html文件

<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>

</body>

</html>

安装Beautiful Soup

$ sudo apt-get install python-bs4

导入Beautiful Soup包

import bs4

导入html构造BeautifulSoup对象

soup = bs4.BeautifulSoup(html)

Beautiful Soup将复杂HTML文档转换成一个复杂的树形结构,每个节点都是Python对象,所有对象可以归纳为4种: Tag , NavigableString , BeautifulSoup , Comment。其中Comment 对象是一个特殊类型的 NavigableString 对象，Comment为html中的注释，如<!--Hey-->

格式化打印整个html源码，所谓格式化指打印时带有<html>等标签，输出格式如本文档开始样例

print soup.prettify()

打印所有文字内容

print soup.get\_text()

第一次出现的<title></title>标签

soup.title

print type(soup.title)

输出# <class 'bs4.element.Tag'>，说明.title获得Tag类型，type()用于获取变量类型

第一次出现的<title></title>标签的标签名

soup.title.name

print type(soup.title.name)

输出# <type 'str'>，说明.name获得str类型

第一次出现的<title></title>标签的内容

soup.title.string

print type(soup.title.string)

输出# <class 'bs4.element.NavigableString'>，说明.string获得NavigableString类型

第一次出现的<title></title>标签的上级标签，为<head></head>标签

soup.title.parent

第一次出现的<p></p>标签

soup.p

第一次出现的<p></p>标签中class属性的内容

soup.p[‘class’]

第一次出现的<a></a>标签

soup.a

所有出现的<a></a>标签

soup.find\_all(‘a’)

遍历所有出现的<a></a>标签，并打印标签内href属性的内容

for link in soup.find\_all(‘a’)

print link.get(‘href’)

第一次出现id属性的内容为link3的标签

soup.find(id=’link3’)或soup.find(attrs={‘id’:’link3’})

第一次出现标签名为div，并且id属性的内容为link3的标签

soup.find(‘div’, attrs={‘id’:’link3’})

tag = soup.p

获取标签名

tag.name

获取标签class属性的内容

tag['class']或tag.get(‘class’)

获取标签的所有属性，包括属性名和属性内容

tag.attrs

修改class属性的内容

tag['class'] = 'abc'

删除tag的class属性

del tag['class']

在标签中可能存在多个属性名相同的属性称为多值属性，如<p class="story" class="a" class="b"></p>

此时tag[‘class’]将返回一个list包含所有的class属性

将存在多值属性的标签转换为字符串时，如打印时，多值属性将合并为一个值

print soup.p

输出# <p class="story" class="a b"></p>

标签通过.string获得的标签内容不能被编辑修改，只能被替换

tag.string.replace\_with(“abc”)

Tag类型说明：

Tag可看作html中的标签，若想获取第一次出现指定标签名的某一个标签只需soup.点取加标签的名字即可

soup.body.p.b

# <b>The Dormouse's story</b>

.contents 获取Tag的所有子节点，并以列表的形式给出

a1\_tag = soup.body.contents[0].contents[1]