

专栏：004：网页下载器的使用

E - 爬虫技术 爬虫知识

系列爬虫专栏

崇尚的学习思维是：输入，输出平衡，且平衡点不断攀升。

曾经有大神告诫说：没事别瞎写文章；所以，很认真的写的是能力范围内的，看客要是看不懂，不是你的问题，问题在我，得持续输入，再输出。

差不多正式涉及所谓的网页爬虫

1：框架

序号	内容	说明
01	网络爬虫知识概况	概念是理解和精进的第一步
02	urllib	简单说明使用方法
03	request	强烈建议入手
04	代码示例	使用request爬取博客
05	参考及备注	总结与说明

2：网络爬虫

- 概念

网络爬虫：网络蜘蛛（Web spider）也叫网络爬虫（Web crawler）[1]，蚂蚁（ant），自动检索工具（automatic indexer），或者（在FOAF软件概念中）网络疾走（WEB scutter），是一种“自动化浏览网络”的程序，或者说是一种网络机器人。它们被广泛用于互联网搜索引擎或其他类似网站，以获取或更新这些网站的内容和检索方式。它们可以自动采集所有其能够访问到的页面内容，以供搜索引擎做进一步处理（分检整理下载的页面），而使得用户能更快的检索到他们需要的信息。

分解复述：爬虫是一段代码，实现的功能是在网页上解析需要的信息。

- 涉及的名词

URL(Uniform Resource Locator):统一资源定位符,URL可以用一种统一的格式来描述各种信息资源，包括文件、服务器的地址和目录等.

URL的格式由三部分组成：

- 第一部分是协议(或称为服务方式)。
 - 第二部分是存有该资源的主机IP地址(有时也包括端口号)。
 - 第三部分是主机资源的具体地址，如目录和文件名等。
- 第一部分和第二部分用“://”符号隔开，
第二部分和第三部分用“/”符号隔开。
第一部分和第二部分是不可缺少的，第三部分有时可以省略。

如：<http://www.jianshu.com/collection/dfcf1390085c>

网络爬虫就是根据这些URL获取网页信息，再对获取到的网页源代码进行解析出所需要的信息。

3：urllib 库的使用简介

python2 和 python3中使用这个库的方法不一样，具体参考文档说明
在python3中，urllib模块被拆分为urllib.request，urllib.parse 和urllib.error

以python3 为例，别问我为什么使用python3, 遇到编码问题你就懂我的好了。

序号	常用方法	解释说明
01	urllib.request.urlopen()	
02	urllib.request.Request()	

```
# 代码示例
# -*- coding:utf-8 -*-
# To: learn module
# Date:2016.04.28
# Author: wuxiaoshen
import urllib.request

url = "http://www.geekonomics10000.com/author/admin"
html = urllib.request.urlopen(url)
response = html.read().decode('utf-8')
print(response)
```

网页在浏览器下的部分显示截图：

Posted in 科研精神 | 10 Comments



```
<div class="col2">
<ul id="sidebar">
  <a href="http://www.geeconomics10000.com/feed"><h5>RSS:</h5></a>
<a href="https://www.google.com/reader/preview?/feed/http://www.geeconomics10000.com/feed">订阅到 Google Reader</a><br>
<a href="http://www.xuehuixia.com/qdd_channel.php?url=http://http://www.geeconomics10000.com/feed">订阅到 抓虾</a> |
<a href="http://www.xinaguo.com/subscribe.php?url=http://www.geeconomics10000.com/feed">订阅到 鲜果</a><br>
E-Mail: 
<br>
<h5>我的书架</h5>
<p style="text-align: center;"><a href="http://www.geeconomics10000.com/940">
<br>
<a href="http://www.geeconomics10000.com/940">知识分子: 做个冗余的现代入</a>
</p>
<p style="text-align: center;"><a href="http://www.geeconomics10000.com/833"></a>
<br>
<a href="http://www.geeconomics10000.com/833">万万没想到: 用理工科思维理解世界</a>
</p>
```

