Day01回顾

请求模块(urllib.request)

```
req = request.Request(url,headers=headers)
res = request.urlopen(req)
html = res.read().decode('utf-8')
```

编码模块(urllib.parse)

```
1、urlencode({dict})
   urlencode({'wd':'美女','pn':'20'})
   编码后 ： 'wd=%E8%D5XXX&pn=20'

2、quote(string)
   quote('织女')
   编码后 ： '%D3%F5XXX'

3、unquote('%D3%F5XXX')
```

解析模块(re)

使用流程

```
p = re.compile('正则表达式',re.S)
r_list = p.findall(html)
```

贪婪匹配和非贪婪匹配

```
贪婪匹配(默认) ：  .*
非贪婪匹配     ：  .*?
```

正则表达式分组

```
1、想要什么内容在正则表达式中加()
2、多个分组,先按整体正则匹配,然后再提取()中数据。结果：[(),(),(),(),()]
```

抓取步骤

```
1、确定所抓取数据在响应中是否存在（右键 - 查看网页源码 - 搜索关键字）
2、数据存在：查看URL地址规律
3、写正则表达式,来匹配数据
4、程序结构
    1、使用随机User-Agent
    2、每爬取1个页面后随机休眠一段时间
```

```python
# 程序结构
class xxxSpider(object):
    def __init__(self):
        # 定义常用变量,url,headers及计数等

    def get_html(self):
        # 获取响应内容函数,使用随机User-Agent

    def parse_html(self):
        # 使用正则表达式来解析页面，提取数据

    def write_html(self):
        # 将提取的数据按要求保存，csv、MySQL数据库等

    def main(self):
        # 主函数，用来控制整体逻辑

if __name__ == '__main__':
    # 程序开始运行时间戳
    start = time.time()
    spider = xxxSpider()
    spider.main()
    # 程序运行结束时间戳
    end = time.time()
    print('执行时间:%.2f' % (end-start))
```

spider-day02笔记

作业讲解

正则分组练习

```python
import re
```

```
2
3  html = '''<div class="animal">
4      <p class="name">
5          <a title="Tiger"></a>
6      </p>
7
8      <p class="content">
9          Two tigers two tigers run fast
10     </p>
11 </div>
12
13 <div class="animal">
14     <p class="name">
15         <a title="Rabbit"></a>
16     </p>
17
18     <p class="content">
19         Small white rabbit white and white
20     </p>
21 </div>'''
22
23 # 问题1
24 p = re.compile('<div class="animal">.*?title="(.*?)".*?content">(.*?)</p>.*?</div>',re.S)
25 r_list = p.findall(html)
26 print(r_list)
27
28 # 问题2
29 for rt in r_list:
30     print('动物名称:',rt[0].strip())
31     print('动物描述:',rt[1].strip())
32     print('*' * 50)
```

猫眼电影top100抓取案例

确定URL网址

```
1  猫眼电影 - 榜单 - top100榜
```

目标

```
1  电影名称、主演、上映时间
```

操作步骤

- **1、确定响应内容中是否存在所需数据**

```
1  右键 - 查看网页源代码 - 搜索关键字 - 存在！！
```

- **2、找URL规律**

```
第1页：https://maoyan.com/board/4?offset=0
第2页：https://maoyan.com/board/4?offset=10
第n页：offset=(n-1)*10
```

- **3、正则表达式**

```
<div class="movie-item-info">.*?title="(.*?)".*?class="star">(.*?)</p>.*?releasetime">(.*?)</p>
```

- **4、编写程序框架，完善程序**

```
from urllib import request
import time
import re
import random

class MaoyanSpider(object):
    def __init__(self):
        self.url = 'https://maoyan.com/board/4?offset={}'
        self.ua_list = [
            'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/72.0.3626.119 Safari/537.36',
            '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'
        ]
        # 爬取页数计数
        self.page = 1

    # 获取页面
    def get_page(self,url):
        # 访问不同页面使用随机的User-Agent
        headers = {'User-Agent':random.choice(self.ua_list)}
        req = request.Request(url,headers=headers)
        res = request.urlopen(req)
        html = res.read().decode('utf-8')
        # 直接调用解析函数
        self.parse_page(html)

    # 解析页面
    def parse_page(self,html):
        # 正则解析
        p = re.compile('<div class="movie-item-info">.*?title="(.*?)".*?class="star">(.*?)</p>.*?releasetime">(.*?)</p>',re.S)
        r_list = p.findall(html)
        # r_list : [('霸王别姬','张国荣','1993'),(),()]
        self.write_page(r_list)

    # 保存数据(从终端输出)
    def write_page(self,r_list):
        # r_list : [(),(),()]
        film_dict = {}
        for rt in r_list:
            film_dict['name'] = rt[0].strip()
```

```
            film_dict['star'] = rt[1].strip()
            film_dict['time'] = rt[2].strip()

            print(film_dict)


    # 主函数
    def main(self):
        # 用range函数可获取某些查询参数的值
        for offset in range(0,41,10):
            url = self.url.format(offset)
            self.get_page(url)
            print('第{}页爬取成功'.format(self.page))
            self.page += 1
            # 每爬1页随机休眠，控制爬取速率
            time.sleep(random.randint(0,2))

if __name__ == '__main__':
    spider = MaoyanSpider()
    spider.main()
```

