

实验一报告

#注：由于报告在给出格式建议前就已经写好，所以没有按照推荐格式。请见谅。

1.返回某网页中所有超链接

通过解析某网页的HTML源码，我们可以发现超链接的形式可以分为三种：

①形如：`内容`(其中`target="_blank"`表示新窗口打开)

②JS跳转链接：`内容`

③meta标签跳转链接：`<meta http-equiv="refresh" content="3;URL=网址">` (数字3为秒)

在这个实验中，我们只需考虑形式①的超链接方式，那么以网页的HTML内容`content`为参数的`parseURL(content)`函数构造思路如下：

首先调用BeautifulSoup库处理`content`，得到对象`soup`。对于`soup`，调用`findAll`函数，找到所有标签为`'a'`的节点，再遍历这些节点对其使用`get`函数，得到其`'href'`标签，也就是超链接的URL，这些URL储存在`urlset`中，作为函数的返回对象。

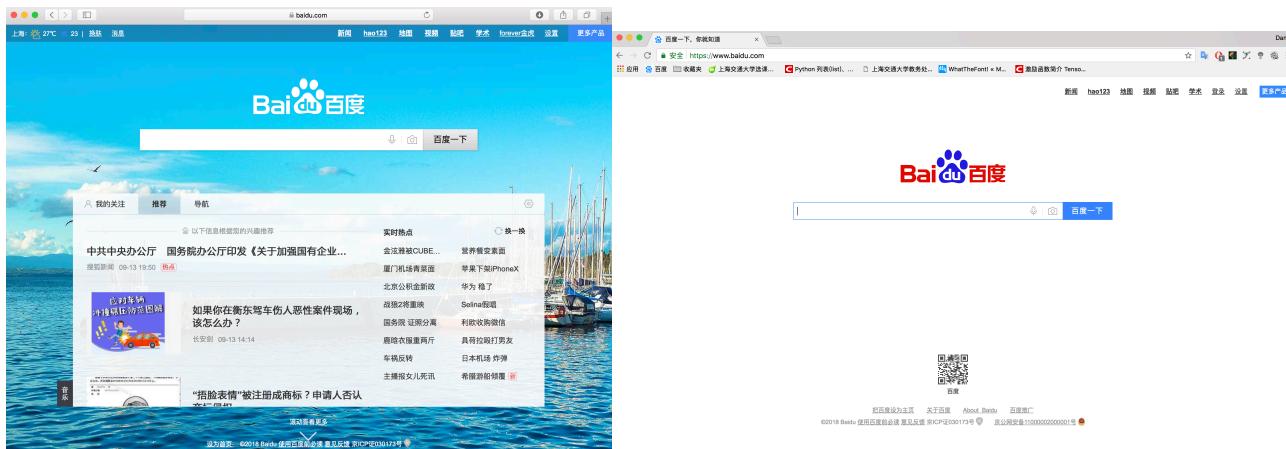
而`write_outputs(urls,filename)`函数则较为简单，仅涉及Python的文件读写问题。

网页的HTML内容`content`的获得，需要调用`urllib`库。由于我安装的是Python3，其中`urllib2`库被取消，等价于`urllib.request`库，因此调用`urllib.request`库即可完成任务。

特别注意的问题：

①分析得到的超链接URL集，我发现部分超链接头为`https://`，如个人中心部分，需要SSL证书认证，部分为`http://`或直接以`//`开头，另外一些为`'/'`或`'javascript:;'`，推测为有动态效果。

②不同浏览器打开的百度首页是不同的，这是什么原因呢？是百度根据收到不同浏览器请求而发出不一样的包吗？



同一时刻不同浏览器打开的百度网页（左为Safari，右为Chrome）

③尝试爬取百度以外的网页，发现许多爬不了，出现SSL:ACCESS FAILURE报错，也就是在第一阶段就没有连接成功。这是由于需要认证以及反爬机制，需要设置请求头，也就是在实验3中会经历的内容。

2.返回某网页中所有图片的地址

具体方法与实验1中基本相同。图片部分的标签为

需要注意的是，在写文件时，发现这些URL都是相对地址（相对于原网页），而我们需要绝对地址，因此我在write_outputs函数中加了一个传入参数url，也就是原网页的网址，再通过字符串处理，将两部分拼接起来得到绝对网址。事实上也可以调用urlparse库中的urljoin函数。

3.返回糗事百科主页中所有图片的网址及说明

首先发现的问题是，如果继续按照实验1、2中的方式爬取网页，会出现无法连接的报错，这是因为网页设置了反爬机制，我们需要设置请求头，把自己模拟成浏览器。这里我首先前往Chrome浏览器查看了自己的请求头header，这里我只需要'User-Agent'值，其他的cookie数据等没有用，则建立header={'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/68.0.3440.106 Safari/537.36'}，

再建立请求：

```
rq=urllib.request.Request(url,headers=header)
```

```
data=urllib.request.urlopen(rq).read()
```

data部分就是爬取的HTML内容。

在构建parseQiushibaikepic函数时，发现图片是储存在标签为'a'的结点里的。但是不是所有标签为'a'的结点都是图片，因此我调用了正则表达式'href':re.compile('^\/article')，也就是该结点的href需要以/article开头。这样初筛过后，得到的结点都是图片相关的。部分图片（主页主图）有连续三个的结点，分别是标题、图片网址、评论区，部分图片（侧边栏推荐部分）只有连续两个的结点。我采取的方法是首先爬取qs_tag，作为一副图片的PrimaryKey，再尝试爬取标题部分，如果结果content不为空，则这是标题部分，执行docs[qs_tag]={'content': content, 'imgurl': imgurl}，也就是在字典中添加字段(qs_tag未存在)，其中imgurl只是一个占位符。而若结果为空，则这是图片网址部分，执行

```
docs[qs_tag]['imgurl']=imgurl。就此，一副图片在docs中的字段已经建立完成，其key为qs_tag，value为其内容和网址。网址是相对URL，采用和试验2中类似的方法补全即可。
```

获取下一页的URL较为容易，也利用正则表达式，其href以'/pic/page/2'开头。