from bs4 import BeautifulSoup

<https://beautifulsoup.readthedocs.io/zh_CN/v4.4.0/>

**基于HTML数据生成BeautifulSoup数据结构**

soup=BeautifulSoup(html数据,’html.parser’)

soup=BeautifulSoup(open(html文件) ,’html.parser’)

**输出结构规范化的HTML数据**

soup.prettify()

**查询Tag类型**

soup.tagname.name

**访问Tag文本内容**

soup.tagname.string #tag仅包含一个NavigableString时，包含多个NavigableString时会返回None

soup.tagname.strings #tag包含多个NavigableString时

soup.stripped\_strings #忽略空行、空字符

#此处的文本内容是BeautifulSoup特有的NavigableString类字符串，功能上与Unicode字符串无异，但仅能在BeautifulSoup中使用，可通过unicode()函数转换为正常字符串

**访问Tag的上一级Tag**

soup.tagname.parent

**访问Tag的所有上级Tag**

soup.tagname.parents

**访问Tag的下一级Tag**

soup.tagname.children #以生成器形式返回

soup.tagname.contents #以列表形式返回

**访问Tag的所有下级Tag**

soup.tagname.descendants

**方通Tag邻近的同级Tag**

soup.tagname.previous\_sibling #邻近的前序同级Tag

soup.tagname.next\_sibling #邻近的后续同级Tag

#实际HTML文档中，一个Tag的邻近同级Tag往往是换行符

**方通Tag所有的同级Tag**

soup.tagname.previous\_siblings #邻近的前序同级Tag

soup.tagname.next\_siblings #邻近的后续同级Tag

**查询Tag所有参数属性**

soup.tagname.attrs

**查询Tag参数属性**

soup.tagname[‘参数名’]

#查询Tag参数时实际上是按照键值对的方式进行的字典查询

**查询Tag是否含有某参数属性**

soup.has\_attr(‘参数名’)

**访问第一个检索到的Tag**

soup.tagname

soup.find(‘tagname’或其它条件)

**访问所有同类Tag**

soup.find\_all(‘tagname’ /基于属性的条件/字符串及字符串列表/正则表达式)

#与re库中的find不同，BeautifulSoup中的find方法只用于寻找标签

**访问所有Tag**

soup.find\_all(True)

**获取所有文字内容**

soup.get\_text(连接不同tag中内容的分隔符)