3/11

05

响应状态码，响应头部

逐个分解使用方法：url = “<http://www.geekonomics10000.com/author/admin>” 会经常被我用来分析爬虫知识。

本人非常喜欢这个博客：[学而时嘻之](#)

`requests` 是第三方python库，需要自己安装。安装出问题？生命不息，折腾不止(暴露了是罗粉？)

- 01：发送请求，继而下载网页源代码

```
# 实现的和urllib代码相同的功能：
# -*- coding:utf-8 -*-
# To: learn module
# Date:2016.04.28
# Author: wuxiaoshen
import requests
url = "http://www.geekonomics10000.com/author/admin"
html = requests.get(url)
response = html.text
print(response)
```

结果部分显示截图：

```
<h3>Posts by 同人于野</h3>
<div id="post-970" class="post-970 post type-post status-publish format-standard hentry category-science">
<h3 id="post-970" class="post-title"><a href="http://www.geekonomics10000.com/970" rel="bookmark" title="Permanent Link to 别指望灵感，还是要靠汗水 ——“创造性思维”的三个迷信">别指望灵感，还是
<small>星期二，四月 5th, 2016</small>
<p> Posted in <a href="http://www.geekonomics10000.com/category/science" rel="category tag">科研精神</a> | <a href="http://www.geekonomics10000.com/970#comments">10 Comments</a></div>
<div id="post-967" class="post-967 post type-post status-publish format-standard hentry category-politics">
<h3 id="post-967" class="post-title"><a href="http://www.geekonomics10000.com/967" rel="bookmark" title="Permanent Link to 特朗普是极右狂人？其实共和党候选人里，他最温和">特朗普是极右狂人？非
<small>星期二，四月 5th, 2016</small>
<p> Posted in <a href="http://www.geekonomics10000.com/category/politics" rel="category tag">科学政治主义</a> | <a href="http://www.geekonomics10000.com/967#comments">6 Comments</a></div>
<div id="post-963" class="post-963 post type-post status-publish format-standard hentry category-self_develop">
<h3 id="post-963" class="post-title"><a href="http://www.geekonomics10000.com/963" rel="bookmark" title="Permanent Link to 超强记忆力是个邪道功夫">超强记忆力是个邪道功夫</a></h3>
<small>星期二，四月 5th, 2016</small>
<p> Posted in <a href="http://www.geekonomics10000.com/category/self_develop" rel="category tag">反求诸己</a> | <a href="http://www.geekonomics10000.com/963#comments">5 Comments</a></div>
<div id="post-940" class="post-940 post type-post status-publish format-standard hentry category-uncategorized">
<h3 id="post-940" class="post-title"><a href="http://www.geekonomics10000.com/940" rel="bookmark" title="Permanent Link to 我的新书《智识分子：做个冗余的现代人》">我的新书《智识分子：做个冗余
<small>星期五，一月 15th, 2016</small>
<p> Posted in <a href="http://www.geekonomics10000.com/category/uncategorized" rel="category tag">Uncategorized</a> | <a href="http://www.geekonomics10000.com/940#comments">20 Co
</div>
<div id="post-934" class="post-934 post type-post status-publish format-standard hentry category-books">
<h3 id="post-934" class="post-title"><a href="http://www.geekonomics10000.com/934" rel="bookmark" title="Permanent Link to 2016新年荐书">2016新年荐书</a></h3>
```

- 02：URL传递参数

你也许经常想为URL的查询字符串(query string)传递某种数据。如果你是手工构建URL，那么数据会以键/值 对的形式置于URL中，跟在一个问号的后面。例如，[httpbin.org/get?](http://httpbin.org/get?key=val)key=val

比如：url = "<http://yanbao.stock.hexun.com/xgq/gsyj.aspx?l=1&page=1>"

你想获取不同的网页，你通过翻页发现，只改动page后面的数字就可以了。

你有可能为了获取更多的url,会这样：

```
url = "http://yanbao.stock.hexun.com/xgq/gsyj.aspx?l=1&page="+str(i)
```