数据持久化存储

数据持久化存储 - csv文件

作用

```
将爬取的数据存放到本地的csv文件中
```

使用流程

```
1、导入模块
2、打开csv文件
3、初始化写入对象
4、写入数据(参数为列表)
import csv

with open('film.csv','w') as f:
    writer = csv.writer(f)
    writer.writerow([])
```

示例代码

创建 test.csv 文件，在文件中写入数据

```python
# 单行写入 (writerow([]))
import csv
with open('test.csv','w') as f:
    writer = csv.writer(f)
    writer.writerow(['步惊云','36'])
    writer.writerow(['超哥哥','25'])

# 多行写入(writerows([(),(),()]
import csv
with open('test.csv','w') as f:
    writer = csv.writer(f)
    writer.writerows([('聂风','36'),('秦霜','25'),('孔慈','30')])
```

练习

猫眼电影数据存入本地 maoyanfilm.csv 文件 - 使用writerow方法实现

```python
# 更改write_page函数
def write_page(self,r_list):
    # r_list : [(),(),()]
    with open('maoyanfilm.csv','a') as f:
        writer = csv.writer(f)
        for rt in r_list:
            one_film_list = [
                rt[0].strip(),
                rt[1].strip(),
                rt[2].strip()
            ]
            writer.writerow(one_film_list)
```

思考：使用 writerows()方法实现?

```python
def write_page(self, r_list):
    film_list = []
    # r_list : [(),(),()]
    with open('maoyanfilm.csv', 'a',newline='') as f:
        writer = csv.writer(f)
        for rt in r_list:
            one_film = (rt[0].strip(), rt[1].strip(), rt[2].strip())
            film_list.append(one_film)
            writer.writerows(film_list)
```

数据持久化存储 - MySQL数据库

- **1、在数据库中建库建表**

```
# 连接到mysql数据库
mysql -h127.0.0.1 -uroot -p123456
# 建库建表
create database maoyandb charset utf8;
use maoyandb;
create table filmtab(
name varchar(100),
star varchar(300),
time varchar(50)
)charset=utf8;
```

- **2、回顾pymysql基本使用**

```
import pymysql

# 创建2个对象
db = pymysql.connect('localhost','root','123456','maoyandb',charset='utf8')
cursor = db.cursor()

# 执行SQL命令并提交到数据库执行
# execute()方法第二个参数为列表传参补位
cursor.execute('insert into filmtab values(%s,%s,%s)',['霸王别姬','张国荣','1993'])
db.commit()

# 关闭
cursor.close()
db.close()
```

来试试高效的executemany()方法?

```
import pymysql

# 创建2个对象
db = pymysql.connect('192.168.153.137','tiger','123456','maoyandb',charset='utf8')
cursor = db.cursor()

# 抓取的数据
film_list = [('月光宝盒','周星驰','1994'),('大圣娶亲','周星驰','1994')]

# 执行SQL命令并提交到数据库执行
# execute()方法第二个参数为列表传参补位
cursor.executemany('insert into filmtab values(%s,%s,%s)',film_list)
db.commit()

# 关闭
cursor.close()
db.close()
```

