

# 实验一 实验报告

517030910294 刘瀚文

## 1 实验准备

### 1.1 环境配置准备

在本次实验之前，依照 PPT 的说明安装了 VirtualBox 虚拟机 (Ubuntu 操作系统)，并且完成了相应的功能和安装包的更新 (pip, vim 等)，以及 Miniconda、Python 和 Pycharm 的安装。

### 1.2 相关语言学习准备

上学期的电工导课程上已经学习了一定的 HTML 的初步知识，同时在课堂上对网络爬虫的技术进行了了解。HTML 标签用<>框起的关键字成对出现，第一个是起始标签，第二个是结束标签。在标签队中往往可以嵌套使用 HTML 元素。

BeautifulSoup：对网站进行结构分析的工具。

BeautifulSoup 是用 Python 写的一个 HTML/XML 的解析器，把 html 纯文本转化为便于程序访问的数据结构。

在从网站获取信息的时候也使用了正则表达式，使搜索更快捷简便。

### 1.3 实验的目的及原理

1.3.1 学习 HTML 语言，了解 HTML 使用标记标签来设计网页的原理。

1.3.2 学习使用 BeautifulSoup，从复杂漫长的 HTML 语言中自动提取出我们所需要的信息。

### 1.3.3 使用 Chrome 浏览器的 F12 案件快速读取 HTML 文本信息。

## 二．实验过程

### 2.1 练习一

#### 2.1.1 实验步骤

第一步：引入相关的 Python 库，为之后的操作做好准备。

```
import sys
import urllib2
from bs4 import BeautifulSoup
```

第二步：定义相关的函数。

第一个函数：爬取网址链接的函数 parseURL()

```
def parseURL(content):
    urlset = set()
    soup = BeautifulSoup(content)
    for i in soup.findAll('a'):
        url = i.get('href', '')
        print(url)
        urlset.add(url)
    return urlset
```

使用 BeautifulSoup 从网页中爬取<a>标签中所有“href”链接的地址，并将所有的链接地址存储在一个 set 中返回。

第二个函数：文件写入函数 write\_outputs():

```
def write_outputs(urls, filename):
    with open(filename, 'w') as f:
        for url in urls:
            f.write(url)
            f.write('\n')
```

打开文件，然后把 url 写入到我们的文件夹内。

```
def main():
    url = 'http://www.baidu.com'
    # url = 'http://www.sjtu.edu.cn'
    if len(sys.argv) > 1:
        url = sys.argv[1]
    content = urllib2.urlopen(url).read()
    urls = parseURL(content)
    print(type(urls))
    write_outputs(urls, 'res1.txt')

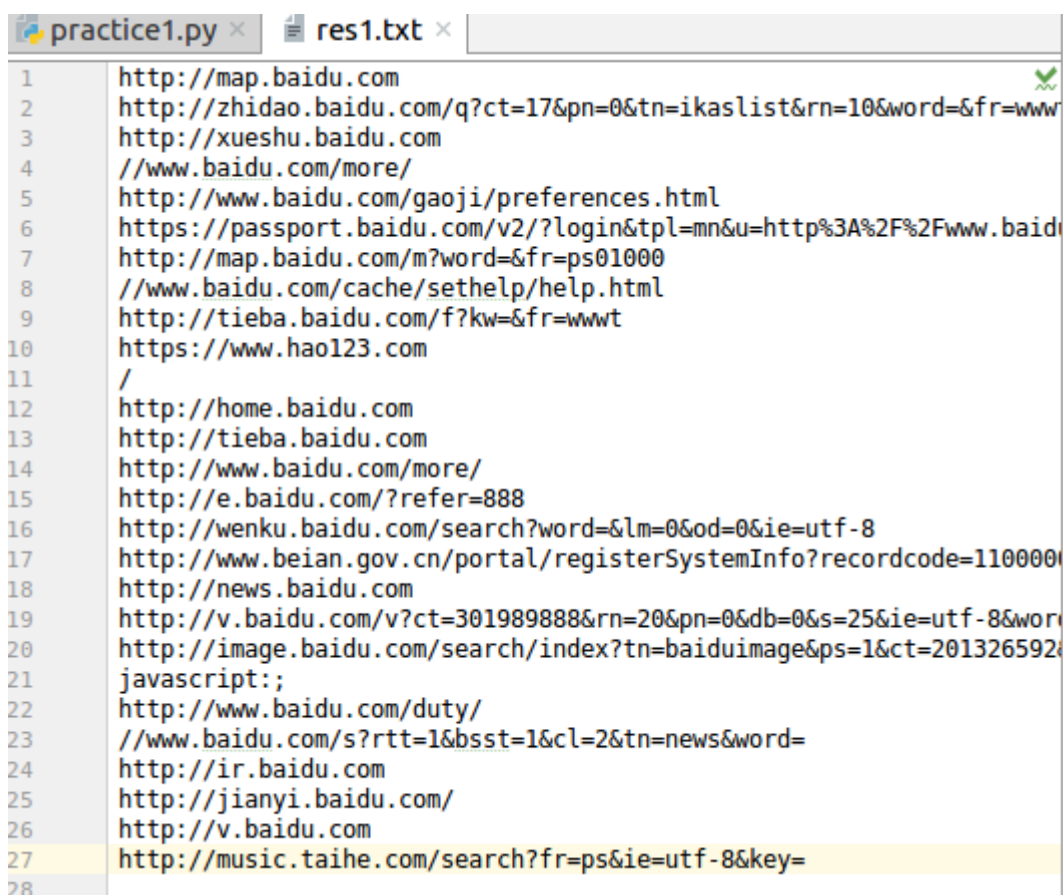
if __name__ == '__main__':
    main()
```

