

Python程序语言入门与应用



python

Life is short, use Python
人生苦短，我用Python



新乡医学院

Python程序语言入门与应用

第十章 Python 爬虫与Web开发

王海蛟

新乡医学院





下周课程&课后作业



第9章 科学计算和可视化

课后练习

 修改实例19，使得效果更佳符合您自己的审美。

代码文件命名：9-1shouhui

 绘制一个自己感兴趣的数学或物理规律图形。

代码文件命名：9-2matplotlib

.py代码文件打包(9.学号+
姓名)发送到
python_xxmu@163.com



下周课程&课后作业



课后练习

代码文件命名：9-1shouhui



修改实例19，使得效果更佳符合您自己的审美。

```
实例19.py > ...
1  #19HandDrawPic.py
2  from PIL import Image
3  import numpy as np
4  vec_el = np.pi/2.2 # 光源的俯视角度，弧度值
5  vec_az = np.pi/4. # 光源的方位角度，弧度值
6  depth = 10. # (0-100)
7  im = Image.open('fcity.jpg').convert('L')
8  a = np.asarray(im).astype('float')
9  grad = np.gradient(a) # 取图像灰度的梯度值
10 grad_x, grad_y = grad # 分别取横纵图像梯度值
11 grad_x = grad_x*depth/100.
12 grad_y = grad_y*depth/100.
13 dx = np.cos(vec_el)*np.cos(vec_az) # 光源对x-轴的影响
14 dy = np.cos(vec_el)*np.sin(vec_az) # 光源对y-轴的影响
15 dz = np.sin(vec_el) # 光源对z-轴的影响
16 A = np.sqrt(grad_x**2 + grad_y**2 + 1.)
17 uni_x = grad_x/A
18 uni_y = grad_y/A
19 uni_z = 1./A
20 a2 = 255*(dx*uni_x + dy*uni_y + dz*uni_z) # 光源归一化
21 a2 = a2.clip(0,255)
22 im2 = Image.fromarray(a2.astype('uint8')) # 重构图像
23 im2.save('fcityHandDraw.jpg')
24
```



下周课程&课后作业



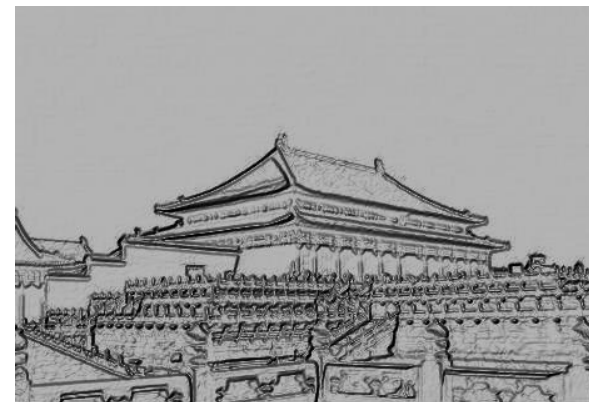
课后练习

代码文件命名：9-1shouhui



修改实例19，使得效果更佳符合您自己的审美。

```
实例19.py > ...
1  #19HandDrawPic.py
2  from PIL import Image
3  import numpy as np
4  a=eval(input('请输入虚拟光源的方位角度(单位:°): '))
5  b=eval(input('请输入虚拟光源的俯视角角度(单位:°): '))
6  vec_el = np.pi*b/180...#光源的方位角度,弧度值
7  vec_az = np.pi*a/180...#光源的俯视角角度,弧度值
8  depth = 10...#(0-100)
9  im = Image.open('fcity.jpg').convert('L')
10 a = np.asarray(im).astype('float')
11 grad = np.gradient(a) #取图像灰度的梯度值
12 grad_x, grad_y = grad #分别取横纵图像梯度值
13 grad_x = grad_x*depth/100.
14 grad_y = grad_y*depth/100.
15 dx = np.cos(vec_el)*np.cos(vec_az) #光源对x-轴的影响
16 dy = np.cos(vec_el)*np.sin(vec_az) #光源对y-轴的影响
17 dz = np.sin(vec_el) #光源对z-轴的影响
18 A = np.sqrt(grad_x**2 + grad_y**2 + 2.)
19 uni_x = grad_x/A
20 uni_y = grad_y/A
21 uni_z = 1./A
22 a2 = 255*(dx*uni_x + dy*uni_y + dz*uni_z) #光源归一化
23 a2 = a2.clip(0,255)
24 im2 = Image.fromarray(a2.astype('uint8')) #重构图像
25 im2.save('fcityHandDraw.jpg')
26
```





下周课程&课后作业



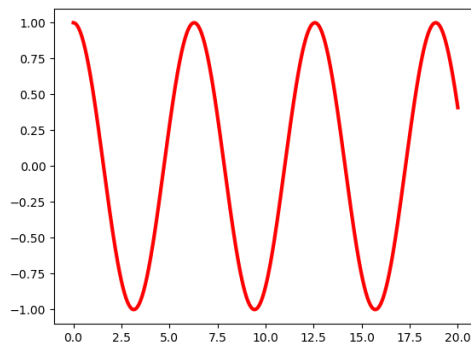
课后练习

代码文件命名：9-2matplotlib



绘制一个自己感兴趣的数学或物理规律图形。

```
2.py > ...  
1  import numpy as np  
2  import matplotlib.pyplot as plt  
3  x = np.linspace(0, 20, 1000)  
4  y = np.cos(x)  
5  plt.plot(x, y, 'k', color='r', linewidth=3, linestyle="-")  
6  plt.show()  
7
```





