上机实验报告（12）

|  |  |
| --- | --- |
| 姓名：邱文静 | 学号：117060400119 |
| 班级：17应统(1)班 | 指导老师：林卫中老师 |

# 实验名称：网络爬虫巩固练习

# 实验目的：

* 掌握网络爬虫的基本方法
* 运用requests库编写基本URL访问过程
* 运用baeutifulsoup4库解析和处理HTML
* 掌握向搜索引擎自动提交关键词并获取返回结果的方法

# 实验过程及结果

1. 按照省份输出中国大学排名，请分别输出江西省和山东省的高校排名。大学排名网址：

<http://www.zuihaodaxue.cn/zuihaodaxuepaiming2018.html>

1. 按江西省份的高校排名

**源代码：**

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("{:^4}{:^10}{:^5}{:^8}{:^10}".format("排名","学校名称","省市","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[6]))

def main(p):

url = 'http://www.zuihaodaxue.cn/zuihaodaxuepaiming2016.html'

html = getHTMLText(url)

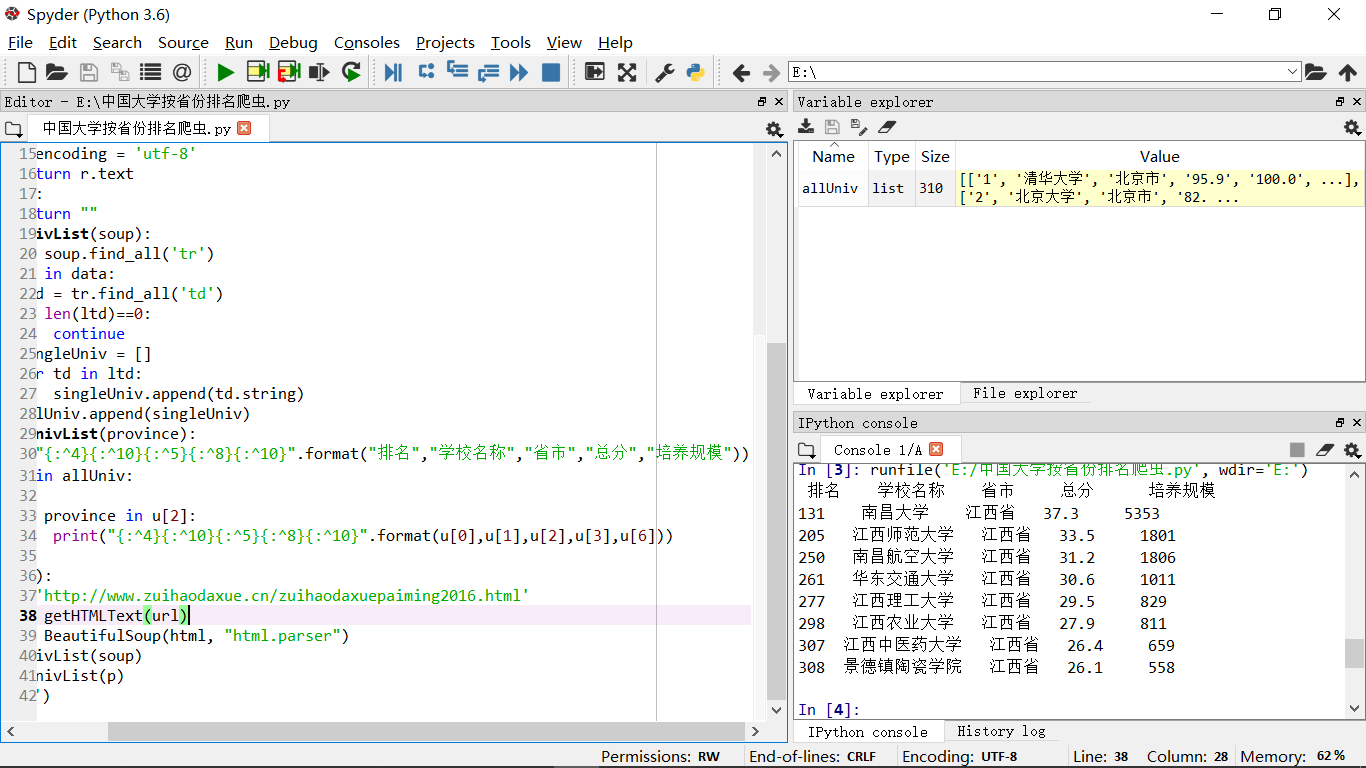
soup = BeautifulSoup(html, "html.parser")

fillUnivList(soup)

printUnivList(p)

main('江西')

