**环境安装**

pip install bs4

pip install lxml

**流程**

实例化BeautifulSoup对象，并将页面源码存入其中

使用BeautifulSoup对象的属性或函数得到想要数据

**实例化对象**

**用本地的html文件实例化对象**

from bs4 import BeautifulSoup

fp=open('./test.html','r',encoding='utf-8')#这是获取本地html文件

soup=BeautifulSoup(fp,'lxml')#这个就可以返回存入fp文件的BeautifulSoup对象(lxml是解析方式)

**用互联网上的页面源码实例化对象**

from bs4 import BeautifulSoup

page\_text=response.text

soup=BeautifulSoup(page\_text,'lxml')#这个就可以返回存入页面响应源码的BeautifulSoup对象(lxml是解析方式)

**相关属性和方法**

**tagName(其实就是在bs对象后直接用标签名做属性)**

#可以返回第一个此标签

soup.a#返回第一个a标签内容

soup.div#返回第一个div标签内容

#标签都可以这么用

**find()和find\_all()**

soup.find('div',class\_='song')#可以匹配有class=song的第一个div标签的内容

soup.find\_all('div',class\_='song')#可以匹配有class=song的全部div标签的内容(列表)

**select()返回的是列表**

soup.select('.tamg')#返回有class='tamg'的所有标签内容(也可以用其他选择器)

soup.select('.tamg > div a')#返回有class='tamg'的标签且下一级有div标签，且这个div下不管层级有一个a标签的内容

#'>':表示下一级;' ':表示下面不管多少级

**获取标签间的文本信息**

#三者都可以获取符合条件第一个标签之间的文本信息

soup.select('.tamg')[0].get\_text()

soup.select('.tamg')[0].text

soup.select('.tamg')[0].string#string只可以获取该标签下直系文本内容，上面两个都是全部获取

**获取标签中的属性**

soup.select('.tamg')[0]['str']#可以获取符合条件的标签的str属性的内容