王伟超

wangweichao@tedu.cn

DAY01

网络爬虫概述

- ### *定义*

```
网络蜘蛛、网络机器人，抓取网络数据的程序
其实就是用Python程序模仿人点击浏览器并访问网站，而且模仿的越像越好，让Web站点无法发现你不是人
```

- ### *爬取数据目的*

```
1、公司项目测试数据
2、公司业务部门及其他部门所需数据
3、数据分析
```

- ### *企业获取数据方式*

```
1、公司自有数据
2、第三方数据平台购买(数据堂、贵阳大数据交易所)
3、爬虫爬取数据
```

- ### *Python做爬虫优势*

```
1、Python ： 请求模块、解析模块丰富成熟,强大的Scrapy网络爬虫框架
2、PHP ： 对多线程、异步支持不太好
3、JAVA：代码笨重,代码量大
4、C/C++：虽然效率高,但是代码成型慢
```

- ## *爬虫分类*

```
1、通用网络爬虫(搜索引擎使用,遵守robots协议)
  robots协议 ： 网站通过robots协议告诉搜索引擎哪些页面可以抓取,哪些页面不能抓取,通用网络爬虫需要遵守robots协议（君子协议）
  https://www.taobao.com/robots.txt
2、聚焦网络爬虫 ： 自己写的爬虫程序
```

- ## *爬虫爬取数据步骤*

```
1、确定需要爬取的URL地址
2、由请求模块向URL地址发出请求,并得到网站的响应
3、从响应内容中提取所需数据
   1、所需数据,保存
   2、页面中有其他需要继续跟进的URL地址,继续第2步去发请求，如此循环
```

爬虫请求模块一

模块名及导入

```
1、模块名：urllib.request
2、导入方式：
   1、import urllib.request
   2、from urllib import request
```

常用方法详解

urllib.request.urlopen

- **作用**

向网站发起请求并获取响应对象

- **参数**

```
1、url：需要爬取的URL地址
2、timeout：设置等待超时时间,指定时间内未得到响应抛出超时异常
```

- **第一个爬虫程序 - 01_urlopen.py**

打开浏览器，输入百度地址(http://www.baidu.com/)，得到百度的响应

```
import urllib.request

# urlopen() ：  向URL发请求,返回响应对象
response=urllib.request.urlopen('http://httpbin.org/get')
# 提取响应内容
html = response.read().decode('utf-8')
# 打印响应内容
print(html)
```

- **响应对象（response）方法**

```
1、bytes = response.read() # read()得到结果为 bytes 数据类型
2、string = response.read().decode() # decode() 转为 string 数据类型
3、url = response.geturl() # 返回实际数据的URL地址
4、code = response.getcode() # 返回HTTP响应码
# 补充
5、string.encode() # bytes -> string
6、bytes.decode()  # string -> bytes
```

思考：网站如何来判定是人类正常访问还是爬虫程序访问？？？

```
# 向测试网站：  http://httpbin.org/get 发请求,查看自己请求头
# 代码如下
import urllib.request
response=urllib.request.urlopen('http://httpbin.org/get')
html = response.read().decode()
print(html)

# html中的请求头headers如下
"headers": {
    "Accept-Encoding": "identity",
    "Host": "httpbin.org",
    "User-Agent": "Python-urllib/3.6"
  },
发现请求头中User-Agent竟然是:Python-urllib/3.6!!!!!!!!!!!!!!!!!!!
我们需要重构User-Agent,发请求时带着User-Agent过去,但是 urlopen()方法不支持重构User-Agent,那我们怎么办？请看下面的方法！！！
```

urllib.request.Request

- **作用**

创建请求对象(包装请求，重构User-Agent，使程序更像正常人类请求)

- **参数**

```
1、url：请求的URL地址
2、headers：添加请求头（爬虫和反爬虫斗争的第一步）
```

- **使用流程**

```
1、构造请求对象(重构User-Agent)
  req = urllib.request.Request(
      url = 'http://httpbin.org/get'
      headers={'User-Agent':'Mozilla/5.0'}
    )
2、发请求获取响应对象(urlopen)
  res = urllib.request.urlopen(req)
3、获取响应对象内容
  html = res.read().decode('utf-8')
```

- **示例 - 02_Request.py**

向测试网站（http://httpbin.org/get）发起请求，构造请求头并从响应中确认请求头信息

```
from urllib import request

# 定义常用变量: URL 、headers
url = 'http://httpbin.org/get'
headers = {
  'User-Agent':'Opera/9.80 (Windows NT 6.1; U; zh-cn) Presto/2.9.168 Version/11.50'
}
# 1. 创建请求对象 - 包装,并没有真正发请求
req = request.Request(url=url,headers=headers)
# 2. 获取响应对象
res = request.urlopen(req)
# 3. 提取响应内容
html = res.read().decode('utf-8')

print(html)
```

URL地址编码模块

模块名及导入

- **模块**

```
# 模块名
urllib.parse

# 导入
import urllib.parse
from urllib import parse
```

