# 第三课 Beautifulsoup4模块的使用

Beautifulsoup作用：

将网页（非结构化内容）转化成结构化内容

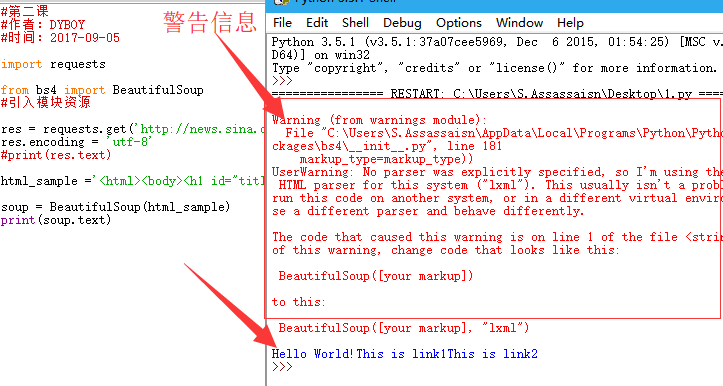
.text 取得bs对象的文字内容（去除HTML标签）

现在令一个新的字符串：

html sample ='<html><body><h1 id="title">Hello World!</h1><a href="#" class="link">This is link1</a><a href="#link2" class="link">This is link2</body></html>'

将字符串转化为bs对象：

soup = BeautifulSoup(html\_sample)

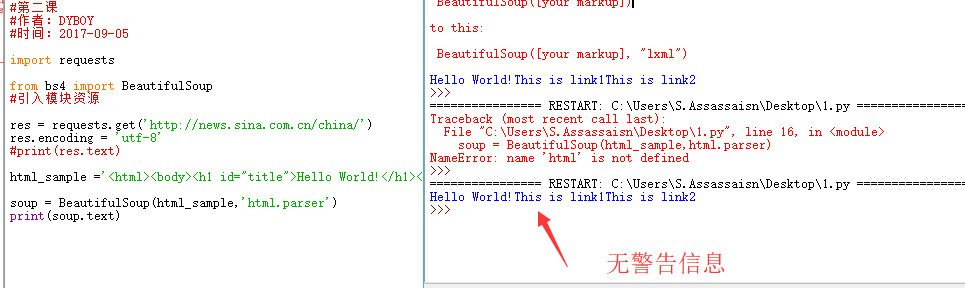


去除警告信息的方法：

原因：因为没有指定“剖析器”

解决：

soup = BeautifulSoup(html\_sample,'html.parser')

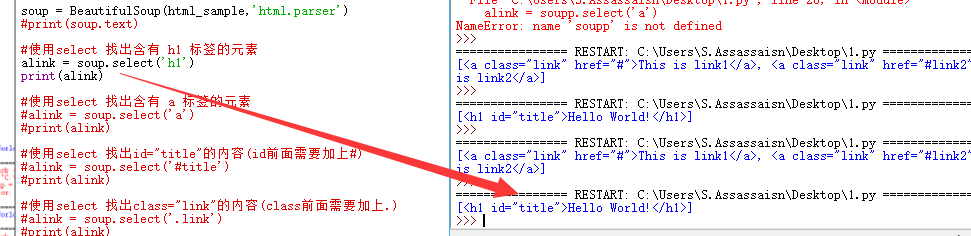


找寻特定元素：

#使用select 找出含有 h1 标签的元素

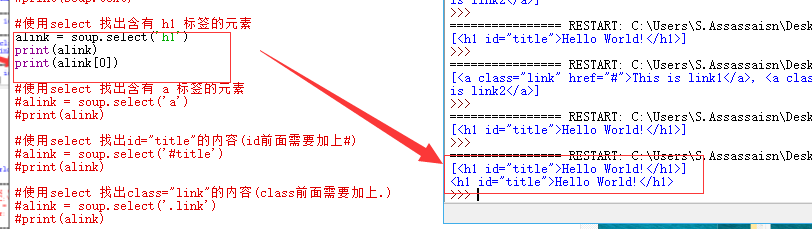
alink = soup.select('h1')

print(alink)



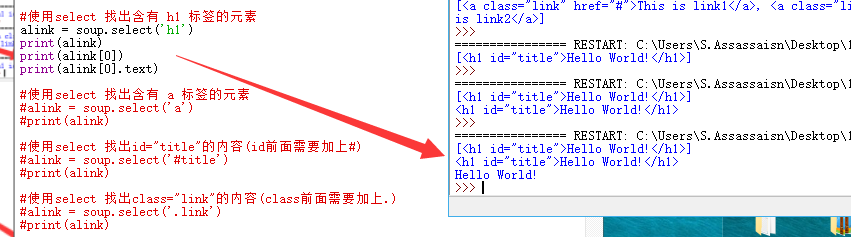
Print(alink[0])

