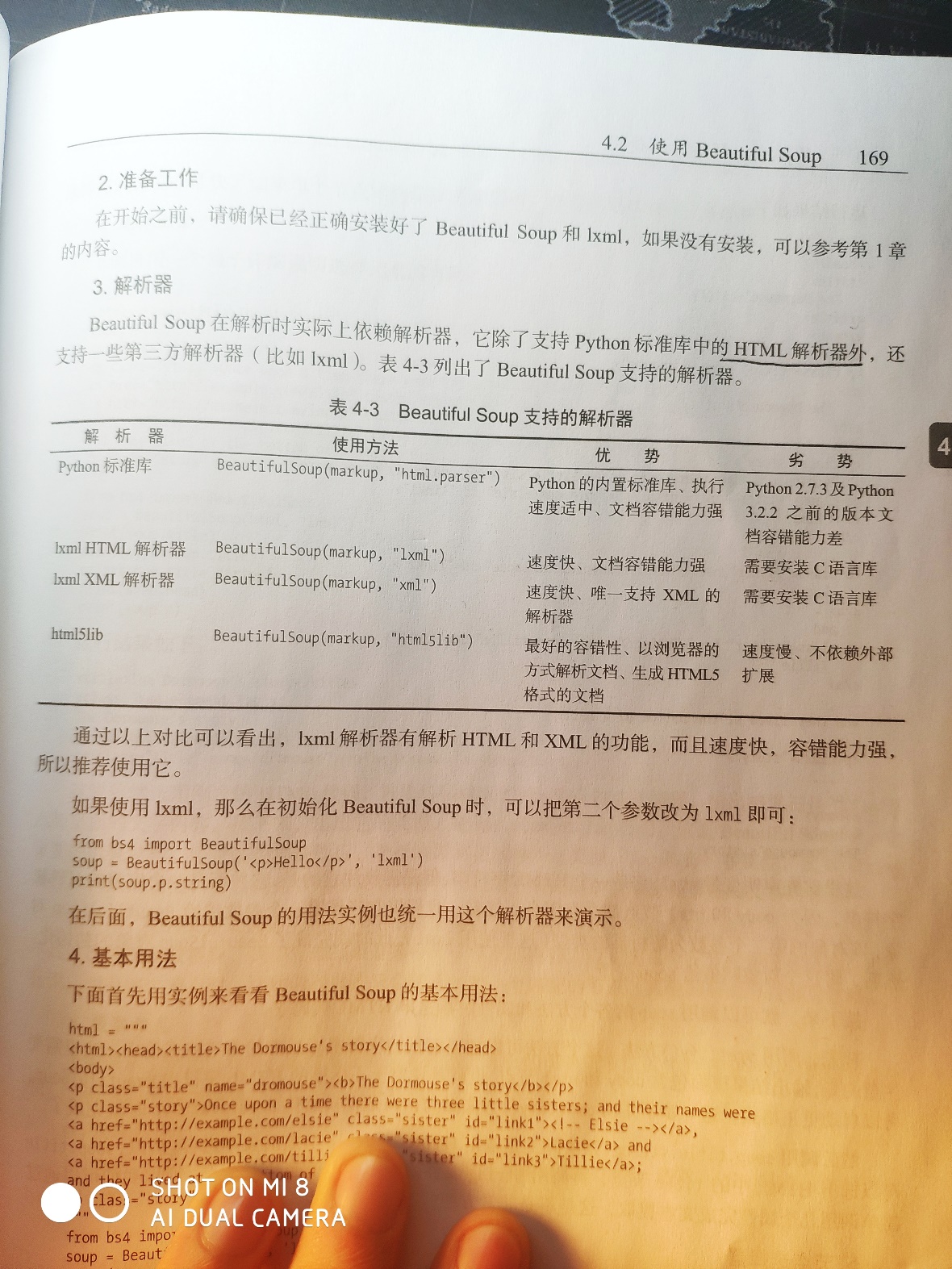
# Beautiful Soup 的使用

## 一、简介

Beautiful Soup 是python的一个HTML或XML的解析库，可以方便的从网页中提取数据。

## 二、 解析器



## 三、基本用法

参考

4.2.4.1.py

## 四、节点选择器

直接调用节点的名称可以选择节点元素，再调用string属性就可以得到节点内的文本了

### 选择元素

参考：

4.2.4.2.py

### 2、提取信息

前面调用了string属性来获取文本的值，如何获取节点属性的值？获取节点名呢？

#### (1)、获取名称

利用name属性获取节点的名称

例如：print(soup.title.name) 输出为：title

#### (2)、获取属性

每个节点可能多个属性，比如id和class等，选择这个节点元素后，可以调用attrs获取所有属性。返回是一个字典。

参考：

4.2.4.3.py

#### (3)、获取内容

string 属性

Print(soup.p.string)