- **作用**

给URL地址中查询参数进行编码

```
编码前: https://www.baidu.com/s?wd=美女
编码后: https://www.baidu.com/s?wd=%E7%BE%8E%E5%A5%B3
```

常用方法

urllib.parse.urlencode({dict})

- **URL地址中一个查询参数**

```
# 查询参数: {'wd' : '美女'}
# urlencode编码后: 'wd=%e7%be%8e%e5%a5%b3'

# 示例代码
query_string = {'wd' : '美女'}
result = urllib.parse.urlencode(query_string)
# result: 'wd=%e7%be%8e%e5%a5%b3'
```

- **URL地址中多个查询参数**

```
from urllib import parse
params = {
  'wd' : '美女',
  'pn' : '50'
}
params = parse.urlencode(query_string_dict)
url = 'http://www.baidu.com/s?{}'.format(params)
print(url)
```

- **拼接URL地址的3种方式**

```
# 1、字符串相加
  baseurl = 'http://www.baidu.com/s?'
  params = 'wd=%E7XXXX&pn=20'
  url = baseurl + params

# 2、字符串格式化（占位符）
  params = 'wd=%E7XXXX&pn=20'
  url = 'http://www.baidu.com/s?%s'% params

# 3、format()方法
  url = 'http://www.baidu.com/s?{}'
  params = 'wd=#E7XXXX&pn=20'
  url = url.format(params)
```

- **练习** 在百度中输入要搜索的内容，把响应内容保存到本地文件

```
请输入搜索内容：赵丽颖
# 最终保存到本地文件 - 赵丽颖.html
```

代码实现 - 03_parse_baidu.py

```
from urllib import request
from urllib import parse

# 拼接URL地址
```

```python
def get_url(word):
  url = 'http://www.baidu.com/s?{}'
  # params: wd=%E7XXXXX
  params = parse.urlencode({'wd':word})
  url = url.format(params)

  return url


# 发请求,保存本地文件
def request_url(url,filename):
  headers = {'User-Agent':'Mozilla/5.0'}
  # 请求对象 + 响应对象 + 提取内容
  req = request.Request(url=url,headers=headers)
  res = request.urlopen(req)
  html = res.read().decode('utf-8')
  # 保存数据
  with open(filename,'w',encoding='utf-8') as f:
    f.write(html)

# 主程序入口
if __name__ == '__main__':
  word = input('请输入搜索内容:')
  url = get_url(word)
  filename = word + '.html'
  request_url(url,filename)
```

quote(string)编码

- **示例1**

```python
from urllib import parse

string = '美女'
print(parse.quote(string))
# 结果: %E7%BE%8E%E5%A5%B3
```

改写之前urlencode()代码，使用quote()方法实现

```python
from urllib import parse

url = 'http://www.baidu.com/s?wd={}'
word = input('请输入要搜索的内容:')
query_string = parse.quote(word)
print(url.format(query_string))
```

unquote(string)解码

- **示例**

```python
from urllib import parse

string = '%E7%BE%8E%E5%A5%B3'
result = parse.unquote(string)
print(result)
```

总结

```python
# 1、urllib.request
req = urllib.request.Request(url,headers)
res = urllib.request.urlopen(req)
html = res.read().decode('utf-8')

# 2、响应对象res方法
res.read()
res.getcode()
res.geturl()

# 3、urllib.parse
urllib.parse.urlencode({})
urllib.parse.quote(string)
urllib.parse.unquote(string)
```

百度贴吧数据抓取案例

要求

```
1、输入贴吧名称:赵丽颖吧
2、输入起始页:1
3、输入终止页:3
4、保存到本地文件
   赵丽颖吧-第1页.html、赵丽颖吧-第2页.html ...
```

实现步骤

- **1、查看是否为静态页面**

```
右键 - 查看网页源代码 - 搜索数据关键字
```

- **2、找URL规律**

```
第1页:http://tieba.baidu.com/f?kw=? ? &pn=0
第2页:http://tieba.baidu.com/f?kw=? ? &pn=50
第n页:pn=(n-1)*50
```

- **3、获取网页内容**
- **4、提取所需数据**
- **5、保存(本地文件、数据库)**

代码实现 - 04_tieba_spider.py

```python
from urllib import request,parse
import time
import random
from useragents import ua_list

class BaiduSpider(object):
  def __init__(self):
    self.url = 'http://tieba.baidu.com/f?kw={}&pn={}'

# 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')

    return html

  # 解析
  def parse_html(self):
    pass

  # 保存
  def write_html(self,filename,html):
    with open(filename,'w',encoding='utf-8') as f:
      f.write(html)

# 入口函数
  def run(self):
    name = input('请输入贴吧名:')
    begin = int(input('请输入起始页:'))
    end = int(input('请输入终止页:'))

    # 查询参数编码

    params = parse.quote(name)
    for page in range(begin,end+1):
      # URL拼接
      pn = (page-1)*50
      pn = (page-1)*50
      url = self.url.format(params,pn)

      # 调用类内函数进行页面抓取+保存
      filename = '{}-第{}页.html'.format(name,page)
      html = self.get_html(url)
      self.write_html(filename,html)
      # 控制爬取速度
      time.sleep(random.randint(1,3))
```

```
        print('第%d页爬取完成' % page)

if __name__ == '__main__':
    start = time.time()
    spider = BaiduSpider()
    spider.run()
    end = time.time()
    print('执行时间:%.2f' % (end-start))
```

