昨晚的作业：

|  |
| --- |
| *#导入库* **from** urllib **import** request,parse *#产生随机数* **import** random **import** json ua\_list = [  **'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.1 (KHTML, like Gecko) Chrome/14.0.835.163 Safari/535.1'**,  **'Mozilla/5.0 (Windows NT 6.1; WOW64; rv:6.0) Gecko/20100101 Firefox/6.0'**,  **'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/534.50 (KHTML, like Gecko) Version/5.1 Safari/534.50'**,  **'Opera/9.80 (Windows NT 6.1; U; zh-cn) Presto/2.9.168 Version/11.50'**,  **'Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0; .NET CLR 2.0.50727; SLCC2; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; InfoPath.3; .NET4.0C; Tablet PC 2.0; .NET4.0E)'**,  **'Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64; Trident/5.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; InfoPath.3; .NET4.0C; .NET4.0E; SE 2.X MetaSr 1.0)'**,  **'Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; InfoPath.3; .NET4.0C; .NET4.0E)'**,  **'Mozilla/5.0 (Windows NT 6.1) AppleWebKit/535.1 (KHTML, like Gecko) Chrome/13.0.782.41 Safari/535.1 QQBrowser/6.9.11079.201'**, ] **def** loadPage(url,word):  *#从列表中随机选取一个useragent* UserAgent = random.choice(ua\_list)  *#构造请求头* headers ={  **"User-Agent"**:UserAgent  }  *#构造form\_data* data ={  **"i"**: word,  **"from"**: **"AUTO"**,  **"to"**: **"AUTO"**,  **"smartresult"**: **"dict"**,  **"client"**: **"fanyideskweb"**,  **"salt"**: **"15722503603964"**,  **"sign"**: **"dc51ee2395f1bc847784ebcdab2093ce"**,  **"ts"**: **"1572250360396"**,  **"bv"**: **"d39271655f6b7da338c7d58f74b0fad8"**,  **"doctype"**: **"json"**,  **"version"**: **"2.1"**,  **"keyfrom"**: **"fanyi.web"**,  **"action"**: **"FY\_BY\_REALTlME"** }  *#发送请求* rep = request.Request(url=url,headers=headers)  *#转换data格式* data = parse.urlencode(data).encode(**'utf-8'**)  *#进行相应* resp = request.urlopen(rep,data=data)  *#读取内容* html = resp.read().decode(**'utf-8'**)  **return** html **def** loadres():  url =**'http://fanyi.youdao.com/translate?smartresult=dict&smartresult=rule'  while True**:  *#input* word = input(**"请输入你想要翻译的词："**)  **if** word ==**"exit"**:  **break** *#调用请求的方法* html = loadPage(url,word)  *#转换格式* res\_json = json.loads(html)  print(res\_json[**"translateResult"**][0][0][**"tgt"**])   **if** \_\_name\_\_ == **'\_\_main\_\_'**:  loadres() |

