

爬虫数据提取——Xpath

爬虫数据提取——方法总结

我们获取了想要的html页面之后，接下来的问题就是如何将我们需要的数据给提取下来，一般来说有三种方式，分别是Xpath语法，正则表达式和bs4库。

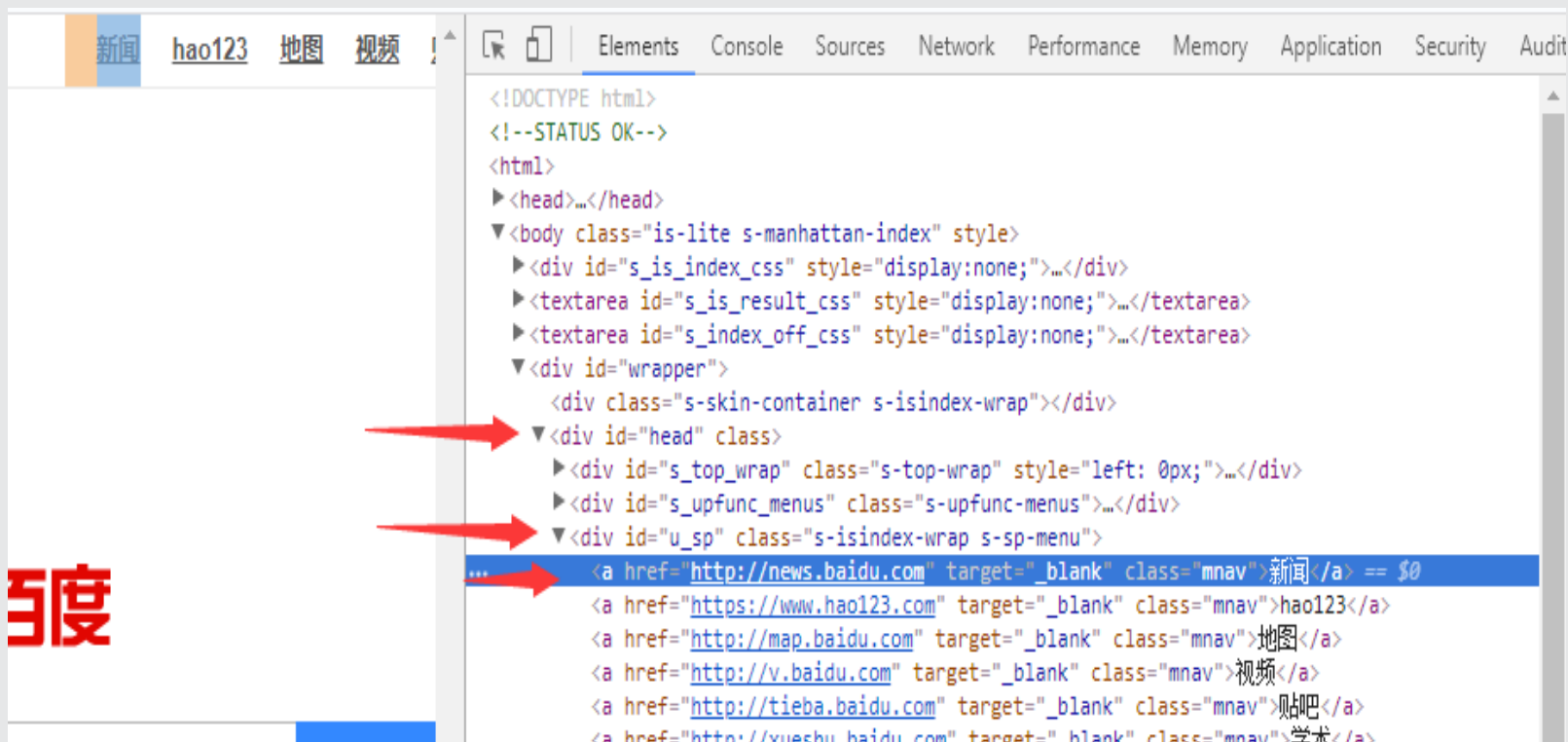
解析方式	解析速度	难度
Xpath	快	中等
bs4	慢	容易
re(正则表达式)	最快	困难

