# Python实验报告10

## **实验十： 网络爬虫**

## 实验目的：学习网络爬虫编写，获取数据

## 实验要求：

1. 能运用request库和beautifulsoup4库访问URL并解析获取的HTML

2. 能向百度等搜索引擎自动提交关键词并获取返回结果

## **实验内容：**

1.程序练习题10.1 修改教材实例代码20.1中的25-27行即可，只输出给定的省份的学校。要求输出江西省和北京市的高校

**编写代码：**

import requests

from bs4 import BeautifulSoup

allUniv=[]

def getHTMLText(url):

try:

r = requests.get(url, timeout=30)

r.raise\_for\_status()

r.encoding = 'utf-8'

return r.text

except:

return ""

def fillUnivList(soup):

data = soup.find\_all('tr')

for tr in data:

ltd = tr.find\_all('td')

if len(ltd ) == 0:

continue

singleUniv = []

for td in ltd:

singleUniv.append(td.string)

allUniv.append(singleUniv)

def printUnivList(province):

print("{1:^2}{2:{0}^10}{3:{0}^6}{4:{0}^4}{5:{0}^10}".format(chr(12288),"排名","学校名称","省市","总分","培养规模"))

for u in allUniv:

if province in u[2]:

print("{1:^2}{2:{0}^10}{3:{0}^6}{4:{0}^4}{5:{0}^10}".format(chr(12288),u[0],u[1],u[2],u[3],u[4]))

def main(p):

url = 'http://www.zuihaodaxue.cn/zuihaodaxuepaiming2018.html'

html = getHTMLText(url)

soup = BeautifulSoup(html,'html.parser')

fillUnivList(soup)

printUnivList(p)

main('江西')

2.程序练习题10.2 实现USNEWS美国大学排名的爬虫，并打印结果。

提示：

1）、美国大学排名网站：<https://www.usnews.com/best-colleges/rankings/national-universities>

2）、上述网站只列出了排名前10的高校，当鼠标往下翻滚时才会加载后10个高校，仔细分析html源代码，找到显示后10个高校的网址。

**编写代码：**

import requests

import re

from bs4 import BeautifulSoup

def getHTMLText(url):

send\_headers = {

"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/61.0.3163.100 Safari/537.36",

"Connection": "keep-alive",

"Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8",

"Accept-Language": "zh-CN,zh;q=0.8"}

try:

r = requests.get(url, headers=send\_headers)

r.raise\_for\_status()

print(r.status\_code)

r.encoding = 'utf-8'

return r.text

except:

return ""

def fillUnivList(soup, allUniv):

data = soup.find\_all('div',{'class':re.compile('shadow-dark')})

for div in data:

singleUniv = []

div1 = div.find('div',{'style':'margin-left: 2.5rem;'})

rank = div1.get\_text().strip()

singleUniv.append(rank.split(' ')[0])

h3 = div.find('h3')

singleUniv.append(h3.get\_text().strip())

ldiv = div.find\_all('div',{'style':'padding-right: 0.5rem;'})

singleUniv.append(ldiv[0].strong.string)

singleUniv.append(ldiv[1].strong.string)

allUniv.append(singleUniv)

def printUnivList(allUniv):

print("{:<6}{:<20}{:<6}{:<10}".format("排名","学校名称","学费","培养规模"))

for u in allUniv:

print("{:<6}{:<20}{:<10}{:<10}".format(u[0],u[1],u[2],u[3]))

def main(num):

allUniv = []

url = 'https://www.usnews.com/best-colleges/rankings/national-universities'

for i in range(1, num+1):

ri = url + '?\_page=' + str(i)

html = getHTMLText(ri)

soup = BeautifulSoup(html, 'html.parser')

fillUnivList(soup, allUniv)

printUnivList(allUniv)

main(4)

3. 分析百度图片搜索返回结果的HTML代码，编写爬虫抓取图片并下载形成专题图片库。

提示：

1）下载网页上的图片代码如下函数：imgUrl是图片的网址；destUrl是存储在本地的地址；fname是图片保存时的名字，默认值为空，图片保存时按网络上图片的名字保存

def downloadImageFile(imgUrl, destUrl, fname=''):

local\_filename = imgUrl.split('/')[-1]

print('Download Image File={}'.format(local\_filename))

try:

r = requests.get(imgUrl, stream=True)

r.raise\_for\_status()

if len(fname) == 0:

fname = local\_filename

print('fname={}'.format(fname))

with open(destUrl + "/" + fname, 'wb') as f:

for chunk in r.iter\_content(chunk\_size=1024):

if chunk:

f.write(chunk)

f.flush()

f.close()

return r.status\_code

except:

return r.status\_code

2）注意分析百度图片的搜索结果，图片的结果是保存在json格式中，关键字为thumbURL所对应的值就是图片的网址。

3）百度图片返回的搜索结果只有30幅图像，只有当鼠标下移才能看到后面30张图像，依此类推。分析HTML，找到相关网址下载后30张图像。

4）选择一个自己喜欢的明星，下载TA的90张照片。

**编写代码：**

import requests

import json

import re

import os

index = 1

def downloadImageFile(imgUrl, destUrl, fname=''):

    local\_filename = imgUrl.split('/')[-1]

    print('Download Image File={}'.format(local\_filename))

    try:

        r = requests.get(imgUrl, stream=True)

        r.raise\_for\_status()

        if len(fname) == 0:

            fname = local\_filename

        print('fname={}'.format(fname))

        with open(destUrl + "/" + fname, 'wb') as f:

