

Python爬虫学习记录（5）——python mongodb + 爬虫 + web.py 的acfun视频排行榜  
202.120.39.152:8888

然后你可能需要模拟用户登录，爬取需要登录的网站（比如人人，新浪微博）。如果只是小规模爬虫建议用浏览器cookie模拟登录

Python爬虫学习记录（0）——Python 爬虫抓站 记录（虾米，百度，豆瓣，新浪微博）

=====

想说的是，不要为了学而学，可以看看有什么以前觉着很麻烦的操作，是不是能用爬虫简化。爬下来的数据是不是有排序筛选分析的价值。

2015-8-31，在csdn上更新了之前失效的百度空间链接，可能有些代码因为网站的改版不适用了，这里主要还是提供一些应用的想法。


编辑于 2015-08-31    31 条评论    感谢    分享    收藏  
• 没有帮助 • 举报 • 作者保留权利

▲  
47  
▼

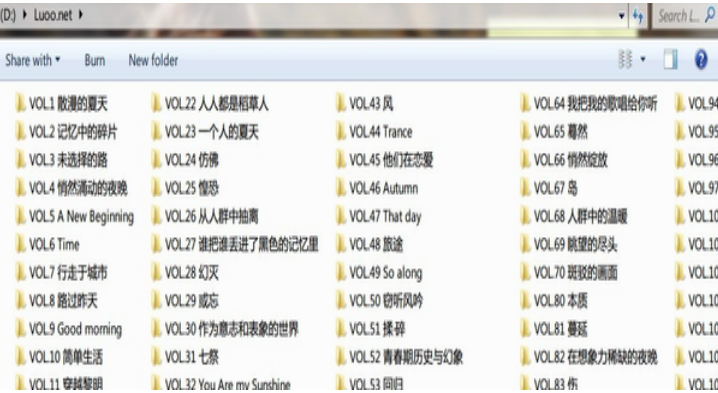
**shadow**，水水的文艺程序猿一枚

知乎用户、Walson、万申生 等人赞同

刚开始只是简单的看了下 Python，简单的写些小程序，小工具之类的，感受到了它的简洁与强大。



突然就想写个爬虫试试，那就爬我最喜欢的音乐网站落网，爬了从第一期到到现在的所有音乐，包括每期的图片。还另外写了一个自动下载当前最新一期的所有歌曲的脚本。还试着用工具 PyInstaller 打包成了 exe，分享给了我几个同样喜欢落网的朋友。↓这是我爬的成果。



VOL8 路过昨天	VOL29 遗忘	VOL50 窃听风吟	VOL80 本质	VOL10
VOL9 Good morning	VOL30 作为意志和表象的世界	VOL51 揉碎	VOL81 蔓延	VOL10
VOL10 简单生活	VOL31 七祭	VOL52 青春历史与幻象	VOL82 在想象力稀缺的夜晚	VOL10
VOL11 穿越黎明	VOL32 You Are my Sunshine	VOL53 回归	VOL83 伤	VOL10
VOL12 Sonic Youth	VOL33 默然于	VOL54 侵袭	VOL84 The Sky	VOL15
VOL13 四月	VOL34 狗日的青春	VOL55 寂静的飞翔	VOL85 依然	VOL15
VOL14 PICTURE	VOL35 you and me	VOL56 人们的梦	VOL86 沉默的存在	VOL15
VOL15 迷失	VOL36 我们都不真实	VOL57 殊途同归	VOL87 再见，昨天。	VOL15
VOL16 轮回	VOL37 我们终究还是要回去	VOL58 我不怕荆棘，倘若你是玫瑰	VOL88 我路过这画面	VOL15
VOL17 忆	VOL38 追逐还是 放逐？	VOL59 远行吧远行	VOL89 Far Away	VOL15
VOL18 被放大的旧照片	VOL39 Love Will Tear Us Apart	VOL60 冬至	VOL90 Your Whisper	VOL15
VOL19 哭泣着的疼痛	VOL40 颂歌	VOL61 让我静静的	VOL91 Black Star	VOL15
VOL20 伤痛的瞬间	VOL41 我们那些小想法	VOL62 我将在天堂里收获庄稼	VOL92 No Stopping	VOL15
VOL21 雨季	VOL42 悲伤的不止是你	VOL63 谁不曾有青春年少时	VOL93 在路上	VOL15

