2.2 BeautifulSoup 装载 HTML 文档

HTML 文档节点的查找工具很多，其中 BeautifulSoup 是功能强大的十分流行的查找工具之一，本节我们介绍 BeautifulSoup 的基本应用。

2.2.1 BeautifulSoup 的安装

BeautifulSoup 是第三方的工具，它包含在一个名称为 bs4 的文件包中，需要另外安装。安装也很简单，在命令行窗体中进入 Python 的安装目录（例如 Python 在 c:\Python36)，再进入 Scripts 子目录，找到 pip 程序，执行：

pip install bs4

我们可看到安装过程进行，成功安装后就可以在 Python 的命令窗体中执行语句：

from bs4 import BeautifulSoup

如果这条语句执行没有报错，就说明安装成功了。

2.2.2 BeautifulSoup 装载 HTML 文档

如果 doc 是一个 HTML 文档，通过：

from bs4 import BeautifulSoup

soup=BequtifulSoup(doc,"lxml")

就可以创建一个名称为 soup 的 BeautifulSoup 对象，其中 doc 是一个 HTML 文档字符串，"lxml"是一个参数，表示创建的是一个通过"lxml"解析器解析的文档。BeautifulSoup 有多种解析器，其中"lxml"是最常用的一个。

通过调用:

soup.prettify()

可以把 soup 对象的文档树变成一个字符串。

例 2-2-1：用 BeautifulSoup 装载 HTML 文档，显示文档的树状结构

```
from bs4 import BeautifulSoup
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>
</body>
</html>
'''
soup=BeautifulSoup(doc,"lxml")
```

```
s=soup.prettify()
print(s)
```

执行后输出的结构如下：

```
<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 class="sister" href="http://example.com/elsie" id="link1">
    Elsie
   </a>
   ,
   <a class="sister" href="http://example.com/lacie" id="link2">
    Lacie
   </a>
   and
   <a class="sister" href="http://example.com/tillie" id="link3">
    Tillie
   </a>
   ;
and they lived at the bottom of a well.
  </p>
  <p class="story">
   ...
  </p>
 </body>
</html>
```

由此可见 BeautifulSoup 装载了 HTML 文档，最后通过 prettify()函数把文档树转为字符串的格式。

BeautifulSoup 装载文档的功能十分强大，它在装载的过程中如果发现 HTML 文档中的元素有缺失的情况，它会尽可能地对文档进行修复，使得最后的文档树是一棵完整的树。这一点十分重要，因为我们面临的大多数网页都或多或少有些元素是缺失的，BeautifulSoup 都能正确装载它们。

非禁复制

例 2-2-2： BeautifulSoup 装载有缺失的 HTML 文档

```
from bs4 import BeautifulSoup
doc='''
<title>有缺失元素的 HTML 文档</title>
<div>
<A href='one.html'>one</a>
<p>
<a href='two.html'>two</a>
</DIV>
'''
soup=BeautifulSoup(doc,"lxml")
s=soup.prettify()
print(s)
```

程序结果如下：

```
html>
 <head>
  <title>
   有缺失元素的 HTML 文档
  </title>
 </head>
 <body>
  <div>
   <a href="one.html">
    one
   </a>
   <p>
    <a href="two.html">
     two
    </a>
   </p>
  </div>
 </body>
</html>
```

由此可见在 HTML 缺失<html>根元素时 BeautifulSoup 会自动补上；它发现<title>元素没有<head>元素后自动增加用<head>元素包含<title>元素；它发现没有<body>元素也自动补上；它还把<A href='one.html'>one改成one；还有它把：

```
<p>
<a href='two.html'>two</a>
```

改成：

```
    <p>
     <a href="two.html">
      two
```

严禁复制

```
    </a>
   </p>
```

最后的</DIV>改成 ，默认情况下所有的 tag 元素的名称都转为小写。通过 BeautifulSoup 的修正后这课 HTML 树就比较完整了。

值得注意的是 BeautifulSoup 虽然功能强大能修正一些缺失的 HTML 元素，但是它还没有智能到能完全修复所有 HTML 文档错误的程度。

严禁复制