那么传递参数是怎么整的？

```
# -*- coding:utf-8 -*-
# To: learn module
# Date:2016.04.28
# Author: wuxiaoshen
import requests

url = "http://yanbao.stock.hexun.com/xgq/gsyj.aspx"
data = {"1": 1, "page": 4}
html = requests.get(url, params=data)
print(html.url)

# output
http://yanbao.stock.hexun.com/xgq/gsyj.aspx?page=4&1=1
别问我为什么后面的位置反了，又没影响正常访问。
好吧。因为字典是无序的。
```

• 03 : 响应内容

读取服务器响应的内容：

```
# -*- coding:utf-8 -*-
# To: learn module
# Date:2016.04.28
# Author: wuxiaoshen
import requests

url = "http://www.geekonomics10000.com/author/admin"
html = requests.get(url)
response_1 = html.text          #
response_2 = html.content       # 以字节的方式访问请求响应体，对于非文本请求
response_3 = html.raw           # 原始响应
print(type(response_1))
print(type(response_2))
print(type(response_3))

# output
<class 'str'>
<class 'bytes'>
<class 'requests.packages.urllib3.response.HTTPResponse'>

# 一般选择第一种 text 响应...
```

• 04 : 响应头部

防盗链和伪装成浏览器访问：

防盗链就是需要在请求的头部加入Referer字段, Referer 指的是HTTP头部的一个字段, 用来表示从哪儿链接到目前的网页, 采用的格式是URL。换句话说, 借着 HTTP Referer 头部网页可以检查访客从哪里而来, 这也常被用来对付伪造的跨网站请求。

某些网站做了限制, 进制爬虫的访问, 此时我们可以更改HTTP的header

HTTP状态码HTTP状态码 (英语 : HTTP Status Code) 是用以表示网页服务器HTTP响应状态的3位数字代码。

比较常见的是200响应成功。403禁止访问。

2xx成功

3xx重定向

4xx客户端错误

5xx服务器错误

```
# -*- coding:utf-8 -*-
# To: learn module
# Date:2016.04.28
# Author: wuxiaoshen
import requests

url = "http://blog.csdn.net/pongba"    # 刘未鹏的CSDN博客地址
html = requests.get(url)
print(html.status_code)

# output:
403

---
# 添加头部信息:
# -*- coding:utf-8 -*-
# To: learn module
# Date:2016.04.28
# Author: wuxiaoshen
import requests

url = "http://blog.csdn.net/pongba"

headers = {"User-Agent": 'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/49.0.2623.110 Safari/537.36',
           "Referer": 'http://blog.csdn.net/pongba/article/details/7911997'}
html = requests.get(url, headers=headers)
print(html.status_code)

# output
200
```

如何获取头部信息：截图演示：
chrome 浏览器，右键，检查。



5：实战抓取博文

获取 刘未鹏 博客：[\[BetterExplained\]如何有效地记忆与学习](#) 的全部博文
[文章地址](#)


```
# -*- coding:utf-8 -*-
# To: learn module
# Date:2016.04.28
# Author: wuxiaoshen
import requests
import re
import codecs

class LiuweipengBlog(object):
    def __init__(self):
        self.url = "http://blog.csdn.net/pongba/article/details/4033477"
        self.header = {"User-Agent": 'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/49.0.2623.110 Safari/537.36',
                        "Referer": 'http://blog.csdn.net/pongba/article/details/7911997'}

        self.pattern_content = r'<div id="article_content" class="article_content">(.*?)</div>'
        self.pattern_word = r'<strong>(.*?)</strong>'

    pass

    def download(self):
        html = requests.get(self.url, headers=self.header)
        try:
            if html.status_code == 200:
                return html.text
        except:
            print("Something with it.")

    pass

    def parse_content(self, content):
        passage = re.findall(self.pattern_content, content, re.S)
        words = re.findall(self.pattern_word, str(passage), re.S)
        print(words)
        return words

    pass

    def save_content(self, passage):
        filename = "blog.txt"
        with codecs.open(filename, 'w', encoding='utf8') as f:
            f.write(str(passage))
```

pass

```
if __name__ == "__main__":
    Blog_passage = LiuweipengBlog()
    content = Blog_passage.download()
    passage = Blog_passage.parse_content(content)
    Blog_passage.save_content(passage)
```