新乡医学院

Python程序语言入门与应用

第十章 Python 爬虫与Web开发

王海蛟

新乡医学院



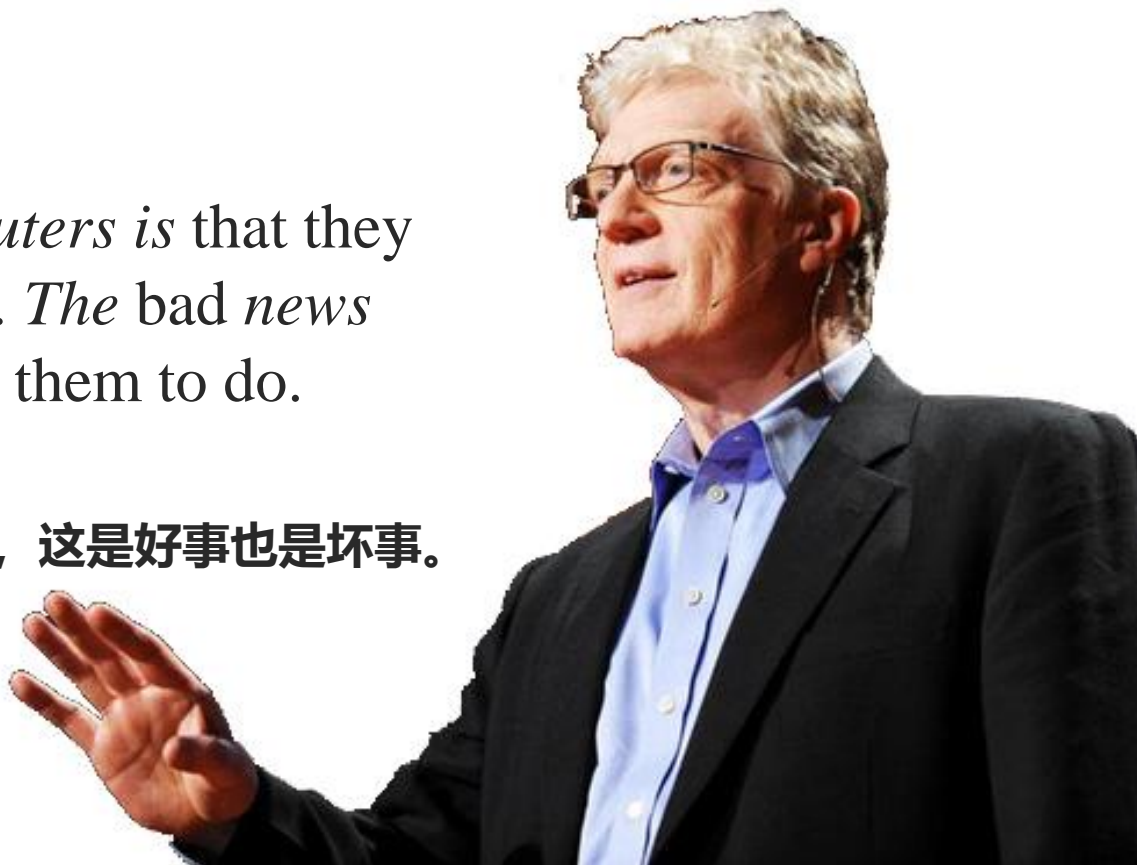


学习目标



The good news about computers is that they do what you tell them to do. The bad news is that they do what you tell them to do.

计算机按照你给出的指令去执行，这是好事也是坏事。



Ted Nelson, 1937年出生于纽约，美国人，网传比特币创始人。



本课概要



🔗 第10章 Python 爬虫与Web开发

🔗 10.1 网络爬虫概述

🔗 10.2 模块10: requests库的使用

🔗 10.3 模块11: beautifulsoup4库的使用

🔗 10.4 实例24: 中国大学排名爬虫

🔗 10.5 实例25: 搜索关键词自动提交



学习目标



基本要求

☞ 掌握

- ☞ 网络爬虫设计的基本方法。
- ☞ 运用requests库编写基本URL访问过程
- ☞ 运用beautifulsoup4库解析和处理HTML
- ☞ 向搜索引擎自动提交关键词并获取返回结果的方法

☞ 理解&了解

- ☞ requests库和beautifulsoup4库进行爬虫分析。



10.1 网络爬虫概述



10.1 网络爬虫概述

- ❖ 万维网的快速发展，带来了大量获取和提交网络信息的需求，这就产生了一系列“网络爬虫”应用程序。
- ❖ 美国的谷歌公司在搜索引擎后端采用python语言进行链接处理和开发，这是该语言成熟的重要标志。
- ❖ Python语言的简洁性和脚本特性非常适合连接和网络处理。
- ❖ Python语言提供了很多类似的函数库，比如urllib，urllib2、urllib3、wget、scrapy、requests等。这些库作用不同，使用方式也不同，用户体验也不同。
- ❖ 对于网络爬虫，可以通过request、beautifulaoup4等函数库来处理。



10.1 网络爬虫概述



10.1 网络爬虫概述



网络爬虫一般分为两个步骤：

(1) 通过网络连接获取网页内容

(2) 对获得的网页内容进行处理。

这两个步骤分别使用requests库和beautifulsoup4库进行处理。



10.1 网络爬虫概述



10.1 网络爬虫概述

 Requests库和beautifulsoup4库的安装:

pip install requests
or
pip3 install requests



10.1 网络爬虫概述



10.1 网络爬虫概述



Python语言实现网络爬虫和信息提交是非常简单的事情，无需学习者知道网络底层的专业知识。但是，肆意爬去网络信息，特别是一些竞争性信息是非常不文明的现象。就像推销电话一样，无视接听者意愿。



为了规范爬虫行为，互联网设计了Robots爬虫协议。网站管理员可以在网站根目录放置一个robots.txt文件，里面列出那些连接不允许爬虫。该协议并非强制协议，但一般搜索引擎都会遵守。



10.2模块13request库的使用



10.1request库概述

- 这个库建立在Python 语言的urllib3 库基础上，类似这种在其他函数库之上再封装功能提供更友好函数的方式在Python 语言中十分常见。在Python 的生态圈里，任何人都有通过技术创新或体验创新发表意见和展示才华的机会。



10.2模块13requests库的使用



10.1requests库概述

- request 库支持非常丰富的链接访问功能，包括：国际域名和URL 获取、HTTP 长连接和连接缓存、HTTP 会话和Cookie 保持、浏览器使用风格的SSL 验证、基本的摘要认证、有效的键值对Cookie 记录、自动解压缩、自动内容解码、文件分块上传、HTTP(S)代理功能、连接超时处理、流数据下载等。有关requests 库的更多介绍请访问：
- <http://docs.python-requests.org>



