



## 2.4 BeautifulSoup遍历文档元素

深圳信息职业技术学院

Shenzhen Institute Of Information Technology

教师：黄锐军

# 目录

COMPANY

2.4.1 获取元素节点的父节点

2.4.2 获取元素节点的直接子元素节点

2.4.3 获取元素节点的所有子孙元素节点

2.4.4 获取元素节点的兄弟节点

# PART ONE

## 获取元素节点的父节点

## 获取元素节点的父节点

---



BeautifulSoup通过：

`tag.parent`

获取tag节点的父节点，其中根节点<html>的父节点是名称为[document]的节点，这个[document]节点的父节点是None。



**例2-4-1：找出文档中<p class="title"><b>The Dormouse's story</b></p>的<b>元素节点的所有父节点的名称。**

```
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")
```

```
print(soup.name)
```

```
tag=soup.find("b")
```

```
while tag:
```

```
    print(tag.name)
```

```
    tag=tag.parent
```



程序结果：

[document]

b

p

body

html

[document]

由此可见<b>节点的父节点依次为<p>、<body>、<html>

# PART TWO

获取元素节点的直接子元素节点



## 获取元素节点的直接子元素节点



BeautifulSoup通过：

`tag.children`

获取tag节点的所有直接子节点，包括element、text等类型的节点。

### 例2-4-2：获取<p>元素的所有直接子元素节点

```
from bs4 import BeautifulSoup
```

```
doc = '''
```

```
<html> <head> <title>The Dormouse's story</title> </head>
```

```
<body>
```

```
<p class="title"> <b>The <i>Dormouse's</i> story</b> Once upon a  
time ...</p>
```

```
</body>
```

```
</html>
```



...

```
soup=BeautifulSoup(doc,"lxml")
tag=soup.find("p")
for x in tag.children:
    print(x)
```

程序结果：

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

Once upon a time ...

<p>节点下面有2个直接子节点元素，一个是element类型的节点

<b>The <i>Dormouse's</i> story</b>，另外一个text类型的节点Once upon a time...

# PART Three

获取元素节点的所有子孙元素节点

# 获取元素节点的所有子孙元素节点

---



BeautifulSoup通过：

tag. **desendants**

获取tag节点的所有子孙节点元素，包括element、text等类型的节点。



### 例2-4-3：获取<p>元素的所有子孙元素节点

```
from bs4 import BeautifulSoup
```

```
doc = '''
```

```
<html> <head> <title>The Dormouse's story</title> </head>
```

```
<body>
```

```
<p class="title"> <b>The <i>Dormouse's</i> story</b> Once upon  
a time ...</p>
```

```
</body>
```

```
</html>
```

```
'''
```

```
soup = BeautifulSoup(doc, "lxml")
```

```
tag = soup.find("p")
```

```
for x in tag.descendants:
```

```
    print(x)
```



程序结果：

**The *Dormouse's* story**

The

*Dormouse's*

Dormouse's

story

Once upon a time ...

由此可见<p>元素节点下面有下面几个子孙节点：

The: 这是一个text孙子节点，它是<b>的子节点；

<i>Dormouse's</i>: 这是<p>下面的一个element孙子节点，是<b>的子节点；

Dormouse's: 这是<p>下面的孙子text节点，即<i>Dormouse's</i>的子节点；

story：这是<p>下面的孙子text节点，它是<b>的子节点；



# PART Four

## 获取元素节点的兄弟节点



## 获取元素节点的兄弟节点



BeautifulSoup通过：

`tag.next_sibling`

`tag.previous_sibling`

来获取下一个和前一个兄弟节点，其中`tag.next_sibling`是tag的临近的下一个兄弟节点，`tag.previous_sibling`是tag的临近的前一个兄弟节点。





## 例2-4-4：查找前后兄弟节点

```
from bs4 import BeautifulSoup
doc=''
<html> <head> <title>The Dormouse's story</title> </head>
<body>
<p class="title"> <b>The <i>Dormouse's</i> story</b> Once upon a
time ...</p>
</body>
</html>
'''

soup=BeautifulSoup(doc,"lxml")
tag=soup.find("b")
print(tag.previous_sibling)
```



```
print(tag.next_sibling)
tag=soup.find("i")
print(tag.previous_sibling)
print(tag.next_sibling)
```

程序结果：

None

Once upon a time ...

The

story

由此可见<b>节点的前面兄弟节点为None，下一个兄弟节点是text节点"Once upon a time .."，<i>节点的前一个兄弟节点是text节点"The"，下一个是text节点"story"。



THANK YOU