- **3、将电影信息存入MySQL数据库（尽量使用executemany方法）**

```
from urllib import request
import time
import re
import csv
```

```python
import random
import pymysql

class MaoyanSpider(object):
    def __init__(self):
        self.url = 'https://maoyan.com/board/4?offset={}'
        self.ua_list = [
            'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/72.0.3626.119 Safari/537.36',
            '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'
        ]
        # 爬取页数计数
        self.page = 1

        # 创建2个对象
        self.db = pymysql.connect(
            '192.168.153.130','tiger','123456','maoyandb',
            charset='utf8'
        )
        self.cursor = self.db.cursor()

    # 获取页面
    def get_page(self,url):
        # 访问不同页面使用随机的User-Agent
        headers = {'User-Agent':random.choice(self.ua_list)}
        req = request.Request(url,headers=headers)
        res = request.urlopen(req)
        html = res.read().decode('utf-8')
        # 直接调用解析函数
        self.parse_page(html)

    # 解析页面
    def parse_page(self,html):
        # 正则解析
        p = re.compile('<div class="movie-item-info">.*?title="(.*?)".*?class="star">(.*?)</p>.*?releasetime">(.*?)</p>',re.S)
        r_list = p.findall(html)
        # r_list : [('霸王别姬','张国荣','1993'),(),()]
        self.write_page(r_list)

    # 保存数据(存到mysql数据库)
    def write_page(self,r_list):
        # 存放1页电影数据的空列表
        one_page_list = []
        ins = 'insert into filmtab(name,star,time) values(%s,%s,%s)'
        for rt in r_list:
            one_film_list = [
                rt[0].strip(),
                rt[1].strip(),
                rt[2].strip()[5:15]
             ]
            one_page_list.append(one_film_list)

        self.cursor.executemany(ins,one_page_list)
        # 提交到数据库执行
```

```
        self.db.commit()

    # 主函数
    def main(self):
        # 用range函数可获取某些查询参数的值
        for offset in range(0,41,10):
            url = self.url.format(offset)
            self.get_page(url)
            print('第{}页爬取成功'.format(self.page))
            self.page += 1
            # 每爬1页随机休眠，控制爬取速率
            time.sleep(random.randint(0,2))

if __name__ == '__main__':
    start = time.time()
    spider = MaoyanSpider()
    spider.main()
    end = time.time()
    print('执行时间:%.2f' % (end-start))
```

- **4、做个SQL查询**

```
1、查询20年以前的电影的名字和上映时间
   select name,time from film where time<=(now()-interval 20 year);
2、查询1990-2000年的电影名字和上映时间
   select name,time from film where time>='1990-01-01' and time<='2000-12-31';
```

电影天堂案例 - 二级页面抓取

领取任务

```
# 地址
电影天堂 - 2019年新片精品 - 更多
# 目标
电影名称、下载链接

# 分析
*********一级页面需抓取**********
        1、电影名称
        2、电影链接

*********二级页面需抓取**********
        1、下载链接
```

实现步骤

- **1、确定响应内容中是否存在所需抓取数据**
- **2、找URL规律**

```
第1页 : https://www.dytt8.net/html/gndy/dyzz/list_23_1.html
第2页 : https://www.dytt8.net/html/gndy/dyzz/list_23_2.html
第n页 : https://www.dytt8.net/html/gndy/dyzz/list_23_n.html
```

- **3、写正则表达式**

```
1、一级页面正则表达式
   <table width="100%".*?<td width="5%".*?<a href="(.*?)".*?ulink">(.*?)</a>.*?</table>
2、二级页面正则表达式
   <td style="WORD-WRAP.*?>.*?>(.*?)</a>
```

4、代码实现

```python
from urllib import request
import re


class FilmSpider(object):
    def __init__(self):
        self.url = 'https://www.dytt8.net/html/gndy/dyzz/list_23_{}.html'
        self.headers = {'User-Agent': 'Mozilla/5.0'}

    # 获取页面
    def get_page(self, url):
        req = request.Request(url, headers=self.headers)
        res = request.urlopen(req)
        html = res.read().decode('gb18030', 'ignore')
        return html

    # 解析一级页面
    def parse_one_page(self, html):
        p = re.compile('<table width="100%".*?<td width="5%".*?<a href="(.*?)".*?ulink">(.*?)</a>.*?</table>', re.S)
        film_list = p.findall(html)
        # [('/html/gndy/dyzz/20190523/58629.html', '2019年爱情喜剧《最佳男友进化论》HD国语中字'),]
        for film_info in film_list:
            film_name = film_info[1]
            # 拼接详情页链接
            film_link = 'https://www.dytt8.net{}'.format(film_info[0].strip())
            # 获取二级页面的函数
            down_link = self.get_download_link(film_link)
            film = {
                '电影名称': film_name,
                '下载链接': down_link[0].strip()
            }
            print(film)

    # 获取二级页面的数据
    def get_download_link(self, film_link):
        html = self.get_page(film_link)
        p = re.compile('<td style="WORD-WRAP.*?>.*?>(.*?)</a>', re.S)
        download_link_list = p.findall(html)
        return download_link_list
```

```
40
41      def main(self):
42          for i in range(1,11):
43              url = self.url.format(i)
44              html = self.get_page(url)
45              self.parse_one_page(html)
46
47  if __name__ == '__main__':
48      spider = FilmSpider()
49      spider.main()
```

