2021/6/10--正则提取

笔记本:	reptile_draft		
创建时间:	2021/6/10 8:23	**更新时间:**	2021/7/13 18:14
作者:	134exetj717		
URL:	about:blank		

3.3.2 正则提取

◆ 正则表达式

正则表达式，通常被用来检索、替换那些符合某个模式（规则）的文本。正则表达式是对字符串操作的一种逻辑公式，就是用事先定义好的一些特定字符及这些特定字符的组合，组成一个“规则字符串”，这个“规则字符串”用来表达对字符串的一种过滤逻辑。Python中使用re模块操作正则表达式。


```
findLink=re.compile(r'<a href="(.*?)">')#找到影片详情链接
findImgSrc=re.compile(r'<img.*src="(.*?)"',re.S)#找到影片图片
findTitle=re.compile(r'<span class="title">(.*)</span>')#找到片名
#找到评分
findRating=re.compile(r'<span class="rating_num" property="v:average">(.*)</span>')
#找到评价人数
findJudge=re.compile(r'<span>(\d*)人评价</span>')
#找到概况
findInq=re.compile(r'<span class="inq">(.*)</span>')
#找到影片相关内容：导演，主演，年份，地区，类别
findBd=re.compile(r'<p class="">(.*?)</p>',re.S)
```

正则提取影片特定数据内容：

```
import urllib
from bs4 import BeautifulSoup


#影片详情链接的规则
findLink = re.compile(r'<a href="(.*?)">')      #生成正则表达式对象，表示规则（字符串的模式）
#影片图片
findImgSrc = re.compile(r'<img.*src="(.*?)"',re.S)  #re.S    让换行符包含在字符中
#影片片名
findTitle = re.compile(r'<span class="title">(.*)</span>')
#影片评分
findRating = re.compile(r'<span class="rating_num" property="v:average">(.*?)
</span')
#找到评价人数
findJudge = re.compile(r'<span>(\d*)人评价</span>')
#找到概况
findInq = re.compile(r'<span class="igq">(.*)</span>')
#找到影片的相关内容
findBD = re.compile(r'<p class="">(.*?)</p>',re.S)


#爬取网页
def getData(baseurl):
    datalist = []
    for i in range(0,10):      #调用获取页面信息的函数10次
```

```
        url = baseurl + str(i*25)
        html = askURL(url)      #保存获取到的网页源码
        #print(html)
        #2.逐一解析数据
        soup = BeautifulSoup(html,'html.parser')
        for item in soup.find_all('div',class_='item'): #查找符合要求的字符串，形成
列表（括号中可以有多个条件）,class 加上 _ 表示查找对应的属性
            #print(item)     #测试：查看电影item的全部信息
            data = [] #保存一部电影的所有信息
            item = str(item)


            link = re.findall(findLink,item)[0]     #re库用来通过正则表达式查找指定
的字符串
            data.append(link)       #添加链接


            imgSrc = re.findall(findImgSrc,item)[0]
            data.append(imgSrc)         #添加图片


            titles = re.findall(findTitle, item)  # 片名可能只有一个中文名，没有外国
名
            if (len(titles) == 2):
                ctitle = titles[0]
                data.append(ctitle)
                otitle = titles[1].replace('/', '')  # 去掉无关符号
                data.append(otitle)  # 添加外国名
            else:
                data.append(titles[0])
                data.append(' ')  # 外国名字留空


            rating = re.findall(findRating, item)[0]
            data.append(rating)  # 添加评分


            judgeNum = re.findall(findJudge, item)[0]
            data.append(judgeNum)  # 添加评价人数


            inq = re.findall(findInq, item)
            if len(inq) != 0:
                inq = inq[0].replace('。', '')  # 去掉句号
                data.append(inq)  # 添加概述
            else:
                data.append(' ')  # 留空


            bd = re.findall(findBD, item)[0]
            bd = re.sub('<br(\s+)?/>(/s+)?', ' ', bd)  # 去掉<br/>
            bd = re.sub('/', ' ', bd)  # 替换/
            data.append(bd.strip())  # 去掉前后的空格
            datalist.append(data)  # 把处理好的一部电影信息放入datalist
    print(datalist)
    return datalist


#得到指定一个URL的网页内容
def askURL(url):        #模拟浏览器头部信息，向豆瓣服务器发送消息
    head = {
        'User-Agent':   'Mozilla / 5.0(Windows NT  10.0;  Win64;  x64; rv: 89.0)
Gecko / 20100101 Firefox / 89.0'
    }
                        #用户代理，表示告诉豆瓣服务器，我们是什么类型的机器、浏览器（本
质上是告诉浏览器我们可以接受什么水平的内容
    request = urllib.request.Request(url,headers=head)
    try:
        response = urllib.request.urlopen(request)
        html = response.read().decode('utf-8')
```

```
        return html
    except urllib.error.URLError as e:
        if hasattr(e,'code'):
            print(e.code)
        if hasattr(e,'reason'):
            print(e.reason)



if __name__ == '__main__':
    getData('https://movie.douban.com/top250?start=')
```