

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. 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 from urllib import request
2 import time
3 import re
4 import csv
5 import random
6
7 class MaoyanSpider(object):
8     def __init__(self):
9         self.url = 'https://maoyan.com/board/4?offset={}'
10        self.ua_list = [
11            'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
12            Chrome/72.0.3626.119 Safari/537.36',
13            'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.1 (KHTML, like Gecko)
14            Chrome/14.0.835.163 Safari/535.1',
15            'Mozilla/5.0 (Windows NT 6.1; WOW64; rv:6.0) Gecko/20100101 Firefox/6.0'
16        ]
17        # 爬取页数计数
18        self.page = 1
19
20    # 获取页面
21    def get_page(self, url):
22        # 访问不同页面使用随机的User-Agent
23        headers = {'User-Agent': random.choice(self.ua_list)}
24        req = request.Request(url, headers=headers)
25        res = request.urlopen(req)
26        html = res.read().decode('utf-8')
27        # 直接调用解析函数
28        self.parse_page(html)
29
30    # 解析页面
31    def parse_page(self, html):
32        # 正则解析
33        p = re.compile('<div class="movie-item-info">.*?title="(.*?)".*?class="star">(.*?)
34        </p>.*?releasetime">(.*?)</p>', re.S)
35        r_list = p.findall(html)
36        # r_list : [('霸王别姬', '张国荣', '1993'), (), ()]
37        self.write_page(r_list)
38
39    # 保存数据(从终端输出)
40    def write_page(self, r_list):
41        # r_list : [(), (), ()]
42        print(r_list)
43        film_dict = {}
44        for rt in r_list:
45            film_dict['name'] = rt[0].strip()
46            film_dict['star'] = rt[1].strip()
47            film_dict['time'] = rt[2].strip()
48
49            print(film_dict)
```

```

49     # 主函数
50     def main(self):
51         # 用range函数可获取某些查询参数的值
52         for offset in range(0,41,10):
53             url = self.url.format(offset)
54             self.get_page(url)
55             print('第{}页爬取成功'.format(self.page))
56             self.page += 1
57             # 每爬1页随机休眠, 控制爬取速率
58             time.sleep(random.randint(0,2))
59
60 if __name__ == '__main__':
61     spider = MaoyanSpider()
62     spider.main()

```

练习

猫眼电影数据存入本地 maoyanfilm.csv 文件

```

1     def write_page(self,r_list):
2         # r_list : [(,),(),()]
3         with open('maoyanfilm.csv','a') as f:
4             writer = csv.writer(f)
5             for rt in r_list:
6                 one_film_list = [
7                     rt[0].strip(),
8                     rt[1].strip(),
9                     rt[2].strip()
10                ]
11                writer.writerow(one_film_list)

```

思考: 使用 writerows()方法实现?

```

1     def write_page(self, r_list):
2         film_list = []
3         # r_list : [(,),(),()]
4         with open('maoyanfilm.csv', 'a',newline='') as f:
5             writer = csv.writer(f)
6             for rt in r_list:
7                 one_film = (rt[0].strip(), rt[1].strip(), rt[2].strip())
8                 film_list.append(one_film)
9             writer.writerows(film_list)

```

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()

```

让我们来回顾一下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 from urllib import request
2 import time
3 import re
4 import csv
5 import random
6 import pymysql
7
8 class MaoyanSpider(object):
9     def __init__(self):
10         self.url = 'https://maoyan.com/board/4?offset={}'
11         self.ua_list = [
12             'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
13             Chrome/72.0.3626.119 Safari/537.36',
14             'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.1 (KHTML, like Gecko)
15             Chrome/14.0.835.163 Safari/535.1',

```

