DAY02

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**

```
pattern = re.compile('正则表达式',re.S)
r_list = pattern.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))
```

Day02笔记

作业讲解

作业1 - 正则分组练习

页面结构如下:

```
<div class="animal">
    <p class="name">
    <a title="Tiger"></a>
    </p>
    <p class="content">
    Two tigers two tigers run fast
    </p>
</div>

<div class="animal">
    <p class="name">
    <a title="Rabbit"></a>
    </p>

    <p class="content">
    Small white rabbit white and white
    </p>
</div>
```

从以上html代码结构中完成如下内容信息的提取：

```
# 问题1
[('Tiger',' Two...'),('Rabbit','Small..')]
# 问题2
动物名称 ： Tiger
动物描述 ： Two tigers two tigers run fast
**************************************
动物名称 ： Rabbit
动物描述 ： Small white rabbit white and white
```

代码实现

```
import re

html = '''
<div class="animal">
    <p class="name">
      <a title="Tiger"></a>
    </p>
    <p class="content">
      Two tigers two tigers run fast
    </p>
</div>

<div class="animal">
    <p class="name">
      <a title="Rabbit"></a>
    </p>

    <p class="content">
      Small white rabbit white and white
    </p>
</div>
'''
```

```
pattern = re.compile(r'<div class="animal">.*?<a title="(.*?)".*?content">(.*?)</p>',re.S)
r_list = pattern.findall(html)
# 问题1
if r_list:
  print(r_list)
# r_list: [('Tiger','\n\t\t Two tigers xxx'),()]
# 问题2
if r_list:
    for r in r_list:
      print('动物名称:',r[0].strip())
      print('动物描述:',r[1].strip())
      print('*' * 50)
else:
    print('未匹配到数据')
```

猫眼电影top100抓取案例

```
猫眼电影 - 榜单 - top100榜
电影名称、主演、上映时间
```

数据抓取实现

- **1、确定响应内容中是否存在所需数据**

```
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 re
import time
import random
from useragents import ua_list

class MaoyanSpider(object):
  def __init__(self):
    self.url = 'https://maoyan.com/board/4?offset={}'
    # 计数
    self.num = 0
```

```
12
13    # 获取
14    def get_html(self,url):
15        headers = {
16            'User-Agent' : random.choice(ua_list)
17        }
18        req = request.Request(url=url,headers=headers)
19        res = request.urlopen(req)
20        html = res.read().decode('utf-8')
21        # 直接调用解析函数
22        self.parse_html(html)
23
24    # 解析
25    def parse_html(self,html):
26        re_bds = r'<div class="movie-item-info">.*?title="(.*?)".*?class="star">(.*?)</p>.*?releasetime">(.*?)</p>'
27        pattern = re.compile(re_bds,re.S)
28        # film_list: [('霸王别姬','张国荣','1993'),()]
29        film_list = pattern.findall(html)
30        # 直接调用写入函数
31        self.write_html(film_list)
32
33    def write_html(self,film_list):
34        item = {}
35        for film in film_list:
36            item['name'] = film[0].strip()
37            item['star'] = film[1].strip()
38            item['time'] = film[2].strip()[5:15]
39            print(item)
40
41            self.num += 1
42
43    def main(self):
44        for offset in range(0,31,10):
45            url = self.url.format(offset)
46            self.get_html(url)
47            time.sleep(random.randint(1,2))
48        print('共抓取数据:',self.num)
49
50if __name__ == '__main__':
51    start = time.time()
52    spider = MaoyanSpider()
53    spider.main()
54    end = time.time()
55    print('执行时间:%.2f' % (end-start))
```

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

```
# 单行写入 (writerow([]))
import csv
with open('test.csv','w',newline='') 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',newline='') 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

```
# 存入csv文件 - writerow()
def write_html(self,film_list):
  with open('film.csv','a') as f:
    # 初始化写入对象,注意参数f别忘了
    writer = csv.writer(f)
    for film in film_list:
      L = [
        film[0].strip(),
        film[1].strip(),
        film[2].strip()[5:15]
      ]
      # writerow()参数为列表
      writer.writerow(L)
```

思考：使用 writerows()方法实现?

```
# 存入csv文件 - writerows()
```

```
def write_html(self,film_list):
  L = []
  with open('film.csv','a') as f:
    # 初始化写入对象,注意参数f别忘了
    writer = csv.writer(f)
    for film in film_list:
      t = (
        film[0].strip(),
        film[1].strip(),
        film[2].strip()[5:15]
      )
      L.append(t)
    # writerows()参数为列表
    writer.writerows(L)
```

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
ins = 'insert into filmtab values(%s,%s,%s)'
cursor.execute(ins,['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
```

```
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  # mysql - executemany([(),(),()])
2  def write_html(self, film_list):
3    L = []
4    ins = 'insert into filmtab values(%s,%s,%s)'
5    for film in film_list:
6      t = (
7        film[0].strip(),
8        film[1].strip(),
9        film[2].strip()[5:15]
10     )
11     L.append(t)
12
13     self.cursor.executemany(ins, L)
14     # 千万别忘了提交到数据库执行
15     self.db.commit()
```

