

# 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: [(,),(),(),(),()]
```

## 抓取步骤

- 1 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 2、数据存在：查看URL地址规律
- 3 3、写正则表达式，来匹配数据
- 4 4、程序结构
- 5     1、使用随机User-Agent
- 6     2、每爬取1个页面后随机休眠一段时间

```
1 # 程序结构
2 class xxxSpider(object):
3     def __init__(self):
4         # 定义常用变量,url,headers及计数等
5
6     def get_html(self):
7         # 获取响应内容函数,使用随机User-Agent
8
9     def parse_html(self):
10        # 使用正则表达式来解析页面,提取数据
11
12    def write_html(self):
13        # 将提取的数据按要求保存, csv、MySQL数据库等
14
15    def main(self):
16        # 主函数,用来控制整体逻辑
17
18 if __name__ == '__main__':
19     # 程序开始运行时间戳
20     start = time.time()
21     spider = xxxSpider()
22     spider.main()
23     # 程序运行结束时间戳
24     end = time.time()
25     print('执行时间:%.2f' % (end-start))
```

# spider-day02笔记

## 作业讲解

### 正则分组练习

1

## 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

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>`

#### ■ 4、编写程序框架，完善程序

1

## 数据持久化存储

### 数据持久化存储 - csv文件

### 作用

1 将爬取的数据存放到本地的csv文件中

### 使用流程

```
1 1、导入模块
2 2、打开csv文件
3 3、初始化写入对象
4 4、写入数据(参数为列表)
5 import csv
6
7 with open('film.csv','w') as f:
8     writer = csv.writer(f)
9     writer.writerow([])
```

## 示例代码

创建 test.csv 文件，在文件中写入数据

```
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')])
```

## 练习

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

```
1 |
```

思考：使用 writerows()方法实现？

```
1 |
```

## 数据持久化存储 - MySQL数据库

### ■ 1、在数据库中建库建表

```

1 # 连接到mysql数据库
2 mysql -h127.0.0.1 -uroot -p123456
3 # 建库建表
4 create database maoyandb charset utf8;
5 use maoyandb;
6 create table filmtab(
7 name varchar(100),
8 star varchar(300),
9 time varchar(50)
10 )charset=utf8;

```

## ■ 2、回顾pymysql基本使用

```

1 import pymysql
2
3 # 创建2个对象
4 db = pymysql.connect('localhost','root','123456','maoyandb',charset='utf8')
5 cursor = db.cursor()
6
7 # 执行SQL命令并提交到数据库执行
8 # execute()方法第二个参数为列表传参补位
9 cursor.execute('insert into filmtab values(%s,%s,%s)',['霸王别姬','张国荣','1993'])
10 db.commit()
11
12 # 关闭
13 cursor.close()
14 db.close()

```

## 来试试高效的executemany()方法?

```

1 import pymysql
2
3 # 创建2个对象
4 db = pymysql.connect('192.168.153.137','tiger','123456','maoyandb',charset='utf8')
5 cursor = db.cursor()
6
7 # 抓取的数据
8 film_list = [('月光宝盒','周星驰','1994'),('大圣娶亲','周星驰','1994')]
9
10 # 执行SQL命令并提交到数据库执行
11 # execute()方法第二个参数为列表传参补位
12 cursor.executemany('insert into filmtab values(%s,%s,%s)',film_list)
13 db.commit()
14
15 # 关闭
16 cursor.close()
17 db.close()

```

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

```

1 |

```

## ■ 4、做个SQL查询

- 1 1、查询20年以前的电影的名字和上映时间
- 2
- 3 2、查询1990-2000年的电影名字和上映时间
- 4

## 电影天堂案例 - 二级页面抓取

### 领取任务

- 1 # 地址
- 2 电影天堂 - 2019年新片精品 - 更多
- 3 # 目标
- 4 电影名称、下载链接
- 5
- 6 # 分析
- 7 \*\*\*\*\*一级页面需抓取\*\*\*\*\*
- 8 1、电影名称
- 9 2、电影链接
- 10
- 11 \*\*\*\*\*二级页面需抓取\*\*\*\*\*
- 12 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

- 1 第1页 : [https://www.dytt8.net/html/gndy/dyzz/list\\_23\\_1.html](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](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](https://www.dytt8.net/html/gndy/dyzz/list_23_n.html)

- 3、写正则表达式

- 1 1、一级页面正则表达式
- 2 `<table width="100%".*?<td width="5%".*?<a href="(.*?)".*?uLink">(.*?)</a>.*?</table>`
- 3 2、二级页面正则表达式
- 4 `<td style="WORD-WRAP.*?>.*?>(.*?)</a>`

### 4、代码实现

1 |

- 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 | 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、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

- 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
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
5 3、查找li节点下所有title节点的,class属性的值
6
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