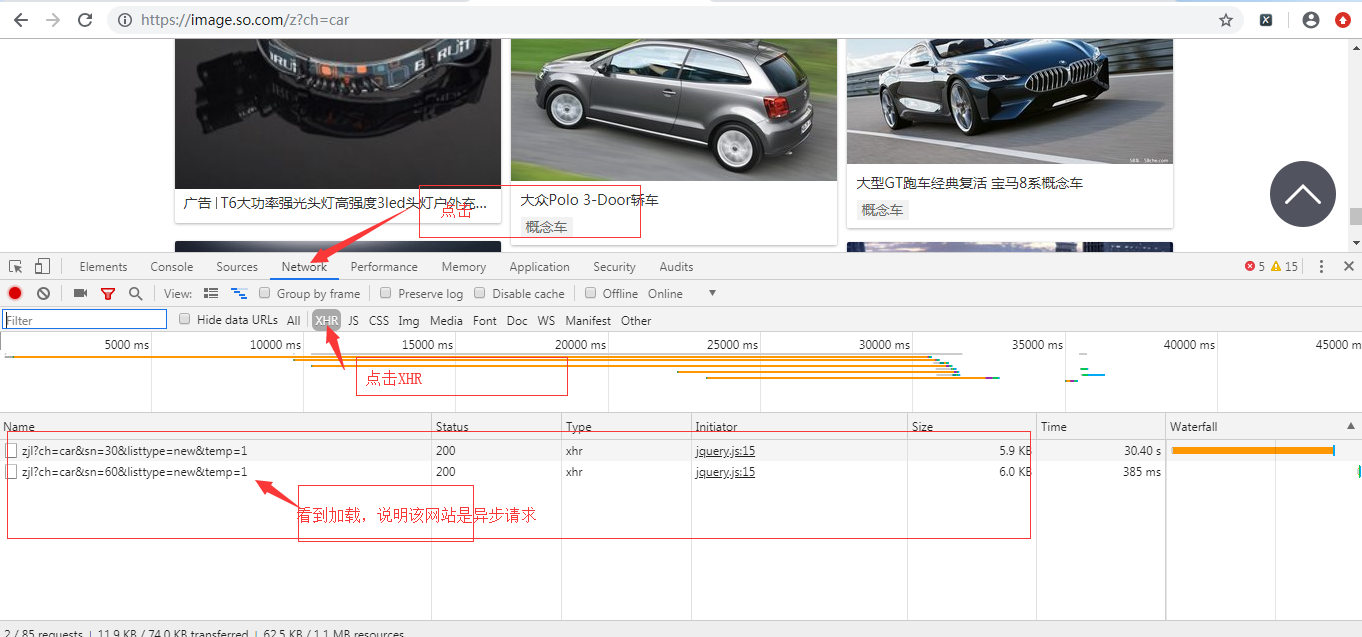
爬取360图片

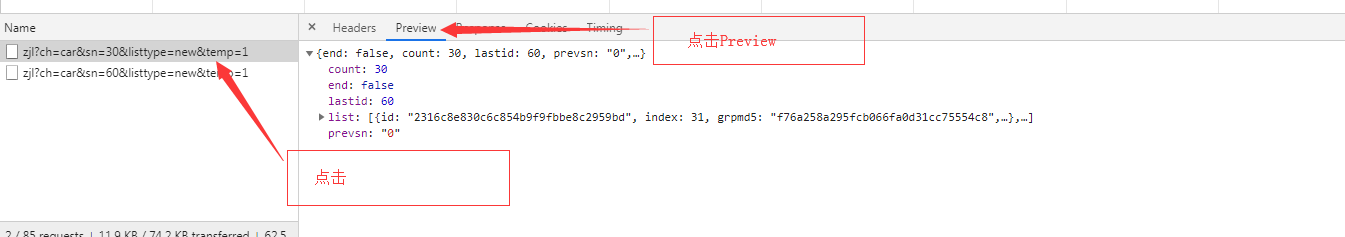
1. 访问<https://image.so.com/z?ch=car>
2. 采用xpath提取对应图片的链接

|  |
| --- |
| *#导入对应包* **from** urllib **import** request **import** random **from** lxml **import** etree ua\_list = [  **'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.1 (KHTML, like Gecko) Chrome/14.0.835.163 Safari/535.1'**,  **'Mozilla/5.0 (Windows NT 6.1; WOW64; rv:6.0) Gecko/20100101 Firefox/6.0'**,  **'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/534.50 (KHTML, like Gecko) Version/5.1 Safari/534.50'**,  **'Opera/9.80 (Windows NT 6.1; U; zh-cn) Presto/2.9.168 Version/11.50'**,  **'Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0; .NET CLR 2.0.50727; SLCC2; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; InfoPath.3; .NET4.0C; Tablet PC 2.0; .NET4.0E)'**,  **'Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64; Trident/5.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; InfoPath.3; .NET4.0C; .NET4.0E; SE 2.X MetaSr 1.0)'**,  **'Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; InfoPath.3; .NET4.0C; .NET4.0E)'**,  **'Mozilla/5.0 (Windows NT 6.1) AppleWebKit/535.1 (KHTML, like Gecko) Chrome/13.0.782.41 Safari/535.1 QQBrowser/6.9.11079.201'**, ] **def** loadPage(url):  *#随机的选取ua* ua = random.choice(ua\_list)  *#构造请求头* headers = {  **"User-Agent"**:ua  }  *#对服务器进行请求* res = request.Request(url=url,headers=headers)  *#进行响应* result = request.urlopen(url=res)  *#读取结果* html = result.read().decode(**'utf-8'**)  *#转换* html\_xpath =etree.HTML(html)  href = html\_xpath.xpath(**'//ul[@class="wf\_position"]//a//img/@src'**)  print(href)  **if** \_\_name\_\_ == **'\_\_main\_\_'**:  url = **'https://image.so.com/z?ch=car'** loadPage(url) |

