

实验报告1 PARSER

518030910303 纪喆

2019 年 9 月 21 日

1 实验准备

1.1 实验环境

本次实验在 **Ubuntu 14.04 LTS**系统中进行，使用的是**Python3**编程语言。本次实验主要用到的工具有：

- vim
- Python3
- pycharm

本次实验使用的库有：

- sys
- urllib
- beautifulsoup4
- requests
- re

对于实验环境的准备问题，详见README.md

1.2 实验目的

本次实验分为三个小问题，实验目的（功能）分别为

- 给定URL，提取对应网页中的所有URL并打印

- 给定URL，提取对应网页中的所有图片资源并打印
- 针对“知乎日报”这一网页，提取其主要信息，按照“图片-标题-详情网页”的顺序打印

1.3 实验原理

对于本次的三个问题，大体框架相同，均分为三步

1. 发送HTML请求获取网页信息
2. 解析网页信息，提取目标内容
3. 将目标内容打印至指定文档

具体的，用`requests`库发送HTML请求，接收返回信息。用`beautifulsoup`库对返回内容进行解析，寻找目标内容。最终以常规方式写入文件。

2 实验过程

本部分按照程序的运行顺序，采用代码+解释的方式展示本实验。

2.1 练习一

```
if len(sys.argv) > 1:
    url = sys.argv[1]
```

这两行代码使用了`sys`提供的接口，检查Terminal中传入的参数个数，以此判断用户是否输入了URL。如果有URL输入，则更新变量`url`的内容。否则使用默认URL“`http://www.sjtu.edu.cn`”

```
r = requests.get(url, headers = {'User-Agent': 'Chrome'})
r.raise_for_status()
urls = parseURL(url, r.text)
```

这一部分使用`requests`库中的`get`函数发出HTTP请求。其中，`headers`的设置是让程序模拟Chrome浏览器浏览页面，从而能够访问更多的页面，拓宽程序的使用范围。

第二行的 `r.raise_for_status()` 则是用来判断是否成功获得网页内容。其具体作用为，当`r`的状态码不是200时，产生一个 `HTTPError`。只有当`r`的

状态码为200时，也就是内容成功获取时，才会进行后续工作。第三行则是调用了本练习的核心函数parseURL，参数传入了网站本身URL以及r的返回网页的内容。函数作用就是把第二个参数传入的网页内容中的所有URL都寻找出来，以列表形式返回。parseURL函数的核心代码如下：

```
soup = bs4.BeautifulSoup(content, 'html.parser')
for a in soup.find_all('a', {'href':re.compile('^http|^/')}):
    urlset.add(urljoin(url, a.get('href')))
```

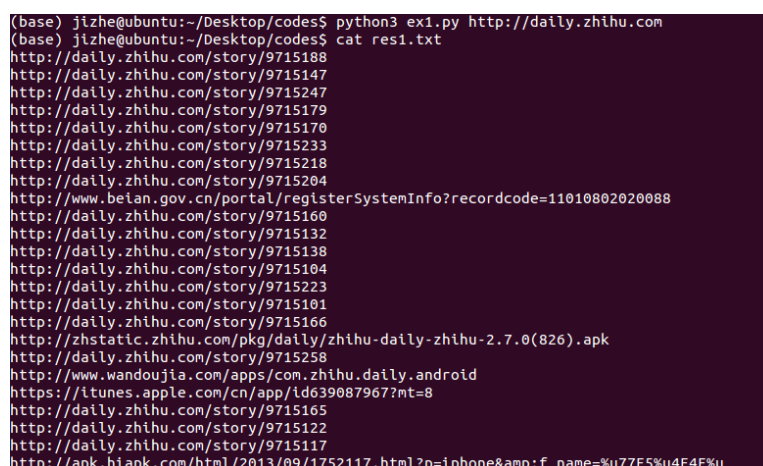
第一行为建立一个BeautifulSoup实体。第一个参数是网页的内容，第二个为指定的PARSER。实际上不指定PARSER也可以解析，但是有一定的可能造成在不同环境下的运行结果有差异。

第二行开始的for循环为核心部分。利用 *BeautifulSoup* 的find_all方法，利用正则表达式库检查href的内容，找出属性 href 是绝对路径或者相对路径的< a >标签。用urljoin函数将所有路径规范成绝对URL后，加入列表中。最终函数返回整个 urlset

```
for url in urls:
    f.write(url)
    f.write('\n')
```