一般来说，解析越快的，用起来肯定越难，解析越慢的，用起来肯定更简单一些，得到了一方面的性能，就要损失一些东西，所谓“鱼与熊掌不可兼得”。

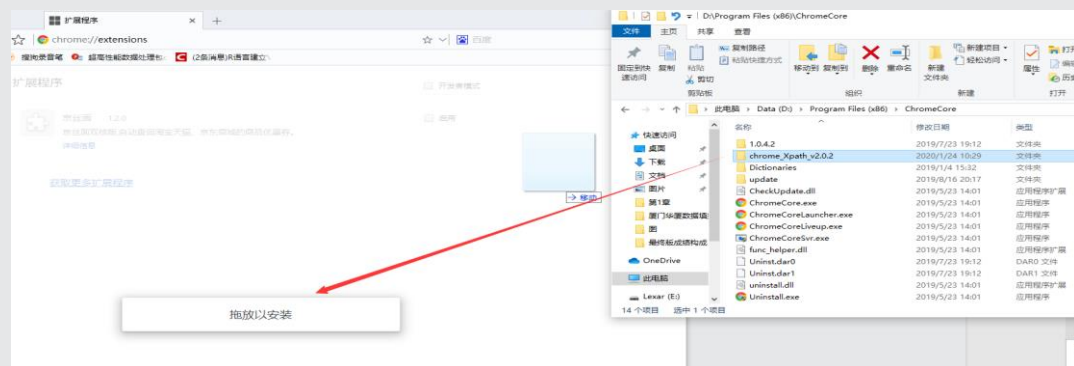
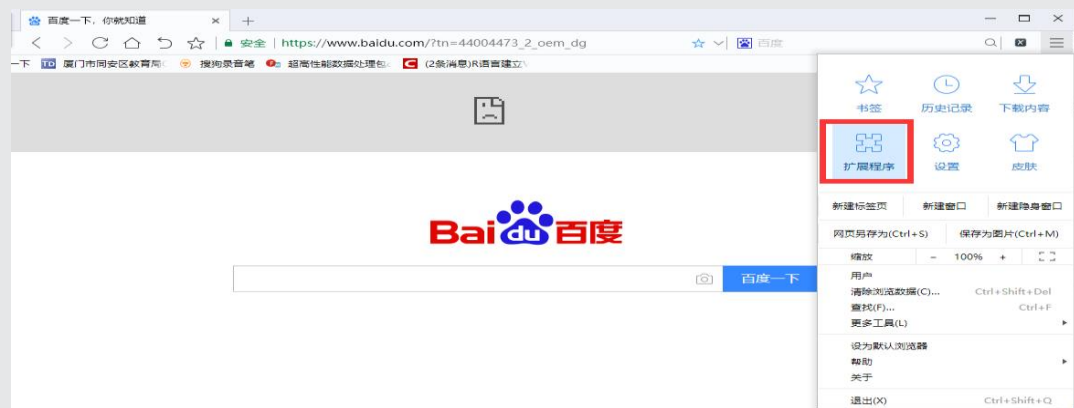
爬虫数据提取——Xpath环境配置

Xpath (XML Path Language) 是一门在XML和HTML文档中查找信息的语言, 可用在XML和HTML文档中对元素和属性进行遍历。简单来说, 我们的数据是超文本数据, 想要获取超文本数据里面的内容, 就要按照一定规则来进行数据的获取, 这种规则就叫做Xpath语法。

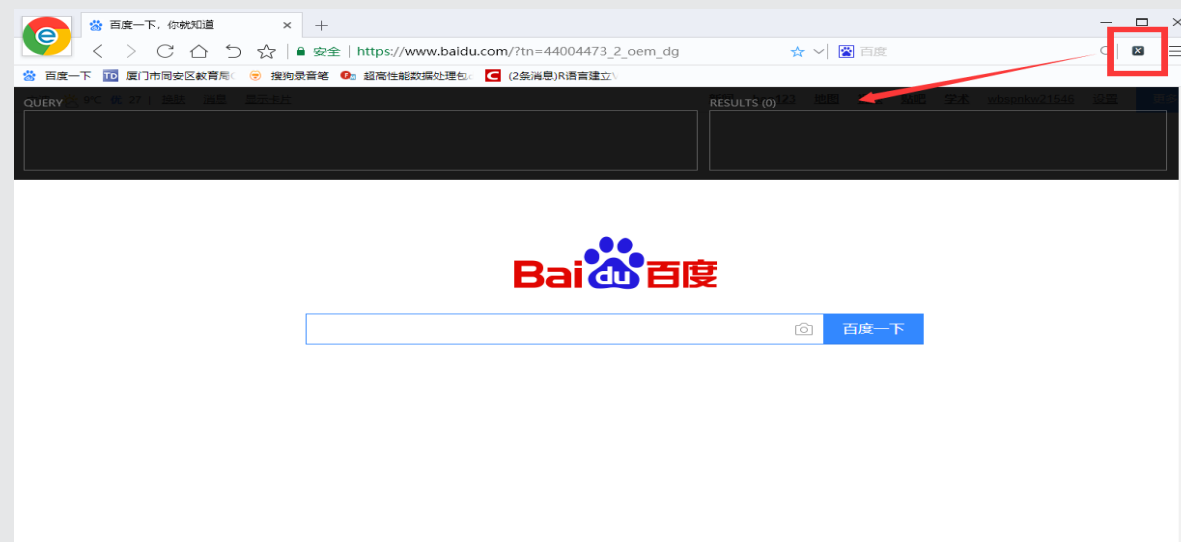
XPath 用于在 HTML 文档中通过元素【HTML标签】和属性【HTML标签的属性】进行数据的定位。



