学习爬虫，怎么也绕不开requests库和BeautifulSoup库，先说下BeautifulSoup库：

BeautifulSoup库通俗来说是【解析、遍历、维护“标签树”(例如html、xml等格式的数据对象)的功能库 】  
一个简单的使用BeautifulSoup库的demo：

[复制代码](javascript:void(0);)

# coding:utf-8

from bs4 import BeautifulSoup

import requests

url = 'http://python123.io/ws/demo.html'

r = requests.get(url)

demo = r.text # 服务器返回响应

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

"""

demo 表示被解析的html格式的内容

html.parser表示解析用的解析器

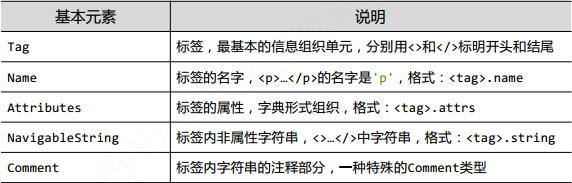
"""

print(soup) # 输出响应的html对象

print(soup.prettify()) # 使用prettify()格式化显示输出

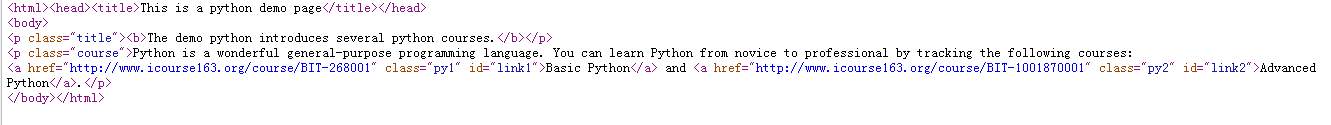
[复制代码](javascript:void(0);)

得到一个BeautifulSoup对象后，一般通过BeautifulSoup类的基本元素来提取html中的内容



2.提取html中的信息

 demo中的html内容如下：



（1）

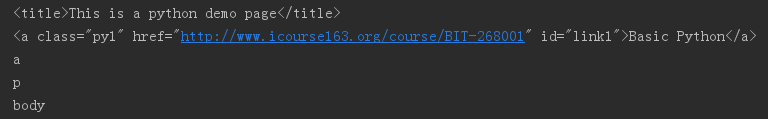
print(soup.title) # 获取html的title标签的信息

print(soup.a) # 获取html的a标签的信息(soup.a默认获取第一个a标签，想获取全部就用for循环去遍历)

print(soup.a.name) # 获取a标签的名字

print(soup.a.parent.name) # a标签的父标签(上一级标签)的名字

print(soup.a.parent.parent.name) # a标签的父标签的父标签的名字



（2）

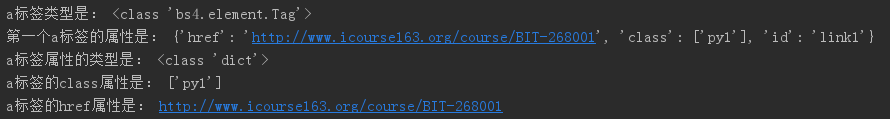
print('a标签类型是：', type(soup.a)) # 查看a标签的类型

print('第一个a标签的属性是：', soup.a.attrs) # 获取a标签的所有属性(注意到格式是**字典**)

print('a标签属性的类型是：', type(soup.a.attrs)) # 查看a标签属性的类型

print('a标签的class属性是：', soup.a.attrs['class']) # 因为是字典，通过**字典**的方式获取a标签的class属性

print('a标签的href属性是：', soup.a.attrs['href']) # 同样，通过**字典**的方式获取a标签的href属性

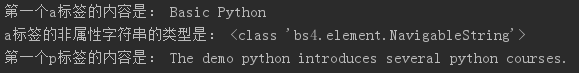


（3）

print('第一个a标签的内容是：', soup.a.string) # a标签的非属性字符串信息，表示尖括号之间的那部分字符串

print('a标签的非属性字符串的类型是：', type(soup.a.string)) # 查看标签string字符串的类型

print('第一个p标签的内容是：', soup.p.string) # p标签的字符串信息(注意p标签中还有个b标签，但是打印string时并未打印b标签，说明string类型是可跨越多个标签层次)



介绍一下find\_all()方法：

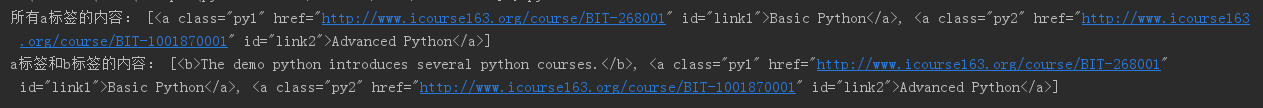
常用通过find\_all()方法来查找标签元素：<>.find\_all(name, attrs, recursive, string, \*\*kwargs) ，返回一个**列表**类型，存储查找的结果

• name：对标签名称的检索字符串  
• attrs：对标签属性值的检索字符串，可标注属性检索  
• recursive：是否对子孙全部检索，默认True  
• string：<>…</>中字符串区域的检索字符串

