Task 05爬虫入门与综合应用

5.0 前言

对于自动化办公而言，网络数据的批量获取完数据可以节约相当的时间，因此爬虫在自动化办公中占据了一个比较重要的位置。

因而本节针对爬虫项目进行一个介绍，力求最大程度还原实际的办公场景。

5.1 Requests简介

Requests是一款目前非常流行的http请求库，使用python编写，能非常方便的对网页Requests进行爬取，也是爬虫最常用的发起请求第三方库。

安装方法：

```
pip install requests
或者conda安装
conda install requests
```

```
re.status_code 响应的HTTP状态码
re.text 响应内容的字符串形式
rs.content 响应内容的二进制形式
rs.encoding 响应内容的编码
```

5.1.1 访问百度

试一试对百度首页进行数据请求：

项目难度：☆

```
import requests
# 发出http请求
re=requests.get("https://www.baidu.com")
# 查看响应状态
print(re.status_code)
#输出：200
#200就是响应的状态码，表示请求成功
#我们可以通过res.status_code的值来判断请求是否成功。
```

res.text 返回的是服务器响应内容的字符串形式，也就是文本内容

5.1.2 下载txt文件

例：用爬虫下载孔乙己的文章，网址是https://apiv3.shanbay.com/codetime/articles/mnvdu

我们打开这个网址 可以看到是鲁迅的文章

我们尝试着用爬虫保存文章的内容

项目难度：☆

```
import requests
# 发出http请求
re = requests.get('https://apiv3.shanbay.com/codetime/articles/mnvdu')
# 查看响应状态
print('网页的状态码为%s'%re.status_code)
with open('鲁迅文章.txt', 'w') as file:
  # 将数据的字符串形式写入文件中
  print('正在爬取小说')
  file.write(re.text)
```

re.txt就是网页中的内容，将内容保存到txt文件中

5.1.3 下载图片

re.text用于文本内容的获取、下载
re.content用于图片、视频、音频等内容的获取、下载

项目难度：☆☆

```
import requests
# 发出http请求
#下载图片
res=requests.get('https://img-blog.csdnimg.cn/20210424184053989.PNG')
# 以二进制写入的方式打开一个名为 info.jpg 的文件
with open('datawhale.png','wb') as ff:
    # 将数据的二进制形式写入文件中
    print('爬取图片')
    ff.write(res.content)
```

re.encoding 爬取内容的编码形似，常见的编码方式有 ASCII、GBK、UTF-8 等。如果用和文件编码不同的方式去解码，我们就会得到一些乱码。

5.2 HTML解析和提取

浏览器工作原理：

向浏览器中输入某个网址，浏览器回向服务器发出请求，然后服务器就会作出响应。其实，服务器返回给浏览器的这个结果就是HTML代码，浏览器会根据这个HTML代码将网页解析成平时我们看到的那样

比如我们来看看百度的html页面

```
import requests
res=requests.get('https://baidu.com')
print(res.text)
```

将会看到很多带有标签的信息

HTML(Hyper Text Markup Language)是一种超文本标记语言，是由一堆标记组成。

例如

```
<html>
  <head>
    <title>我的网页</title>
  </head>
  <body>
    Hello，World
  </body>
</html>
```

上面即为一个最简单的html，我们所需要的信息就是夹在标签中

想对html有根据深入的了解，可以html菜鸟教程

https://www.runoob.com/html/html-tutorial.html

那么我们如何解析html页面呢?

5.3 BeautifulSoup简介

我们一般会使用BeautifulSoup这个第三方库

安装方法：

```
pip install bs4
或
conda install bs4
```

我们来解析豆瓣读书 Top250

它的网址是：https://book.douban.com/top250

项目难度：☆☆

```
import io
import sys
import requests
from bs4 import BeautifulSoup
###运行出现乱码时可以修改编码方式
#sys.stdout = io.TextIOWrapper(sys.stdout.buffer,encoding='gb18030')
###
headers = {
  'user-agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3809.132 Safari/537.36'
}
res = requests.get('https://book.douban.com/top250', headers=headers)
soup = BeautifulSoup(res.text, 'lxml')
print(soup)
```

python 打印信息时会有限制 我们将打印的编码改成gb18030

headers表示我们的请求网页的头，对于没有headers的请求可能会被服务器判定为爬虫而拒绝提供服务

通过 from bs4 import BeautifulSoup 语句导入 BeautifulSoup

然后使用 BeautifulSoup(res.text, lxmlr') 语句将网页源代码的字符串形式解析成了 BeautifulSoup 对象

解析成了 BeautifulSoup 对象可以较为方便的提取我们需要的信息

那么如何提取信息呢?