输出的是内容，没有中括号



print(alink[0].text)

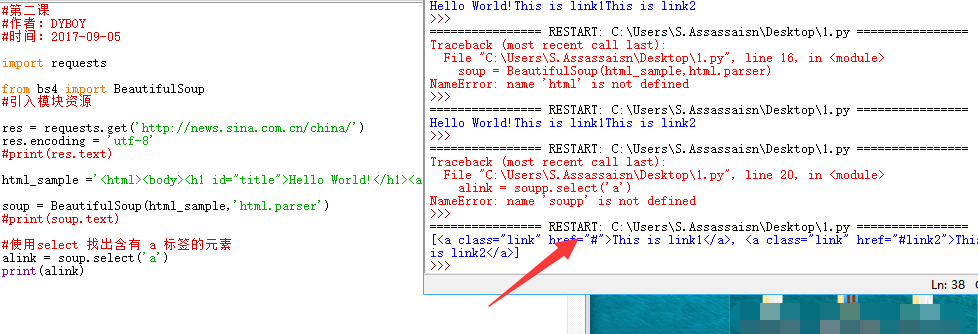
输出标签内的文字



#使用select 找出含有 a 标签的元素

alink = soup.select('a')

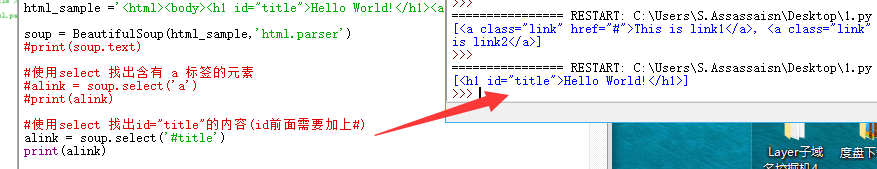
print(alink)



#使用select 找出id="title"的内容(id前面需要加上#)感觉就类似CSS语法

alink = soup.select('#title')

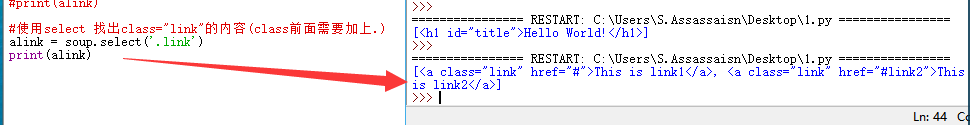
print(alink)



#使用select 找出class="link"的内容(class前面需要加上.)

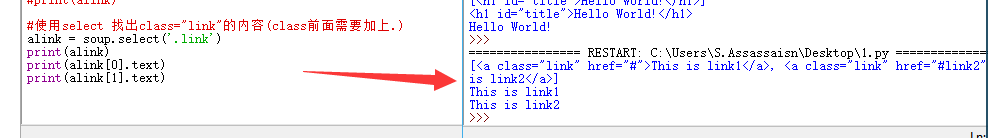
alink = soup.select('.link')

print(alink)



我们看到输出多个结果，以列表的形式存储

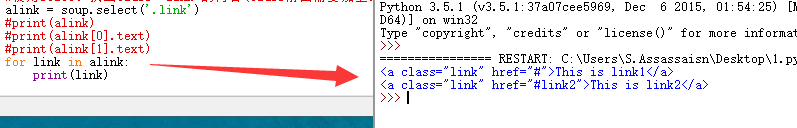
相信这样大家就会更加清晰明白了：



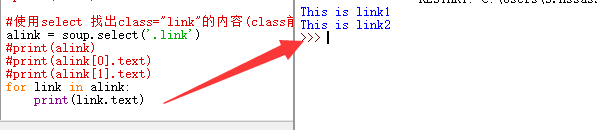
循环输出：

for link in alink:

print(link)



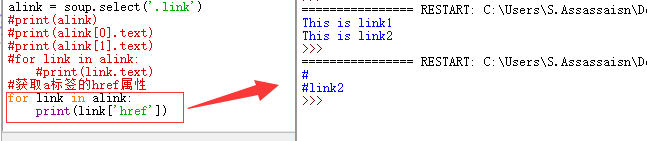
.text的利用



获取a标签的href属性：

for link in alink:

print(link['href'])



会把href等属性包装成一个字典故很方便地取到数据！

取属性值：

html\_sample2 = '<a href="#" id="testlink" title="link"> hello world!</a><a href="#2" id="testlink2" title="link2"> hello world2!</a>'

soup2 = BeautifulSoup(html\_sample2,'html.parser')

print(soup2.select('a')[0]['id'])

print(soup2.select('a')[1]['id'])