            for chunk in r.iter\_content(chunk\_size=1024):

                if chunk:

                    f.write(chunk)

                    f.flush()

            f.close()

        return r.status\_code

    except:

        return r.status\_code

def download(urls,path):

    global index

    for url in urls:

        print("Download Image from page:{}".format(url))

        status = downloadImageFile(url,path,str(index)+".jpg")

        try:

            if str(status)[0] == '4':

                print("未下载成功{}".format(url))

                continue

        except Exception as e:

          print("未下载成功{}".format(url))

        index += 1

pagestr = '<https://image.baidu.com/search/acjson?tn=resultjson_com&ipn=rj>'\

       '&ct=

        '&copyright=&ie=utf-8&oe=utf-8&adpicid=&st=&z=&ic='\

        '&word={}&s=&se=&tab=&width=&height=&face=&istype='\

        '&qc=&nc=&fr=&pn={}&rn=30&gsm=1e&

for i in range(1,11):

    page = pagestr.format('海贼王','海贼王',i\*30)

    print(page)

    try:

        rsp = requests.get(page,timeout=10)

        rsp.raise\_for\_status()

    except:

        print('对不起，百度图片访问失败！程序退出')

    imgdata=json.loads(rsp.text)

    imgs = imgdata['data']

    urls=[]

    for im in imgs:

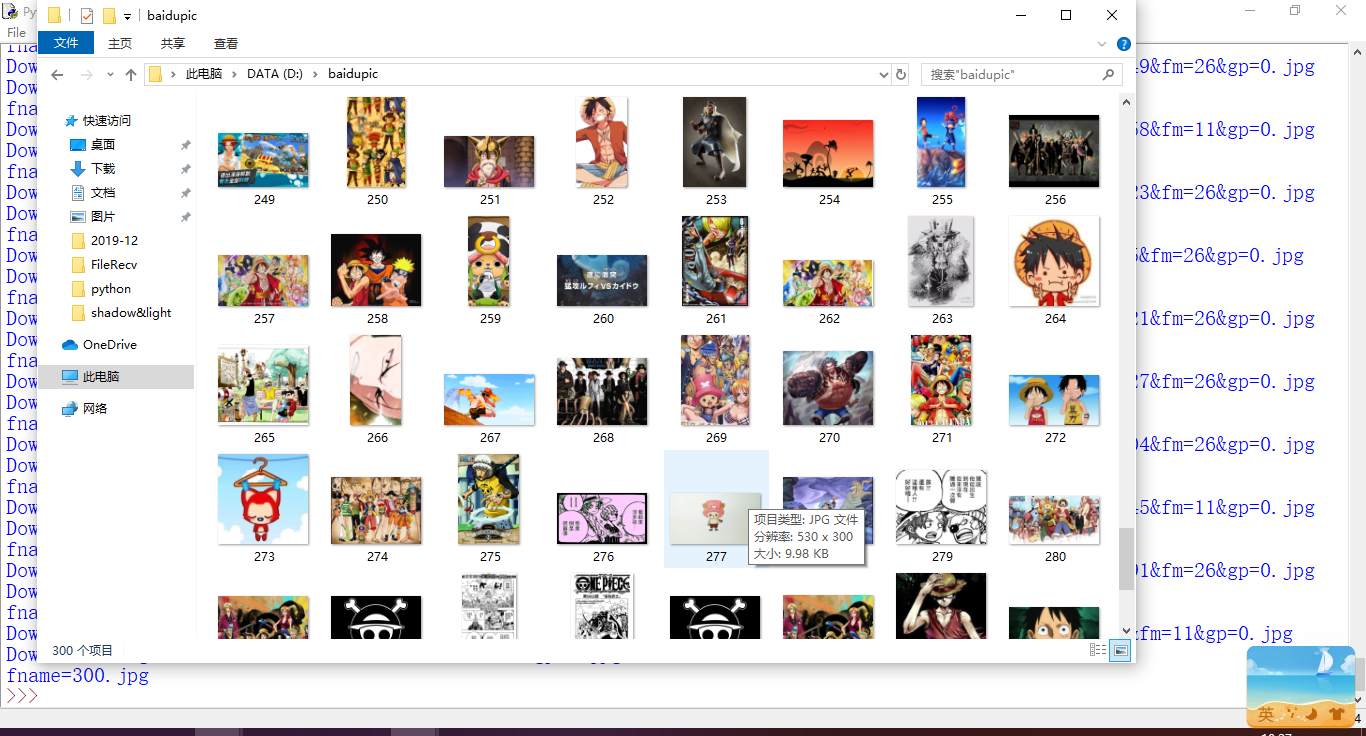
        url = im.get('thumbURL')

        print(url)

        if url is not None:

            urls.append(im.get('thumbURL'))

    download(urls, 'd:/baidupic')



## 实验思考：

网络爬虫应用一般分为两个步骤：1.通过网络链接获取网页内容；2.对获得的网页内容进行处理。这两个步骤分别使用不同的函数库：request和beautifulsoup4.

采用pip或pip3安装requests库和beautifulsoup4库

pip3 install requests

pip3 install beautifulsoup4

网络爬虫，相对于前面实验的基础理论知识，其更加贴近生活和实际应用。例如可以爬取试卷信息、获取网络的数据。

通过实验，我认为在进行网络爬虫操作之前，必须先得学会读取HTML语言，能根据参数找到对应标签，返回列表类型。 例如在爬取图片程序题过程中，找到thumbURL和acjson?中pn={}是关键。