**执行结果：**



1. 按山东省的高校排名

**源代码：**

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("{:^4}{:^10}{:^5}{:^8}{:^10}".format("排名","学校名称","省市","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[6]))

def main(p):

url = 'http://www.zuihaodaxue.cn/zuihaodaxuepaiming2016.html'

html = getHTMLText(url)

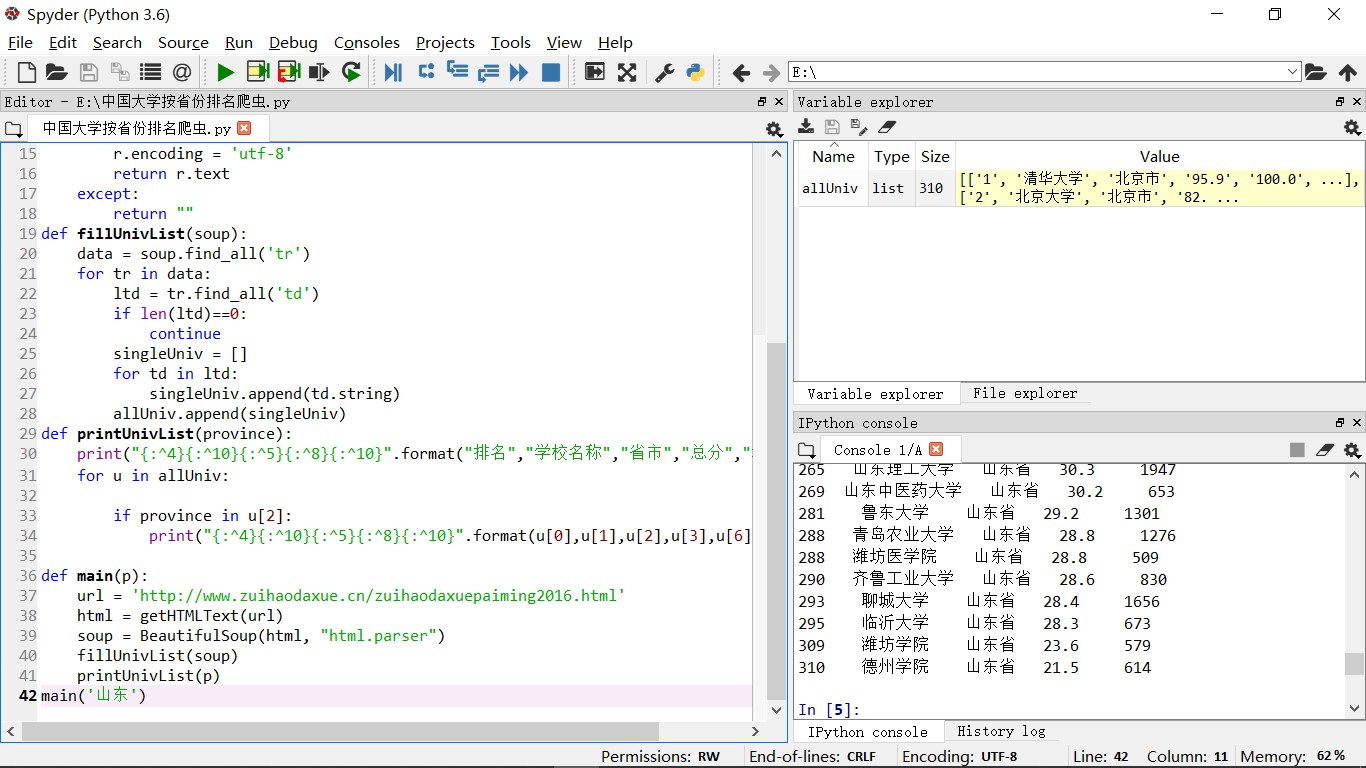
soup = BeautifulSoup(html, "html.parser")

fillUnivList(soup)

printUnivList(p)

main('山东')