Xpath环境配置安装

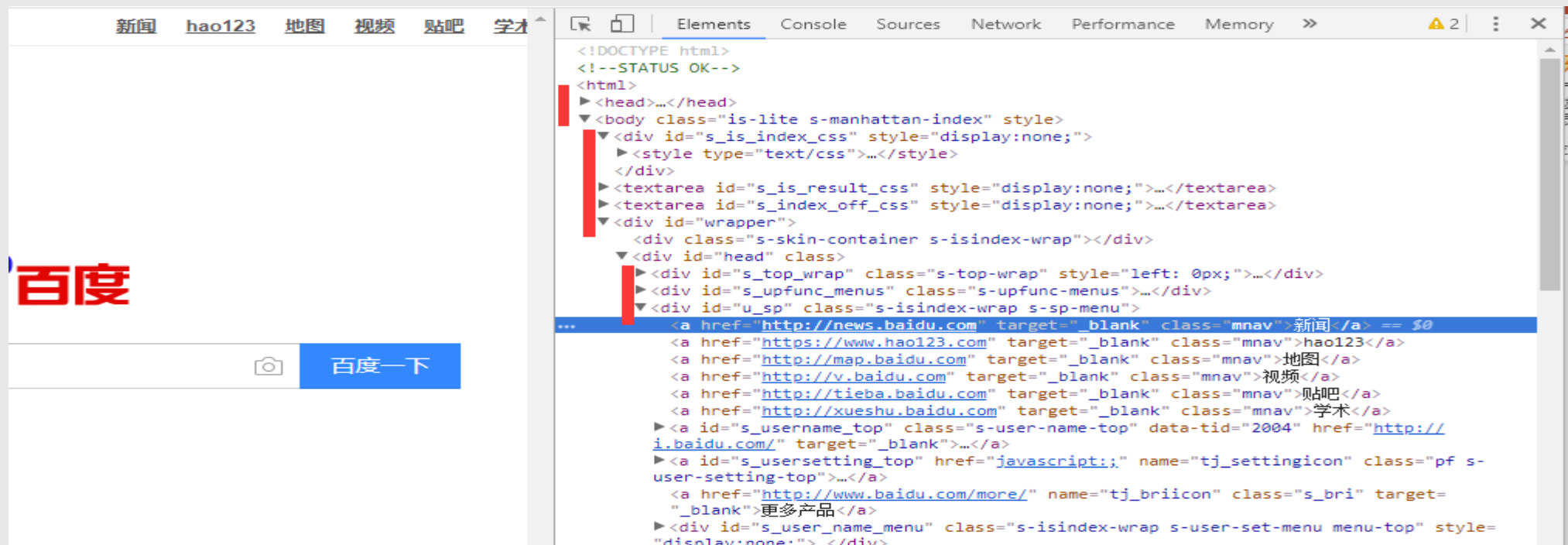


注：chrome_Xpath_v2.0.2文件的删除或者移动，都会导致Xpath无法使用，请注意存放位置和误删！



Xpath语法选取数据逻辑

HTML页面是由标签构成的，这些标签就像整个族谱一样排列有序



总结：所有的HTML标签都有很强的层级关系，正是基于这种层级关系，Xpath语法能够选择出我们想要的数据。

```

<html>
  <head>...</head>
  <body class="is-lite s-manhattan-index" style>
    <div id="s_is_index_css" style="display:none;">
      <style type="text/css">...</style>
    </div>
    <textarea id="s_is_result_css" style="display:none;">...</textarea>
    <textarea id="s_index_off_css" style="display:none;">...</textarea>
    <div id="wrapper">
      <div class="s-skin-container s-isindex-wrap"></div>
      <div id="head" class>
        <div id="s_top_wrap" class="s-top-wrap" style="left: 0px;">...</div>
        <div id="s_upfunc_menus" class="s-upfunc-menus">...</div>
        <div id="u_sp" class="s-isindex-wrap s-sp-menu">
          ...
          <a href="http://news.baidu.com" target="_blank" class="mnav">新闻</a> == $0
          <a href="https://www.hao123.com" target="_blank" class="mnav">hao123</a>
          <a href="http://map.baidu.com" target="_blank" class="mnav">地图</a>
          <a href="http://v.baidu.com" target="_blank" class="mnav">视频</a>
          <a href="http://tieba.baidu.com" target="_blank" class="mnav">贴吧</a>
          <a href="http://xueshu.baidu.com" target="_blank" class="mnav">学术</a>
          <a id="s_username_top" class="s-user-name-top" data-tid="2004" href="http://i.baidu.com/" target="_blank">...</a>
          <a id="s_usersetting_top" href="javascript:;" name="tj_settingicon" class="pf s-user-setting-top">...</a>
          <a href="http://www.baidu.com/more/" name="tj_briicon" class="s_bri" target="_blank">更多产品</a>
          <div id="s_user_name_menu" class="s-isindex-wrap s-user-set-menu menu-top" style="display:none;">...</div>
          <div id="s_user_setting_menu" class="s-isindex-wrap s-user-set-menu menu-top" style="display:none;">...</div>
        </div>
        <style>...</style>
        <div class="clear"></div>
        <div id="head_wrapper" class="s-isindex-wrap head_wrapper s-title-img s-ps-islite">...</div>
        <div id="s_wrap" class="s-isindex-wrap">...</div>
        <div id="bottom_layer" class="s-bottom-layer">...</div>
      </div>
    <div class="s_tab" id="s_tab">...</div>
    <div id="wrapper_wrapper"></div>
  </body>
</html>

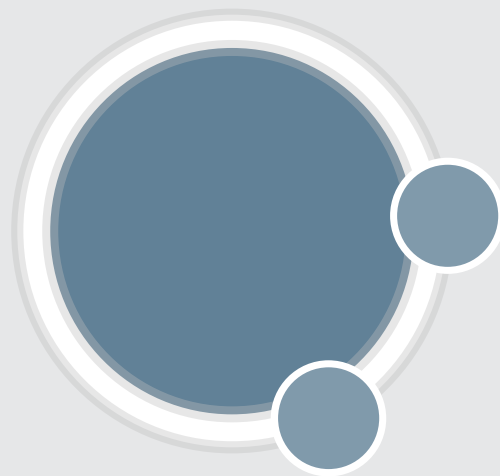
```