10.2模块13requests库的使用



10.1requests库中的网页请求函数

函数	描述
<code>get(url [, timeout=n])</code>	对应于 HTTP 的 GET 方式，获取网页最常用的方法，可以增加 <code>timeout=n</code> 参数，设定每次请求超时时间为 n 秒
<code>post(url, data = {'key': 'value'})</code>	对应于 HTTP 的 POST 方式，其中字典用于传递客户数据
<code>delete(url)</code>	对应于 HTTP 的 DELETE 方式
<code>head(url)</code>	对应于 HTTP 的 HEAD 方式
<code>options(url)</code>	对应于 HTTP 的 OPTIONS 方式
<code>put(url, data = {'key': 'value'})</code>	对应于 HTTP 的 PUT 方式，其中字典用于传递客户数据



10.2模块13requests库的使用



10.1requests库中的网页请求函数

函数	描述
<code>get(url [, timeout=n])</code>	对应于 HTTP 的 GET 方式，获取网页最常用的方法，可以增加 <code>timeout=n</code> 参数，设定每次请求超时时间为 <code>n</code> 秒

Get()函数是获取网页最常用的方法，在调用requests.get()函数之后，返回的网页内容保存为一个Response对象。其中，get () 函数的参数url必须采用http或者https访问方式。例如：

```
>>>import requests

>>>r=requests.get("http://www.baidu.com") #使用 get 方法打开百度链接

>>>type(r)

<class 'requests.models.Response'>      #返回 Response 对象
```



10.2模块13requests库的使用



10.1requests库中的网页请求函数

函数	描述
<code>get(url [, timeout=n])</code>	对应于 HTTP 的 GET 方式，获取网页最常用的方法，可以增加 <code>timeout=n</code> 参数，设定每次请求超时时间为 n 秒

- 和浏览器的交互过程一样，`requests.get()`代表请求过程，它返回的`Response` 对象代表响应。返回内容作为一个对象更便于操作，`Response` 对象的属性如下表所示，需要采用`<a>.`形式使用。



10.2模块13requests库的使用



10.2 Response属性

属性	描述
status_code	HTTP 请求的返回状态，整数，200 表示连接成功，404 表示失败
text	HTTP 响应内容的字符串形式，即，也是 url 对应的页面内容
encoding	HTTP 响应内容的编码方式
content	HTTP 响应内容的二进制形式



10.2 Response属性

21



10.2模块13requests库的使用



10.2 Response对象方法

方法	描述
<code>json()</code>	如果 HTTP 响应内容包含 JSON 格式数据，该方法解析 JSON 数据
<code>raise_for_status()</code>	如果不是 200，那么这个方法就会产生异常

`Json()`方法能够在http响应内容中解析存在的JSON数据，这将为HTTP的解析带来便利。

- `raise_for_status()`方法能在非成功响应后产生异常，即只要返回的请求状态`status_code` 不是200，这个方法会产生一个异常，用于`try...except` 语句。使用异常处理语句可以避免设置一堆复杂的`if` 语句，只需要在收到响应调用这个方法，就可以避开状态字200 以外的各种意外情况。



10.2模块13requests库的使用



10.2 Response对象方法

方法	描述
<code>json()</code>	如果 HTTP 响应内容包含 JSON 格式数据，该方法解析 JSON 数据
<code>raise_for_status()</code>	如果不是 200，那么这个方法就会产生异常

- requests 会产生几种常用异常。当遇到网络问题时，如：DNS 查询失败、拒绝连接等，requests 会抛出 `ConnectionError` 异常；遇到无效HTTP 响应时，requests 则会抛出 `HTTPError` 异常；若请求url 超时，则抛出 `Timeout` 异常；若请求超过了设定的最大重定向次数，则会抛出一个 `TooManyRedirects` 异常



10.2模块13requests库的使用



10.2 Response对象方法

例如通过如下代码获取一个网页内容：

```
1  import requests
2  def getHTMLText():
3      try:
4          r = requests.get(url, timeout=30)
5          r.raise_for_status() #如果状态不是 200, 引发异常
6          r.encoding = 'utf-8' #无论原来用什么编码, 都改成 utf-8
7          return r.text
8      except:
9          return ""
10 url = "http://www.baidu.com"
11 print(getHTMLText(url))
```



10.2模块13requests库的使用



10.2 HTTP的get和post

- HTTP 协议定义了客户端与服务器交互的不同方法，最基本的方法是GET 和POST。顾名思义，GET 可以根据某链接获得内容，POST 用于发送内容。然而，GET 也可以向链接提交内容
- 1) GET 方式可以通过URL 提交数据，待提交数据是URL 的一部分；采用POST 方式，待提交数据放置在HTML HEADER 内；



10.2模块13requests库的使用



10.2 HTTP的get和post




- 2) GET 方式提交的数据最多不超过1024 字节，POST 没有对提交内容的长度限制。
- 3) 安全性问题。如(1)所述，使用GET 时参数会显示在URL 中，而POST不会。所以，如果这些数据是非敏感数据，那么使用GET；如果提交数据是敏感数据，建议采用POST 方式。



10.3模块14beautifulsoup4库的使用



10.3 beautifulsoup4库概述

-  BeautifulSoup4库是一个解析和处理HTML和XML的第三方库。
-  Requests库获取HTML页面并将其转换成字符串后，需要进一步解析HTML页面的格式，提取有用的信息，这就需要处理HTML和XML的函数库。
-  BeautifulSoup4库将专业的HTML格式解析部分封装为函数，提供了很多简单有用的网页处理函数。



10.3模块14beautifulsoup4库的使用



10.3 beautifulsoup4库概述

- 在使用beautifulsoup4 库之前，需要进行引用，由于这个库的名字非常特殊且采用面向对象方式组织，可以用 from...import 方式从库中直接引用 BeautifulSoup 类，方法如下。

```
>>>from bs4 import BeautifulSoup
```



10.3模块14beautifulsoup4库的使用



10.3 beautifulsoup4库概述

- beautifulsoup4 库中最主要的是BeautifulSoup 类，每个实例化的对象相当于一个页面。采用 from...import 导入库中类后，使用BeautifulSoup() 创建一个BeautifulSoup对象。