- **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    select name,time from filmtab 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 filmtab where time>='1990-01-01' and time<='2000-12-31';
```

数据持久化存储 - MongoDB数据库

pymongo操作mongodb数据库

```python
import pymongo

# 1.数据库连接对象
conn=pymongo.MongoClient('localhost',27017)
# 2.库对象
db = conn['库名']
# 3.集合对象
myset = db['集合名']
# 4.插入数据
myset.insert_one({字典})
```

思考

```
1、能否到电影详情页把评论抓取下来?
2、能否到电影详情页把电影图片抓取下来?  - 并按照电影名称分别创建文件夹
```

代码实现

```python
from urllib import request
import re
import time
import random
from useragents import ua_list
import os

class MaoyanSpider(object):
  def __init__(self):
    self.url = 'https://maoyan.com/board/4?offset={}'
    # 计数
    self.num = 0

  # 获取
  def get_html(self,url):
    headers = {
      'User-Agent' : random.choice(ua_list)
    }
    req = request.Request(url=url,headers=headers)
    res = request.urlopen(req)
    html = res.read().decode('utf-8','ignore')

    return html

  def re_func(self,re_bds,html):
    pattern = re.compile(re_bds,re.S)
    r_list = pattern.findall(html)

    return r_list

  # 解析
  def parse_html(self,url):
    re_bds = r'<div class="movie-item-info">.*?href="(.*?)".*?title="(.*?)".*?class="star">
(.*?)</p>.*?releasetime">(.*?)</p>'
    # html获取 + re解析
    html = self.get_html(url)
```

```
    film_list = self.re_func(re_bds,html)
    # 直接调用写入函数
    self.write_html(film_list)

  def write_html(self,film_list):
    film_dict = {}
    for film in film_list:
      film_dict['name'] = film[1].strip()
      film_dict['star'] = film[2].strip()
      film_dict['time'] = film[3].strip()[5:15]
      two_url = 'https://maoyan.com{}'.format(film[0].strip())
      film_dict['comment'] = self.get_comment(two_url)
      self.save_image(two_url,film)
      print(film_dict)

      self.num += 1

  def get_comment(self,two_url):
    # 获取 + 解析
    html = self.get_html(two_url)
    re_bds = r'<div class="comment-content">(.*?)</div>'
    comment_list = self.re_func(re_bds,html)

    return comment_list

  def save_image(self,two_url,film):
    re_bds = r'<div class="img.*?"><img class="default-img" data-src="(.*?)" alt=""></div>'
    html = self.get_html(two_url)
    img_link_list = self.re_func(re_bds,html)


    for img_link in img_link_list:
      req = request.Request(img_link)
      res = request.urlopen(req)
      html = res.read()
      # 处理文件名
      directory = 'D:\\猫眼\\{}\\'.format(film[1].strip())
      if not os.path.exists(directory):
        os.makedirs(directory)

      filename=directory + img_link.split('/')[-1].split('@')[0]
      with open(filename,'wb') as f:
        f.write(html)


  # 入口函数
  def run(self):
    for offset in range(0,31,10):
      url = self.url.format(offset)
      self.parse_html(url)
      time.sleep(random.randint(1,2))
    print('共抓取数据:',self.num)

if __name__ == '__main__':
  start = time.time()
  spider = MaoyanSpider()
  spider.run()
```

```
    end = time.time()
    print('执行时间:%.2f' % (end-start))
```

电影天堂二级页面抓取案例

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、电影详情页链接

*********二级页面需抓取**********
        1、电影名称
        2、电影下载链接
```

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">.*?</table>
2、二级页面正则表达式
   <div class="title_all"><h1><font color=#07519a>(.*?)</font></h1></div>.*?<td style="WORD-WRAP.*?>.*?>(.*?)</a>
```

- **4、代码实现**

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

class FilmSkySpider(object):
```

```python
  def __init__(self):
    # 一级页面url地址
    self.url = 'https://www.dytt8.net/html/gndy/dyzz/list_23_{}.html'

  # 获取html功能函数
  def get_html(self,url):
    headers = {
      'User-Agent':random.choice(ua_list)
    }
    req = request.Request(url=url,headers=headers)
    res = request.urlopen(req)
    # 通过网站查看网页源码,查看网站charset='gb2312'
    # 如果遇到解码错误,识别不了一些字符,则 ignore 忽略掉
    html = res.read().decode('gb2312','ignore')

    return html

  # 正则解析功能函数
  def re_func(self,re_bds,html):
    pattern = re.compile(re_bds,re.S)
    r_list = pattern.findall(html)

    return r_list

  # 获取数据函数 - html是一级页面响应内容
  def parse_page(self,one_url):
    html = self.get_html(one_url)
    re_bds = r'<table width="100%".*?<td width="5%".*?<a href="(.*?)".*?ulink">.*?</table>'
    # one_page_list: ['/html/xxx','/html/xxx','/html/xxx']
    one_page_list = self.re_func(re_bds,html)

    for href in one_page_list:
      two_url = 'https://www.dytt8.net' + href
      self.parse_two_page(two_url)
      # uniform: 浮点数,爬取1个电影信息后sleep
      time.sleep(random.uniform(1, 3))


  # 解析二级页面数据
  def parse_two_page(self,two_url):
    item = {}
    html = self.get_html(two_url)
    re_bds = r'<div class="title_all"><h1><font color=#07519a>(.*?)</font></h1></div>.*?<td style="WORD-WRAP.*?>.*?>(.*?)</a>'
    # two_page_list: [('名称1','ftp://xxxx.mkv')]
    two_page_list = self.re_func(re_bds,html)

    item['name'] = two_page_list[0][0].strip()
    item['download'] = two_page_list[0][1].strip()

    print(item)


  def main(self):
    for page in range(1,201):
      one_url = self.url.format(page)
      self.parse_page(one_url)
```

```
        # uniform: 浮点数
        time.sleep(random.uniform(1,3))

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

- **5、练习**

  把电影天堂数据存入MySQL数据库 - 增量爬取