关于如何学，我只是个新手，谈不上指导，就说说我自己怎么做的吧：

1. 首先基本的 Python 语法你要了解吧，推荐一本书《Python 基础教程》，很适合入门。
2. 其次分析你的爬虫需求。程序具体流程是怎么样的？把程序的大致框架搭起来。另外可能还有哪些难点？
3. 然后了解一下一般写爬虫需要用哪些库，这些库可以很好的帮助你解决很多问题。推荐强悍的 [Requests: HTTP for Humans](#) 另外还有其它一些库如 urllib2 BeautifulSoup 都可以了解下。
4. 动手开始写了，遇到问题 Google 即可，Google 不行就上知乎问，我遇到的一个问题就是知乎私信大牛解决的。在写的过程中还会学到很多相关的知识，比如说HTTP协议，多线程等等。

或者你也可以直接用别人的框架，如其它人提到的 [Scrapy](#)，不用重复造轮子。

编辑于 2014-05-06 10 条评论 感谢 分享 收藏

• 没有帮助 • 举报 • 作者保留权利



10

知乎用户，热爱Python，Data Debugger，机器学习进...

sean、沙德轩、知乎用户 等人赞同



最开始看了写Simplecd那个人的博客开始了解原来Python写爬虫这么牛逼。

然后写了脚本1W+豆瓣电影的评分页面。之后实验室有项目，写了脚本爬了50W条微博，中途熟悉如何模拟登录骗过服务器是最有趣的部分。

总结，你看看之前人的博客就行了，简单的爬虫用不到多高深的技术，无非就是几个：

- 1.熟悉一下urllib的使用
- 2.了解基本的html解析，通常来说最基本的正则就够用了

发布于 2013-09-04    6 条评论    感谢    分享    收藏

• 没有帮助 • 举报 • 作者保留权利

▲  
4

知乎用户，IT 自学日记 rcdisk.com



uuz、知乎用户、邵烟球 等人赞同

▼

网络上很多你需要重复去做的，都可以写python脚本去做。  
比如知乎的一些好文章，你要保存下来，或者自动定期发送到kindle电子书

[python 抓取知乎推荐话题的所有答案 知乎爬虫](#)

[批量获取色影无忌图片 Python\\_小组](#)

[暴力破解wordpress后台管理员密码](#)

[用python破解某211大学BBS论坛用户密码](#)

感觉都是自己为了完成某种目的而且做得，这样动机会更加明确。目前准备爬取股票信息，做研究使用（炒股啦）

更多 [Python自学\\_小组](#)

编辑于 2015-11-17    4 条评论    感谢    分享    收藏 •

没有帮助 • 举报 • 作者保留权利

▲  
48

知乎用户，lining0806.com/



高小天、知乎用户、知乎用户 等人赞同

▼

Python学习网络爬虫主要分3个大的版块：**抓取，分析，存储**  
另外，比较常用的爬虫框架[Scrapy](#)，这里最后也详细介绍一下。

首先列举一下本人总结的相关文章，这些覆盖了入门网络爬虫需要的基本概念和技巧：[宁哥的小站-网络爬虫](#)

当我们在浏览器中输入一个url后回车，后台会发生什么？比如说你输入[firing的数据空间](#)，你就会看到宁哥的小站首页。

简单来说这段过程发生了以下四个步骤：

- 查找域名对应的IP地址。
- 向IP对应的服务器发送请求。
- 服务器响应请求，发回网页内容。
- 浏览器解析网页内容。

网络爬虫要做的，简单来说，就是实现浏览器的功能。通过指定url，直接返回给用户所需要的数据，而不需要一步步人工去

操纵浏览器获取。

抓取

这一步，你要明确要得到的内容是什么？是HTML源码，还是Json格式的字符串等。

### 1. 最基本的抓取

抓取大多数情况属于get请求，即直接从对方服务器上获取数据。

首先，Python中自带urllib及urllib2这两个模块，基本上能满足一般的页面抓取。另外，[requests](#) 也是非常有用的包，与此类似的，还有[httplib2](#) 等等。

```
Requests:
import requests
response = requests.get(url)
content = requests.get(url).content
print "response headers:", response.headers
print "content:", content

Urllib2:
import urllib2
response = urllib2.urlopen(url)
content = urllib2.urlopen(url).read()
print "response headers:", response.headers
print "content:", content

Httpplib2:
import httpplib2
http = httpplib2.Http()
response_headers, content = http.request(url, 'GET')
print "response headers:", response_headers
print "content:", content
```

