实验 1-HTML_parser

一、实验概览

通过对信息检索，网络搜索的基本概念，HTML 语言的结构、基础、常见 Tag 的初步了解以及对 HTML/XML 解析器 BeautifulSoup 的初步学习，尝试爬取任意网页超链接的 URL、所有图片链接以及相对应的文本内容。

二、实验环境

1. 个人笔记本电脑
2. 操作系统：windows10 专业版
3. 使用软件：Visual Studio Code；Docker Desktop

三、实验过程

练习一：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。建议参考 example1.py。

【解】

对这个问题首先可以进行宏观的分析以及问题模型的建构，进而得到练习一的整体流程，这种类型的模块化思想是计算机编程中非常重要的一块：

1. 加载网页内容

2. 爬全部超链接

3. 输出打印结果

只要完善、充实这个问题模型，练习一就可以得到比较好的解答。

1. 加载网页内容

首先，利用 python 模拟浏览器抓取 HTML 网页，此处对百度网页进行抓取：

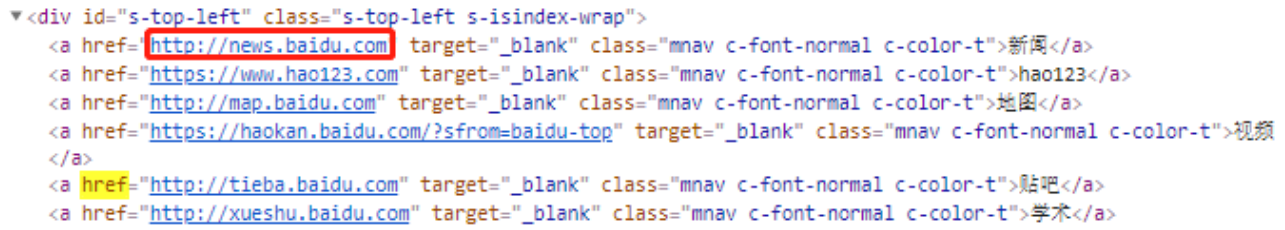
```
1.  content = urllib.request.urlopen("http://www.baidu.com").read()
```

对于得到的内容，利用 BeautifulSoup 的解析器将 content 转化为相应的数据结构：

```
2.  soup = BeautifulSoup(content)
```

2. 爬全部超链接

通过对百度搜索主页的 HTML 源码的查看，可以注意到，在该源码中，超链接是在<a>标签下且这个标签下的 href 值：



因而，我们只需要找到满足给定标签<a>标签下的所有 href 值即可提取出超链接：

```
for i in soup.findAll('a'):
    t = i.get('href','')
    urlset.add(t)
```

这里的 urlset 是我们用来存储超链接的一个集合：

```
urlset = set()
```

3. 输出打印结果

练习中要求输出到'res1.txt'中，根据 python 的文件操作与集合操作的知识，我们可以直接得到输出打印结果的代码程序：

```
file = open(filename, 'w', encoding='utf-8')
for i in urls:
    file.write(i)
    file.write('\n')
file.close()
```

从而，我们的整个任务就算完成了。

练习二：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。

【解】

该题与练习一的问题类似，故我们可以利用类似于练习一的问题框架来解决练习二的问题，其中 *1. 加载网页内容* 与 *3. 输出打印结果* 和练习一相同，故在此处不赘述。

2. 爬全部超链接

通过对百度搜索主页的 HTML 源码的查看，可以注意到，在该源码中，超链接是在<img>标签下且这个标签下的 src 值：

因而，我们只需要找到满足给定标签<img>标签下的所有 src 值即可提取出超链接：

```
for i in soup.findAll('img'):
    t = i.get('src','')
    urlset.add(t)
```

进而，我们的任务也就完成了，完整代码提供在 *example2.py* 中。

练习三： 给定知乎日报的 url，返回网页中的图片和相应文本，以及每个图片对应的超链接网址。并将图片地址，相应文本，超链接网址以下述格式打印至 res3.txt 中，每一行对应一个图片地址，相应文本和超链接网址，格式为：图片地址\t 相应文本\t 超链接网址。参考 *example3.py*。

【解】

练习三可以认为是练习一和二的一个提升，在 *1. 加载网页内容* 与 *3. 输出打印结果* 和练习一、二几乎相同，主要的不同体现在第二点的爬取相应内容。

题目中要求的格式为*图片地址\t 相应文本\t 超链接网址*作为一个项目，故这里我通过建立一个空列表集，先将图片地址作为一个列表元素添加到这个列表中，再第二次爬取时，选择相应的文本，逐一对应到相应的图片地址中：

```
#对于收集到的图片地址，利用列表来封装起来，方便后面的项目直接 append
    s = []
    s.append(t)
    zhihulist.append(s)
#其次爬取文本内容，为保证一对一，此处设立了计数器 cccnt 将文本内容 append 到相应的图片网址后
    zhihulist[cccnt].append(t[0])
    cccnt += 1
#最后爬取网页链接，同样 cnt 计数，由于这里得到的链接为/story...故需要与知乎的网站连接成绝对地址
    zhihulist[cnt].append(urllib.parse.urljoin(url, t))
    cnt += 1
```

这里通过对知乎日报的 HTML 源码进行检查，可以发现，我们需要的信息条目实际上是在同一个标签下的具体不同条目（即图片地址与相应文本都归属于<a>标签且 class 为 link-button 的标签下），且每个条目都有具体的 class 属性（如<a>的为 link-button、<img>的为 preview-image、的为 title），方便我们进行直接查找。

