Python 网络爬虫与信息提取课程笔记

1. requests 库方法

requests.request(method, url, **kwargs)	构造一个请求，支撑以下各种基础方法
requests.get(url,params=None,**kwargs)	获取 HTML 网页，对应于 HTTP 的 GET
requests.head(url,**kwargs)	获取 HTML 头信息，对应于 HTTP 的 HEAD
requests.post(url, data=None, json=None,**kwargs)	向 HTML 提交 POST 请求，对应 POST
requests.put(url,data=None,**kwargs)	向 HTML 提交 PUT 请求，对应 PUT
requests.patch(url,data=None,**kwargs)	向 HTML 提交局部修改请求，对应 PATCH
request.delete(url,**kwargs)	向 HTML 提交删除请求，对应 DELETE

user →<get, header>→ Internet →<post, put, patch, delete>→ user

URL: http://host[:port][path]

host-Internet 主机域名或 IP 地址；port-端口号；path-请求资源路径

URL 是通过 HTTP 协议存取资源的 Internet 路径。

Robots 协议: https://www.jd.com/robots.txt

2. r = requests.get(url,params=None) # response

r.status_code	返回状态，200 表示连接成功
r.text	url 页面内容，字符串形式
r.content	HTTP 响应内容的二进制形式
r.encoding	header 中猜测的相应内容编码方式
r.apparent_encoding	从内容分析出的编码方式
r.request.headers	返回 headers
r.request.url	

3. requests 库异常

requests.ConnectionError	网络连接错误
requests.HTTPError	HTTP 错误异常
requests.URLRequired	URL 缺失异常
requests.TooManyRedirects	超过最大重定向次数
requests.ConnectTimeout	连接远程服务器超时
requests.Timeout	请求 URL 超时

通用框架：异常处理

```python
import requests
def getHTMLText(url):
    try:
        r = requests.get(url, timeout=30)
        r.raise_for_status()
        r.encoding = r.apparent_encoding
        return r.text
    except:
        return 'Error!'

if __name__ == '__main__':
    url = 'https://www.baidu.com'
    print(getHTMLText(url))
```

4. examples

- 搜索关键词 https://www.baidu.com/s?wd=keyword https://www.google.com/search?q=keyword

```
1.        r = requests.get(url='https://www.baidu.com/s',params={'wd':'python'})
2.    r = requests.get(url='https://www.google.com/search',params={'q':'python'})
```

- 下载图片

```
1.    path = 'C:/Users/zhpy/Desktop/pic.jpg'  # 新建图片
2.    url = 'https://www.thebeaverton.com/wp-content/uploads/2019/03/cat-800x600.jpg'
3.    r = requests.get(url)
4.    with open(path,'wb') as f:
5.        f.write(r.content)   # 二进制格式存储
6.        f.close()
```

5. BeautifulSoup

- bs4 解析库

bs4 的 HTML 解析器	BeautifulSoup(mk,'html.parser')
lxml 的 HTML 解析器	BeautifulSoup(mk,'lxml')
lxml 的 XML 解析器	BeautifulSoup(mk,'xml')
html5lib 解析器	beautifulSoup(mk,'html5lib')

```
1.    import requests
2.    from bs4 import BeautifulSoup
3.
4.    url = 'http://shanghai.gongjiao.com/lines_all.html'
5.    r = requests.get(url)
6.    soup = BeautifulSoup(r.text,'html.parser')
7.    print(soup.prettify)
```

- bs 基本元素

Tag	标签，<>与</>表明开头结尾	
Name	标签名称， ，名称为'p'	<tag>.name
Attributes	标签属性	<tag>.attrs
NavigableString	标签内非属性字符串	<tag>.string
Comment	标签内字符串注释	

- 遍历

下行遍历：	soup.tag.contents/children/decendants # tag=html/head/title/body/p/a
上行遍历：	soup.tag.parent/parents
平行遍历：	soup.tag.next_sibling/previous_sibling/next_siblings/previous-siblings

- 信息标记

XML	eXtensible Markup Language	<tag></tag>
JSON	JavaScript Object Notation	{key:value}
YAML	YAML Ain't markup Language	key:value