10.3模块14beautifulsoup4库的使用



10.3 beautifulsoup4库用法

```
>>>
>>> import requests
>>> from bs4 import BeautifulSoup
>>> r = requests.get('http://www.baidu.com')
>>> k=r.text.encode(r.encoding).decode('utf-8')
>>> soup=BeautifulSoup(k)
>>> type(soup)
<class 'bs4.BeautifulSoup'>
>>>
```

创建了一个beautifulsoup类对象



10.3模块14beautifulsoup4库的使用



10.3 beautifulsoup4库用法

- 创建的BeautifulSoup 对象是一个树形结构，它包含HTML 页面里的每一个Tag（标签）元素，如<head>、<body>等。具体来说，HTML 中的主要结构都变成了BeautifulSoup 对象的一个属性，可以直接用<a>.形式获得，其中的名字采用HTML 中标签的名字。



10.3模块14beautifulsoup4库的使用



10.3 BeautifulSoup对象的属性

属性	描述
head	HTML 页面的<head>内容
title	HTML 页面标题，在<head>之中，由<title>标记
body	HTML 页面的<body>内容
p	HTML 页面中第一个<p>内容
strings	HTML 页面所有呈现在 Web 上的字符串，即标签的内容
stripped_strings	HTML 页面所有呈现在 Web 上的非空格字符串



10.3模块14beautifulsoup4库的使用



10.3 BeautifulSoup对象的属性

```
>>> soup.head
<head><meta content="text/html; charset=utf-8" http-equiv="content-type"/><meta content="IE=Edge" http-equiv="X-UA-Compatible"/><meta content="always" name="referrer"/><link href="http://sl.bdstatic.com/r/www/cache/bdorz/baidu.min.css" rel="stylesheet" type="text/css"/><title>百度一下，你就知道</title></head>
>>> title=soup.title
>>> title
<title>百度一下，你就知道</title>
>>> type(title)
<class 'bs4.element.Tag'>
>>> soup.p
<p id="lh"> <a href="http://home.baidu.com">关于百度</a> <a href="http://ir.baidu.com">About Baidu</a> </p>
>>>
```



10.3模块14beautifulsoup4库的使用



10.3 beautifulsoup对象的属性

- 每一个Tag 标签在beautifulsoup4 库中也是一个对象，称为Tag 对象。上例中，title 是一个标签对象。每个标签对象在HTML 中都有类似的结构：

```
<a class="mnav" href="http://www.nuomi.com">糯米</a>
```



10.3模块14beautifulsoup4库的使用



10.3 beautifulsoup对象的属性

- 其中，尖括号（<>）中的标签的名字是name，尖括号内其他项是attrs，尖括号之间的内容是string。
因此，可以通过Tag 对象的name、 attrs 和string 属性获得相应内容，采用<a>.的语法形式。
- 标签Tag 有4 个常用属性



10.3模块14beautifulsoup4库的使用



10.3 标签对象的常用属性

属性	描述
name	字符串，标签的名字，比如 div
attrs	字典，包含了原来页面 Tag 所有的属性，比如 href
contents	列表，这个 Tag 下所有子 Tag 的内容
string	字符串，Tag 所包围的文本，网页中真实的文字



10.3模块14beautifulsoup4库的使用



10.3 标签对象的常用属性

```
>>> title
<title>百度一下，你就知道</title>
```

```
>>> p
<p id="lh"> <a href="http://home.baidu.com">关于百度</a> <a href="http://ir.baidu.com">About Baidu</a> </p>
```

```
>>> title.name
'title'
>>> title.attrs
{'id': 'lh'}
>>> title.contents
['百度一下，你就知道']
>>> title.string
'百度一下，你就知道'
>>>
>>> p=soup.p
>>> p.name
'p'
>>> p.attrs
{'id': 'lh'}
>>> p.contents
['', <a href="http://home.baidu.com">关于百度</a>, '', <a href="http://ir.baidu.com">About Baidu</a>, '']
>>> p.string
>>> print(p.string)
SyntaxError: invalid syntax
>>> print(p.string)
None
>>>
```



10.3模块14beautifulsoup4库的使用



10.3 标签对象的常用属性

- 由于HTML 语法可以在标签中嵌套其他标签，所以，string 属性的返回值遵循如下原则：
 - 如果标签内部没有其他标签，string 属性返回其中的内容；
 - 如果标签内部有其他标签，但只有一个标签，string 属性返回最里面标签的内容；
 - 如果标签内部有超过1 层嵌套的标签，string 属性返回None（空字符串）。



10.3模块14beautifulsoup4库的使用



10.3 标签对象的常用属性

- HTML 语法中同一个标签会有很多内容，例如<a> 标签，百度首页一共有13 处，直接调用soup.a 只能返回第一个。
- 当需要列出标签对应的所有内容或者需要找到非第一个标签时，需要用到BeautifulSoup 的find()和find_all()方法。这两个方法会遍历整个HTML 文档，按照条件返回标签内容。



10.3模块14beautifulsoup4库的使用



10.3 标签对象的常用属性

`BeautifulSoup.find_all(name, attrs, recursive, string, limit)`

作用：根据参数找到对应标签，返回列表类型。

参数：

name：按照 Tag 标签名字检索，名字用字符串形式表示，例如：div, li;

attrs：按照 Tag 标签属性值检索，需要列出属性名称和值，采用 JSON 表示；

recursive：设置查找层次，只查找当前标签下一层时使用 recursive=False;

string：按照关键字检索 string 属性内容，采用 string=开始;

limit：返回结果的个数，默认返回全部结果。



10.3模块14beautifulsoup4库的使用



10.3 标签对象的常用属性

```

>>> a= soup.find_all('a')
>>> len(a)
11
>>> a
[<a class="mnav" href="http://news.baidu.com" name="tj_trnews">新闻</a>, <a class="mnav" href="http://www.hao123.com" name="tj_trhao123">hao123</a>, <a class="mnav" href="http://map.baidu.com" name="tj_trmap">地图</a>, <a class="mnav" href="http://v.baidu.com" name="tj_trvideo">视频</a>, <a class="mnav" href="http://tieba.baidu.com" name="tj_trtieba">贴吧</a>, <a class="lb" href="http://www.baidu.com/bdorz/login.gif?login&tpl=mn&u=http%3A%2F%2Fwww.baidu.com%2f%3fbdorz_come%3dl" name="tj_login">登录</a>, <a class="bri" href="//www.baidu.com/more/" name="tj_briicon" style="display: block;">更多产品</a>, <a href="http://home.baidu.com">关于百度</a>, <a href="http://ir.baidu.com">About Baidu</a>, <a href="http://www.baidu.com/duty/">使用百度前必读</a>, <a class="cp-feedback" href="http://jianyi.baidu.com/">意见反馈</a>]
```