BeautifulSoup 为我们提供了一些方法

find()方法和**find_all()方法**:

- **find()** 返回符合条件的**首个**数据
- **find_all()** 返回符合条件的**所有**数据

```
import io
import sys
import requests
from bs4 import BeautifulSoup
#如果出现了乱码报错，可以修改编码形式
#sys.stdout = io.TextIOWrapper(sys.stdout.buffer,encoding='gb18030')
#
headers = {
  'user-agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3809.132 Safari/537.36'
}
res = requests.get('https://book.douban.com/top250', headers=headers)
soup = BeautifulSoup(res.text, 'lxml')
print(soup.find('a'))
#<a class="nav-login" href="https://accounts.douban.com/passport/login?
source=book" rel="nofollow">登录/注册</a>
print(soup.find_all('a'))
#返回一个列表 包含了所有的<a>标签
```

除了传入 HTML 标签名称外，BeautifulSoup 还支持熟悉的定位

```
# 定位div开头 同时id为'doubanapp-tip的标签
soup.find('div', id='doubanapp-tip')
# 定位a抬头 同时class为rating_nums的标签
soup.find_all('span', class_='rating_nums')
#class是python中定义类的关键字，因此用class_表示HTML中的class
```

HTML定位方法：https://www.cnblogs.com/bosslv/p/8992410.html

理论看百遍，不如上手一练

5.4 实践项目1：自如公寓数据抓取

> 首先是先说一声抱歉，在课程设计时，没有想到自如公寓在价格上增加一定程度的反爬措施，因此自如公寓的价格在本节不讨论，在以后的课程中，我们会详细讲解相关的方法。
>
> 本节内容为作者原创的项目，整体爬取过程有4星的难度，建议读者跟着课程一步一步的来，如果有不明白的地方，可以在群里面与其他伙伴进行交流。
>
> 在输出本节内容时，请注明来源，Datawhale自动化办公课程，谢谢~

日前，国务院办公厅印发《关于加快培育和发展住房租赁市场的若干意见》，你是某新媒体公司的一名员工，老板希望对武汉的租房情况进行深度调研与分析，你想调查自如公寓的数据情况。根据工作的安排，你调研的是自如公寓武汉房屋出租分析的任务。

项目难度：☆☆☆☆

自如公寓官网：https://wh.ziroom.com/z/z/

通过观察官网你发现

第1页的网页为：https://wh.ziroom.com/z/p1/

第2页的网页为：https://wh.ziroom.com/z/p2/

第3页的网页为：https://wh.ziroom.com/z/p3/

...

第50页的网页为：https://wh.ziroom.com/z/p50/

你继续观察，发现

房屋的信息网页为类似于：https://wh.ziroom.com/x/741955798.html

即：https://wh.ziroom.com/x/XXXX.html

因此你有了思路，通过访问自如公寓的网站，获取每个房间后面的数字号 然后通过数字号访问房屋的直接信息，然后抓取房屋的信息保存在excel中

于是你访问了房屋的网页：https://wh.ziroom.com/x/741955798.html

通过观察房屋的网页，你发现是这些信息是你需要的

房屋的名称，房屋的面积，房屋的朝向，房屋的户型，房屋的位置，房屋的楼层，是否有电梯，房屋的年代，门锁情况，绿化情况

但是你遇到了困难，不知道这些信息的标签信息，不能用beautifulsoup对他们进行定位

通过百度查询，浏览器按F12时能进入源代码模式 或者 点击右键进入审查元素

点击左上角的箭头，可以定位到元素的位置

方法掌握后你开始写代码了

```
import requests
from bs4 import BeautifulSoup
import random
import time
import csv
```

