2.5 BeautifulSoup 使用 CSS 语法查找元素

2.5.1 使用 CSS 语法

BeautifulSoup 除了可以用 find 与 find_all 函数查找 HTML 文档树的节点元素外，还可以采用 CSS 类似的语法来查询，规则是：

tag.select(css)

其中 tag 是一个 bs4.element.Tag 对象，即 HTML 中的一个 element 节点元素，select 是它的查找方法，css 是类似 css 语法的一个字符串，一般结构如下：

[tagName][attName[=value]]

其中[...]部分是可选的；

tagName 是元素名称，如果没有指定就是所有元素；

attName=value 是属性名称，value 是它对应的值，可以不指定属性，在指定了属性后也可以不指定值；

tag.select(css)返回一个 bs4.element.Tag 的列表，哪怕只有一个元素也是一个列表；

例 2-5-1：soup.select("a") 查找文档中所有<a>元素节点；

soup.select("p a") 查找文档中所有<p>节点下的所有<a>元素节点；

soup.select("p[class='story'] a") 查找文档中所有属性 class="story"的<p>节点下的所有<a>元素节点；

soup.select("p[class] a") 查找文档中所有具有 class 属性的<p>节点下的所有<a>元素节点；

soup.select("a[id='link1']") 查找属性 id="link1"的<a>节点；

soup.select("body head title") 查找<body>下面<head>下面的<title>节点；

soup.select("body [class] ") 查找<body>下面所有具有 class 属性的节点；

soup.select("body [class] a") 查找<body>下面所有具有 class 属性的节点下面的<a>节点；

<head>下面的<title>节点；

例 2-5-2： 查找 HTML 文档中所有<p>下面的<a>的链接

```
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")
tags=soup.select("p[class='story'] a")
for tag in tags:
    print(tag["href"])
```

程序结果：

```
http://example.com/elsie
http://example.com/lacie
http://example.com/tillie
```

另外我们通过

```
tags=soup.select("p a")
tags=soup.select("a")
tags=soup.select("p[class] a")
```

等也可以得到一样的结果。

2.5.2 属性的语法规则

在 CSS 结构中的[attName=value]中表示属性 attrName 与 value 相等，也可以指定不等、包含等运算关系，具体运算如下表：

选择器	描述
［attName］	用于选取带有指定属性的元素。
［attName=value］	用于选取带有指定属性和值的元素。
［attNameˆ=value］	匹配属性值以指定值开头的每个元素。
［attName$=value］	匹配属性值以指定值结尾的每个元素。
［attrName*=value］	匹配属性值中包含指定值的每个元素。

因此：

soup.select("a[href='http://example.com/elsie']") 查找 href="http://example.com/elsie"的<a>节点；

soup.select("a[href$='sie']") 查找 href 以"sie"结尾的<a>节点；

soup.select("a[href^='http://example.com']") 查找 href 以"http://example.com"开始的<a>节点；

soupselect("a[href*='example']") 查找 href 的值中包含"example"字符串的<a>节点；

2.5.3 select 查找子孙节点

在 select(css)中的 css 有多个节点时，节点元素之间用空格分开，就是查找子孙节点，例如 soup.select("div p")是查找所有 节点下面的所有子孙<p>节点。

例 2-5-3：查找子孙节点

```
from bs4 import BeautifulSoup
doc="<div><p>A</p><span><p>B</p></span></div><div><p>C</p></div>"
soup=BeautifulSoup(doc,"lxml")
tags=soup.select("div p")
```

```
for tag in tags:
    print(tag)
```

程序结果：

```
<p>A</p>
<p>B</p>
<p>C</p>
```

其中 tags=soup.select("div p")是查找 下面的所有子孙节点 ，因此包含下面的 B 。

2.5.4 select 查找直接子节点

在 select(css)中的 css 有多个节点时，节点元素之间用" > "分开(注意>的前后至少包含一个空格)，就是查找直接子节点，例如 soup.select("div > p")是查找所有 节点下面的所有直接子节点 ，不包含孙节点。

例 2-5-4：查找直接子节点

```
from bs4 import BeautifulSoup
doc="<div><p>A</p><span><p>B</p></span></div><div><p>C</p></div>"
soup=BeautifulSoup(doc,"lxml")
tags=soup.select("div > p")
for tag in tags:
    print(tag)
```

程序结果：

```
<p>A</p>
<p>C</p>
```

其中 tags=soup.select("div > p")是查找 下面的直接子节点 ，因此不包含下面的 B 。

2.5.5 select 查找兄弟节点

在 select 中用" ~ "连接两个节点表示查找前一个节点后面的所有同级别的兄弟节点（注意~号前后至少有一个空格），例如 soup.select("div ~ p")查找 后面的所有同级别的 兄弟节点。

在 select 中用" + "连接两个节点表示查找前一个节点后面的第一个同级别的兄弟节点（注意+号前后至少有一个空格）

例 2-5-5：查找兄弟节点

```
from bs4 import BeautifulSoup
doc="<body>demo<div>A</div><b>X</b><p>B</p><span><p>C</p></span><p>D</p></div></body>"
soup=BeautifulSoup(doc,"lxml")
print(soup.prettify())
tags=soup.select("div ~ p")
for tag in tags:
```

```
        print(tag)
    print()
    tags=soup.select("div + p")
    for tag in tags:
        print(tag)
```

程序结果：

```
<p>B</p>
<p>D</p>
```

其中 tags=soup.select("div ~ p")找到 后面同级别的所有 节点，不包含中的 C ，因为它与 不同级别。 而 tags=soup.select("div + p")要找 的下一个兄弟节点 ，但是 的下一个兄弟节点是X，不是 节点，因此没有找到，注意结果不是 B 。