xxx >> 太爷爷 >> 爷

爷 >> 爸爸 >> 儿子 >>

孙子 >> xxx

【如果在中间有分叉，可以理解为家族的旁系，一个爷爷有两个儿子，然后分家了】

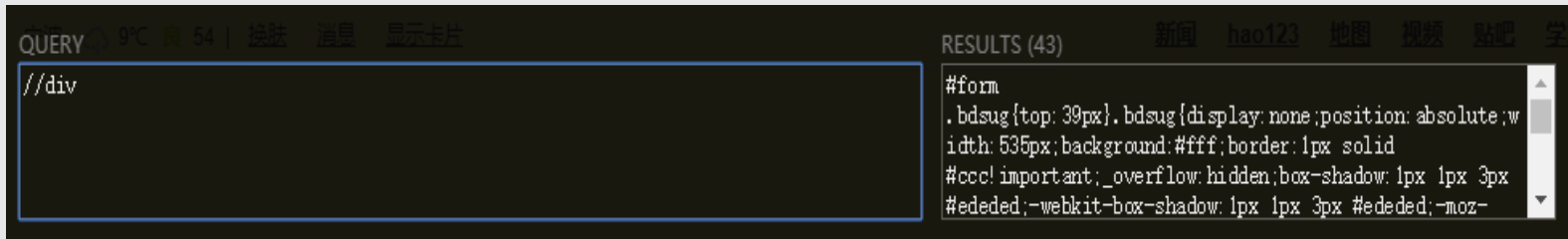


Xpath语法

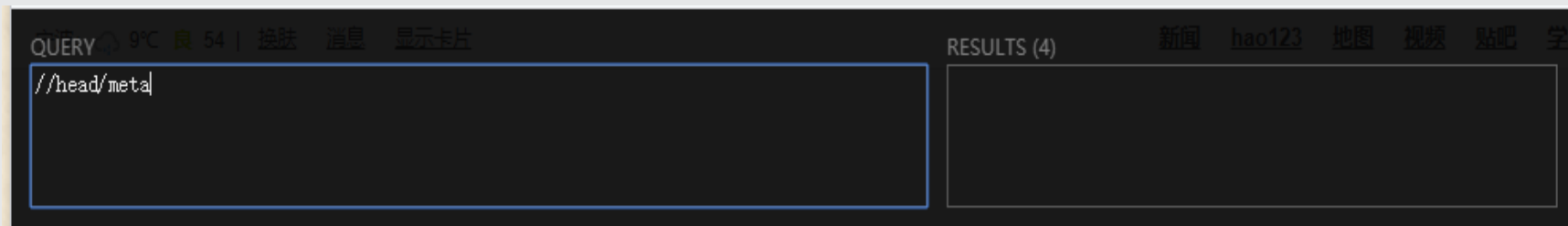
1.节点选取

表达式	描述	用法	说明
nodename	选取此节点的所有子节点	div	选取div下的所有标签
//	从全局节点中选择节点，任意位置均可	//div	选取整个HTML页面的所有div标签
/	选取某个节点下的节点	//head/title	选取head标签下的title标签
@	选取带某个属性的节点	//div[@id]	选择带有id属性的div标签
.	当前节点下	./span	选择当前节点下的span标签【代码中威力强大】

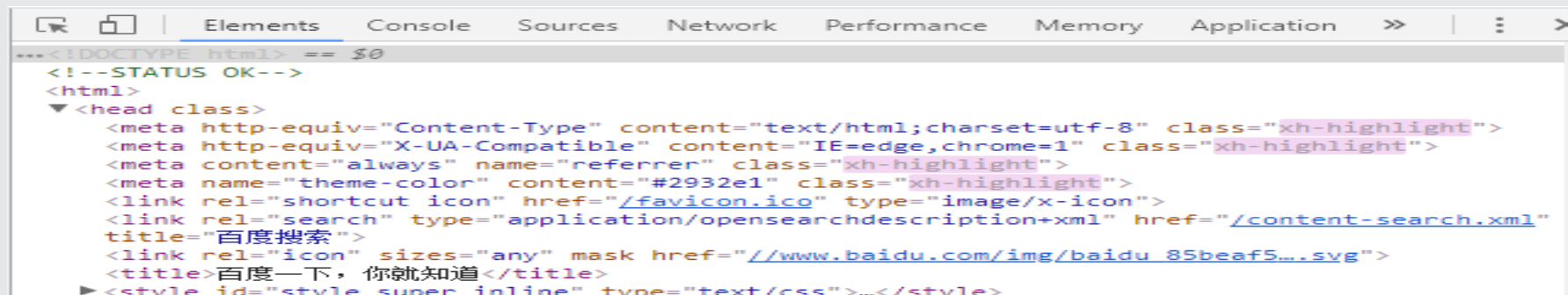
以百度为例，我们来学习Xpath语法。



// 表示全局搜索，//div即全局搜索所有的div标签



/表示标签下的局部搜索，//head/meta即表示head标签下所有的meta标签



观察html页面，将head下的“百度一下，你就知道”提取出来。

```
<!DOCTYPE html>
<!--STATUS OK-->
<html>
  <head class>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" class>
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" class>
    <meta content="always" name="referrer" class>
    <meta name="theme-color" content="#2932e1" class>
    <link rel="shortcut icon" href="/favicon.ico" type="image/x-icon">
    <link rel="search" type="application/opensearchdescription+xml" href="/content-search.xml"
    title="百度搜索"> == $0
    <link rel="icon" sizes="any" mask href="//www.baidu.com/img/baidu_85beaf5....svg">
    <title class="xh-highlight">百度一下，你就知道</title>
    <style id="style_super_inline" type="text/css">...</style>
    <style id="style_super_head_inline" type="text/css">...</style>
```