此外，对于带有查询字段的url，get请求一般会将来请求的数据附在url之后，以?分割url和传输数据，多个参数用&连接。

```
data = {'data1': 'XXXXX', 'data2': 'XXXXX'}
Requests: data为dict, json
import requests
response = requests.get(url=url, params=data)

Urllib2: data为string
import urllib, urllib2
data = urllib.urlencode(data)
full_url = url+'?' + data
response = urllib2.urlopen(full_url)
```

相关参考：[网易新闻排行榜抓取回顾](#)

参考项目：[网络爬虫之最基本的爬虫：爬取网易新闻排行榜](#)

### 2. 对于登陆情况的处理

#### 2.1 使用表单登陆

这种情况属于post请求，即先向服务器发送表单数据，服务器

再将返回的cookie存入本地。

```
data = {'data1':'XXXXX', 'data2':'XXXXX'}
Requests: data为dict, json
import requests
response = requests.post(url=url, data=data)
Urllib2: data为string
import urllib, urllib2
data = urllib.urlencode(data)
req = urllib2.Request(url=url, data=data)
response = urllib2.urlopen(req)
```

## 2.2 使用cookie登陆

使用cookie登陆，服务器会认为你是一个已登陆的用户，所以就会返回给你一个已登陆的内容。因此，需要验证码的情况可以使用带验证码登陆的cookie解决。

```
import requests
requests_session = requests.session()
response = requests_session.post(url=url_login, data=
```

若存在验证码，此时采用response = requests\_session.post(url=url\_login, data=data)是不行的，做法应该如下：

```
response_captcha = requests_session.get(url=url_login)
response1 = requests.get(url_login) # 未登陆
response2 = requests_session.get(url_login) # 已登陆，
response3 = requests_session.get(url_results) # 已登陆
```

相关参考：[网络爬虫-验证码登陆](#)

参考项目：[网络爬虫之用户名密码及验证码登陆：爬取知乎网站](#)

## 3. 对于反爬虫机制的处理

### 3.1 使用代理

适用情况：限制IP地址情况，也可解决由于“频繁点击”而需要输入验证码登陆的情况。

这种情况最好的办法就是维护一个代理IP池，网上有很多免费的代理IP，良莠不齐，可以通过筛选找到能用的。对于“频繁点击”的情况，我们还可以通过限制爬虫访问网站的频率来避免被网站禁掉。

```
proxies = {'http':'http://XX.XX.XX.XX:XXXX'}
Requests:
import requests
response = requests.get(url=url, proxies=proxies)
Urllib2:
import urllib2
proxy_support = urllib2.ProxyHandler(proxies)
```

```
opener = urllib2.build_opener(proxy_support, urllib2.install_opener(opener) # 安装opener, 此后该
response = urllib2.urlopen(url)
```

### 3.2 时间设置

适用情况：限制频率情况。

Requests, Urllib2都可以使用time库的sleep()函数：

```
import time
time.sleep(1)
```

### 3.3 伪装成浏览器，或者反“反盗链”

有些网站会检查你是不是真的浏览器访问，还是机器自动访问的。这种情况，加上User-Agent，表明你是浏览器访问即可。有时还会检查是否带Referer信息还会检查你的Referer是否合法，一般再加上Referer。

```
headers = {'User-Agent': 'XXXXX'} # 伪装成浏览器访问，适
headers = {'Referer': 'XXXXX'}
headers = {'User-Agent': 'XXXXX', 'Referer': 'XXXXX'}
Requests:
    response = requests.get(url=url, headers=headers)
Urllib2:
    import urllib, urllib2
    req = urllib2.Request(url=url, headers=headers)
    response = urllib2.urlopen(req)
```

## 4. 对于断线重连

不多说。

```
def multi_session(session, *arg):
    while True:
        retryTimes = 20
        while retryTimes>0:
            try:
                return session.post(*arg)
            except:
                print '.',
                retryTimes -= 1
```

或者

```
def multi_open(opener, *arg):
    while True:
        retryTimes = 20
        while retryTimes>0:
            try:
                return opener.open(*arg)
            except:
                print '.',
                retryTimes -= 1
```

这样我们就可以使用multi\_session或multi\_open对爬虫抓取的session或opener进行保持。

## 5. 多进程抓取

这里针对[华尔街见闻](#) 进行并行抓取的实验对比：[Python多进程抓取](#) 与 [Java单线程和多线程抓取](#)