**执行结果：**



1. USNEWS美国大学排名爬虫。

美国大学排名网址如下： https://www.usnews.com/best-colleges/rankings/national-universities注意：由于该网站有反爬虫机制，需要模拟浏览器行为进行爬虫，参考代码如下：

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 ""

**源代码：**

import requests

from bs4 import BeautifulSoup

import re

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())

lstrong = div.find\_all('strong')

singleUniv.append(lstrong[0].string)

singleUniv.append(lstrong[1].string)

allUniv.append(singleUniv)

def printUnivList():

print("{:<6}{:<40}{:<10}{:<10}".format("排名","学校","学费","培养规模"))

for u in allUniv:

print("{:<6}{:<40}{:<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)

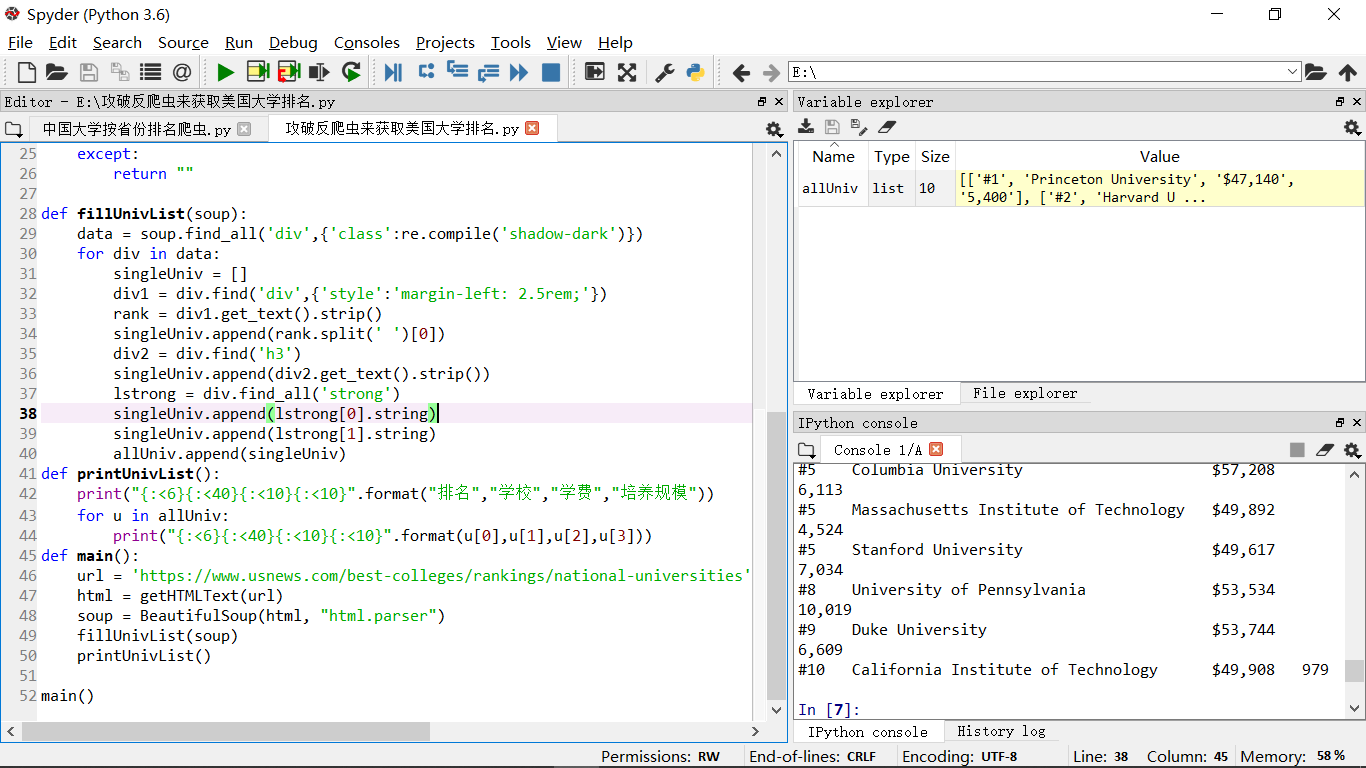
soup = BeautifulSoup(html, "html.parser")

fillUnivList(soup)

printUnivList()

main()