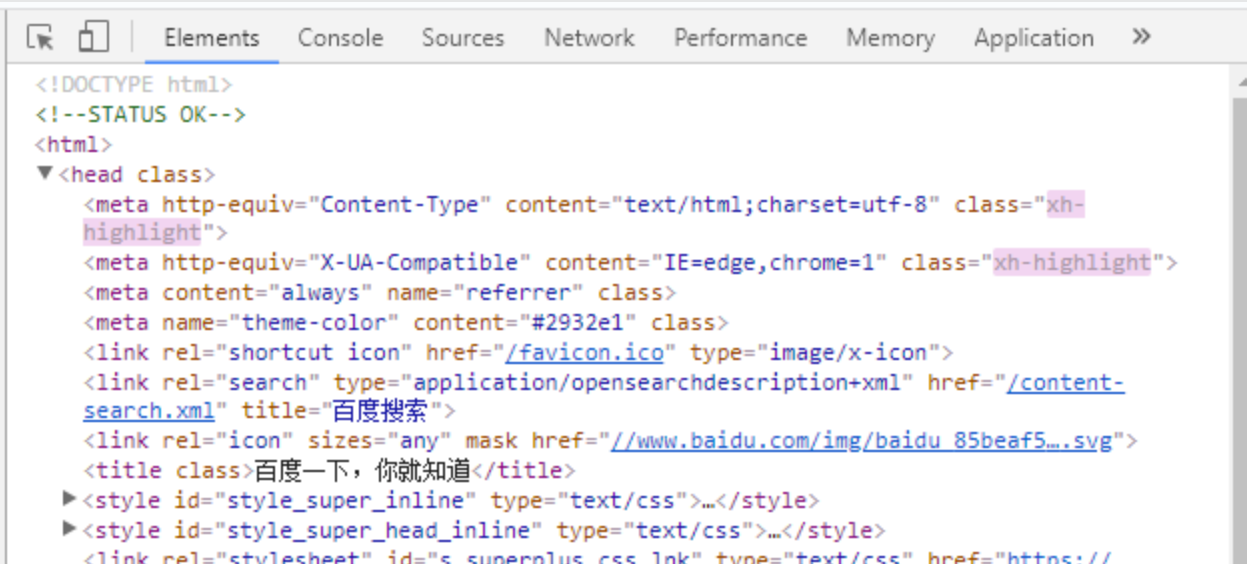
```
QUERY 9°C 54 | 换肤 消息 显示卡片 RESULTS (1) 新闻 hao123 地图 视频 贴吧 学
//head/title
百度一下，你就知道
```



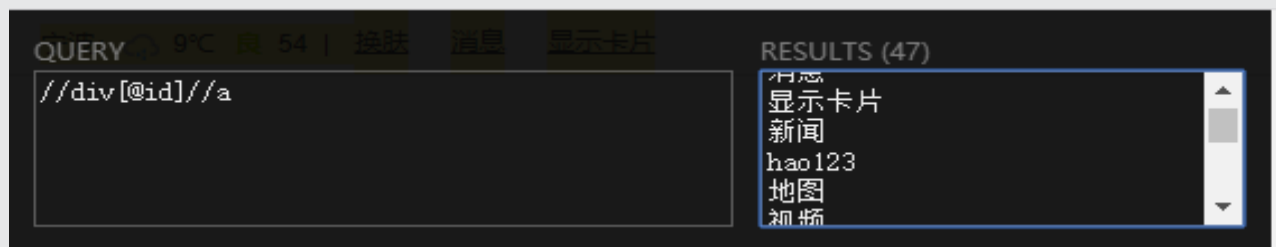
用[]括号内添加@，将标签属性填入进去，将含有该标签属性的部分提取出来。



//head/meta[@http-equiv], 将head下的meta下的所有http-equiv标签的内容提取出来。



思考：如何将百度页面右上方的这些信息通过Xpath提取出来？



2.谓语句

表达式	用法说明
<code>//head/meta[1]</code> <code>//head/meta[k]</code>	选择所有head下的第一个meta标签 选择所有head下的第k个meta标签
<code>//head/meta[last()]</code>	选择所有head下的最后一个meta标签
<code>//head/meta[position()<3]</code>	选择所有head下的前两个meta标签
<code>//div[@id]</code>	选择带有id属性的div标签
<code>//div[@id='u1']</code>	选择所有拥有id=u1的div标签

注：[k]表示当前位置下的第k个标签

QUERY 9°C 良 54 | 换肤 消息 显示卡片

//head/meta[1]

RESULTS (1)

Elements Console Sources Network Performance Memory Application

<!DOCTYPE html>
<!--STATUS OK-->
...<html> == \$0
▼<head class>
 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" class="xh-highlight">
 <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" class>
 <meta content="always" name="referrer" class>
 <meta name="theme-color" content="#2932e1" class>
 <link rel="shortcut icon" href="/favicon.ico" type="image/x-icon">
 <link rel="search" type="application/opensearchdescription+xml" href="/content-search.xml" title="百度搜索">
 <link rel="icon" sizes="any" mask href="//www.baidu.com/img/baidu_85beaf5...svg">
 <title class>百度一下，你就知道</title>
▶<style id="style_super_inline" type="text/css">...</style>
▶<style id="style_super_head_inline" type="text/css">...</style>