（1）

print('所有a标签的内容：', soup.find\_all('a')) # 使用find\_all()方法通过标签名称查找a标签,返回的是一个列表类型

print('a标签和b标签的内容：', soup.find\_all(['a', 'b'])) # 把a标签和b标签作为一个列表传递，可以一次找到a标签和b标签



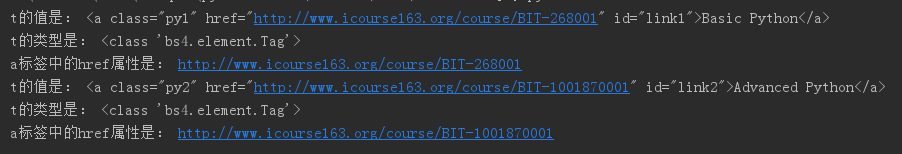
（2）

for t in soup.find\_all('a'): # for循环遍历所有a标签，并把返回列表中的内容赋给t

print('t的值是：', t) # link得到的是标签对象

print('t的类型是：', type(t))

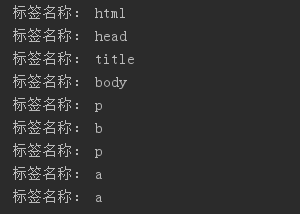
print('a标签中的href属性是：', t.get('href')) # 获取a标签中的url链接



（3）

for i in soup.find\_all(True): # 如果给出的标签名称是**True**，则找到所有标签

print('标签名称：', i.name) # 打印标签名称

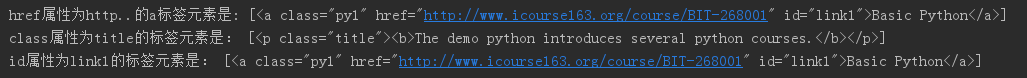


（4）

print('href属性为http..的a标签元素是:', soup.find\_all('a', href='http://www.icourse163.org/course/BIT-268001')) # 标注属性检索

print('class属性为title的标签元素是：', soup.find\_all(class\_='title')) # 指定属性，查找class属性为title的标签元素，注意因为class是python的关键字，所以这里需要加个下划线'\_'

print('id属性为link1的标签元素是：', soup.find\_all(id='link1')) # 查找id属性为link1的标签元素



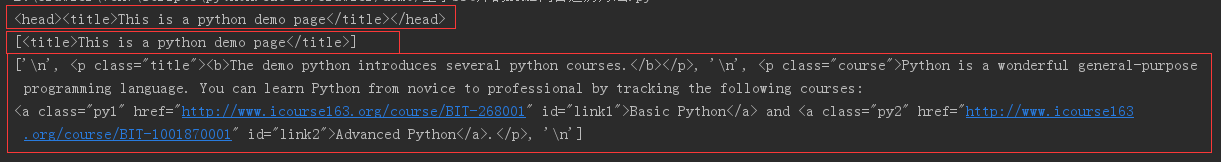
 （5）

print(soup.head) # head标签

print(soup.head.contents) # head标签的儿子标签，contents返回的是列表类型

print(soup.body.contents) # body标签的儿子标签

"""对于一个标签的儿子节点，不仅包括标签节点，也包括字符串节点，比如返回结果中的 \n"""



（6）

print(len(soup.body.contents)) # 获得body标签儿子节点的数量

print(soup.body.contents[1]) # 通过列表索引获取第一个节点的内容

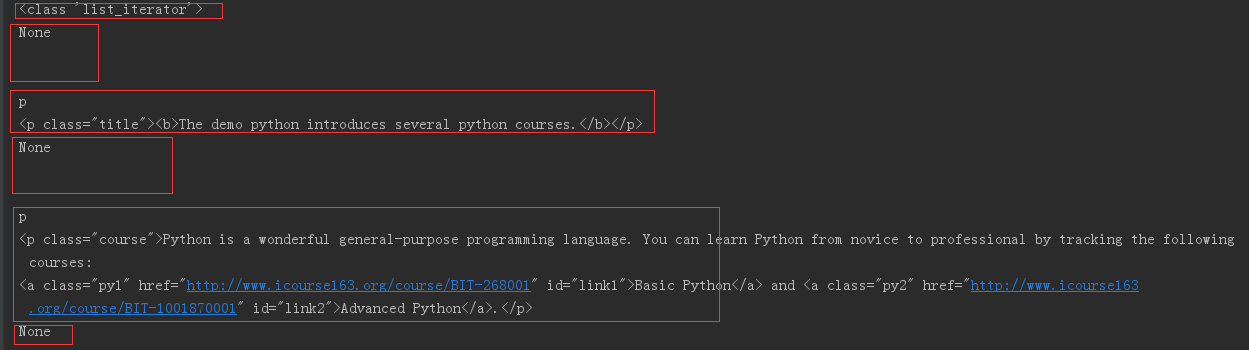


（7）

print(type(soup.body.children)) # children返回的是一个迭代对象，只能通过for循环来使用，不能直接通过索引来读取其中的内容

for i in soup.body.children: # 通过for循环遍历body标签的儿子节点

print(i.name) # 打印节点的名字



分类: [05.python爬虫学习](https://www.cnblogs.com/hanmk/category/1193015.html)