写到这里的时候，你想到，我多次访问自如的官网，如果只用一个UA头岂不是很容易被反爬虫识别

你想到，我可以做很多个UA头，然后每次访问的时候可以随机选一个，想到这里，你直呼自己是个天才

于是，你到网上找到了很多UA头信息

```
#这里增加了很多user_agent
#能一定程度能保护爬虫
user_agent = [
    "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_6_8; en-us) AppleWebKit/534.50 (KHTML, like Gecko) Version/5.1 Safari/534.50",
    "Mozilla/5.0 (Windows; U; Windows NT 6.1; en-us) AppleWebKit/534.50 (KHTML, like Gecko) Version/5.1 Safari/534.50",
    "Mozilla/5.0 (Windows NT 10.0; WOW64; rv:38.0) Gecko/20100101 Firefox/38.0",
    "Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; .NET4.0C; .NET4.0E; .NET CLR 2.0.50727; .NET CLR 3.0.30729; .NET CLR 3.5.30729; InfoPath.3; rv:11.0) like Gecko",
    "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)",
    "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.0; Trident/4.0)",
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0)",
    "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)",
    "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.6; rv:2.0.1) Gecko/20100101 Firefox/4.0.1",
    "Mozilla/5.0 (Windows NT 6.1; rv:2.0.1) Gecko/20100101 Firefox/4.0.1",
    "Opera/9.80 (Macintosh; Intel Mac OS X 10.6.8; U; en) Presto/2.8.131 Version/11.11",
    "Opera/9.80 (Windows NT 6.1; U; en) Presto/2.8.131 Version/11.11",
    "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_0) AppleWebKit/535.11 (KHTML, like Gecko) Chrome/17.0.963.56 Safari/535.11",
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Maxthon 2.0)",
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; TencentTraveler 4.0)",
```

```
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)",
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; The World)",
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; SE 2.X
MetaSr 1.0; SE 2.X MetaSr 1.0; .NET CLR 2.0.50727; SE 2.X MetaSr 1.0)",
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; 360SE)",
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Avant Browser)"]
```

现在开始正式开始爬取数据了

房屋的名称，房屋的价格，房屋的面积，房屋的朝向，房屋的户型，房屋的位置，房屋的楼层，是否有电梯，房屋的年代，门锁情况，绿化情况

你思考爬取的信息应该保存到csv文件中，于是你导入了csv包 并简单的了解了CSV包的用法

第一步，是要获取房屋的数字标签

于是你打开了自如的官网，用浏览器的元素进行定位

发现房屋的信息标签都是这个

< a href="dd//wh.ziroom.com/x/741955798.html" target="_blank"> 房屋名称< /a >

聪明的你，随手写下了这个代码,便能爬取自如前50页

```
def get_info():
    csvheader=['名称','面积','朝向','户型','位置','楼层','是否有电梯','建成时间',' 门
锁','绿化']
    with open('wuhan_ziru.csv', 'a+', newline='') as csvfile:
        writer  = csv.writer(csvfile)
        writer.writerow(csvheader)
        for i in range(1,50):  #总共有50页
            print('正在爬取自如第%s页'%i)
            timelist=[1,2,3]
            print('有点累了，需要休息一下啦（￣㉨￣）')
            time.sleep(random.choice(timelist))   #休息1-3秒，防止给对方服务器过大的
压力！！！
            url='https://wh.ziroom.com/z/p%s/'%i
            headers = {'User-Agent': random.choice(user_agent)}
            r = requests.get(url, headers=headers)
            r.encoding = r.apparent_encoding
            soup = BeautifulSoup(r.text, 'lxml')
            all_info = soup.find_all('div', class_='info-box')
            print('开始干活咯(๑>◡<๑）')
            for info in all_info:
                href = info.find('a')
                if href !=None:
                    href='https:'+href['href']
                    try:
                        print('正在爬取%s'%href)
                        house_info=get_house_info(href)
                        writer.writerow(house_info)
                    except:
                        print('出错啦，%s进不去啦( •̀ ω •́ )'%href)
```

