Experiment Report 1

Name : 王嘉璐

Class: F1503023

ID: 515030910558

HTML

What is HTML?

HTML is a **markup** language for **describing** web documents (web pages).

- HTML stands for **H**yper **T**ext **M**arkup **L**anguage
- A markup language is a set of **markup tags**
- HTML documents are described by **HTML tags**
- Each HTML tag **describes** different document content

HTML tags

HTML tags are **keywords** (tag names) surrounded by **angle brackets**:

```
<tagname>content goes here...</tagname>
```

- HTML tags normally come in pairs like
  and

- The first tag in a pair is the start tag, the second tag is the end tag
- The end tag is written like the start tag, but with a forward slash inserted before the tag name

Here're some examples for HTML tags.

> - The <!DOCTYPE html> declaration defines this document to be HTML5
> - The text between <html> and </html> describes an HTML document
> - The text between <head> and </head> provides information about the document
> - The text between <title> and </title> provides a title for the document
> - The text between <body> and </body> describes the visible page content
> - The text between <h1> and </h1> describes a heading
> - The text between and describes a paragraph

Web Browser

The purpose of a web browser (Chrome, IE, Firefox, Safari) is to read HTML documents and display them.

The browser does not display the HTML tags, but uses them to determine how to display the document.

For example, when we input a URL the in address bar, the web browser send a request to the web server. The web server process the request and response the HTML document. Then the web browser will parse the HTML document. As a result, we see the final

website.

BeautifulSoup

Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.

Use Python to Fetch HTML document

We can use *urllib2* to fetch HTML document.

```
import urllib2
response = urllib2.urlopen('http://www.baidu.com')
content = response.read()
print content
```

Howerver, I's prefer to use another library called *requests*, which is more simple and flexible.

```
import requests
r = requests.get("http://www.baidu.com")
content = r.text
print content
```

Both pieces of code are the same.

Use BeautifulSoup to Parse Website

1. Making the Soup

```
from BeautifulSoup import BeautifulSoup
soup = BeautifulSoup(open('index.html'))
soup = BeautifulSoup("<html>data</html>")
```

2. Use Tag as members

```
print soup.head
print soup.head.title
print str(soup.head.title).decode('utf8')
print str(soup.head.title.string).decode('utf8')
soup.head.meta['content']
soup.head.meta.get('content','')
```

3. Navigating the Tree

Here's an example:

```
html_doc = """
<html><head><title>The Dormouse's story</title></head>
<body>
<p class="title"><b>The Dormouse's story</b></p>

<p class="story">Once upon a time there were three little sisters; and their names were
<a href="http://example.com/elsie" class="sister" id="link1">Elsie</a>,
<a href="http://example.com/lacie" class="sister" id="link2">Lacie</a> and
<a href="http://example.com/tillie" class="sister" id="link3">Tillie</a>;
and they lived at the bottom of a well.</p>

<p class="story">...</p>
"""

from BeautifulSoup import BeautifulSoup
soup = BeautifulSoup(html_doc, 'html.parser')

p = soup.head
# <head><title>The Dormouse's story</title></head>

soup.title
p = soup.title

p = soup.find_all('a')
# [<a class="sister" href="http://example.com/elsie" id="link1">Elsie</a>,
#  <a class="sister" href="http://example.com/lacie" id="link2">Lacie</a>,
#  <a class="sister" href="http://example.com/tillie" id="link3">Tillie</a>]
```

Parent:

```
p.parent
p.parent.name
head = p.parent
```

Contents:

```
print head.contents
print str(head.contents[0]).decode('utf8') #第一个子节点
print str(head.contents[1]).decode('utf8')  #第二个子节点
```

nextSibling and previousSibling:

```
p.nextSibling.name
p.previousSibling.name
```

4. Regex

We use this regex to match url:

```
import re
p = re.compile('<a.*?href="(.*?)".*?<\/a>')
m = p.match(content)
```

Exercise

Problem 1

> 给定任意网页内容，返回网页中所有链接地址（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。

Here's my code.

```
import requests
from BeautifulSoup import BeutifulSoup

def parseURL(content):
    urlset = set()
    soup = BeautifulSoup(content)
    for a in soup.findAll('a'):
        urlset.add(str(a.get('href', '')))
    if len(urlset) > 0:
        return urlset
    else :
        return -1


def main():
    headers = {
    'User-Agent': 'Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.1.5) Gecko/20091102 Firefox
/3.5.5 (.NET CLR 3.5.30729)',
    'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8',
    'Accept-Language': 'en-us,en;q=0.5',
    'Accept-Encoding': 'gzip,deflate',
    'Accept-Charset': 'ISO-8859-1,utf-8;q=0.7,*;q=0.7'
}
    r = requests.get("http://www.github.com", headers)
    urlset = parseURL(r.text)
    if urlset != -1:
        fp = open("res1.txt",'w')
        for u in urlset:
            fp.write(str(u))
        fp.close()
        print "Parse URLs successfully!"
    else:
        print "No URL fetched!"

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

Problem2

> 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。

Similarly to problem 1, just change one line.

```
import requests
import re
from BeautifulSoup import BeautifulSoup


def parseIMG(content):
    imgset = set()
    soup = BeautifulSoup(content)
    for img in soup.findAll('img'):
        imgset.add(str(img.get('src', '')))
    if len(imgset) > 0:
        return imgset
    else :
        return -1


def main():
    headers = {
    'User-Agent': 'Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.1.5) Gecko/20091102 Firefox
/3.5.5 (.NET CLR 3.5.30729)',
    'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8',
    'Accept-Language': 'en-us,en;q=0.5',
    'Accept-Encoding': 'gzip,deflate',
    'Accept-Charset': 'ISO-8859-1,utf-8;q=0.7,*;q=0.7'
}
    r = requests.get("http://www.github.com", headers)
    imgset = parseIMG(r.text)
    if imgset != -1:
        fp = open("res2.txt",'w')
        for u in imgset:
            fp.write(str(u))
        fp.close()
        print "Parse IMGs successfully!"
    else:
        print "No IMG fetched!"