- **5、练习**

  把电影天堂数据存入MySQL数据库

```
1
```

requests模块

安装

- **Linux**

```
1  sudo pip3 install requests
```

- **Windows**

```
1  # 方法一
2     进入cmd命令行 : python -m pip install requests
3  # 方法二
4     右键管理员进入cmd命令行 : pip install requests
```

常用方法

requests.get()

- **作用**

```
1  # 向网站发起请求,并获取响应对象
2  res = requests.get(url,headers=headers)
```

- **参数**

```
1、url ： 需要抓取的URL地址
2、headers ： 请求头
3、timeout ： 超时时间，超过时间会抛出异常
```

- **响应对象(res)属性**

```
1、encoding ： 响应字符编码
   res.encoding = 'utf-8'
2、text ： 字符串
3、content ： 字节流
4、status_code ： HTTP响应码
5、url ： 实际数据的URL地址
```

- **非结构化数据保存**

```
with open('xxx.jpg','wb') as f:
    f.write(res.content)
```

示例

保存赵丽颖图片到本地

```
import requests

url='http://hbimg.b0.upaiyun.com/ac0a5f64360b9c55a6ea4ba395203543d48a8e401bcf7-6q2JJL_fw658'
headers = {'User-Agent':'Mozilla/5.0'}

# 获取响应内容bytes
html = requests.get(url,headers=headers).content
# 写文件
with open('颖宝.jpg','wb') as f:
    f.write(html)
```

练习

```
1、将猫眼电影案例改写为 requests 模块实现
2、将电影天堂案例改写为 requests 模块实现
```

Chrome浏览器安装插件

- 安装方法

```
1、把下载的相关插件（对应操作系统浏览器）后缀改为 .zip
2、解压,打开Chrome浏览器 -> 右上角设置 -> 更多工具 -> 扩展程序 -> 点开开发者模式
3、把相关插件文件夹 拖拽 到浏览器中，释放鼠标即可安装
```

- 需要安装插件

```
1、Xpath Helper：轻松获取HTML元素的xPath路径
2、Proxy SwitchyOmega：Chrome浏览器中的代理管理扩展程序
3、JsonView：格式化输出json格式数据
```

xpath解析

- **定义**

```
XPath即为XML路径语言，它是一种用来确定XML文档中某部分位置的语言，同样适用于HTML文档的检索
```

- **示例HTML代码**

```
<ul class="book_list">
    <li>
        <title class="book_001">Harry Potter</title>
        <author>J K. Rowling</author>
        <year>2005</year>
        <price>69.99</price>
    </li>

    <li>
        <title class="book_002">Spider</title>
        <author>Forever</author>
        <year>2019</year>
        <price>49.99</price>
    </li>
</ul>
```

- **匹配演示**

```
1、查找所有的li节点
   //li
2、查找li节点下的title子节点中,class属性值为'book_001'的节点
   //li/title[@class="book_001"]
3、查找li节点下所有title节点的,class属性的值
   //li//title/@class

# 只要涉及到条件,加 []
# 只要获取属性值,加 @
```

- **选取节点**

```
1、// ：从所有节点中查找（包括子节点和后代节点）
2、@  ：获取属性值
   # 使用场景1（属性值作为条件）
     //div[@class="movie"]
   # 使用场景2（直接获取属性值）
     //div/a/@src
```

- **匹配多路径（或）**

```
xpath表达式1 | xpath表达式2 | xpath表达式3
```

- **常用函数**

```
1、contains() : 匹配属性值中包含某些字符串节点
   # 查找class属性值中包含"book_"的title节点
     //title[contains(@class,"book_")]
2、text() : 获取节点的文本内容
   # 查找所有书籍的名称
     //ul[@class="book_list"]/li/title/text()
```

lxml解析库

- **安装**

```
sudo pip3 install lxml
```

- **使用流程**

```
1、导模块
   from lxml import etree
2、创建解析对象
   parse_html = etree.HTML(html)
3、解析对象调用xpath
   r_list = parse_html.xpath('xpath表达式')
```

今日作业

```
1、把之前所有代码改为 requests 模块
2、抓取链家二手房房源信息（房源名称、总价）,把结果存入到MySQL数据库
3、把电影天堂用xpath实现
```