10.4实例24 中国大学排名爬虫



10.4 中国大学排名爬虫

这里通过抓取上海交通大学研发的“软科中国最好大学排名2019”为例，编写一个大学排名爬虫，爬去名单上面的大学排名数据并打印。

www.zuihaodaxue.com/zuihaodaxuepaiming2019.html

常用网址 京东商城 天猫11.11

最好大学网

ZUIHAODAXUE.COM

网站首页 中国大学排名 世界大学排名 原创分析 要闻资讯 院校信息 会议

首页 / 中国大学排名 / 软科中国最好大学排名2019

软科中国最好大学排名2019

Powered by: Scopus

2019

“中国最好大学排名”的排名范围是教育部公布的全国普通高等学校名单中，1243所办学层次为本科的大学。这其中公办大学826所、民办大学153所、独立学院264所。

[查看排名方法](#)

排名	学校名称	省市	总分	指标得分
				生源质量（新生高考成绩得分）
1	清华大学	北京	94.6	100.0
2	北京大学	北京	76.5	95.2
3	浙江大学	浙江	72.9	84.2
4	上海交通大学	上海	72.1	91.1
5	复旦大学	上海	65.6	91.6
6	中国科学技术大学	安徽	60.9	91.1
7	华中科技大学	湖北	58.9	80.1
7	南京大学	江苏	58.9	86.2



10.4实例24 中国大学排名爬虫



10.4 中国大学排名爬虫

- 大学排名爬虫的构建需要三个重要步骤：
- 第一，从网络上获取网页内容；
- 第二，分析网页内容并提取有用数据到恰当的数据结构中；
- 第三，利用数据结构展示或进一步处理数据。
 - 由于大学排名是一个典型的二维数据，因此，采用二维列表存储该排名所涉及的表单数据。



10.4实例24 中国大学排名爬虫



10.4 中国大学排名爬虫

- 具体来说，采用requests 库爬取网页内容，使用beautifulsoup4 库分析网页中数据，提取310 个学校的排名及相关数据，存储到二维列表中，最后采用用户偏好的方式打印出来。



10.4实例24 中国大学排名爬虫



10.4 中国大学排名爬虫

- 为了解析网页上数据，首先需要程序编写者观察爬虫页面的特点，即找到拟获取数据在HTML 页面中的格式。打开大学排名页面，在浏览器菜单中选择“查看网页源代码”，该选项在所有浏览器中都存在，得到的HTML 源代码如下图所示（为了便于阅读，该源代码做过一定排版）



10.4实例24 中国大学排名爬虫



10.4 中国大学排名爬虫

```
<tbody class="hidden_zhpm" style="text-align:center;">
  <tr class="alt">
    <td>1</td><td><div align="left">清华大学</div></td> <td>北京市</td><td>95.9</td><td class="hidden-xs need-hidden indicator5">100.0</td><td class="hidden-xs need-hidden indicator6" style="display:none;">97.90%</td><td class="hidden-xs need-hidden indicator7" style="display:none;">37342</td><td class="hidden-xs need-hidden indicator8" style="display:none;">1.298</td><td class="hidden-xs need-hidden indicator9" style="display:none;">1177</td><td class="hidden-xs need-hidden indicator10" style="display:none;">109</td><td class="hidden-xs need-hidden indicator11" style="display:none;">1137711</td><td class="hidden-xs need-hidden indicator12" style="display:none;">1187</td><td class="hidden-xs need-hidden indicator13" style="display:none;">593522</td>
  </tr>
  <tr>
    <td>2</td><td><div align="left">北京大学</div></td> <td>北京市</td><td>82.6</td><td class="hidden-xs need-hidden indicator5">98.9</td><td class="hidden-xs need-hidden indicator6" style="display:none;">95.96%</td><td class="hidden-xs need-hidden indicator7" style="display:none;">36137</td><td class="hidden-xs need-hidden indicator8" style="display:none;">1.294</td><td class="hidden-xs need-hidden indicator9" style="display:none;">986</td><td class="hidden-xs need-hidden indicator10" style="display:none;">87</td><td class="hidden-xs need-hidden indicator11" style="display:none;">439403</td><td class="hidden-xs need-hidden indicator12" style="display:none;">799</td><td class="hidden-xs need-hidden indicator13" style="display:none;">7343</td>
  </tr>
  <tr class="alt">
    <td>3</td><td><div align="left">浙江大学</div></td> <td>浙江省</td><td>80</td><td class="hidden-xs need-hidden indicator5">88.8</td><td class="hidden-xs need-hidden indicator6" style="display:none;">96.46%</td><td class="hidden-xs need-hidden indicator7" style="display:none;">41188</td><td class="hidden-xs need-hidden indicator8" style="display:none;">1.059</td><td class="hidden-xs need-hidden indicator9" style="display:none;">803</td><td class="hidden-xs need-hidden indicator10" style="display:none;">86</td><td class="hidden-xs need-hidden indicator11" style="display:none;">959511</td><td class="hidden-xs need-hidden indicator12" style="display:none;">833</td><td class="hidden-xs need-hidden indicator13" style="display:none;">64392</td>
  </tr>
```




10.4实例24 中国大学排名爬虫



10.4 中国大学排名爬虫

- 对比两张图，每个大学排名的数据信息被封装在一个<tr></tr>之间的结构中。这是HTML 语言表示表格中一行的标签，在这行中，每列内容采用<td></td>表示。以“清华大学”为例，它对应一行信息的HTML 代码如下。