因此，在查找时我们可以根据这几个属性来设立相应的查找方式，便可以轻松得到相应标签下我们所需要的内容：

```
for i in soup.findAll('img',{"class":"preview-image"}):
for i in soup.findAll('span',{"class":"title"}):
for i in soup.findAll('a',{"class":"link-button"}):
```

从而我们的练习三可以得到很好的解决。

四、实验问题及解决

问题 1：在实验过程中，我发现爬取出来的链接存在这两种情况：

①存在空链接

②存在无意义链接 *javascript:;*

要怎么做以排除无效链接而得到有效链接呢？

【解】为得到有效的链接，此处需要利用正则表达式对获取的链接进行处理。经过观察，可以发现有效链接的形式应该是形如：'http://…'、'https://…'、'//…'之类的（由于在百度主页的网站上爬取下来的链接没有以'www.…'的形式存在的链接，故本处不进行考虑），从而我们可以确定正则表达式的形式应该为'^http.*$'或'^//.*$'，并修改相应的代码：

```
#t 是具体的链接、文字等
    p1 = re.compile('^http.*$')
    m1 = p1.match(t)
    p2 = re.compile('^//.*$')
    m2 = p2.match(t)
    if(m1 or m2):
        urlset.add(t)
```

通过这样的操作，我们可以得到百度主页中所有合法超链接的URL（不包括图片地址）并存储在 urlset 之中，等待最后的输出结果，而这个方法在练习一、二、三中均是通用的。

问题 2：爬取内容的过程中，我想要将图片所对应的文本爬取下来，但是这样爬取的条目是一个完整的条目，即*给尸体加心脏起搏器和正常供氧供血，它会不腐烂吗？*，多出了前缀和后缀，，怎么进行删除呢？

【解】通过查阅网络相关资料[1]后发现，可以利用 i.string 或者 i.contents 的方式来获取文本内容，前者得到的是一个字符串类型，后者得到的是列表类型，故如果想要直接获取文本信息，可以利用这样的代码来实现：

```
t = i.string
zhihulist[cccnt].append(t)

t = i.contents
zhihulist[cccnt].append(t[0])
```

二者均可以得到满意的结果。

五、实验结果

练习一（节选）：

```
http://ss.bdimg.com/static/superman/img/topnav/baobaozhidao@2x-af409f9dbe.png
http://ss.bdimg.com/static/superman/img/topnav/tupian@2x-482fc011fc.png
http://ss.bdimg.com/static/superman/img/topnav/wenku@2x-f3aba893c1.png
http://ss.bdimg.com/static/superman/img/qrcode/qrcode-hover@2x-f9b106a848.png
//www.baidu.com/img/flexible/logo/pc/result@2.png
```

练习二（节选）：

```
https://zhidao.baidu.com
http://wenku.baidu.com/search?lm=0&od=0&ie=utf-8
//www.baidu.com/cache/setindex/index.html
//help.baidu.com/newadd?prod_id=1&category=4
//home.baidu.com
//www.baidu.com/duty
http://xueshu.baidu.com
http://tieba.baidu.com/f?fr=wwwt
```

练习三（节选）：

```
https://pic1.zhimg.com/v2-cc40738743ef6e02aaacb9504f9764a3.jpg?source=8673f162  给尸体加心脏起搏器和正常供氧供血，它会不腐烂吗?          http://daily.zhihu.com/story/9727766
https://pic2.zhimg.com/v2-419a5bcd4689fa01356d49e5861fa53a.jpg?source=8673f162  就目前的法律层面而言，外卖骑手的安全是如何保障的?        http://daily.zhihu.com/story/9727753
https://picb.zhimg.com/v2-9df187e9180ed180367d42151394e24e.jpg?source=8673f162 秦朝法律真的很严苛吗?          http://daily.zhihu.com/story/9727756
https://pic2.zhimg.com/v2-f04bb86da37d2fa0da51b2af4538e567.jpg?source=8673f162  不可忽略的头号杀手：心脑血管疾病        http://daily.zhihu.com/story/9727749
https://pic3.zhimg.com/v2-b67b1cd0ad0d964499109fc41b362bac.jpg?source=8673f162 把这枚奥运金牌，献给我的亡妻 http://daily.zhihu.com/story/9727768
https://pic2.zhimg.com/v2-59d11d929232029f6b681b475660fdec.jpg?source=8673f162 瞄扯 · 如何正确地吐槽          http://daily.zhihu.com/story/9727746
```

六、拓展思考

我爬取到的 href 链接的形式有：①http://… ②https://… ③//… ④javascript:;

我在***四、实验问题与解决***中通过正则表达式的方式来筛选合法链接。

七、实验体会

这是本课程的第一次实验作业，难度并不会太大，因而我能够较为顺利地完成本次实验的三个练习，并且在实验过程中，我逐渐了解了 HTML，对 BeautifulSoup 这个解析器有了一定的知识基础。此外，这门课能够让我将大一上所学的 python 语言的知识再次复习、巩固，使得我对该部分的知识有了更深层次的了解。最后，对于练习中遇到的问题，我能够及时在网上或其他地方寻求帮助，这对我学习态度的端正有着很好的帮助！

当然，本次实验也存在不足之处，主要是由于刚开始学习这样的一个新方向，导致了我在一些细节上的处理可能还比较不尽人意，希望在接下来的几次实验中我能够越做越好！

[1] Python 爬虫利器二之 Beautiful Soup 的用法：https://cuiqingcai.com/1319.html