相关参考：[关于Python和Java的多进程多线程计算方法对比](#)

## 6. 对于Ajax请求的处理

对于“加载更多”情况，使用Ajax来传输很多数据。

它的工作原理是：从网页的url加载网页的源代码之后，会在浏览器里执行JavaScript程序。这些程序会加载更多的内容，“填充”到网页里。这就是为什么如果你直接去爬网页本身的url，你会找不到页面的实际内容。

这里，若使用Google Chrome分析”请求“对应的链接(方法：右键→审查元素→Network→清空，点击”加载更多“，出现对应的GET链接寻找Type为text/html的，点击，查看get参数或者复制Request URL)，循环过程。

- 如果“请求”之前有页面，依据上一步的网址进行分析推导第1页。以此类推，抓取抓Ajax地址的数据。
- 对返回的json格式数据(str)进行正则匹配。json格式数据中，需从'\uxxxx'形式的unicode\_escape编码转换成u'\uxxxx'的unicode编码。

## 7. 自动化测试工具Selenium

Selenium是一款自动化测试工具。它能实现操纵浏览器，包括字符填充、鼠标点击、获取元素、页面切换等一系列操作。总之，凡是浏览器能做的事，Selenium都能够做到。

这里列出在给定城市列表后，使用selenium来动态抓取[去哪儿网](#) 的票价信息的代码。

参考项目：[网络爬虫之Selenium使用代理登陆：爬取去哪儿网站](#)

## 8. 验证码识别

对于网站有验证码的情况，我们有三种办法：



- 使用代理，更新IP。
- 使用cookie登陆。
- 验证码识别。

使用代理和使用cookie登陆之前已经讲过，下面讲一下验证码识别。

可以利用开源的Tesseract-OCR系统进行验证码图片的下载及识别，将识别的字符传到爬虫系统进行模拟登陆。如果不成功，可以再次更新验证码识别，直到成功为止。

参考项目：[Captcha1](#)

**爬取有两个需要注意的问题：**

- 如何监控一系列网站的更新情况，也就是说，如何进行增量式爬取？
- 对于海量数据，如何实现分布式爬取？

分析

抓取之后就是对抓取的内容进行分析，你需要什么内容，就从中提炼出相关的内容来。

常见的分析工具有[正则表达式](#)，[BeautifulSoup](#)，[lxml](#)等等。

存储

分析出我们需要的内容之后，接下来就是存储了。

我们可以选择存入文本文件，也可以选择存入[MySQL](#) 或 [MongoDB](#) 数据库等。

**存储有两个需要注意的问题：**

- 如何进行网页去重？
- 内容以什么形式存储？

Scrapy

Scrapy是一个基于Twisted的开源的Python爬虫框架，在工业中应用非常广泛。

相关内容可以参考[基于Scrapy网络爬虫的搭建](#)，同时给出这篇文章介绍的[微信搜索](#)爬取的项目代码，给大家作为学习参考。

参考项目：[使用Scrapy或Requests递归抓取微信搜索结果](#)

编辑于 2015-12-17    3 条评论    感谢    分享    收藏

• 没有帮助 • 举报 • 作者保留权利



▼ 只推薦一個库不解释：

### Requests: HTTP for Humans

发布于 2013-09-04    添加评论    感谢    分享    收藏

• 没有帮助 • 举报 • 作者保留权利

▲ 佯良，不会。不会。不会。



4

高野良、小罗、知乎用户 等人赞同

▼ 爬了个知乎日报,本来是用php写的,后来学了python就改用了python

[Python 获得知乎日报数据](#)

php版本爬虫的知乎日报:[直呼日报 2.0 Beta](#)

界面做的很搓,但是能用,最近没时间改进,有时间再说

发布于 2015-02-09    1 条评论    感谢    分享    收藏

• 没有帮助 • 举报 • 作者保留权利

▲ 知乎用户，linux dvb



3

梁风、知乎用户、知乎用户 赞同

▼ 去年，我嫌我的笔记本笨了，想买个thinkpad。该死的官方网站，产品型号那么多，连个基本的比较功能都没有做，一怒之下就写了个爬虫，下载所有产品网页，同一个产品的不同型号生成一个表格网页，用不同的颜色区分不同的参数。  
如果有人打算买thinkpad，可以试试啊。

[wayneming/thinkpadcrawler](#) · [GitHub](#)

发布于 2015-06-30    1 条评论    感谢    分享    收藏

• 没有帮助 • 举报 • 作者保留权利

▲ 张方舟，ZhihuHot.sinaapp.com    知乎会火问题精选...