通过研究发现了你需要定位的信息 通过标签头 h1 li span 和class的值对标签进行定位

```
<h1 class="Z_name"><i class="status iconicon_sign"></i>自如友家·电建地产盛世江城·4居
室-05卧</h1>
```

```
----
<div class="Z_home_info">
<div class="Z_home_b clearfix">
    <dl class="">
        <dd>8.4㎡</dd>
        <dt>使用面积</dt>
    </dl>
    <dl class="">
        <dd>朝南</dd>
        <dt>朝向</dt>
    </dl>
    <dl class="">
        <dd>4室1厅</dd>
        <dt>户型</dt>
    </dl>
</div>
</div>
----
<ul class="Z_home_o">
    <li>
        <span class="la">位置</span><span class="va">
        <span class="ad">小区距2号线长港路站步行约231米</span>
     </li>
        <span class="la">楼层</span><span class="va">6/43</span>
    </li>
    <li>
        <span class="la">电梯</span><span class="va">有</span>
    </li>
    <li>
        <span class="la">年代</span><span class="va">2016年建成</span>
    </li>
    <li>
        <span class="la">门锁</span><span class="va">智能门锁</span>
    </li>
    <li>
        <span class="la">绿化</span><span class="va">35%</span>
    </li>
</ul>
```

通过对上面标签的研究你完成了所有的代码

```
import requests
from bs4 import BeautifulSoup
import random
import time
import csv

#这里增加了很多user_agent
#能一定程度能保护爬虫
user_agent = [
    "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_6_8; en-us) AppleWebKit/534.50 (KHTML, like Gecko) Version/5.1 Safari/534.50",
    "Mozilla/5.0 (Windows; U; Windows NT 6.1; en-us) AppleWebKit/534.50 (KHTML, like Gecko) Version/5.1 Safari/534.50",
    "Mozilla/5.0 (Windows NT 10.0; WOW64; rv:38.0) Gecko/20100101 Firefox/38.0",
```

```
    "Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; .NET4.0C; .NET4.0E; .NET
CLR 2.0.50727; .NET CLR 3.0.30729; .NET CLR 3.5.30729; InfoPath.3; rv:11.0) like
Gecko",
    "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)",
    "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.0; Trident/4.0)",
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0)",
    "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)",
    "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.6; rv:2.0.1) Gecko/20100101
Firefox/4.0.1",
    "Mozilla/5.0 (Windows NT 6.1; rv:2.0.1) Gecko/20100101 Firefox/4.0.1",
    "Opera/9.80 (Macintosh; Intel Mac OS X 10.6.8; U; en) Presto/2.8.131
Version/11.11",
    "Opera/9.80 (Windows NT 6.1; U; en) Presto/2.8.131 Version/11.11",
    "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_0) AppleWebKit/535.11 (KHTML,
like Gecko) Chrome/17.0.963.56 Safari/535.11",
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Maxthon 2.0)",
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; TencentTraveler 4.0)",
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)",
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; The World)",
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; SE 2.X
MetaSr 1.0; SE 2.X MetaSr 1.0; .NET CLR 2.0.50727; SE 2.X MetaSr 1.0)",
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; 360SE)",
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Avant Browser)"]

def get_info():
    csvheader=['名称','面积','朝向','户型','位置','楼层','是否有电梯','建成时间',' 门
锁','绿化']
    with open('wuhan_ziru.csv', 'a+', newline='') as csvfile:
        writer  = csv.writer(csvfile)
        writer.writerow(csvheader)
        for i in range(1,50):  #总共有50页
            print('正在爬取自如第%s页'%i)
            timelist=[1,2,3]
            print('有点累了，需要休息一下啦（￣㉨￣）')
            time.sleep(random.choice(timelist))   #休息1-3秒，防止给对方服务器过大的
压力！！！
            url='https://wh.ziroom.com/z/p%s/'%i
            headers = {'User-Agent': random.choice(user_agent)}
            r = requests.get(url, headers=headers)
            r.encoding = r.apparent_encoding
            soup = BeautifulSoup(r.text, 'lxml')
            all_info = soup.find_all('div', class_='info-box')
            print('开始干活咯(๑>◡<๑）')
            for info in all_info:
                href = info.find('a')
                if href !=None:
                    href='https:'+href['href']
                    try:
                        print('正在爬取%s'%href)
                        house_info=get_house_info(href)
                        writer.writerow(house_info)
                    except:
                        print('出错啦，%s进不去啦( •̀ ω •́ )'%href)

def get_house_info(href):
    #得到房屋的信息
    time.sleep(1)
    headers = {'User-Agent': random.choice(user_agent)}
```

```
    response = requests.get(url=href, headers=headers)
    response=response.content.decode('utf-8', 'ignore')
    soup = BeautifulSoup(response, 'lxml')
    name = soup.find('h1', class_='Z_name').text
    sinfo=soup.find('div', class_='Z_home_b clearfix').find_all('dd')
    area=sinfo[0].text
    orien=sinfo[1].text
    area_type=sinfo[2].text
    dinfo=soup.find('ul',class_='Z_home_o').find_all('li')
    location=dinfo[0].find('span',class_='va').text
    loucen=dinfo[1].find('span',class_='va').text
    dianti=dinfo[2].find('span',class_='va').text
    niandai=dinfo[3].find('span',class_='va').text
    mensuo=dinfo[4].find('span',class_='va').text
    lvhua=dinfo[5].find('span',class_='va').text
    ['名称','面积','朝向','户型','位置','楼层','是否有电梯','建成时间',' 门锁','绿化']
    room_info=
[name,area,orien,area_type,location,loucen,dianti,niandai,mensuo,lvhua]
    return room_info

if __name__ == '__main__':
    get_info()
```