10.4实例24 中国大学排名爬虫



10.4 中国大学排名爬虫

```
<tr class="alt"><td>1</td><td><div align="left">清华大学</div></td><td>北京</td><td>94.6</td><td class="hidden-xs need-hidden indicator5">100.0</td><td class="hidden-xs need-hidden indicator6" style="display: none;">98.30%</td><td class="hidden-xs need-hidden indicator7" style="display: none;">1589319</td><td class="hidden-xs need-hidden indicator8" style="display: none;">48698</td><td class="hidden-xs need-hidden indicator9" style="display: none;">1.512</td><td class="hidden-xs need-hidden indicator10" style="display: none;">1810</td><td class="hidden-xs need-hidden indicator11" style="display: none;">126</td><td class="hidden-xs need-hidden indicator12" style="display: none;">1697330</td><td class="hidden-xs need-hidden indicator13" style="display: none;">302898</td><td class="hidden-xs need-hidden indicator14" style="display: none;">6.81%</td></tr>
```



10.4实例24 中国大学排名爬虫



10.4 中国大学排名爬虫

- 这个代码中每个td 标签包含大学排名表格的一个列数值，与表头一一对应。因此，如果要获得其中的数据，需要首先找到<tr> </tr> 标签，并遍历其中每个<td> </td> 标签，获取其值写入程序的数据结构中，这个代码封装成函数表示如下：



10.4实例24 中国大学排名爬虫



10.4 中国大学排名爬虫

```
1 allUniv=[] #存储全部表格数据, 二维列表
2 def fillUnivList(soup):
3     data = soup.find_all('tr') #找到所有 tr 标签
4     for tr in data:
5         singleUniv = []
6         ltd = tr.find_all('td') #在每个 tr 标签中找到所有 td 标签
7         for td in ltd:
8             singleUniv.append(td.string) #提取 td 标签中信息
9         allUniv.append(singleUniv)
```



10.4实例24 中国大学排名爬虫



10.4 中国大学排名爬虫

- 上述逻辑尽管不错，却不完全。HTML 页面中除了显示大学排名的地方，其他位置也可能有表格和<tr></tr>标签，应该尽量剔除这种情况。由于爬虫针对特定网页，程序编写也不必考虑所有情况，只要能应对当前页面即可。在这个大学排名页面中，还有一处用到了表格，包含<tr>标签，但这个标签内部不包括<td>标签。因此，可以通过增加一个判断语句剔除这种情况，观察下面代码的第6 行和第7 行。



10.4实例24 中国大学排名爬虫



10.4 中国大学排名爬虫

```
1 allUniv=[] #存储全部表格数据，二维列表
2 def fillUnivList(soup):
3     data = soup.find_all('tr') #找到所有 tr 标签
4     for tr in data:
5         ltd = tr.find_all('td') #在每个 tr 标签中找到所有 td 标签
6         if len(ltd)==0:
7             continue
8         singleUniv = []
9         for td in ltd:
10             singleUniv.append(td.string) #提取 td 标签中信息
11         allUniv.append(singleUniv)
```



10.4实例24 中国大学排名爬虫



10.4 中国大学排名爬虫

```
e: > 课程 > python > 第10次课 > 实例 > 24.py > ...
1  #e24CrawUnivRanking.py
2  import requests
3  from bs4 import BeautifulSoup
4  allUniv = []
5  def getHTMLText(url):
6      try:
7          r = requests.get(url, timeout=30)
8          r.raise_for_status()
9          r.encoding = 'utf-8'
10         return r.text
11     except:
12         return ""
13  def fillUnivList(soup):
14      data = soup.find_all('tr')
15      for tr in data:
16          ltd = tr.find_all('td')
17          if len(ltd)==0:
18              continue
19          singleUniv = []
20          for td in ltd:
21              singleUniv.append(td.string)
22          allUniv.append(singleUniv)
```



10.4实例24 中国大学排名爬虫



10.4 中国大学排名爬虫

```
23 def printUnivList(num):
24     print("{:^4}{:^10}{:^5}{:^8}".format("排名", "学校名称", "省市", "总分"))
25     for i in range(num):
26         u=allUniv[i]
27         print("{:^4}{:^10}{:^5}{:^8}".format(u[0],u[1],u[2],u[3]))
28 def main():
29     url = 'http://www.zuihaodaxue.com/zuihaodaxuepaiming2019.html'
30     html = getHTMLText(url)
31     soup = BeautifulSoup(html, "html.parser")
32     fillUnivList(soup)
33     printUnivList(10)
34 main()
```



10.4实例24 中国大学排名爬虫



10.4 中国大学排名爬虫

排名	学校名称	省市	总分
1	清华大学	北京	94.6
2	北京大学	北京	76.5
3	浙江大学	浙江	72.9
4	上海交通大学	上海	72.1
5	复旦大学	上海	65.6
6	中国科学技术大学	安徽	60.9
7	华中科技大学	湖北	58.9
7	南京大学	江苏	58.9
9	中山大学	广东	58.2
10	哈尔滨工业大学	黑龙江	56.7



10.5实例25搜索关键词自动提交



10.5 向搜索引擎提供关键词并获取返回结果

- 搜索引擎是日常工作常用的工具，也是访问互联网的门户。有时候需要自动向搜索引擎提交关键字并获得查询结果。
- 本节以百度为例介绍搜索关键字自动提交并获得返回结果的方法。



10.5实例25搜索关键词自动提交



10.5 向搜索引擎提供关键词并获取返回结果

- 百度搜索引擎首页为：<http://www.baidu.com>，当输入一个待查询关键词keyword时，百度程序将这个查询自动转换为链接：<http://www.baidu.com/s?wd=keyword>。读者可以在浏览器上手工输入这个链接，将keyword 换成任意想查询的关键字，都能获得查询结果。



10.5实例25搜索关键词自动提交



10.5 向搜索引擎提供关键词并获取返回结果

- 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。这个问题的IPO 描述如下：
 - 输入：待查询关键字
 - 处理：自动获得百度搜索结果页面，并对页面内容解析处理
 - 输出：返回链接的标题列表



10.5实例25搜索关键词自动提交



10.5 分析搜索引擎返回的结果

- 首先人工分析百度查询结果页面HTML 代码，部分片段参考下图。
- 由于这些HTML 代码由机器自动生成，可读性较差，需要对比网页上的搜索结果和代码仔细寻找。