运行的结果：href获取到为空

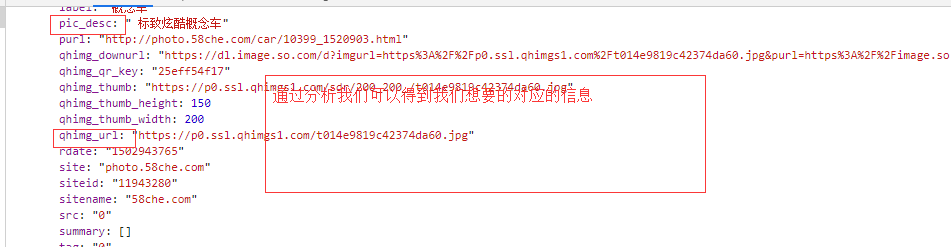
这时我们可以看看该网站是否是通过ajax请求，再一次对网站进行分析。





通过分析我们可以看到对应的信息





分析完之后我们可以编写对应的代码：

|  |
| --- |
| *#导入对应包* **from** urllib **import** request **import** random **import** json **from** lxml **import** etree ua\_list = [  **'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.1 (KHTML, like Gecko) Chrome/14.0.835.163 Safari/535.1'**,  **'Mozilla/5.0 (Windows NT 6.1; WOW64; rv:6.0) Gecko/20100101 Firefox/6.0'**,  **'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/534.50 (KHTML, like Gecko) Version/5.1 Safari/534.50'**,  **'Opera/9.80 (Windows NT 6.1; U; zh-cn) Presto/2.9.168 Version/11.50'**,  **'Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0; .NET CLR 2.0.50727; SLCC2; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; InfoPath.3; .NET4.0C; Tablet PC 2.0; .NET4.0E)'**,  **'Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64; Trident/5.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; InfoPath.3; .NET4.0C; .NET4.0E; SE 2.X MetaSr 1.0)'**,  **'Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; InfoPath.3; .NET4.0C; .NET4.0E)'**,  **'Mozilla/5.0 (Windows NT 6.1) AppleWebKit/535.1 (KHTML, like Gecko) Chrome/13.0.782.41 Safari/535.1 QQBrowser/6.9.11079.201'**, ] **def** loadPage(url):  *#随机的选取ua* ua = random.choice(ua\_list)  *#构造请求头* headers = {  **"User-Agent"**:ua  }  *#对服务器进行请求* res = request.Request(url=url,headers=headers)  *#进行响应* result = request.urlopen(url=res)  *#读取结果* html = result.read().decode(**'utf-8'**)  *# print(type(josn\_html))*  *# 转换成json格式*  json\_html =json.loads(html)  #取list的值 list\_json = json\_html[**'list'**]  #取第一张图片的连接 href = list\_json[0][**'qhimg\_url'**]  #获取名字 name = list\_json[0][**'pic\_desc'**] print(href,name)  **return** html   **if** \_\_name\_\_ == **'\_\_main\_\_'**:   *# https: // image.so.com / zjl?ch = car & sn = 30 & listtype = new & temp = 1* loadPage() |