**执行结果：**



1. 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。

参考代码如下：

#下载图片

#imgUrl: 图片的URL

#destUrl: 图片保存的本地文件夹名字

#fname: 图片保存到本地时的另存名，如果为空，则默认用原来的文件名

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

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

爬取范冰冰图片

**源代码：**

import requests

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))

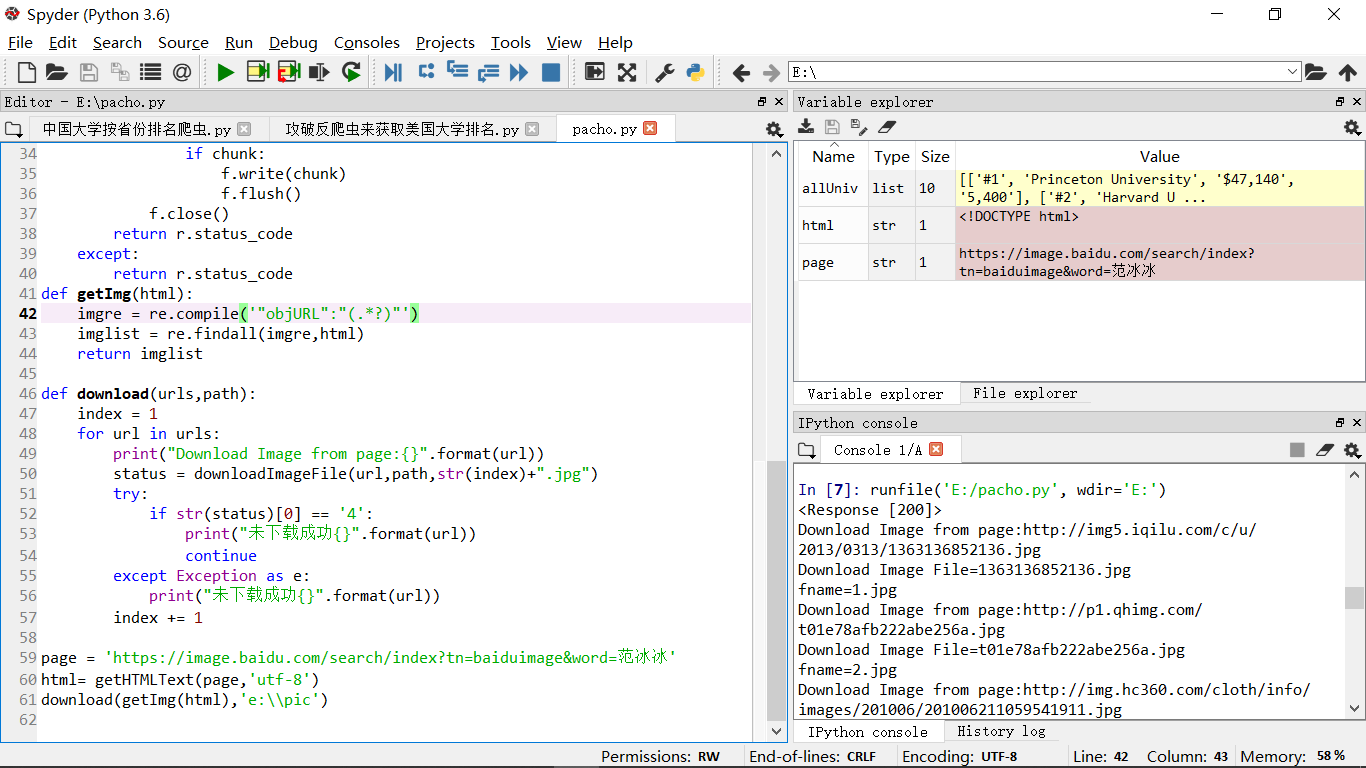
index += 1

page = 'https://image.baidu.com/search/index?tn=baiduimage&word=范冰冰'

html= getHTMLText(page,'utf-8')

download(getImg(html),'e:\\pic')

**执行结果：**



# 实验心得：

说实话，最近关于爬虫学的很累。上课一定要认真听讲，要有耐心，多自己去练习爬虫，加油！一定能学会爬虫的！并尝试学好！