```
# 思路
# 1、MySQL中新建表 urltab,存储所有爬取过的链接的指纹
# 2、在爬取之前,先判断该指纹是否爬取过,如果爬取过,则不再继续爬取
```

练习代码实现

```
# 建库建表
create database filmskydb charset utf8;
use filmskydb;
create table request_finger(
finger char(32)
)charset=utf8;
create table filmtab(
name varchar(200),
download varchar(500)
)charset=utf8;
```

```
from urllib import request
import re
from useragents import ua_list
import time
import random
import pymysql
from hashlib import md5


class FilmSkySpider(object):
    def __init__(self):
        # 一级页面url地址
        self.url = 'https://www.dytt8.net/html/gndy/dyzz/list_23_{}.html'
        self.db = pymysql.connect('192.168.153.151', 'tiger', '123456', 'filmskydb',
charset='utf8')
        self.cursor = self.db.cursor()

    # 获取html功能函数
    def get_html(self, url):
        headers = {
            'User-Agent': random.choice(ua_list)
        }
        req = request.Request(url=url, headers=headers)
        res = request.urlopen(req)
        # 通过网站查看网页源码,查看网站charset='gb2312'
        # 如果遇到解码错误,识别不了一些字符,则 ignore 忽略掉
        html = res.read().decode('gb2312', 'ignore')
```

```
        return html

    # 正则解析功能函数
    def re_func(self, re_bds, html):
        pattern = re.compile(re_bds, re.S)
        r_list = pattern.findall(html)

        return r_list

    # 获取数据函数
    def parse_page(self, one_url):
        html = self.get_html(one_url)
        re_bds = r'<table width="100%".*?<td width="5%".*?<a href="(.*?)".*?ulink">.*?</table>'
        # one_page_list: ['/html/xxx','/html/xxx','/html/xxx']
        one_page_list = self.re_func(re_bds, html)

        for href in one_page_list:
            two_url = 'https://www.dytt8.net' + href
            # 生成指纹 - md5加密
            s = md5()
            s.update(two_url.encode())
            two_url_md5 = s.hexdigest()
            # 判断链接是否需要抓取
            if self.is_go_on(two_url_md5):
                self.parse_two_page(two_url)
                # 爬取完成此链接后将指纹放到数据库表中
                ins = 'insert into request_finger values(%s)'
                self.cursor.execute(ins, [two_url_md5])
                self.db.commit()
                # uniform: 浮点数,爬取1个电影信息后sleep
                time.sleep(random.uniform(1, 3))


    def is_go_on(self, two_url_md5):
        # 爬取之前先到数据库中查询比对
        sel = 'select finger from request_finger where finger=%s'
        # 开始抓取之前,先来判断该链接之前是否抓取过
        result = self.cursor.execute(sel, [two_url_md5])
        if not result:
            return True


    # 解析二级页面数据
    def parse_two_page(self, two_url):
        item = {}
        html = self.get_html(two_url)
        re_bds = r'<div class="title_all"><h1><font color=#07519a>(.*?)</font></h1></div>.*?<td style="WORD-WRAP.*?>.*?>(.*?)</a>'
        # two_page_list: [('名称1','ftp://xxxx.mkv')]
        two_page_list = self.re_func(re_bds, html)

        item['name'] = two_page_list[0][0].strip()
        item['download'] = two_page_list[0][1].strip()

        ins = 'insert into filmtab values(%s,%s)'
```

```
        film_list = [
            item['name'], item['download']
        ]
        self.cursor.execute(ins, film_list)
        self.db.commit()
        print(film_list)


    def main(self):
        for page in range(1, 201):
            one_url = self.url.format(page)
            self.parse_page(one_url)
            # uniform: 浮点数
            time.sleep(random.uniform(1, 3))


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

今日作业

```
1、电影天堂数据,存入MySQL、MongoDB、CSV文件
2、百度图片抓取：输入要抓取的图片内容,抓取首页的30张图片,保存到对应的文件夹，比如：
   你想要谁的照片，请输入：赵丽颖
   创建文件夹到指定目录：赵丽颖  并把首页30张图片保存到此文件夹下
3、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,MongoDB数据库,CSV文件
  # 小区名 、总价 、单价
```