注：与python中记数的区别。Python中第1个是0还是1？

[last()]

QUERY 9°C 54 | 换肤 消息 显示卡片

RESULTS (1)

```
//head/meta[last()]
```

Baidu

Elements Console Sources Network Performance Memory Application >>

```
<!DOCTYPE html>
<!--STATUS OK-->
...<html> == $0
▼<head class>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" class>
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" class>
  <meta content="always" name="referrer" class>
  <meta name="theme-color" content="#2932e1" class="xh-highlight">
  <link rel="shortcut icon" href="/favicon.ico" type="image/x-icon">
  <link rel="search" type="application/opensearchdescription+xml" href="/content-search.xml" title="百度搜索">
  <link rel="icon" sizes="any" mask href="//www.baidu.com/img/baidu_85beaf5...svg">
  <title class>百度一下, 你就知道</title>
  ▶<style id="style_super_inline" type="text/css">...</style>
  ▶<style id="style_super_head_inline" type="text/css">...</style>
```

QUERY 9°C 54 | 换肤 消息 显示卡片

RESULTS (2)

```
//head/meta[position()<3]
```

Baidu

Elements Console Sources Network Performance Memory Application >>

```
<!DOCTYPE html>
<!--STATUS OK-->
...<html> == $0
▼<head class>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" class="xh-highlight">
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" class="xh-highlight">
  <meta content="always" name="referrer" class>
  <meta name="theme-color" content="#2932e1" class>
  <link rel="shortcut icon" href="/favicon.ico" type="image/x-icon">
  <link rel="search" type="application/opensearchdescription+xml" href="/content-search.xml" title="百度搜索">
  <link rel="icon" sizes="any" mask href="//www.baidu.com/img/baidu_85beaf5...svg">
  <title class>百度一下, 你就知道</title>
```

思考：如何将百度页面右上方的这些信息通过Xpath提取出来？



QUERY

//div[@id="u_sp"]

设为首页 关于百度 About Baidu 百度推广 使用百度前必读 意见反馈 帮助中心

RESULTS (1)

新闻hao123地图视频贴吧学术wbspnkW21546设置更多产品个人中心帐号设置退出搜索设置高级搜索搜索历史意见反馈

显示卡片 ©2020 Baidu (京)经营性-2017-0020

`//div[@id="u_sp"]`
将所有的id=“u_sp”的内容提取，因此只有一个结果

QUERY

//div[@id="u_sp"]/a

设为首页 关于百度 About Baidu 百度推广 使用百度前必读 意见反馈 帮助中心

RESULTS (9)

新闻
hao123
地图
视频
贴吧

显示卡片 ©2020 Baidu (京)经营性-2017-0020

`//div[@id="u_sp"]/a`
将所有的id=“u_sp”的内容中所有a标签提取，因此有九个结果

QUERY

//div[@id="u_sp"]/a[1]

设为首页 关于百度 About Baidu 百度推广 使用百度前必读 意见反馈 帮助中心

RESULTS (1)

新闻

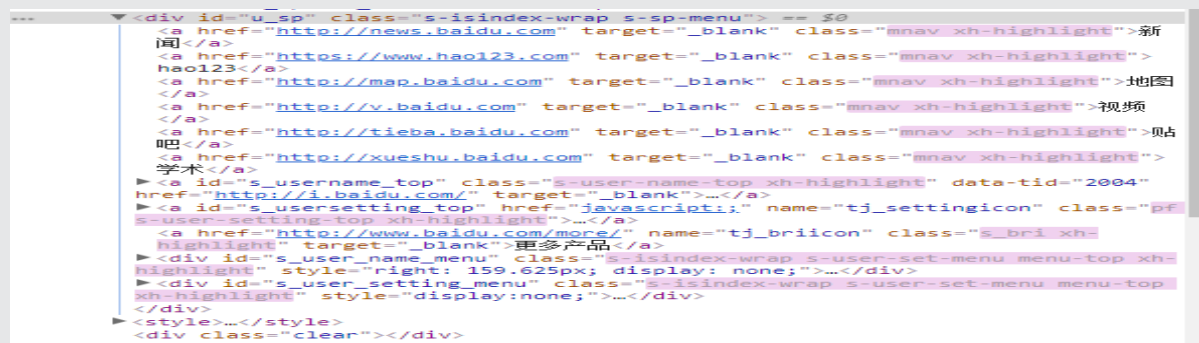
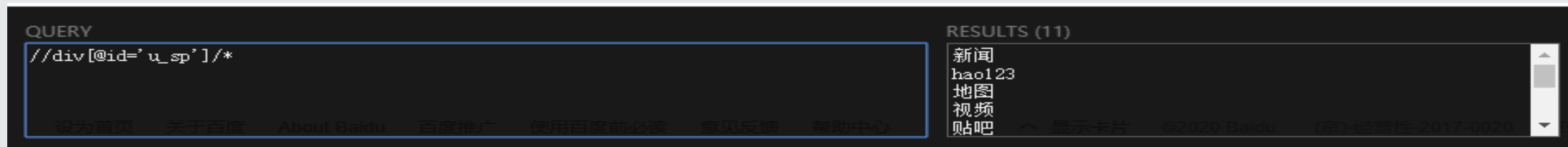
显示卡片 ©2020 Baidu (京)经营性-2017-0020