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

Problem 3

> 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

At first, I observed the source code and inspect the element. So I can use BeautifulSoup to fetch the imgurl and the text easily.

```
import codecs
import requests
import re
from BeautifulSoup import BeautifulSoup

def parseQiushiBaikePic(content):
    docs = {}
    nextPage = ''
    soup = BeautifulSoup(content)
    for div in soup.findAll('div', {'id' : re.compile('qiushi_tag_[\d]+')}):
        qiushi_tag_list = div.get('id','').split('_')
        qiushi_tag = qiushi_tag_list[-1]
        sentence = str(div.contents[3].contents[1].contents[1].contents[0]).decode('utf-8')
        print sentence
        imgurl = div.contents[5].contents[1].contents[1].get('src','')
        docs[qiushi_tag] = {'content': sentence, 'imgurl': imgurl}
    nextPage = soup.find('span', {'class':'next'}).parent.get('href','')
    return docs,  nextPage

def main():
    headers = {
    'User-Agent': 'Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.1.5) Gecko/20091102 Firefox
/3.5.5 (.NET CLR 3.5.30729)',
    'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8',
    'Accept-Language': 'en-us,en;q=0.5',
    'Accept-Encoding': 'gzip,deflate',
    'Accept-Charset': 'ISO-8859-1,utf-8;q=0.7,*;q=0.7'
}
    r = requests.get("http://www.qiushibaike.com/pic/", headers)
    docs, nextPage = parseQiushiBaikePic(r.text)
    fp = codecs.open("res3.txt","w","utf-8")
    for tag in docs.values():
        txt = tag["imgurl"] + u'\t' + tag["content"] + u'\n'
        fp.write(txt)
    fp.write(str(nextPage))
    fp.close()

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

There's something wrong if I use `open("res3.txt","w")` directly, for python use ascii to write files as default. So here I use *codecs* library to support unicode.

> http://pic.qiushibaike.com/system/pictures/11755/117557612/medium/app117557612.jpg 别的地方看的，感觉身体被掏空了
> http://pic.qiushibaike.com/system/pictures/11755/117557417/medium/app117557417.jpg 小东西。挺厉害的吗
> http://pic.qiushibaike.com/system/pictures/11755/117559479/medium/app117559479.jpg 老司机才能看懂
> http://pic.qiushibaike.com/system/pictures/11755/117559352/medium/app117559352.jpg 生命中的人来来往往 没有什么来日方长 你看不惯我 就滚好吗 我 为什么要取悦别人
> http://pic.qiushibaike.com/system/pictures/11755/117558194/medium/app117558194.jpg 看糗事百科三年了。从来没发过。心情不好 发一次 希望大神给我过了吧。跟糗友唠叨同时希望能给我点建议
> http://pic.qiushibaike.com/system/pictures/11755/117558281/medium/app117558281.jpg 这个车牌也不错~
> http://pic.qiushibaike.com/system/pictures/11755/117558941/medium/app117558941.jpg 有没有人吃过这个，蜜蜂沾馒头吃，美味哦，城里人吃不到的 http://pic.qiushibaike.com/system/pictures/11755/117559279/medium/app117559279.jpg 一会要剪个什么发型呢？！好紧张啊！ http://pic.qiushibaike.com/system/pictures/11755/117558464/medium/app117558464.jpg 眼熟吗？反正今天我是第一次见看看有多少人知道[憨笑]

> http://pic.qiushibaike.com/system/pictures/11755/117558689/medium/app117558689.jpg 美女，来张合影呗，好啊...
> http://pic.qiushibaike.com/system/pictures/11755/117559285/medium/app117559285.jpg 过百留扣 不留不是人
> http://pic.qiushibaike.com/system/pictures/11755/117559680/medium/app117559680.jpg 我就是：怪不得。
> http://pic.qiushibaike.com/system/pictures/11755/117559654/medium/app117559654.jpg 贴吧看见的。有愿意当接盘侠的么?
> http://pic.qiushibaike.com/system/pictures/11755/117557405/medium/app117557405.jpg 喜欢吗。小鳄鱼
> http://pic.qiushibaike.com/system/pictures/11755/117558622/medium/app117558622.jpg 如果拉了会怎样，会不会火会不会上新闻能不能做网红！会坐牢吗，知道的告知一下在线急等。
> http://pic.qiushibaike.com/system/pictures/11755/117559520/medium/app117559520.jpg 同样的蜕皮，不一样的说法。
> http://pic.qiushibaike.com/system/pictures/11755/117558562/medium/app117558562.jpg 风吹树枝黄叶飞，
> http://pic.qiushibaike.com/system/pictures/11755/117559663/medium/app117559663.jpg 这个车牌还是可以啊
> http://pic.qiushibaike.com/system/pictures/11755/117558022/medium/app117558022.jpg 不想了，姐姐睡吧，可以了
> http://pic.qiushibaike.com/system/pictures/11755/117559587/medium/app117559587.jpg 第一次发嘿嘿嘿
> /pic/page/2?s=4914184

There's a little problem: the next page's url is a relative location. So I had to fix it in line 18:

```
nextPage = 'http://www.qiushibaike.com' + soup.find('span', {'class':'next'}).parent.get('href','')
```

All the codes of exercises can be seen in the attachment.