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：[(),(),(),(),()]
```

---

spider-day02笔记

csv模块

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)
```

示例代码

创建 test.csv 文件，在文件中写入2条数据(01_csv_example.py)

```
# 单行写入 (writerow([]))
import csv
with open('test.csv','w') as f:
    writer = csv.writer(f)
    writer.writerow(['大旭','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([('大旭','36'),('超哥哥','25'),('小泽','30')])
```

猫眼电影top100抓取案例

确定URL网址

猫眼电影 - 榜单 - top100榜 **目标**

电影名称、主演、上映时间 **操作步骤**

- 1. 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
```

- 2. 正则正则表达式

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

- 3. 编写程序框架，完善程序(02_maoyan_film.py)

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

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

    # 获取页面
    def get_page(self,url):
        req = request.Request(url,headers=self.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 : [(),(),()]
        for rt in r_list:
            film = [
                rt[0].strip(),
                rt[1].strip(),
                rt[2].strip()
            ]
            print(film)


    # 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.baseurl + str(offset)
            self.get_page(url)
            print('第%d页爬取成功' % self.page)
```

```
            self.page += 1
            time.sleep(1)

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

练习

猫眼电影数据存入本地 maoyanfilm.csv 文件

```
# 保存数据(存到csv文件)
def write_page(self,r_list):
    # r_list : [(),(),()]
    with open('猫眼.csv','a') as f:
        writer = csv.writer(f)
        for rt in r_list:
            film = [
                rt[0].strip(),
                rt[1].strip(),
                rt[2].strip()
            ]
            writer.writerow(film)
```

思考：使用 writerows()方法实现?

```
# 保存数据(存到csv文件)
def write_page(self,r_list):
    film_list = []
    # r_list : [(),(),()]
    with open('maoyanfilm.csv','a') as f:
        writer = csv.writer(f)
        for rt in r_list:
            film = ( rt[0].strip(),rt[1].strip(),rt[2].strip() )
            film_list.append(film)
        writer.writerows(film_list)
```

数据持久化存储(mongodb)

- 1. **MongoDB数据库**

让我们来回顾一下pymongo模块的使用

```
conn = pymongo.MongoClient('IP',27017)
db = conn['库名']
myset = db['集合名']
myset.insert_one({})
```

示例代码（03_pymongo.py）

```
import pymongo
```

```
2
3  db_name = 'maoyandb'
4  set_name = 'film'
5
6  # 1. 连接对象
7  conn = pymongo.MongoClient('localhost',27017)
8  # 2. 库对象
9  db = conn[db_name]
10 # 3. 集合对象
11 myset = db[set_name]
12
13 # 4. 执行插入语句
14 myset.insert_one({'name':'Tiechui'})
15 # 5、insert_many()的用法?
```

MongoDB命令行操作

```
1  show dbs
2  use 库名
3  show collections
4  db.集合名.find().pretty()
5  db.集合名.count()
6  db.dropDatabase()
```

练习：把猫眼电影案例中电影信息存入mongodb数据库中（04_maoyan_mongo.py)

- 2. **MySQL数据库**

让我们来回顾一下pymysql模块的基本使用（05_pymysql.py）

```
1  import pymysql
2
3  db = pymysql.connect('localhost','root','123456','db1',charset='utf8')
4  cursor = db.cursor()
5  # execute()方法第二个参数为列表传参补位
6  cursor.execute('insert into film values(%s,%s)',['霸王别姬','1993'])
7  # 提交到数据库执行
8  db.commit()
9  # 关闭
10 cursor.close()
11 db.close()
```

让我们来回顾一下pymysql中executemany()的用法(06_pymysql_executemany.py)

```
1  import pymysql
2
3  # 数据库连接对象
4  db = pymysql.connect(
5      'localhost','root','123456',charset='utf8'
6  )
7  # 游标对象
8  cursor = db.cursor()
9  # 存放所有数据的大列表
10 ins_list = []
```

```python
for i in range(2):
    name = input('请输入第%d个学生姓名:' % (i+1))
    age = input('请输入第%d个学生年龄:' % (i+1))
    ins_list.append([name,age])
# 定义插入语句
ins = 'insert into t3 values(%s,%s)'
# 一次数据库的IO操作可插入多条语句，提升性能
cursor.executemany(ins,ins_list)
# 提交到数据库执行
db.commit()
cursor.close()
db.close()
```

练习：把猫眼电影案例中电影信息存入MySQL数据库中（尽量使用executemany方法）(07_maoyan_mysql.py)

```python
from urllib import request
import time
import re
import pymysql

class MaoyanSpider(object):
    def __init__(self):
        self.baseurl = 'https://maoyan.com/board/4?offset='
        self.headers = {
            'User-Agent':'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/72.0.3626.119 Safari/537.36'
        }
        # 爬取页数计数
        self.page = 1
        self.data_list = []
        # 创建2个对象
        self.db = pymysql.connect(
            'localhost','root','123456','spider',
            charset='utf8'
        )
        self.cursor = self.db.cursor()


    # 获取页面
    def get_page(self,url):
        req = request.Request(url,headers=self.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):
        ins = 'insert into film(name,star,time) values(%s,%s,%s)'
        for rt in r_list:
            film_list = [
                rt[0].strip(),
                rt[1].strip(),
                rt[2].strip()[5:15]
             ]
            self.data_list.append(film_list)

        self.cursor.executemany(ins,self.data_list)
        # 提交到数据库执行
        self.db.commit()
        # 爬取1页后清除列表
        self.data_list.clear()

    # 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.baseurl + str(offset)
            self.get_page(url)
            print('第%d页爬取成功' % self.page)
            self.page += 1
            time.sleep(1)
        # 等所有页面爬完后再关闭
        self.cursor.close()
        self.db.close()

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

让我们来做个SQL命令查询

```
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）

- 确定URL地址

```
百度搜索 ： 电影天堂 - 2019年新片 - 更多
```

- 目标

```
*********一级页面***********
        1、电影名称
        2、电影链接

*********二级页面***********
        1、下载链接
```

- 步骤

1. 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
```

2. 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>
```

3. 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_1.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]
            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

if __name__ == '__main__':
    spider = FilmSpider()
    html = spider.get_page(spider.url)
    spider.parse_one_page(html)
```

练习 让我们来把电影天堂数据存入MongoDB数据库

```

```

让我们来把电影天堂数据存入MySQL数据库

```

```

requests模块

安装

- Linux

```
sudo pip3 install requests
```

- Windows

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

常用方法

requests.get()

- 作用

```
# 向网站发起请求,并获取响应对象
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)
```

Chrome浏览器安装插件

- 安装方法

```
# 方法1
1、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
2、把相关插件 拖拽 到浏览器中，释放鼠标即可安装

# 方法2
1、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
2、把下载的插件  插件名.crx 重命名，后缀改为  .rar，并解压
3、在浏览器中点击 ： 加载已解压的扩展程序 -> 选中解压后的插件文件夹
4、重启浏览器
```

- 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>
            <title class="book_001">J K. Rowling</title>
        </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）,把结果成存入到MySQL Mongo
3、把电影天堂用xpath实现