新闻

3.通配符

通配符	描述	示例	结果
*	匹配任意节点	//div[@id='u1']/*	选择所有拥有id=u1的div标签下的所有节点
@*	匹配节点中的任何属性	//meta[@*] //a[@*]	选择所有拥有属性的meta标签 选择所有拥有属性的a标签

//div[@id= 'u_sp ']/*, 在div中所有id= 'u_sp '的内容

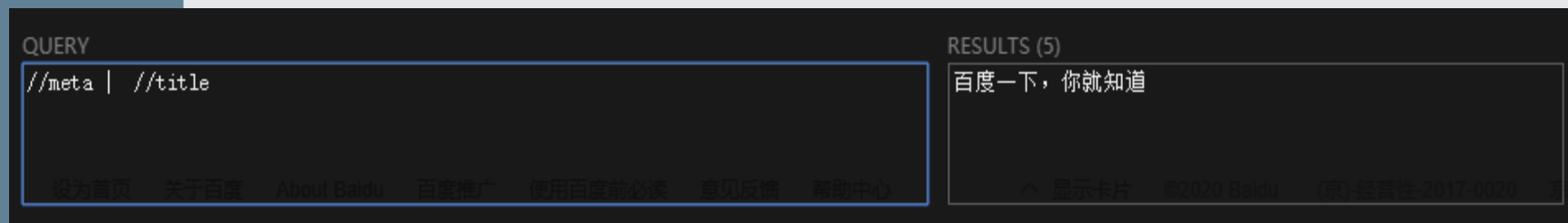


//title[@*]



注：若标签下，没有标签属性，则不会被提取出来，会以空格返回

4.选取多个路径



使用 | 来表示选择多个路径：
eg: //meta | // title -> > 选择所有的meta和title.

等价于“和”或者“或”的效果。

注意：上述的结果中有5个。理解 | 的等价效果。

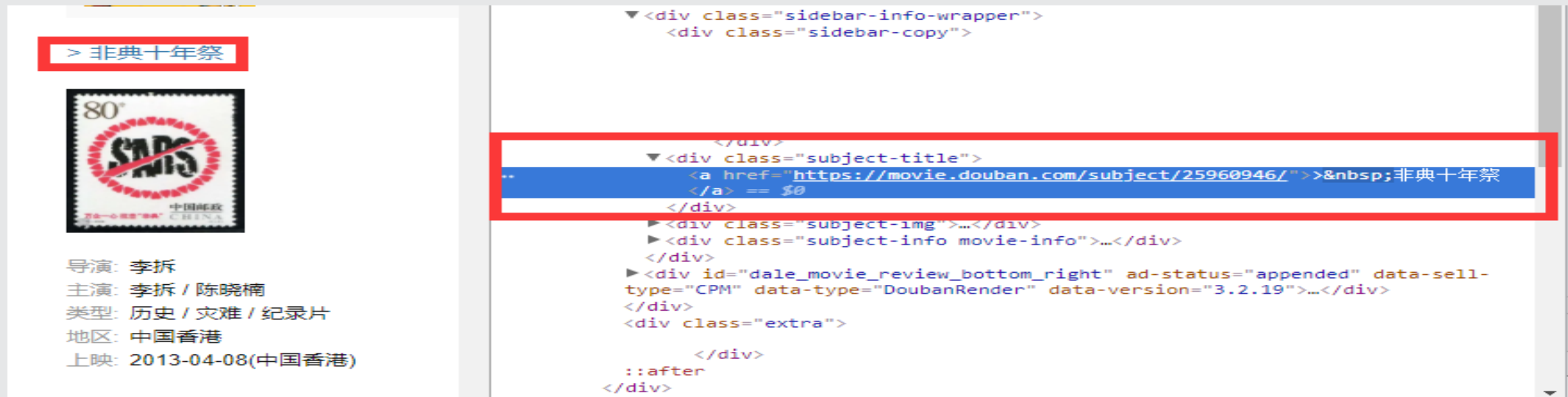
Xpath语法应用实战

我们在豆瓣电影的影评中，随意找一部电影或者影评来抓取相关信息。

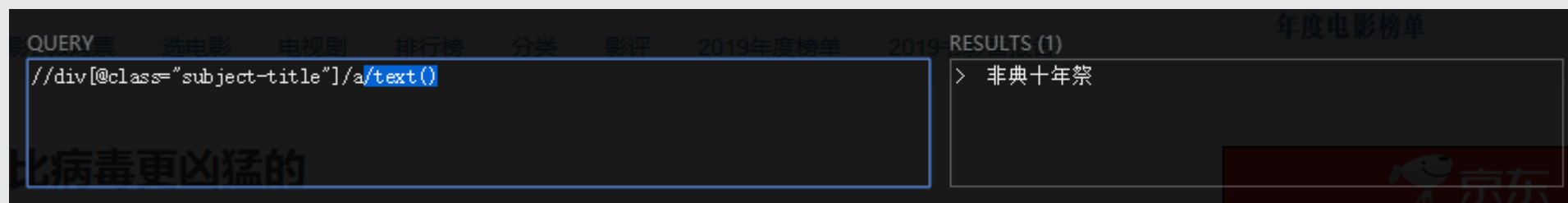


<https://movie.douban.com/review/12181209/>





- 抓取电影名
- 在html定位到相关位置
- 如果这个标签是在html中唯一的，那么我们就可以利用Xpath将它抓下来，



注：在代码中我们需要添加`/text()`获取值，这里并不是必须的。

抓取导演和主演等相关信息

导演: 李拆

主演: 李拆 / 陈晓楠

类型: 历史 / 灾难 / 纪录片

地区: 中国香港

上映: 2013-04-08(中国香港)

```
><div class="subject-img">...</div>
▼<div class="subject-info movie-info"> == $0
  ▼<ul class="info-list">
    ▼<li class="info-item">
      <span class="info-item-key">导演:</span>
      <span class="info-item-val">李拆</span>
    </li>
    ▼<li class="info-item">
      <span class="info-item-key">主演:</span>
      <span class="info-item-val">李拆 / 陈晓楠</span>
    </li>
    ▶<li class="info-item">...</li>
    ▶<li class="info-item">...</li>
    ▶<li class="info-item">...</li>
  </ul>
</div>
</div>
```