```

14         'Mozilla/5.0 (Windows NT 6.1; WOW64; rv:6.0) Gecko/20100101 Firefox/6.0'
15     ]
16     # 爬取页数计数
17     self.page = 1
18
19     # 创建2个对象
20     self.db = pymysql.connect(
21         '192.168.153.130', 'tiger', '123456', 'maoyandb',
22         charset='utf8'
23     )
24     self.cursor = self.db.cursor()
25
26     # 获取页面
27     def get_page(self, url):
28         # 访问不同页面使用随机的User-Agent
29         headers = {'User-Agent': random.choice(self.ua_list)}
30         req = request.Request(url, headers=headers)
31         res = request.urlopen(req)
32         html = res.read().decode('utf-8')
33         # 直接调用解析函数
34         self.parse_page(html)
35
36     # 解析页面
37     def parse_page(self, html):
38         # 正则解析
39         p = re.compile('<div class="movie-item-info">.*?title="(.*?)".*?class="star">(.*?)</p>.*?releasetime">(.*?)</p>', re.S)
40         r_list = p.findall(html)
41         # r_list : [('霸王别姬', '张国荣', '1993'), (), ()]
42         self.write_page(r_list)
43
44     # 保存数据(存到mysql数据库)
45     def write_page(self, r_list):
46         # 存放1页电影数据的空列表
47         one_page_list = []
48         ins = 'insert into maoyanfilm(name,star,time) values(%s,%s,%s)'
49         for rt in r_list:
50             one_film_list = [
51                 rt[0].strip(),
52                 rt[1].strip(),
53                 rt[2].strip()[5:15]
54             ]
55             one_page_list.append(one_film_list)
56
57         self.cursor.executemany(ins, one_page_list)
58         # 提交到数据库执行
59         self.db.commit()
60
61     # 主函数
62     def main(self):
63         # 用range函数可获取某些查询参数的值
64         for offset in range(0, 41, 10):
65             url = self.url.format(offset)
66             self.get_page(url)
67             print('第{}页爬取成功'.format(self.page))
68             self.page += 1
69         # 每爬1页随机休眠, 控制爬取速率

```

```

70         time.sleep(random.randint(0,2))
71
72     if __name__ == '__main__':
73         spider = MaoyanSpider()
74         spider.main()

```

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      <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>

```

3. 代码实现

```

1  from urllib import request
2  import re
3
4

```

```

5 class FilmSpider(object):
6     def __init__(self):
7         self.url = 'https://www.dytt8.net/html/gndy/dyzz/list_23_{}.html'
8         self.headers = {'User-Agent': 'Mozilla/5.0'}
9
10    # 获取页面
11    def get_page(self, url):
12        req = request.Request(url, headers=self.headers)
13        res = request.urlopen(req)
14        html = res.read().decode('gb18030', 'ignore')
15        return html
16
17    # 解析一级页面
18    def parse_one_page(self, html):
19        p = re.compile('<table width="100%".*?<td width="5%".*?<a href="(.*?)".*?uLink">(.*?)</a>.*?</table>', re.S)
20        film_list = p.findall(html)
21        # [('/html/gndy/dyzz/20190523/58629.html', '2019年爱情喜剧《最佳男友进化论》HD国语中字'),]
22        for film_info in film_list:
23            film_name = film_info[1]
24            # 拼接详情页链接
25            film_link = 'https://www.dytt8.net{}'.format(film_info[0].strip())
26            # 获取二级页面的函数
27            down_link = self.get_download_link(film_link)
28            film = {
29                '电影名称': film_name,
30                '下载链接': down_link[0].strip()
31            }
32            print(film)
33
34    # 获取二级页面的数据
35    def get_download_link(self, film_link):
36        html = self.get_page(film_link)
37        p = re.compile('<td style="WORD-WRAP.*?>.*?>(.*?)</a>', re.S)
38        download_link_list = p.findall(html)
39        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()

```

练习

让我们来把电影天堂数据存入MySQL数据库

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 import requests
2
3 url='http://hbimg.b0.upaiyun.com/ac0a5f64360b9c55a6ea4ba395203543d48a8e401bcf7-6q2JJL_fw658'
4 headers = {'User-Agent':'Mozilla/5.0'}
5
6 # 获取响应内容bytes
7 html = requests.get(url,headers=headers).content
8 # 写文件
9 with open('颖宝.jpg','wb') as f:
10     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、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 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 //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
```