运行完成后，会在文件夹中看到刚才爬取好的信息保存在wuhan_ziru.csv中

5.5 实践项目2：36kr信息抓取与邮件发送

> 本节内容为作者原创的项目，课程难度为5星，建议读者跟着课程一步一步的来，如果有不明白的地方，可以在群里面与其他伙伴进行交流。
>
> 在输出本节内容时，请注明来源，Datawhale自动化办公课程，谢谢~
>
> 如果没有多个邮箱，可以百度搜索临时邮箱进行实践学习

项目难度：☆☆☆☆☆

完成了上面的实践项目1后，你膨胀到不行，觉得自己太厉害了。通过前面的学习，你了解到使用python进行电子邮件的收发，突然有一天你想到，如果我用A账户进行发送，同时用B账户进行接受，在手机上安装一个邮件接受的软件，这样就能完成信息从pc端投送到移动端。

在这样的思想上，就可以对动态变化的信息进行监控，一旦信息触发了发送的条件，可以将信息通过邮件投送到手机上，从而让自己最快感知到。

具体路径是：

python爬虫-->通过邮件A发送-->服务器--->通过邮件B接收

因此我们本节的内容就是爬取36kr的信息然后通过邮件发送

36kr官网：https://36kr.com/newsflashes

通过python发送邮件需要获得pop3的授权码

具体获取方式可参考：

https://blog.csdn.net/wateryouyo/article/details/51766345

接下来就爬取36Kr的网站

通过观察我们发现 消息的标签为

```
<a class="item-title" rel="noopener noreferrer" target="_blank"
href="/newsflashes/1218249313424001" sensors_operation_list="page_flow">中国平安：推动新方正集团聚集医疗健康等核心业务发展</a>
```

因此我们爬取的代码为

需要注意的是，邮箱发送消息用的HTML的模式，而HTML模式下换行符号为 < br>

```
def main():
    print('正在爬取数据')
    url = 'https://36kr.com/newsflashes'
    headers = {'User-Agent': random.choice(user_agent)}
    response = requests.get(url, headers=headers)
    response=response.content.decode('utf-8', 'ignore')
    soup = BeautifulSoup(response, 'lxml')
    news = soup.find_all('a', class_='item-title')
    news_list=[]
    for i in news:
        title=i.get_text()
        href='https://36kr.com'+i['href']
        news_list.append(title+'<br>'+href)
    info='<br></br>'.join(news_list)
```

接下来就是配置邮箱的发送信息

```
smtpserver = 'smtp.qq.com'

# 发送邮箱用户名密码
user = ''
password = ''

# 发送和接收邮箱
sender = ''
receive = ''

def send_email(content):
    # 通过QQ邮箱发送
    title='36kr快讯'
    subject = title
    msg = MIMEText(content, 'html', 'utf-8')
    msg['Subject'] = Header(subject, 'utf-8')
    msg['From'] = sender
    msg['To'] = receive
    # SSL协议端口号要使用465
    smtp = smtplib.SMTP_SSL(smtpserver, 465)  # 这里是服务器端口!
    # HELO 向服务器标识用户身份
    smtp.helo(smtpserver)
    # 服务器返回结果确认
    smtp.ehlo(smtpserver)
    # 登录邮箱服务器用户名和密码
    smtp.login(user, password)
    smtp.sendmail(sender, receive, msg.as_string())
    smtp.quit()
```