分析过程显示：正则为什么那样写：
网页源代码唯一标示啊，然后再在这里面分析，大部分文字在 `(.*?)`
注意到写的正则没有使用很复杂的表达式，就使用了 `(.*?)` 就完成了大部分任务。

```
251         </label>
252     </div>
253 </div>
254 <script type="text/javascript" src="http://static.blog.csdn.net/scripts/category.js"></script>
255 <div class="blog_copyright">
256     <p class="copyright_p"> 版权声明：本文为博主原创文章，未经博主允许不得转载。 </p>
257 </div>
258
259
260
261
262
263
264
265 <div id="article_content" class="article_content">
266 <p><strong>你所拥有的知识并不取决于你记得多少，而在于它们能否在恰当的时候被回忆起来。</strong></p>
267 <p>让我稍微说得再详细一点：学习新知识并将其存放于大脑中，<strong>最终的目的是要在恰当的时候能够想得起来去使用</strong>。因此，学习的有效性<strong>显然应该这样来衡量</strong>：当遇到需要用到学过的知
268 识的时候，相关的知识<strong>是否会自动从你脑海中“蹦”出来</strong>，最起码“Andash.Andash.能否通过有意识的搜索将它们提取出来。</p>
269 <p>这可不像它听上去那么简单，否则就不会有“掉书袋”、“读死书”这种修辞手法了。</p>
270 <p>为了更深入地说明这一点，以下是几个著名的关于学习与记忆机制的实验：</p>
271 <p>《找寻逝去的自我》上提到这样一个例子：</p>
```

效果显示：
网页的文章开头：

原

[BetterExplained]如何有效地记忆与学习

标签： distance 语言 娱乐 音乐 生活 任务

2009-03-29 11:56 71551人阅读 评论(82) 收藏 举报

分类： 片面思考 (36)

版权声明：本文为博主原创文章，未经博主允许不得转载。

你所拥有的知识并不取决于你记得多少，而在于它们能否在恰当的时候被回忆起来。

抓取的开头：

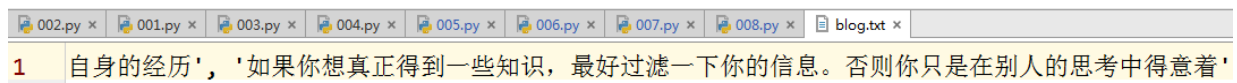
002.py x 001.py x 003.py x 004.py x 005.py x 006.py x 007.py x 008.py x blog.txt x

1 你所拥有的知识并不取决于你记得多少，而在于它们能否在恰当的时候被回忆起来。

网页的结尾：

观察、阅读，并别忘带着你的理性去审视（包括本文），弄清娱乐是娱乐，知识是知识，如果你想真正得到一些知识，最好过滤一下你的信息。否则你只是在别人的思考中得意着。

抓取的结尾：



代码还存在好些值得优化的地方(不写注释的程序员, 不是个好吃货)。你懂的。因为...我还有事。。

可以先直观的看看实现过程。

6：参考及说明

参考资料1：

[requests文档](#)

[urllib文档](#)

[正则表达式参考教程：](#)

[爬虫系列教程](#)

关于本人：

国内小硕，跌跌撞撞的IT学习者。

兴趣领域：爬虫及数据科学

本人正在构建一个爬虫学习付费社群。付费是为了降低信噪比。社群的理念是：思维，不断的精进。

有兴趣的可以私信，限制30名。群内鼓励原创教程，不断交流精进，目前已经有小伙伴参加。