运行的结果：



取到对应的图片连接之后我们可以进行对图片下载

|  |
| --- |
| **from** urllib **import** request **import** random **import** json **from** lxml **import** etree ua\_list = [  **'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.1 (KHTML, like Gecko) Chrome/14.0.835.163 Safari/535.1'**,  **'Mozilla/5.0 (Windows NT 6.1; WOW64; rv:6.0) Gecko/20100101 Firefox/6.0'**,  **'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/534.50 (KHTML, like Gecko) Version/5.1 Safari/534.50'**,  **'Opera/9.80 (Windows NT 6.1; U; zh-cn) Presto/2.9.168 Version/11.50'**,  **'Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0; .NET CLR 2.0.50727; SLCC2; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; InfoPath.3; .NET4.0C; Tablet PC 2.0; .NET4.0E)'**,  **'Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64; Trident/5.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; InfoPath.3; .NET4.0C; .NET4.0E; SE 2.X MetaSr 1.0)'**,  **'Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; InfoPath.3; .NET4.0C; .NET4.0E)'**,  **'Mozilla/5.0 (Windows NT 6.1) AppleWebKit/535.1 (KHTML, like Gecko) Chrome/13.0.782.41 Safari/535.1 QQBrowser/6.9.11079.201'**, ] **def** loadPage(url):  *#随机的选取ua* ua = random.choice(ua\_list)  *#构造请求头* headers = {  **"User-Agent"**:ua  }  *#对服务器进行请求* res = request.Request(url=url,headers=headers)  *#进行响应* result = request.urlopen(url=res)  *#读取结果* html = result.read().decode(**'utf-8'**)  *# print(type(josn\_html))*  *# 转换成json格式*  json\_html =json.loads(html)  #取list的值 list\_json = json\_html[**'list'**]  #取第一张图片的连接 href = list\_json[0][**'qhimg\_url'**]  #获取名字 name = list\_json[0][**'pic\_desc'**] print(href,name)  #将图片进行保存  request.urlretrieve(url=href, filename=+ **"**name**.jpg"**)  **return** html **if** \_\_name\_\_ == **'\_\_main\_\_'**:   *# https: // image.so.com / zjl?ch = car & sn = 30 & listtype = new & temp = 1* loadPage() |

批量的进行图片保存，在当前工作目录下新建一个images文件夹

|  |
| --- |
| ...... **def** loadImage():  url = **'https://image.so.com/zjl?ch=car&sn=0&listtype=new&temp=1'** *# 提取键为list的内容* html = loadPage(url)  *#转换成json格式* json\_html =json.loads(html)  list\_json =json\_html[**'list'**]  **for** img **in** range(len(list\_json)):  href =list\_json[img][**'qhimg\_url'**]  name = list\_json[img][**'pic\_desc'**]  request.urlretrieve(url=href,filename=**"./images/"** + name + **".jpg"**)  **if** \_\_name\_\_ == **'\_\_main\_\_'**:   *# https: // image.so.com / zjl?ch = car & sn = 30 & listtype = new & temp = 1* loadImage() |

下载完第一页之后，通过以下分析

|  |
| --- |
| https://image.so.com/zjl?ch=car&sn=0&listtype=new&temp=1  https://image.so.com/zjl?ch=car&sn=30&listtype=new&temp=1  https://image.so.com/zjl?ch=car&sn=60&listtype=new&temp=1 |

变得只有sn，所以我们可以构造多条url

|  |
| --- |
| **def** loadImage():  *#爬取信息的url* **for** i **in** range(3):  url = **f'https://image.so.com/zjl?ch=car&sn={**i\*30**}&listtype=new&temp=1'**  *# 提取键为list的内容* html = loadPage(url)  *#转换成json格式* json\_html =json.loads(html)  list\_json =json\_html[**'list'**]  **for** img **in** range(len(list\_json)):  href =list\_json[img][**'qhimg\_url'**]  name = list\_json[img][**'id'**]  request.urlretrieve(url=href,filename=**"./images/"** + name + **".jpg"**) |