### 3、嵌套选择

我们知道每一个返回结果都是bs4.element.Tag，它可以继续调用节点进行下一步的选择。

参考4.2.4.4.py

### 4、关联选择

在做选择的时候，有时候不能做到一步想要的节点元素，需要先选中一个节点元素，然后以它为基准再选择它的子节点，父节点，兄弟节点等。

#### （1）、子节点和子孙节点

选取元素后，如果想要获取它的直接子节点，可以调用contents属性

#调用childer属性得到相应得结果，为生成器类型  
print(soup.p.children)  
for i, child2 in enumerate(soup.p.children):  
 print(i, child2)  
  
#得到所有子孙节点的话，调用descendants属性,结果为生成器，descendants会递归查询所有子节点  
print(soup.p.descendants)  
for i, child2 in enumerate(soup.p.descendants):  
 print(i, child2)

参考：

###### 4.2.4.5.py

#### （2）、父节点和祖先节点

获取某个节点元素的父节点，可以调用parent属性：

所有的祖先节点，可以调用parents属性

参考:

###### 4.2.4.6.py

#### （3）、获取兄弟节点

next\_sibling和previous\_sibling分别获取节点的下一个和上一个兄弟节点。

next\_siblings和previous\_siblings分别获取节点的前面和后面的所有兄弟节点的生成器。

###### 参考: 4.2.4.7.py

#### （4）、提取信息

前面讲解了关联元素节点的选择方法，如果想要获取它们的一些信息，比如文本，属性等

如果返回的是单个节点，那么可以直接调用string, attrs等属性获得其文本和属性，如果返回的是多个节点的生成器，则可以转化为列表后取出单个元素，然后再调用string，attrs等属性获取其对应的文本和属性。

###### 参考：4.2.4.8.py

## 五、方法选择器

前面所将的选择方法都似乎通过属性来选择，这种方法非常快，但是如果进行比较负责的选择的话，比较麻烦，Beautiful Soup 还为我们提供了一些查询方法比如find\_all()和find()，调用它们，然后传入相应的参数。

### 1、find\_all()

查询所有符合条件的元素，给它传入一些属性和文本，就可以获取符合条件的元素。

十分强大。

API:

find\_all(name,attrs,recursive, text, “”kwargs)

#### （1）、name

根据节点名(ul)来查询元素

例如：

，soup.find\_all(name='ul')

参考：

###### 4.2.4.9.py

#### (2)、attrs

除了根据节点名查询，我们也传入一些属性来查询

例如：

print(soup.find\_all(attrs={"name": 'elements'}))  
#对于一些常用的属性，比如id和class等，我们可以不用attrs来传递  
print(soup.find\_all(id='list-1'))

参考：

###### 4.2.4.10.py

#### (3)、text

text参数可以用来匹配节点的文本，传入的形式可以是字符串，可以是正则表达式。

参考：

###### 4.2.4.11.py

### find()

我们这里返回的是单个元素，也就是匹配到的第一个元素

参考

###### 4.2.4.12.py

还有很多查询方法：

find\_parents()和find\_parent()分别是查询所有祖先节点，后者返回直接父亲节点

find\_next\_siblings()和find\_next\_silbings()分别是返回后面所有的兄弟节点，返回后面的第一个兄弟

find\_previous\_siblings()和find\_previous\_sibling()分别返回前面所有的兄弟节点，返回后

面第一个兄弟节点

find\_all\_next()和find\_next()：分别返回节点后所有符合条件的节点，和返回第一个符合条件的节点

find\_all\_previous()和find\_previous()返回节点前所有符合条件的节点，返回第一个符合条件的节点。

## 六、CSS选择器

Beautiful Soup还提供了另外一种选择器，CSS选择器。

参考：

<http://www.w2school.com.cn/cssref/css_selectors.asp>

使用CSS选择器时，只需要调用select()方法，传入相应的CSS选择器即可

实例：

###### 4.2.4.13.py

### 嵌套选择

select()方法同样也支持嵌套选择，例如先选择所有ul节点，再遍历每个ul节点，选择其li节点。

参考：

###### 4.2.4.14.py

### 获取属性

我们知道节点类型是Tag类型，所以获取属性还可以利用原来的方法。

参考：

###### 4.2.4.15.py

### 获取文本

要获取文本，当然可以利用string属性，还有一个方法，那就是get\_text()方法

###### 参考： 4.2.4.16.py

## 总结：

推荐使用lxml解析库，必要可以使用html.parser

节点选择筛选功能弱小，但是速度快

建议使用find()或者find\_all()查询匹配单个结果或者多个结果

如果对CSS选择器熟悉，可以选择使用select()方法