10.5 分析搜索引擎返回的结果

60



10.5实例25搜索关键词自动提交



10.5 分析搜索引擎返回的结果

- 经过分析发现，页面上返回结果标题被封装在如下结构中：

- `<div...data-tools=`
`'{"title":"...", "url":"..."}'>...</div>`



10.5实例25搜索关键词自动提交



10.5 分析搜索引擎返回的结果

- 利用beautifulsoup4 库找到data-tools 属性值，提取带有title 的字符串，可以看到，data-tools 内部由{}形成的数据是典型的JSON 格式，可以用JSON 库将其转换成字典，便于操作。



10.5实例25搜索关键词自动提交



10.5 分析搜索引擎返回的结果

```
#e25.1AutoKeywordSearch.py
import requests
from bs4 import BeautifulSoup
import re
import json
def getKeywordResult(keyword):
    url = 'http://www.baidu.com/s?wd='+keyword
    try:
        r = requests.get(url, timeout=30)
        r.raise_for_status()
        r.encoding = 'utf-8'
        return r.text
    except:
        return ""
def parserLinks(html):
    soup = BeautifulSoup(html, "html.parser")
    links = []
    for div in soup.find_all('div', {'data-tools': re.compile('title')}):
        data = div.attrs['data-tools'] #获得属性值
        d = json.loads(data)           #将属性值转换成字典
        links.append(d['title'])        #将返回链接的题目返回
    return links
def main():
    html = getKeywordResult('Python语言程序设计基础(第2版)')
    ls = parserLinks(html)
    count = 1
    for i in ls:
        print("[{:~3}] {}".format(count, i))
        count += 1
main()
```



10.5实例25搜索关键词自动提交



10.5 分析搜索引擎返回的结果

```
===== RESTART: E:\课程\python\第10次课\案例\案例20.py =====  
=====  
[ 1 ]Python语言程序设计基础(第2版) 高清pdf扫描版[48MB] 电子书 下载-脚本...  
[ 2 ]《Python语言程序设计基础》第2版-高天 - 『电子书屋』..._吾爱破解  
[ 3 ]Python语言程序设计基础(第2版) (豆瓣)  
[ 4 ]Python语言程序设计基础-高天(第2版)-CSDN下载  
[ 5 ]Python语言程序设计基础(第2版) 高天-CSDN下载  
[ 6 ]《高天、礼欣、黄天羽 著 Python语言程序设计基础(第2版本)》.pdf  
[ 7 ]Python语言程序设计基础(第2版)_高天、礼欣、黄天..._孔夫子旧书网  
[ 8 ]Python语言程序设计基础下载|Python语言程序设计基础(第2版) pdf...  
[ 9 ]《Python语言程序设计基础》(第2版)PDF下载 - weixin_3..._CSDN博客  
>>>
```



10.5实例25搜索关键词自动提交



10.5 分析搜索引擎返回的结果

- 在技术层面，除了搜索引擎，还可以向其它可以查询数据信息的网页提交查询关键词。正是因为有这类自动提交程序，当今开发的服务网站不得不增加图片或声音类型的验证码，用来区分用户是计算机的自动程序还是人。技术是反映人类思想的手段，掌握了所谓“更有能力”和“更先进”的技术没什么大不了的，最为可贵的是去思考如何通过技术手段为人类和世界带来更美好的未来。



10.5实例25搜索关键词自动提交



10.5 分析搜索引擎返回的结果

- CAPTCHA 验证码是“全自动区分计算机和人类的图灵测试”（ Completely Automated Public Turing test to tell Computers and Humans Apart ）的缩写，它是一种区分用户是计算机程序或是人的方法，用于防止程序肆意向网络自动提交请求。



10.5实例25搜索关键词自动提交



10.5 分析搜索引擎返回的结果

PPNIG *syriepel*

just example



10.6拓展实例26 下载SCI杂志社图文摘要



https://pubs.acs.org/action/doSearch?AllField=carbon+dots&pageSize=20&startPage=1

ACS Publications C&EN CAS Find my institution Log In

ACS Publications
Most Trusted. Most Cited. Most Read.

carbon dots



My Activity



Publications



Chapter 682

Article 244

Article 64833

CLE TYPE

Article 57871

Communication 4203

Article 2216

Article 678

Article 403

15

PUBLICATION DATE

Year 5611

3 Months 3210

3 Months 1907

Month 949

Week 356

FOR

Article

Tuning Laccase Catalytic Activity with Phosphate Functionalized Carbon Dots by Visible Light

Hao Li, Sijie Guo, Chuanxi Li, Hui Huang, Yang Liu, and Zhenhui Kang*

ACS Applied Materials & Interfaces 2015, 7, 18, 10004-10012 (Research Article)

Publication Date (Web): April 17, 2015

DOI: 10.1021/acsami.5b02386

Abstract

Full text

PDF

ABSTRACT

Article

Future Perspectives and Review on Organic Carbon Dots in Electronic Applications

Maria Semeniuk, Zhihui Yi, Vida Poursorkhabi, Jimi Tjong, Shaffiq Jaffer, Zheng-Hong Lu, and Mohini Sain*

ACS Nano 2019, 13, 6, 6224-6255 (Review)

