**数据结构实验报告11**

**学号：** 117060400104 **姓名**：熊骏 **班级：**应用统计1班

**指导老师：** 林卫中

**实验要求：**网络爬虫

**实验题目：程序练习题**

**算法实现：**

**一**

**import requests**

**def getHTMLText(url):**

**try:**

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

**r.raise\_for\_status()**

**r.encoding = 'utf-8'**

**return r.text**

**except:**

**return ""**

**text = getHTMLText('http://www.sina.com.cn')**

**print(text)**

二

#e23.1CrawUnivRanking.py

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(num):

print("{:^4}{:^10}{:^5}{:^8}{:^10}".format("排名","学校名称","省市","总分","培养规模"))

for i in range(num):

u=allUniv[i]

print("{:^4}{:^10}{:^5}{:^8}{:^10}".format(u[0],u[1],u[2],u[3],u[6]))

def main():

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

html = getHTMLText(url)

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

fillUnivList(soup)

printUnivList(10)

main()

三

html\_doc = """

<html><head><title>The Dormouse's story</title></head>

<body>

<p class="title"><b>The Dormouse's story</b></p>

<p class="story">Once upon a time there were three little sisters; and their names were

<a href="http://example.com/elsie" class="sister" id="link1">Elsie</a>,

<a href="http://example.com/lacie" class="sister" id="link2">Lacie</a> and

<a href="http://example.com/tillie" class="sister" id="link3">Tillie</a>;

and they lived at the bottom of a well.</p>

<p class="story">...</p>

"""

**四**

html\_doc = """

<html><head><title>The Dormouse's story</title></head>

<body>

<p class="title"><b>The Dormouse's story</b></p>

<p class="story">Once upon a time there were three little sisters; and their names were

<a href="http://example.com/elsie" class="sister" id="link1">Elsie</a>,

<a href="http://example.com/lacie" class="sister" id="link2">Lacie</a> and

<a href="http://example.com/tillie" class="sister" id="link3">Tillie</a>;

and they lived at the bottom of a well.</p>

<p class="story">...</p>

“""

五

import requests

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):

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

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

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

with open(destUrl + "/" + local\_filename, '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

def downloadImageFile(imgUrl, destUrl, fname):

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

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

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

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

六

from PIL import Image

import numpy as np

vec\_el = np.pi/2.2

vec\_az = np.pi/4.

depth = 50.

im = Image.open('t012c852c8f981d6a00.jpg').convert('L')

a = np.asarray(im).astype('float')

grad = np.gradient(a)

grad\_x, grad\_y = grad

grad\_x =grad\_x\*depth/100

grad\_y =grad\_y\*depth/100

dx = np.cos(vec\_el)\*np.cos(vec\_az)

dy = np.cos(vec\_el)\*np.sin(vec\_az)

dz = np.sin(vec\_el)

A = np.sqrt(grad\_x\*\*2 + grad\_y\*\*2 + 1.)

uni\_x = grad\_x/A

uni\_y = grad\_y/A

uni\_z = 1./A

a2 = 255\*(dx\*uni\_x + dy\*uni\_y + dz\*uni\_z)

a2 = a2.clip(0,255)

im2 =Image.fromarray(a2.astype('uint8'))

im2.save(‘fcityHandDraw.jpy')

七

import numpy as np

import matplotlib.pyplot as plt

def squareWave(x,n):

f = np.zeros((x,shape[0],))

k = 1

while k <= n:

f =f + (8\*np.sin((2\*k-1)\*x)/((2\*k-1)\*np.pi))

k = k + 1

return f

x = np.linspace(0.0,2\*np.pi,100)

y = squareWave(x,8)

plt.plot(x,y)

plt.show()