本节知识点概述

使用Xpath解析豆瓣短评

解析神器Xpath

有哪些常用的网页解析

Xpath的安装

Xpath的使用

从浏览器中复制

手写Xpath

实战环节

使用Xpath解析网页

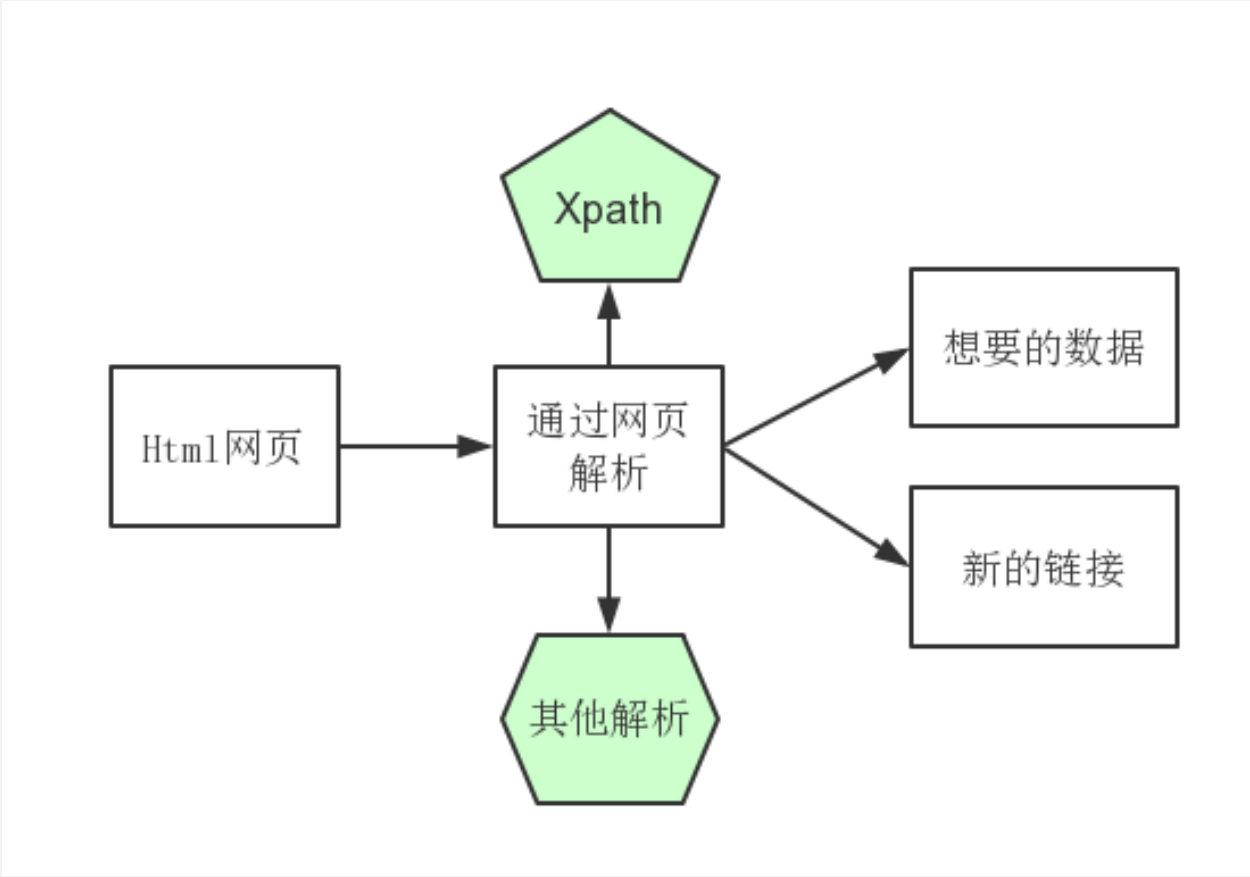
更简洁的写法

解析神器Xpath

你是否还记得
上节课的内容

获取数据	• Requests
解析数据	• Xpath
保存数据	• 保存本地

解析神器Xpath



有哪些常用的网页解析

通常情况下，lxml 是抓取数据的最好选择，这是因为该方法既快速又简单。

而正则表达式和Beautiful Soup 只在某些特定场景下有用 。

Xpath的安装

1、使用pip安装

$ pip install lxml

2、下载whl文件

$ pip install “文件名”

Lxml, a binding for the libxml2 and libxslt libraries.

- lxml-3.7.3-cp27-cp27m-win32.whl
- lxml-3.7.3-cp27-cp27m-win_amd64.whl
- lxml-3.7.3-cp34-cp34m-win32.whl
- lxml-3.7.3-cp34-cp34m-win_amd64.whl
- lxml-3.7.3-cp35-cp35m-win32.whl
- lxml-3.7.3-cp35-cp35m-win_amd64.whl
- lxml-3.7.3-cp36-cp36m-win32.whl
- lxml-3.7.3-cp36-cp36m-win_amd64.whl
- lxml-3.8.0-cp27-cp27m-win32.whl
- lxml-3.8.0-cp27-cp27m-win_amd64.whl
- lxml-3.8.0-cp34-cp34m-win32.whl
- lxml-3.8.0-cp34-cp34m-win_amd64.whl
- lxml-3.8.0-cp35-cp35m-win32.whl
- lxml-3.8.0-cp35-cp35m-win_amd64.whl
- lxml-3.8.0-cp36-cp36m-win32.whl
- lxml-3.8.0-cp36-cp36m-win_amd64.whl

