爬虫 day2 解析库的使用

XPath

XML Path Language XML 路径语言

安装lxml库 (支持HTML和XML解析，支持XPath解析方式)
pip3 install lxml

XPath、XQuery 以及 XSLT 函数

Beautiful Soup

pip3 install beautifulsoup4

解析器

Python 标准库 BeautifulSoup(html, “html.parser”) 速度一般，容错能力好
lxml HTML解析器 BeautifulSoup(html, “lxml”) 速度快，容错好
lxml xml解析器 BeautifulSoup(markup, “xml”) 速度快，唯一支持xml
html5lib BeautifulSoup(markup, “html5lib”) 容错性高，速度慢

引入BeautifulSoup

```
from bs4 import BeautifulSoup
```

获取方法

```
soup = BeautifulSoup(html, "lxml") # 试用lxml解析器构造beautifulsoup
print(soup.prettify())  # 取网页缩进格式化输出
print(soup.title.string) # 取网页title内容
print(soup.head)
print(soup.p)

# 获取节点的名字
print(soup.title.name)
# 获取节点属性
soup.img.attrs["src"]
```

```
print(soup.p.attrs)
print(soup.p.attrs["name"])
print(soup.p["class"])
# 获取节点包含的内容
print(soup.p.string)
```

asdf<*span*>*asdfasdfasdfasdfadsfad*

嵌套选择

<head>
  <title>this is title</title>
</head>

```
# soup的节点都为 bs4.element.Tag类型,可以继续选择
print(soup.head.title.string)
```

关联选择

有些元素没有特征定位，可以先选择有办法定位的，然后以这个节点为准选择它的子节点、父节点、兄弟节点等

```
print(soup.p.contents) # 取p节点下面所有子节点列表
print(soup.p.descendants) #取p节点所有子孙节点
print(soup.a.parent) # 取父节点
print(soup.a.parents) # 取所有祖先节点
print(soup.a.next_sibling) # 同级下一节点
print(soup.a.previous_sibling) # 同级上一节点
print(soup.a.next_siblings) # 同级所有后面节点
print(soup.a.previous_siblings) # 同级所有前面节点
print(list(soup.a.parents)[0].attrs['class'])
```

方法选择器

根据属性和文本进行查找

</*li*>

<*li*>*jjj*<*li*>

```
print(soup.find_all(name="ul"))

for ul in soup.find_all(name="ul"):
  print(ul.find_all(name="li"))
  for li in ul.find_all(name="li"):
    print(li.string)

soup.find_all(attrs={"id": "list-1"})
```

css 选择器

<p id="p1" class="panel"><p class=""><*p*><p>

```
soup.select('.panel .panel_heading')
soup.select('ul li')
soup.select('#id1 .element')
```