爬取豆瓣电影的评分和电影的名字：

Url地址：<https://movie.douban.com/top250>

使用xpath进行对网站信息提取

|  |
| --- |
| *#导入对应的库* **from** urllib **import** request **import** random **from** lxml **import** etree ua\_list = [  **'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.1 (KHTML, like Gecko) Chrome/14.0.835.163 Safari/535.1'**,  **'Mozilla/5.0 (Windows NT 6.1; WOW64; rv:6.0) Gecko/20100101 Firefox/6.0'**,  **'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/534.50 (KHTML, like Gecko) Version/5.1 Safari/534.50'**,  **'Opera/9.80 (Windows NT 6.1; U; zh-cn) Presto/2.9.168 Version/11.50'**,  **'Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0; .NET CLR 2.0.50727; SLCC2; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; InfoPath.3; .NET4.0C; Tablet PC 2.0; .NET4.0E)'**,  **'Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64; Trident/5.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; InfoPath.3; .NET4.0C; .NET4.0E; SE 2.X MetaSr 1.0)'**,  **'Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; InfoPath.3; .NET4.0C; .NET4.0E)'**,  **'Mozilla/5.0 (Windows NT 6.1) AppleWebKit/535.1 (KHTML, like Gecko) Chrome/13.0.782.41 Safari/535.1 QQBrowser/6.9.11079.201'**, ] **def** loadPage(url):  *#随机的选取ua* ua = random.choice(ua\_list)  headers ={  **"User-Agent"**:ua  }  *#发送请求* res = request.Request(url=url,headers=headers)  *#进行响应* content = request.urlopen(res)  html = content.read().decode(**'utf-8'**)  *#转换格式* xpath\_html = etree.HTML(html)  title = xpath\_html.xpath(**"//div[@class='info']//a/span[1]/text()"**)  print(title)  **if** \_\_name\_\_ == **'\_\_main\_\_'**:  url = **'https://movie.douban.com/top250'** loadPage(url=url) |

获取更多的信息

|  |
| --- |
| *......*  *title = mo.xpath(".//a/span[1]/text()") pingfen = mo.xpath(".//div[@class='bd']/div[@class='star']//span[2]/text()")  renshu = mo.xpath(".//div[@class='bd']/div[@class='star']//span[4]/text()") inq = mo.xpath(".//div[@class='bd']/p/span/text()")* |

获取到信息之后我们要对信息进行保存到文件中

|  |
| --- |
| **def** loadPage(url):  *#随机的选取ua* ua = random.choice(ua\_list)  headers ={  **"User-Agent"**:ua  }  *#发送请求* res = request.Request(url=url,headers=headers)  *#进行响应* content = request.urlopen(res)  *#读取内容* html = content.read().decode(**'utf-8'**)  *#转换格式* xpath\_html = etree.HTML(html)  mo\_list = xpath\_html.xpath(**"//div[@class='info']"**)  items = []  **for** mo **in** mo\_list:  *#构造空的字典* item ={   }  item[**'title'**] = mo.xpath(**".//a/span[1]/text()"**)[0]  item[**'pingfen'**] = mo.xpath(**".//div[@class='bd']/div[@class='star']//span[2]/text()"**)[0]  item[**'renshu'**] = mo.xpath(**".//div[@class='bd']/div[@class='star']//span[4]/text()"**)[0]  item[**'inq'**] = mo.xpath(**".//div[@class='bd']/p/span/text()"**)[0]  *#将数据添加到列表中* items.append(item)  print(items)  **return** items  **def** writeInfo():  url = **'https://movie.douban.com/top250'** items = loadPage(url)  **with** open(**"xxx.txt"** ,**'w+'**,encoding=**'utf-8'**) **as** f:  f.write(str(items)) **if** \_\_name\_\_ == **'\_\_main\_\_'**:  writeInfo() |

爬完第一页之后我们可以实现翻页爬取

|  |
| --- |
| **def** writeInfo():  *#第二种写文件的形式* f = open(**"xxx.txt"** ,**'a+'**,encoding=**'utf-8'**)  **for** i **in** range(10):  url = **f'https://movie.douban.com/top250?start={**i \* 25**}&filter='** items = loadPage(url)  *#第一种写文件的方式  # with open("xxx.txt" ,'a+',encoding='utf-8') as f:  # # f.write(str(items))* f.write(items)  f.close() |

作业：1、访问粤嵌的官网<http://www.gec-edu.org/teachers/>

2、提取老师的名字、职称、简介

3、保存到文件中