7

leiwuhen zhang、Rex Zh、士心 等人赞同

▼ 不知题主的编程基础如何，基本上想写Python爬虫只需要看下前人的日志学习下urllib和BeautifulSoup就够用了。

另外如果有千人的实际例子看一下应该能比较直观的明白一个爬虫是怎么跑起来的。

在下不才，写过一个爬知乎数据的爬虫，**开源**于（各位亲走过路过的时候请给点个Star或Fork

呗~）：[MorganZhang100/zhihu-spider](#) · [GitHub](#)

该爬虫爬到的数据应用于：[zhiuhot.sinaapp.com/](#)

简单点说就是一个爬取问题的一些参数，进行分析，找出最可能会火的问题。据此回答问题的话，得赞速度比之前多20倍左右。

我对Python也不是很熟悉，不过一共只有几百行的代码，题主看看应该问题不大。

其实学任何东西，看得教程再多也比不上自己实习写几行代码来得有效果。只要开始写了，你就知道你需要什么了。

发布于 2015-02-27    1 条评论    感谢    分享    收藏

• 没有帮助 • 举报 • 作者保留权利



5

李哲瑞, lizherui.com



钱生生、郝机智、邢远 等人赞同



[github.com/lizherui/spi...](https://github.com/lizherui/spi...)

发布于 2013-09-04    添加评论    感谢    分享    收藏

• 没有帮助 • 举报 • 作者保留权利



2

知乎用户, Coder, 找个喜欢的工作中, 要用python



You are not Mark、知乎用户 赞同



scrapy直接硬上, 不用自己写, 直接关注想要啥。

发布于 2015-02-09    2 条评论    感谢    分享    收藏

• 没有帮助 • 举报 • 作者保留权利



4

神手, get a real life



怀晓伟、thanksdanny、知乎用户 等人赞同



我的博客里有很详细的叙述和源码,python3.4实现。

欢迎交流 [网络资源搜索爬虫\(python 3.4.1实现\)](#) 修改

发布于 2014-10-15    添加评论    感谢    分享    收藏

• 没有帮助 • 举报 • 作者保留权利



23

匿名用户

黄睿昆、arc Miracle、知乎用户 等人赞同



第一次想写爬虫是为了抓取草榴的高点击率视频链接。代码如下。需要翻墙。

```
# -- coding: utf-8 --
```

```
import urllib2
```

```

import sys
from bs4 import BeautifulSoup

reload(sys)
sys.setdefaultencoding('utf8')

BaseUrl = "http://t66y.com/"
j=1

for i in range(1, 100):
    url = "http://t66y.com/thread0806.php?fid=22&search"
    page = urllib2.urlopen(url)
    soup = BeautifulSoup(page, from_encoding="gb18030")
    print("reading page "+ str(i))
    counts = soup.find_all("td", class_="tal f10 y-styl

    for count in counts:
        if int(count.string)>15:
            videoContainer = count.previous_sibling.previous
            video = videoContainer.find("h3")
            print("Downloading link "+ str(j))
            line1 = (video.get_text())
            line2 = BaseUrl+video.a.get('href')
            line3 = "view *" + count.string + "*"
            print line1
            f = open('cao.md', 'a')
            f.write("\n"+"###"+" "+line1+"\n"+"<"+line2+">")
            f.close()
            j+=1

```

发布于 2014-08-31    7 条评论    感谢    分享    收藏

• 没有帮助 • 举报 • 作者保留权利

**时雨苍剑**，ISTP/想成为程序猿的孩子



jason yang 赞同

其实开始动手就好了~

先找个想爬的网页，比如[豆瓣电影TOP250](#)，然后urllib2打开网页，beautifulsoup解析html，结合正则表达式抓取你需要的字段，最后写入文件（print出来也没意见，新手嘛哈哈）爬虫入门并不难哈哈~

如果想深入一点，可以学习scrapy，看[初窥Scrapy](#)（中文文档），把你之前写的爬虫放到框架中试试吧~

PS：我也是新新新新手，感兴趣所以开始学了一点点，欢迎交流哈~

PPS：欢迎光临[ztybuaa \(朱天宇\)](#) · [GitHub](#) 我写的豆瓣电影Top250爬虫（大神们请无视我~），写的并不好，请见谅哈~

发布于 2015-04-18    8 条评论    感谢    分享    收藏

• 没有帮助 • 举报 • 作者保留权利



3

**liujiawen**，你可知希望不会来啊？



小岛民在大陆、郝机智、知乎用户 赞同



有一个网站上有很多论文，想下载，但懒得一个个点。于是写了个爬虫来下载。

发布于 2014-08-31    5 条评论    感谢    分享    收藏

• 没有帮助 • 举报 • 作者保留权利



8

知乎用户，不要邀请我答题了.....知乎的推荐机制太烂...