返回后的urlset 会作为第一个参数传入函数 write_outputs(set,fname)，该函数的第二个参数是为了指定文件名称。由于其作用原理简单，不在此赘述。

最终运行结果如图：



```
(base) jizhe@ubuntu:~/Desktop/codes$ python3 ex1.py http://daily.zhihu.com
(base) jizhe@ubuntu:~/Desktop/codes$ cat res1.txt
http://daily.zhihu.com/story/9715188
http://daily.zhihu.com/story/9715147
http://daily.zhihu.com/story/9715247
http://daily.zhihu.com/story/9715179
http://daily.zhihu.com/story/9715170
http://daily.zhihu.com/story/9715233
http://daily.zhihu.com/story/9715218
http://daily.zhihu.com/story/9715204
http://www.beian.gov.cn/portal/registerSystemInfo?recordcode=11010802020088
http://daily.zhihu.com/story/9715160
http://daily.zhihu.com/story/9715132
http://daily.zhihu.com/story/9715138
http://daily.zhihu.com/story/9715104
http://daily.zhihu.com/story/9715223
http://daily.zhihu.com/story/9715101
http://daily.zhihu.com/story/9715166
http://zhstatic.zhihu.com/pkg/daily/zhihu-daily-zhihu-2.7.0(826).apk
http://daily.zhihu.com/story/9715258
http://www.wandoujia.com/apps/com.zhihu.daily.android
https://itunes.apple.com/cn/app/id639087967?mt=8
http://daily.zhihu.com/story/9715165
http://daily.zhihu.com/story/9715122
http://daily.zhihu.com/story/9715117
http://apk.hiapk.com/html/2013/09/1752117.html?p=iphone&f_name=%u77E5%u4E4E%u
```

图 1: 练习一运行结果

2.2 练习二

此练习中，`main()`函数基本与练习一相同，用来解析的`parseIMG`的结构与练习一的`parseURL`也差别不大。主要差别在于对标签的检索。

```
for img in soup.findAll('img'):  
    if img.get('src'):  
        imgset.add(urljoin(url, img.get('src')))
```

由于假定图片的标签全部为``，因此直接搜索标签名，将属性`src`提取出来后与网页URL合并成绝对URL即可。

其他内容与练习一完全相同，在此不做赘述。

```
(base) jizhe@ubuntu:~/Desktop/codes$ python3 ex2.py http://daily.zhihu.com  
(base) jizhe@ubuntu:~/Desktop/codes$ cat res2.txt  
https://pic4.zhimg.com/v2-fb1adb9ee8bfe2d0d3cdd80bce79cbaf.jpg  
http://daily.zhihu.com/img/new_home_v3/qrcode_top.png  
http://daily.zhihu.com/img/new_home_v3/qrcode_bottom.png  
https://pic1.zhimg.com/v2-4abf0037c84b8be16ba3b04a43fefefc.jpg  
https://pic2.zhimg.com/v2-bb485cfe63264eb52bd509fb11b6afc9.jpg  
https://pic4.zhimg.com/v2-86b9d92197f3ff1b17e2dc70720463.jpg  
https://pic4.zhimg.com/v2-5b04882f31501d5195dc5c58f6523187.jpg  
https://pic1.zhimg.com/v2-1930df119ce0d39eb4895c7944e4e758.jpg  
https://pic1.zhimg.com/v2-b17958de5e8a586812fde37e73876a6c.jpg  
https://pic2.zhimg.com/v2-cdd6b075183532fc1c1de2635aef445.jpg  
https://pic1.zhimg.com/v2-784e12528c683e11a00fa21d0007e7cc.jpg  
https://pic1.zhimg.com/v2-cf3a2b2f21ed2e7a85af34f6eb1aaa34.jpg  
https://pic3.zhimg.com/v2-0ab4432fa18202d1c2cd0aa71dd394b6.jpg  
https://pic1.zhimg.com/v2-7e9289d2e48f0276f571dab2780630ac.jpg  
https://pic4.zhimg.com/v2-11b7ff7dcb6fcd1362a828928436899f.jpg  
https://pic3.zhimg.com/v2-f09bbe130943cac280820f1de4b19f02.jpg  
https://pic1.zhimg.com/v2-a88ed04f1269f4a76243dd9e7d25ef4.jpg  
https://pic2.zhimg.com/v2-db0eb675bc82a52e87bcb86b1027cd55.jpg  
http://daily.zhihu.com/img/new_home_v3/phone_sample.png  
https://pic4.zhimg.com/v2-d71d1d1f375f375f375f375f375f375f.jpg
```

