

1、HTML、XML、CSS

2、BeautifulSoup 中find的使用:

BeautifulSoup 提供的这一系列的 find 方法可以让我们以对象的方式来过滤和筛选文档树的节点，这种方式的优点是学习成本低，而且代码易读性强。

find() 系列方法参考如下表所示。

方 法	说 明
find()	查找返回文档中符合条件的第一个元素
find('p')	按照指定的元素标签名称查找元素并返回第一个符合条件的元素
find(text="newtext")	查找并返回具有指定文本内容的第一个元素
find(attrs={'id': 'value'})	查找并返回与指定属性值匹配的的第一个元素
find(class_='value')	查找具有指定样式类的第一个匹配元素
find_all()	查找并返回所有元素
find_parent()	返回父元素
find_parents()	返回所有的祖先元素
find_sibling()	返回第一个相邻的元素
find_siblings()	返回所有的相邻的元素
find_next()	从当前元素开始查找返回下一个符合条件的元素
find_all_next()	从当前元素开始查找返回下一个符合条件的所有元素
find_previous()	从当前元素开始查找返回上一个符合条件的元素
find_all_previous()	从当前元素开始查找返回下一个符合条件的所有元素

find_all_*方法与 find() 方法具有相同的参数。

3、css选择器

4、

```
import requests
import os
url = "http://image.nationalgeographic.com.cn/2017/0211/20170211061910157.jpg"
root = "D://pics//"
path = root + url.split('/')[-1]
try:
    if not os.path.exists(root):
        os.mkdir(root)
    if not os.path.exists(path):
        r = requests.get(url)
        with open(path, 'wb') as f:
            f.write(r.content)
            f.close()
        print("文件保存成功")
    else:
        print("文件已存在")
except:
    print("爬取失败")
```

