首先需要安装

> pip install beautifulsoup4

- 用法：

```
from bs4 import BeautifulSoup
soup = BeautifulSoup(html字符串, '解析方法')
# 解析方法如 lxml

print(soup.prettify()) # 自动将html字符串的标签补齐
```

标签选择器

选择元素

```
from bs4 import BeautifulSoup
soup = BeautifulSoup(html字符串, '解析方法')

html = """
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title> 胖鸟电影-最新中英字幕电影美剧下载 热门电影推荐 </title>
<meta name="keywords" content="最新电影下载，高清下载，免费在线观看，免费电影，最新美剧，电影推荐" />
<meta name="description" content="胖鸟电影 -每日更新最新电影，中英字幕电影，免费电影在线观看，手机在线观看，分享高清电影下载，百度网盘电影，盘点热门电影合集。">
<meta name="siteBaseUrl" content="http://www.pniao.com" />
<link rel="stylesheet" href="http://www.pniao.com/View/style/main.css?v=55"  />
<link rel="shortcut icon" href="http://www.pniao.com/favicon.ico" type="image/x-icon" />
<script type="text/javascript" src="http://www.pniao.com/Plugin/jquery-1.7.1.min.js"></script>
<script type="text/javascript" src="http://www.pniao.com/Plugin/commonmin.js?v=333"></script>
</head>
<body>
<div class="headerContainer"><div class="mainContainer">
        <div class="logo"><h1><a href="http://www.pniao.com/">胖鸟电影</a></h1></div>
        <div class="headerMenu">
        """

print(soup.title.string) # 输出标题字符串 // title位置的参数必须是标签名，如果有多个标签，只返回第一个结果
print(soup.title.name) # 返回标签名称（输出title）
print(soup.meta.attrs['content']) # 获取meta标签下conten属性的值（标签中用等号的部分）,注意是方括号哦
print(soup.meta['content']) # 也能获取meta标签下conten属性的值
print(soup.head.title.string) # 嵌套，获取head标签下title标签的文字
print(soup.head.contents) # 将head标签下内容返回一个列表，子节点
print(soup.dead.children) # 也是子节点，不顾返回的是一个迭代器，不能直接输出内容，可以用for进行遍历
print(soup.head.descendants) # 子孙节点，可以获取子节点和子节点的子节点，也是返回一个迭代器
print(soup.meta.parent) # 获取meta的父节点
print(soup.meta.parent) # 获取meta的祖先节点，返回一个生成器
print(soup.link.next_siblings) # 获取link标签后面的兄弟标签，返回生成器
print(soup.link.previous_siblings) # 获取link标签前面的兄弟标签，返回生成器
```

标签选择器

- find_all(name, attrs（属性）, recursive（递归的）, text, **kw args（字典键值对？）)
  - 可根据标签名，属性，内容查找文档
  - 返回 `bs4.element.tag` 元素构成的列表，里面的元素可以继续用lxml方法查询
  - 关于查找 `name`

```
print(soup.find_all('div')) # 返回name标签对应的内容
```

- 关于查找 `attrs`

```
print(soup.find_all(attrs = {'name' = 'keywords'})) # 查找含有这个属性的标签，传入的是字典哦
# 或者简化写法：
print(soup.find_all(class_ = 'logo')) # 因为class是python定义的关键字，所以要加下划线
```

- 关于查找 `text`

```
print(soup.find_all(text = '胖鸟电影')) # 返回的就是文字内容，以列表形式
```

- find(name, attrs（属性）, recursive（递归的）, text, **kw args（字典键值对？）)
    - 返回单个元素（第一个），返回的就是对应的值
- 相应的，find和find_all方法还可以查找子节点、父节点、兄弟节点等

CSS选择器

将CSS选择方式传给select

```
print(soup.select('.logo')) # logo是属性值，class的话就在属性值前面加'.'，li的话就在前面加'#'
print(soup.select('head meta')) # 前面什么也不加的话表示标签名，可以层层查询，空格隔开
```

获取属性

- 使用`['属性名']`或者`attrs['属性名']`

```
for i in soup.select('head meta'):
    print(i['content'])  # 输出所有content属性的值
```

获取内容

- 使用`get_text()`方法

```
text_ = soup.select('.logo') # 此时text_是列表，不能进行操作
for i in text_:
    print(i.get_text())
```

——by 秦小炅 2018.9.19