图 2: 练习二运行结果

2.3 练习三

为了对知乎日报网页更有效的提取信息，首先我们需要利用浏览器的检查功能大致了解网页结构。



图 3: 网页的检查结果

从图中可以看出，需要提取的每组信息的结构为：

```
<a href = ... class = "link-button">
  <img src = ... class = ...>
  <span class = ...> ... </span>
</a>
```

因此在解析页面的时候，只需要寻找 `class` 属性为 `link-button` 的 `a` 标签。然后通过搜索标签内的内容，获取 `img` 的 `src` 以及 `span` 的文本。

```
l = soup.find_all('a', {'class': 'link-button'})
total = list()
for m in l:
    ans = [urljoin(url, m.find('img').get('src')), m.
            find('span').string, urljoin(url, m.
            get('href'))]

total.append(ans)
```

上方的代码便是针对这一特定的网页组织结构建立的提取代码。其中 `total` 的每个元素是一个长度为3的 `list`，每个 `list` 里面有图片源，`span` 的文本和故事链接。最后通过 `writeAns(ans, filename)` 函数，按照规则打印答案。

限于篇幅，这里只给出运行结果的一部分。

```
(base) jizhe@ubuntu:~/Desktop/codes$ python3 ex3.py
(base) jizhe@ubuntu:~/Desktop/codes$ cat res3.txt
https://pic3.zhimg.com/v2-82cea5983830972e7130bb138044ba32.jpg
沐浴露是不是新世纪的一场营销骗局？
http://daily.zhihu.com/story/9715261
https://pic3.zhimg.com/v2-1d2aeb2aae51f1c5485e6b7f1fa2a576.jpg
如何评价周杰伦发布的新歌《说好不哭》？
http://daily.zhihu.com/story/9715276
https://pic1.zhimg.com/v2-1930df119ce0d39eb4895c7944e4e758.jpg
速度提升四成的下一代 Wi-Fi 6 标准正式启用，会带来哪些影响？
http://daily.zhihu.com/story/9715267
https://pic2.zhimg.com/v2-bb485cfe63264eb52bd509fb11b6afc9.jpg
瞎扯 · 如何正确地吐槽
http://daily.zhihu.com/story/9715258
https://pic3.zhimg.com/v2-0ab4432fa18202d1c2cd0aa71dd394b6.jpg
VIE 结构是什么？建立的过程中需要注意什么问题？
http://daily.zhihu.com/story/9715247
https://pic2.zhimg.com/v2-cdd6b075183532fc1c1de26353aef445.jpg
喝牛奶会使皮肤变白吗？
http://daily.zhihu.com/story/9715233
https://pic1.zhimg.com/v2-cf3a2b2f21ed2e7a85af34f6eb1aaa34.jpg
《名侦探柯南》按主线时间线来算，现在距离新一变成柯南过了多久？
http://daily.zhihu.com/story/9715238
https://pic1.zhimg.com/v2-a88ed04f1269f4a76243ddd9e7d25ef4.jpg
如何看待《「学渣」儿子，妈妈相信你是来报恩的》的一文？
http://daily.zhihu.com/story/9715251
```

图 4: 练习三运行结果

3 实验总结

3.1 概述

本次实验的前两个练习为最基本的解析动作。第三个练习则是在前两个练习的基础上，在特定的环境下设计代码读取网络内容。在本次实验中，我学到了如何用程序发起`HTTP`请求，如何解析网络页面，提取关键信息。这些内容想必都是日后课程中的基础。本次实验是一个良好的开端，点燃了我对后续实验的热情。

3.2 感想

从基本解析动作到提取网页结构化信息，这样的从简到繁的操作顺序体现了自古以来的科学学习方式。在实践中探索，在试错中发展。我切实感觉到我正在打开一个全新的世界，并对此后的学习充满信心。

3.3 问题与解决

问题： 由于本实验使用的语言为`Python3`，并非`Ubuntu`系统默认`python`，这导致了在安装库的时候会遇到一些麻烦。倘若直接使用命令 `pip install beautifulsoup` 则会把 `bs4` 库装到 `Python2` 的解释器下，当用`Python3` 运行文件时，仍会报错找不到库文件。

解决： 根据资料查找，发现在 `Ubuntu` 系统中，`python3`所对应的 `pip` 对应命令为 `pip3 install ...` 而`Ubuntu` 默认情况是不自带 `pip3`。因此需要先`sudo apt install python3-pip`后再操作