Publication Date (Web): May 30, 2019

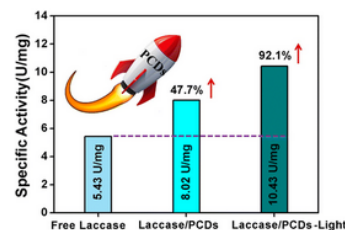
DOI: 10.1021/acs.nano.9b00688

Abstract

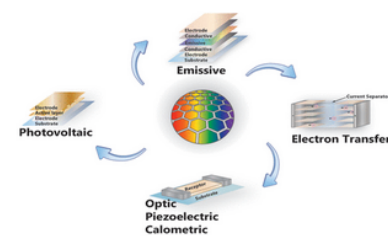
Full text

PDF

ABSTRACT



ACS APPLIED MATERIALS & INTERFACES



ACS NANO


```
import requests
import re
from bs4 import BeautifulSoup
from PIL import Image
a=input('请输入关键词')
no1=input('请输入截取的页数:')
a=a.split(' ')
c=""
no=0
tifaddress='https://pubs.acs.org/'
print(a)
for i in a:
    if c == '':
        c=c+i
    else:
        c=c+' '+str(i)
print(c)
header={
    "User-Agent":"Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:69.0) Gecko/20100101 Firefox/69.0",
    "Cookie":"__cfduid=dd8db6958e9fb2b3a121d76acca227b071573552245; ACSEnt=184138_6105_1573552246042; I2",
    "features":"html.parser"
}
b = 'https://pubs.acs.org/action/doSearch?AllField='+c+'&pageSize=20&startPage='
print(b)
for j in range(eval(no1)):
    r = requests.get(b+str(j), cookies=header)
    print(r.status_code)
    r.raise_for_status
    k=r.text.encode(r.encoding).decode('utf-8')
    soup=BeautifulSoup(k, features="html.parser")
    ls = soup.find_all('img', {'data-original':re.compile('gif')})
    #print(ls)
    for i in ls:
        if r'png' in str(i):
            type(i)
        else:
            no=no+1
            print(type(i))
            gifurl=tifaddress+i.attrs['data-original']
            print(gifurl)
            g=requests.get(gifurl, cookies=header)
            fs=open(str(no)+'.gif', 'wb')
            fs.write(g.content)
            fs.close()
```

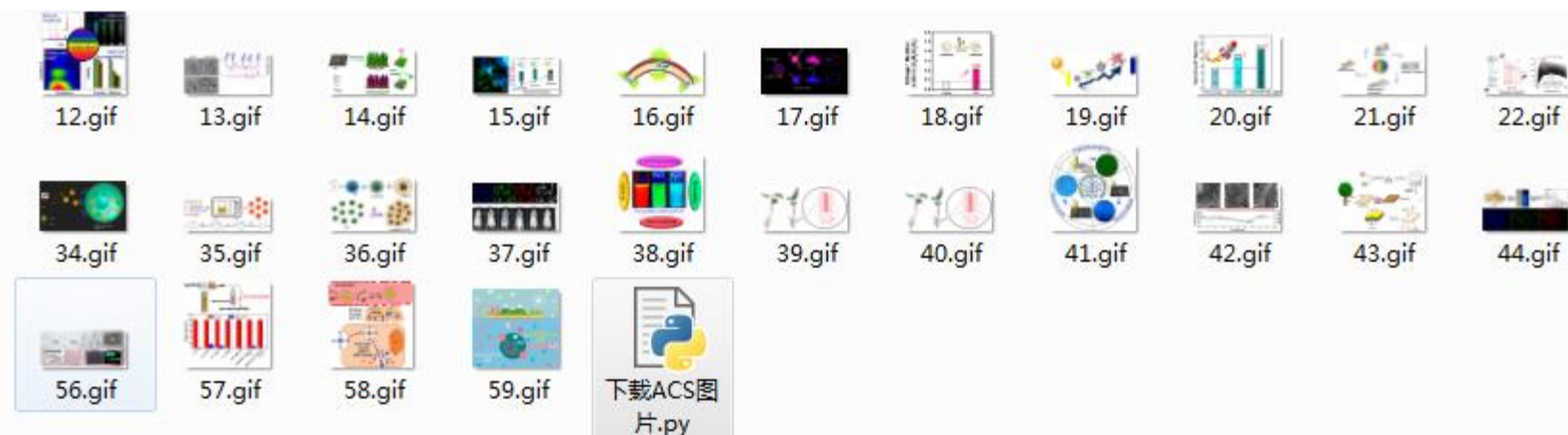
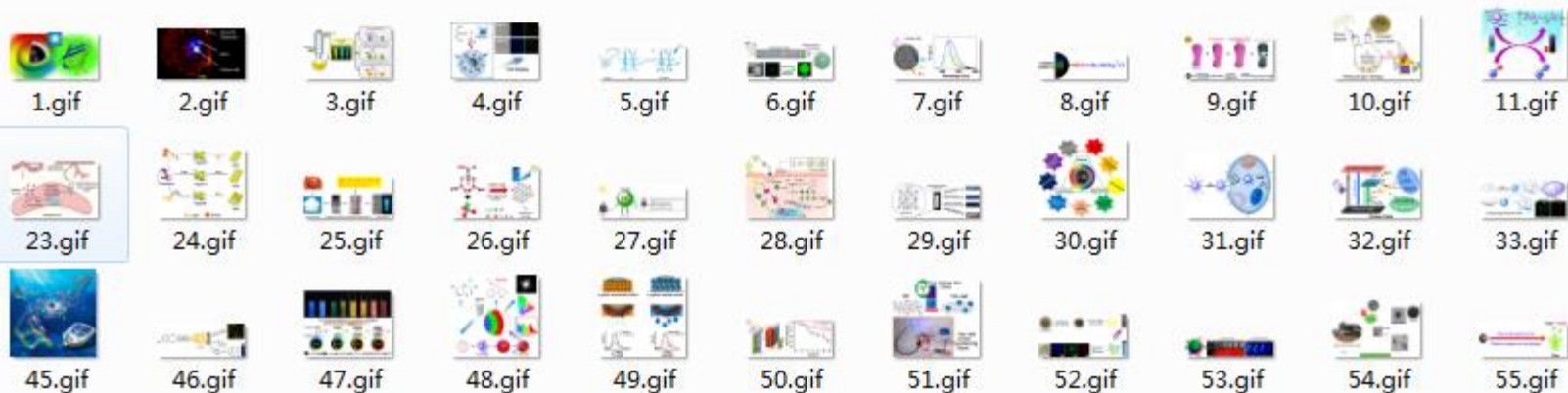
源代码



10.6拓展实例26 下载SCI杂志社图文摘要



请输入关键词carbon dots
请输入截取的页数: 3
['carbon', 'dots']
carbon+dots
<https://pubs.acs.org/action/doSearch?AllField=carbon+dots&page200>
<class 'bs4.element.Tag'>
<https://pubs.acs.org//na101/home/literatum/publisher/achs/journal/2014/ancac3.2014.8.issue-5/rn406628s/production/images/mediu>





下周课程&课后作业



第9章 科学计算和可视化

课后练习

修改实例24，按照省份输出大学排名结果。

代码文件命名：10-1pachongshengfen

修改实例24，编写软科世界一流学科排名2019-化学排名结果。

代码文件命名：10-2pachongchem

.py代码文件打包(10.学号
+姓名)发送到
python_xxmu@163.com

编程辣么好，还等什么？开始学习吧！



Programing is an Art