**实验报告**

姓名：周蓓 学号：117060400109 指导老师：林卫中

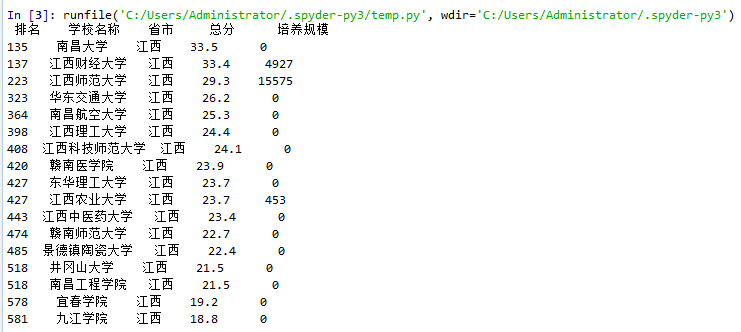
实验名称：爬虫实验指导

实验要求：熟练运用requests和beautifulsoup4函数库爬取数据，利用百度搜索提供的这个链接接口，可以通过requests的get（）函数提交查询，响应结果为百度搜索结果。

实验内容：1、按照省份输出中国大学排名，输出江西省的高校排名。代码如下：

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("{:^4}{:^10}{:^5}{:^8}{:^10}".format("排名","学校名称","省份","总分","培养规模"))  
    for u in allUniv:  
        if province in u[2]:  
            print("{:^4}{:^10}{:^5}{:^8}{:^10}".format(u[0],u[1],u[2],u[3],u[4]))                        
def main(p):  
    url = 'IMG_256http://www.zuihaodaxue.cn/zuihaodaxuepaiming2018.html'  
    html = getHTMLText(url)  
    soup = BeautifulSoup(html, "html.parser")  
    fillUnivList(soup)  
    printUnivList(p)  
main('江西')

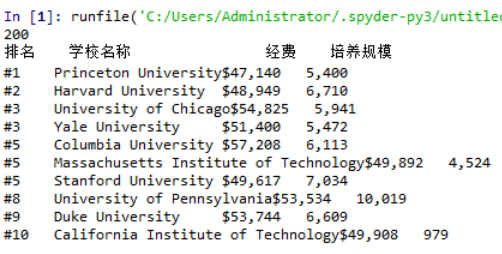
运行结果：



1. USNEWS美国大学排名爬虫。由于该网站有反爬虫机制，需要模拟浏览器行为进行爬虫。代码如下：

import requests  
import re  
from bs4 import BeautifulSoup  
allUniv=[]  
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):  
    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])  
        div2 =div.find('h3')  
        singleUniv.append(div2.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():  
    print("{:<6}{:<20}{:<6}{:<10}".format("排名","学校名称","经费","培养规模"))  
    for u in allUniv:  
        print("{:<6}{:<20}{:<10}{:<10}".format(u[0],u[1],u[2],u[3]))  
def main():  
    url = 'https://www.usnews.com/best-colleges/rankings/national-universities'  
    html = getHTMLText(url)  
    soup = BeautifulSoup(html,'html.parser')  
    fillUnivList(soup)  
    printUnivList()  
main()

运行结果：



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。

from NetSpider import \*  
from bs4 import BeautifulSoup  
import re  
  
def getHTMLText(url,coding='gbk'):  
    try:  
        r = requests.get(url,timeout=30)  
        print(r)  
        r.raise\_for\_status()  
        r.encoding = coding  
        return r.text  
    except:  
        return ""  
  
  
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 getImg(html):  
    imgre = re.compile('"objURL":"(.\*?)"')  
    imglist = re.findall(imgre,html)  
    return imglist  
  
def download(urls,path):  
    index = 1  
    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  
  
page = ''  
html= getHTMLText(page,'utf-8')  
download(getImg(html),'FileURL')

实验总结：本次实践课，继续学习网络爬虫，进一步了解爬虫，分析百度源代码，爬取有用的信息，方便统计查找所需的信息。在操作上需要细心，在源代码查找不正确的话，会导致程序错误。