

# Day01回顾

## 请求模块(urllib.request)

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

## 编码模块(urllib.parse)

```
1 1、urlencode({dict})
2   urlencode({'wd':'美女','pn':'20'})
3   编码后 : 'wd=%E8%D5XXX&pn=20'
4
5 2、quote(string)
6   quote('织女')
7   编码后 : '%D3%F5XXX'
8
9 3、unquote('%D3%F5XXX')
```

## 解析模块(re)

### 使用流程

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

### 贪婪匹配和非贪婪匹配

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

### 正则表达式分组

```
1 1、想要什么内容在正则表达式中加()
2 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)

```
1 # 单行写入 (writerow([]))
2 import csv
3 with open('test.csv', 'w') as f:
4     writer = csv.writer(f)
5     writer.writerow(['步惊云', '36'])
6     writer.writerow(['超哥哥', '25'])
7
8 # 多行写入(writerows([( ), ( ), ( )])
9 import csv
10 with open('test.csv', 'w') as f:
11     writer = csv.writer(f)
12     writer.writerows([('聂风', '36'), ('秦霜', '25'), ('孔慈', '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. 查看是否为动态加载

1 右键 - 查看网页源代码 - 搜索爬取关键字 (查看在源代码中是否存在)

- 2. 找URL规律

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

### 3. 正则表达式

```
1 <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)

```
1 |
```

## 练习

猫眼电影数据存入本地 maoyanfilm.csv 文件

```
1 |
```

思考: 使用 `writerows()`方法实现?

```
1 |
```

## 数据持久化存储 (MySQL数据库)

让我们回顾一下pymysql模块的基本使用

```
1 import pymysql
2
3 db = pymysql.connect('localhost','root','123456','maoyandb',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,%s)', ['霸王别姬', '张国荣', '1993'])
7 # 提交到数据库执行
8 db.commit()
9 # 关闭
10 cursor.close()
11 db.close()
```

让我们回顾一下pymysql中`executemany()`的用法

```
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 = []
11 for i in range(2):
12     name = input('请输入第%d个学生姓名:' % (i+1))
13     age = input('请输入第%d个学生年龄:' % (i+1))
14     ins_list.append([name,age])
15 # 定义插入语句
16 ins = 'insert into t3 values(%s,%s)'
17 # 一次数据库的IO操作可插入多条语句, 提升性能
18 cursor.executemany(ins,ins_list)
19 # 提交到数据库执行
20 db.commit()
21 cursor.close()
22 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)

```

1

```

让我们来做个SQL命令查询

```

1 1、查询20年以前的电影的名字和上映时间
2   select name,time from film where time<=(now()-interval 20 year);
3 2、查询1990-2000年的电影名字和上映时间
4   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

```

1 百度搜索：电影天堂 - 2019年新片 - 更多

```

### ■ 目标

```

1 *****一级页面*****
2     1、电影名称
3     2、电影链接
4
5 *****二级页面*****
6     1、下载链接

```

### ■ 步骤

#### 1. 找URL规律

```

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

```

#### 2. 写正则表达式

```
1 1、一级页面正则表达式
2
3 2、二级页面正则表达式
4
```

### 3. 代码实现

```
1 |
```

### 练习

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

#### ■ 响应对象(res)属性

```
1 1、encoding : 响应字符编码
2   res.encoding = 'utf-8'
3 2、text : 字符串
4 3、content : 字节流
5 4、status_code : HTTP响应码
6 5、url : 实际数据的URL地址
```

#### ■ 非结构化数据保存

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

#### 示例

保存赵丽颖图片到本地

```
1 |
```

#### 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 1、把下载的相关插件（对应操作系统浏览器）后缀改为 .zip
2 2、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 3、把相关插件 拖拽 到浏览器中，释放鼠标即可安装
```

#### ■ 需要安装插件

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

## xpath解析

## ■ 定义

```
1 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

```
1 <ul class="book_list">
2   <li>
3     <title class="book_001">Harry Potter</title>
4     <author>J K. Rowling</author>
5     <year>2005</year>
6     <price>69.99</price>
7   </li>
8
9   <li>
10    <title class="book_002">Spider</title>
11    <author>Forever</author>
12    <year>2019</year>
13    <price>49.99</price>
14  </li>
15 </ul>
```

## ■ 匹配演示

```
1 1、查找所有的li节点
2 //li
3 2、查找li节点下的title子节点中,class属性值为'book_001'的节点
4 //li/title[@class="book_001"]
5 3、查找li节点下所有title节点的,class属性的值
6 //li//title/@class
7
8 # 只要涉及到条件,加 []
9 # 只要获取属性值,加 @
```

## ■ 选取节点

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

## ■ 匹配多路径（或）

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

## ■ 常用函数

```
1 1、contains()：匹配属性值中包含某些字符串节点
2   # 查找class属性值中包含"book_"的title节点
3   //title[contains(@class,"book_")]
4 2、text()：获取节点的文本内容
5   # 查找所有书籍的名称
6   //ul[@class="book_list"]/li/title/text()
```

## lxml解析库

### ■ 安装

```
1 sudo pip3 install lxml
```

### ■ 使用流程

```
1 1、导模块
2   from lxml import etree
3 2、创建解析对象
4   parse_html = etree.HTML(html)
5 3、解析对象调用xpath
6   r_list = parse_html.xpath('xpath表达式')
```

## 今日作业

```
1 1、把之前所有代码改为 requests 模块
2 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 3、把电影天堂用xpath实现
```