最后我们的整个代码文件为

```
import requests
import random
from bs4 import BeautifulSoup
import smtplib  # 发送邮件模块
from email.mime.text import MIMEText  # 定义邮件内容
from email.header import Header  # 定义邮件标题

smtpserver = 'smtp.qq.com'

# 发送邮箱用户名密码
user = ''
password = ''

# 发送和接收邮箱
sender = ''
receive = ''

user_agent = [
    "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_6_8; en-us) AppleWebKit/534.50
(KHTML, like Gecko) Version/5.1 Safari/534.50",
    "Mozilla/5.0 (Windows; U; Windows NT 6.1; en-us) AppleWebKit/534.50 (KHTML,
like Gecko) Version/5.1 Safari/534.50",
    "Mozilla/5.0 (Windows NT 10.0; WOW64; rv:38.0) Gecko/20100101 Firefox/38.0",
    "Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; .NET4.0C; .NET4.0E; .NET
CLR 2.0.50727; .NET CLR 3.0.30729; .NET CLR 3.5.30729; InfoPath.3; rv:11.0) like
Gecko",
    "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)",
    "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.0; Trident/4.0)",
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0)",
    "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)",
    "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.6; rv:2.0.1) Gecko/20100101
Firefox/4.0.1",
    "Mozilla/5.0 (Windows NT 6.1; rv:2.0.1) Gecko/20100101 Firefox/4.0.1",
    "Opera/9.80 (Macintosh; Intel Mac OS X 10.6.8; U; en) Presto/2.8.131
Version/11.11",
    "Opera/9.80 (Windows NT 6.1; U; en) Presto/2.8.131 Version/11.11",
    "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_0) AppleWebKit/535.11 (KHTML,
like Gecko) Chrome/17.0.963.56 Safari/535.11",
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Maxthon 2.0)",
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; TencentTraveler 4.0)",
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)",
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; The World)",
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; SE 2.X
MetaSr 1.0; SE 2.X MetaSr 1.0; .NET CLR 2.0.50727; SE 2.X MetaSr 1.0)",
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; 360SE)",
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Avant Browser)"]

def main():
    print('正在爬取数据')
    url = 'https://36kr.com/newsflashes'
    headers = {'User-Agent': random.choice(user_agent)}
    response = requests.get(url, headers=headers)
    response=response.content.decode('utf-8', 'ignore')
    soup = BeautifulSoup(response, 'lxml')
    news = soup.find_all('a', class_='item-title')
    news_list=[]
    for i in news:
        title=i.get_text()
```

```
            href='https://36kr.com'+i['href']
            news_list.append(title+'<br>'+href)
        info='<br></br>'.join(news_list)
        print('正在发送信息')
        send_email(info)

def send_email(content):
    # 通过QQ邮箱发送
    title='36kr快讯'
    subject = title
    msg = MIMEText(content, 'html', 'utf-8')
    msg['Subject'] = Header(subject, 'utf-8')
    msg['From'] = sender
    msg['To'] = receive
    # SSL协议端口号要使用465
    smtp = smtplib.SMTP_SSL(smtpserver, 465)  # 这里是服务器端口!
    # HELO 向服务器标识用户身份
    smtp.helo(smtpserver)
    # 服务器返回结果确认
    smtp.ehlo(smtpserver)
    # 登录邮箱服务器用户名和密码
    smtp.login(user, password)
    smtp.sendmail(sender, receive, msg.as_string())
    smtp.quit()

if __name__ == '__main__':
    main()
```

Task5 END.

--- By: 牧小熊

> 华中农业大学研究生，Datawhale成员, Datawhale优秀原创作者
>
> 知乎：https://www.zhihu.com/people/muxiaoxiong

关于Datawhale： Datawhale是一个专注于数据科学与AI领域的开源组织，汇集了众多领域院校和知名企业的优秀学习者，聚合了一群有开源精神和探索精神的团队成员。Datawhale 以“for the learner，和学习者一起成长”为愿景，鼓励真实地展现自我、开放包容、互信互助、敢于试错和勇于担当。同时 Datawhale 用开源的理念去探索开源内容、开源学习和开源方案，赋能人才培养，助力人才成长，建立起人与人，人与知识，人与企业和人与未来的联结。 本次数据挖掘路径学习，专题知识将在天池分享，详情可关注Datawhale：