第三个函数：主函数 main()：

在主函数中输入需要爬取的网页（即特异化信息在主函数处输入，并且将网页转换成为 HTML 文本的格式。最后将函数处理后的信息打印在文件中。

### 2.1.2 实验结果

给定任意网页内容，返回网页中所有超链接的 URL（不包括图片地址），并将结果打印至文件 res1.txt 中，每一行为一个链接地址。



```
practice1.py x res1.txt x
1 http://map.baidu.com
2 http://zhidao.baidu.com/q?ct=17&pn=0&tn=ikaslist&rn=10&word=&fr=www
3 http://xueshu.baidu.com
4 //www.baidu.com/more/
5 http://www.baidu.com/gaoji/preferences.html
6 https://passport.baidu.com/v2/?login&tpl=mn&u=http%3A%2F%2Fwww.baidu
7 http://map.baidu.com/m?word=&fr=ps01000
8 //www.baidu.com/cache/sethelp/help.html
9 http://tieba.baidu.com/f?kw=&fr=wwwt
10 https://www.hao123.com
11 /
12 http://home.baidu.com
13 http://tieba.baidu.com
14 http://www.baidu.com/more/
15 http://e.baidu.com/?refer=888
16 http://wenku.baidu.com/search?word=&lm=0&od=0&ie=utf-8
17 http://www.beian.gov.cn/portal/registerSystemInfo?recordcode=1100000
18 http://news.baidu.com
19 http://v.baidu.com/v?ct=301989888&rn=20&pn=0&db=0&s=25&ie=utf-8&word
20 http://image.baidu.com/search/index?tn=baiduimage&ps=1&ct=201326592
21 javascript:;
22 http://www.baidu.com/duty/
23 //www.baidu.com/s?rtt=1&bsst=1&cl=2&tn=news&word=
24 http://ir.baidu.com
25 http://jianyi.baidu.com/
26 http://v.baidu.com
27 http://music.taihe.com/search?fr=ps&ie=utf-8&key=
28
```

虚拟机中 res1.txt 结果展示。

## 2.2 练习二

### 2.2.1 实验步骤（与实验一较为相似）

第一步：

引入相关的 Python 库，为之后的操作做好准备。

第二步：

定义相关的函数：函数一：爬取图片链接的函数

parseIMG(), 与实验一不同的就只是查找的标签是'img', 返

回需要的标签是'src'。函数二：文件写入函数

write\_outputs(), 与实验一相似。主函数：main (), 与实验一相似。

代码展示：

```
1  import sys
2  import urllib2
3  from bs4 import BeautifulSoup
4
5
6  def parseIMG(content):
7      urlset = set()
8      soup = BeautifulSoup(content)
9      for i in soup.findAll('img'):
10         url = i.get('src', '')
11         print(url)
12         urlset.add(url)
13     return urlset
14
15
16 def write_outputs(urls, filename):
17     with open(filename, 'w') as f:
18         for url in urls:
19             f.write(url)
20             f.write('\n')
21
22
23 def main():
24     # url = 'http://www.baidu.com'
25     url = 'http://www.sjtu.edu.cn'
26     if len(sys.argv) > 1:
27         url = sys.argv[1]
28     content = urllib2.urlopen(url).read()
29     imgs = parseIMG(content)
30     write_outputs(imgs, 'res2.txt')
31
32
33 if __name__ == '__main__':
34     main()
35
```