正则解析模块

re模块使用流程

- **方法一**

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

- **方法二**

```
# 1、创建正则编译对象
pattern = re.compile(r'正则表达式',re.S)
r_list = pattern.findall(html)
```

正则表达式元字符

元字符	含义
.	任意一个字符（不包括\n）
\d	一个数字
\s	空白字符
\S	非空白字符
[]	包含[]内容
*	出现0次或多次
+	出现1次或多次

思考：请写出匹配任意一个字符的正则表达式?

```
import re
# 方法一
pattern = re.compile('.',re.S)
# 方法二
pattern = re.compile('[\s\S]')
```

贪婪匹配和非贪婪匹配

- **贪婪匹配（默认）**

```
1、在整个表达式匹配成功的前提下,尽可能多的匹配 *
2、表示方式：  .*
```

- **非贪婪匹配**

```
1、在整个表达式匹配成功的前提下,尽可能少的匹配 *
2、表示方式：.*?
```

示例代码 - 05_re_greed.py

```
import re

html = '''
<div><p>九霄龙吟惊天变</p></div>
<div><p>风云际汇潜水游</p></div>
'''
# 贪婪匹配 ：  .*
pattern = re.compile('<div><p>.*</p></div>',re.S)
r_list = pattern.findall(html)
print(r_list)

# 非贪婪匹配 ：.*?
pattern = re.compile('<div><p>.*?</p></div>',re.S)
r_list = pattern.findall(html)
print(r_list)
```

正则表达式分组

- **作用**

在完整的模式中定义子模式，将每个圆括号中子模式匹配出来的结果提取出来

- **示例**

```
import re

s = 'A B C D'
```

```
p1 = re.compile('\w+\s+\w+')
print(p1.findall(s))
# 结果: ['A B','C D']

p2 = re.compile('(\w+)\s+\w+')
print(p2.findall(s))
# 结果: ['A','C']

p3 = re.compile('(\w+)\s+(\w+)')
print(p3.findall(s))
# 结果: [('A','B'),('C','D')]
```

- **分组总结**

```
1、在网页中,想要什么内容,就加()
2、先按整体正则匹配,然后再提取分组()中的内容
   如果有2个及以上分组(),则结果中以元组形式显示 [('小区1','500万'),('小区2','600万'),()]
```

练习

页面结构如下:

```
# <div class="animal">.*?title="(.*?)".*?
<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 - 06_re_exercise.py

```python
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')
```

今日作业

1、把百度贴吧案例重写一遍,不要参照课上代码 2、爬取猫眼电影信息： 猫眼电影-榜单-top100榜

```
第1步完成:
  猫眼电影-第1页.html
  猫眼电影-第2页.html
  ... ...

第2步完成:
  1、提取数据 ： 电影名称、主演、上映时间
  2、先打印输出,然后再写入到本地文件
```

3、复习任务

```
pymysql、MySQL基本命令
MySQL  ：建库建表普通查询等
```

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

    # 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')
        # 直接调用解析函数
        self.parse_html(html)

    # 解析
    def parse_html(self,html):
        re_bds = r'<div class="movie-item-info">.*?title="(.*?)".*?class="star">(.*?)</p>.*?releasetime">(.*?)</p>'
        pattern = re.compile(re_bds,re.S)
        # film_list: [('霸王别姬','张国荣','1993'),()]
        film_list = pattern.findall(html)
        # 直接调用写入函数
        self.write_html(film_list)

    def write_html(self,film_list):
        item = {}
        for film in film_list:
            item['name'] = film[0].strip()
            item['star'] = film[1].strip()
            item['time'] = film[2].strip()[5:15]
            print(item)

            self.num += 1

    def main(self):
        for offset in range(0,31,10):
            url = self.url.format(offset)
            self.get_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.main()
    end = time.time()
    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()
```

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
```

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 = 'https://timgsa.baidu.com/timg?image&quality=80&size=b9999_10000&sec=1567090051520&di=77e8b97b3280f999cf51340af4315b4b&imgtype=jpg&src=http%3A%2F%2F5b0988e595225.cdn.sohucs.com%2Fimages%2F20171121%2F4e6759d153d04c6badbb0a5262ec103d.jpeg'
headers = {'User-Agent':'Mozilla/5.0'}

html = requests.get(url=url,headers=headers).content
with open('花千骨.jpg','wb') as f:
    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、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
#3、有的插件直接拖拽 .zip 文件释放即可
```

需要安装插件

```
1、Xpath Helper: 轻松获取HTML元素的xPath路径
  # 开启/关闭: Ctrl + Shift + x
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
```

今日作业

```
1、把之前所有代码改为 requests 模块
2、电影天堂数据,存入MySQL、MongoDB、CSV文件
3、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
4、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
```