Xpath的使用


```
from lxml import etree

html =‘’’
#省略
’’’
s = etree.HTML(html)

print(s.xpath())
```

https://zhuanlan.zhihu.com/p/25572729

Xpath的使用

- 获取文本内容用 text()
- 获取注释用 comment()
- 获取其它任何属性用@xx，如：
  - @href
  - @src
  - @value

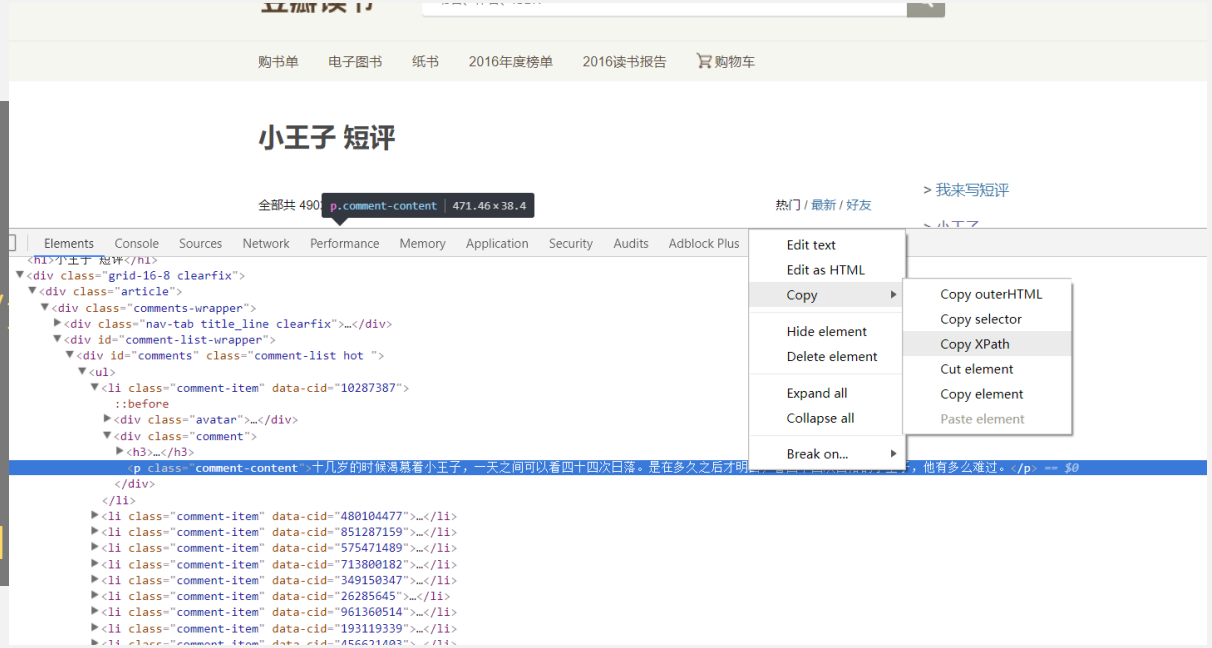
- 想要获取某个标签下所有的文本（包括子标签下的文本），使用string
  - 如 123<a>来获取我啊 ，这边如果想要得到的文本为"123来获取我啊"，则需要使用string
- starts-with 匹配字符串前面相等
- contains 匹配任何位置相等

从浏览器复制

```
import requests
from lxml import etree

url = 'https://book.douban.com/subject/
r = requests.get(url).text

s = etree.HTML(r)
print(s.xpath('//*[@id="comments"]/ul/l
```



手写Xpath

```
import requests
from lxml import etree

url = 'https://book.douban.com/su
r = requests.get(url).text

s = etree.HTML(r)
print(s.xpath('//div[@class="comm
```

实战环节

```
import requests
from lxml import etree

url = 'https://book.douban.com/subject/1084336/comments/'
r = requests.get(url).text

s = etree.HTML(r)

print(s.xpath('//*[@id="comments"]/ul/li[1]/div[2]/p/text()'))

print(s.xpath('//*[@id="comments"]/ul/li[2]/div[2]/p/text()'))

print(s.xpath('//*[@id="comments"]/ul/li[3]/div[2]/p/text()'))
```

通用的规律是什么？

更简洁的写法

```
import requests
from lxml import etree

url = 'https://book.douban.com/subject/1084336/comments/'
r = requests.get(url).text

s = etree.HTML(r)

print(s.xpath('//div[@class="comment"]/p/text()'))
```

思考

为什么这样更简洁

完成作业

- 安装lxml
- 使用xpath解析豆瓣网页
- 勇敢尝试你想爬取的网站

你已经掌握了爬虫基本原理
去犯错吧~

温馨提示

- ✓ 爬取互联网公开数据
- ✓ 尽量放慢你的速度
- ✓ 尽量遵循robots协议
- ✓ 不要用于商业用途
- ✓ 不要公布爬虫程序与数据

爬虫工程师

更多数据科学课程，上DC学院： class.pkbigdata.com

关注 DataCastle

关注造数

造数