## 2.2.2 实验结果

给定任意网页内容，返回网页中所有图片地址，并将结果打印至文件 res2.txt 中，每一行为一个图片地址。

实验结果展示：

```
images/logo120.png
images/ii_r2_c2.png
_mediafile/wwwsjtu2013/2018/05/14/13sfhuqp5y.png
_mediafile/wwwsjtu2013/2018/05/14/1fvzi51p5z.jpg
_mediafile/wwwsjtu2013/2018/05/14/3l2n8y3p5y.jpg
_mediafile/wwwsjtu2013/2018/05/14/2dvf9vep5y.jpg
images/i_r6_c6.png
|
```

虚拟机中 res2.txt 结果展示。

## 2.3 实验三

### 2.3.1 实验步骤

第一步：

引入相关的 Python 库，为之后的操作做好准备。

```
import sys
reload(sys)
sys.setdefaultencoding('utf-8')

import re
import urllib2
import urlparse
from bs4 import BeautifulSoup
```

与前面两个实验不同的是多引入了 re（用于正则表达式），urlparse（实现绝对网址的组合），reload(sys）（用于防止读取文本时中文乱码）

同时添加请求头中的 User-Agent。

```
agent = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.186 Safari/537.36'}
```

第二步，定义相关的函数。

第一个函数：爬取糗事百科的函数 `parseQiushibaikePic(content)`

实现以下几个功能：

- 使用正则表达式查找并返回特定'id'为  
`re.compile('^qiushi_tag_\d+')`
- 使用正则表达式查找并返回特定'class'为  
`re.compile('^content$')`——文本内容
- 使用正则表达式查找并返回特定'class'为  
`re.compile('^thumb$')` ——网站链接，同时使用  
`urlparse` 将其拼接为完整的网址。
- 使用正则表达式查找并返回特定'id'为  
`re.compile('^articleNextLink')`——下一页的网站地址，  
同时也是用 `urlparse` 进行拼接。

代码展示：

```
def parseQiushibaikePic(content):  
    soup = BeautifulSoup(content)  
    for i in soup.findAll('div', {'id': re.compile('^qiushi_tag_\d+')}):  
        global tag  
        tag = i.get('id', '')  
        print(tag)  
    for j in soup.body.findAll('div', {'class': re.compile('^content$')}):  
        for k in soup.body.findAll('div', {'class': re.compile('^thumb$')}):  
            log = 'http://www.'  
            img = urlparse.urljoin(log, k.img['src'])  
            docs = {tag.split('_')[-1]: {'content': j.text.replace("\n", ""), 'imgurl': img}}  
            print(docs)  
    for t in soup.body.findAll('input', {'id': re.compile('^articleNextLink')}):  
        nextpage = t.get('value')  
        url = 'http://www.qiushibaike.com/pic'  
        nextpage = urlparse.urljoin(url, nextpage)  
        return docs, nextpage
```

第二个函数：文件写入函数 `write_outputs()`:

```
def write_outputs(urls, filename):  
    with open(filename, 'w') as f:  
        urls = list(urls)  
        f.write([(urls[0])[tag.split('_')[-1]]['imgurl'] + '\t' + ((urls[0])[tag.split('_')[-1]]['content'] + '\n' + urls[1])])
```

打开文件，然后把要求的数据写入到我们的文件夹内。

第三个函数：主函数 main()：

```
def main():
    url = urllib2.Request('https://www.qiushibaike.com/article/112782870', headers=agent)
    if len(sys.argv) > 1:
        url = sys.argv[1]
    content = urllib2.urlopen(url).read()
    Info = parseQiushibaikePic(content)

    write_outputs(Info, 'res3.txt')

if __name__ == '__main__':
    main()
```

在主函数中输入需要爬取的网页（即特异化信息在主函数处输入，并且将网页转换成为 HTML 文本的格式。最后将函数处理后的信息打印在文件中。

### 2.3.2 实验结果

给定糗事百科有图有真相任意一页内容，返回网页中图片和相应文本，以及下一页的网址，并将图片地址与相应文本以下述格式打印至文件 res3.txt 中，每一行对应一个图片地址与相应文本，格式为：图片地址\t 相应文本

实验结果展示：

```
http://pic.qiushibaike.com/system/pictures/11278/112782870/medium/app112782870.jpg 有图有真相。。。哈哈.....
http://www.qiushibaike.com/article/119639555
```

虚拟机中 res3.txt 结果展示。

## 三．实验感想

本次实验中学习了 HTML 语言，对网页的设计有了基础的了解。学习了使用 BeautifulSoup 对网页信息的提取，同时复习了 python 的语法。