黄睿昆、万浩然、张月 等人赞同



最开始其实想学nutch，后来想想，还是用小一点的项目比较好，开源项目scrapy，还不错，就是文档写得不适合新手看。最近刚完成一个需要登录的爬虫，抓取sis亚洲无码区种子的下载数和红心数.....

[Scrapy Tutorial](#)

发布于 2014-03-14    7 条评论    感谢    分享    收藏

• 没有帮助 • 举报 • 作者保留权利



5

**Python Mingo**，马路牙子~



庖丁解、知乎用户、王月 等人赞同



最开始时候，因为自己用ubuntu，想查单词上网页查找太费劲，就用python的urllib写了一个终端下查字典的脚本，非常方便，也非常开心~之后，我在学校的bbs里写了一个自动正则过滤帖子并发贴的代码，用的python3的urllib模块，开始就是为了好玩~现在更新了好几个版本，版里的盆友们玩的也非常开心~哈哈~主要就是兴趣，觉的好玩，那就搞起，不论神马方法，条条铁路通北京，必然能成~

发布于 2013-09-03    4 条评论    感谢    分享    收藏

• 没有帮助 • 举报 • 作者保留权利



6

**张静宁**，在理工学校学科技传播



thanksdanny、庖丁解、苏仪 等人赞同



动机上：我想把学校教育系统课程信息爬下来

操作上：先了解 HTTP协议，然后学习使用 Python Requests 模块，再实战。

练习上：先在终端直接操作，试着抓抓**baidu.com**，再把自己写的爬虫发布到PIPy...然后，把教务系统2学期2000+门课抓下来，还尝试攻击了一个小伙伴的网站...把它搞挂啦！

具体可以看这篇博客，写一个 Python 爬虫

[jenny42.com/2015/02/wri...](http://jenny42.com/2015/02/wri...)

突然发现我这个其实不算爬虫，顶多算抓取网页。因为我没有学 xPath 和 CSSSelect 那些东西，没有把网站爬个遍...


你会发现，尽管我的爬虫技术还很烂...但是我做的大部分尝试都是有反馈的，比如发布模块可以看到下载次数(心想着这种坑爹模块也有人下载啊)，抓教务系统数据很好玩(我发现全校有250个左右的同学重名)...把小伙伴的网站搞垮再去报Bug...但我觉得这样学习很有趣。

而且我觉得我对之后学：怎么抓需要验证码的网站，怎么把网站爬个遍之类的进阶技能也很感兴趣~

发布于 2015-02-25    4 条评论    感谢    分享    收藏

• 没有帮助 • 举报 • 作者保留权利

---

▲ 罗轩，iOS开发, [luoyibu.com](http://luoyibu.com) 

4

FAN TOM、Chan Nuclear、连坤 等人赞同

▼ 我先开始不太了解爬虫的原理，简单百度了下。然后想着先试一下，就直接找到了scrapy框架。

[Scrapy Tutorial](#) 读了这个文档后，看一下官方的例子，就可以用scrapy框架做一个简单的爬虫了。然后爬虫基本原理又有了点理解，当然关于性能之类的，还需要自己去思考，或者看看一般爬虫框架的源码实现（比如scrapy）。

发布于 2014-05-04    1 条评论    感谢    分享    收藏

• 没有帮助 • 举报 • 作者保留权利

---

▲ 匿名用户

3

周圆方、张建凯、啦啦啦 赞同

▼ 背景.在校大三学生狗，学过基础C++

课题要求需要下载大量的东西作为样本分析。手动下载太麻烦。然后用一个小时学了基本的python。然后看了一下目标网站的源码，看了一下格式。稍微想下方式。然后写出了一个简单的爬虫。

总共用时一个晚上。

只匹配一个网站的格式的最简陋的爬虫。能用。

发布于 2014-11-18    添加评论    感谢    分享    收藏

• 没有帮助 • 举报 • 作者保留权利

更多

我来回答这个问题

写回答...

我要回答