QUERY 选 选电影 电视剧 排行榜 分类 影评 2019年度榜单 2019-RESULTS (5)

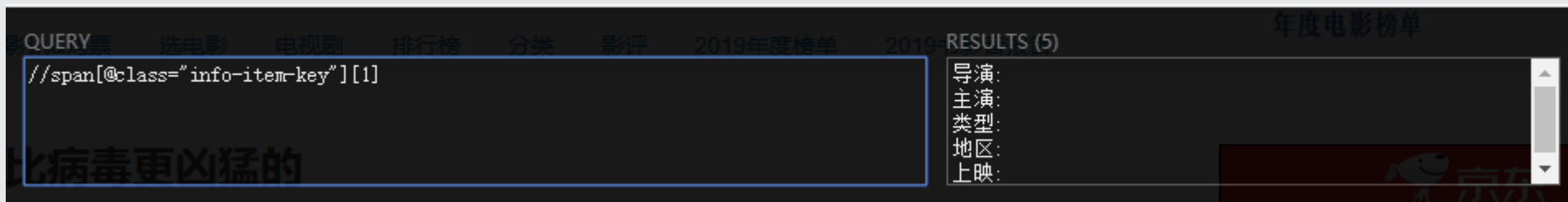
//span[@class="info-item-key"]

导演:
主演:
类型:
地区:
上映:

比病毒更凶猛的

年度电影榜单

如果我想把导演提取出来，怎么办？



```
▼<ul class="info-list">
  ▼<li class="info-item">
    <span class="info-item-key xh-highlight">导演:</span>
    <span class="info-item-val">李拯</span>
  </li>
  ▼<li class="info-item">
    <span class="info-item-key xh-highlight">主演:</span>
    <span class="info-item-val">李拯 / 陈晓楠</span>
  </li>
  ▼<li class="info-item">
    <span class="info-item-key xh-highlight">类型:</span>
    <span class="info-item-val">历史 / 灾难 / 纪录片</span>
  </li>
  ▶<li class="info-item">...</li> == $0
  ▶<li class="info-item">...</li>
</ul>
</div>
</div>
```

因为这里span标签的属性全部一致，因此提取出错。

即：如果这个标签在html中不唯一呢？

儿子闯祸了，怎么办？找双方家长解决问题啊！

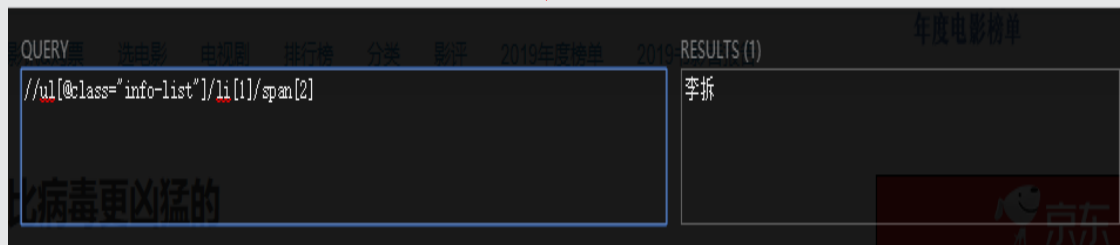
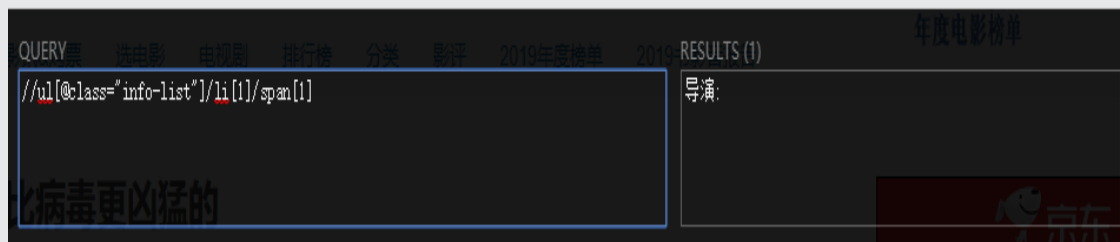
在html页面中，严格的层级关系，能够帮助我们定位和提取。

也就是爸爸解决不了找爷爷，爷爷解决不了，找太爷爷。

```

</div>
▼<div class="subject-title">
  <a href="https://movie.douban.com/subject/25960946/" class>
    >&nbsp;&nbsp;&nbsp;非典十年祭</a>
</div>
▶<div class="subject-img">...</div>
▼<div class="subject-info movie-info">
...
  ▼<ul class="info-list"> == $0
    ▼<li class="info-item">
      ▼<span class="info-item-key xh-highlight">导演:</span>
      <span class="info-item-val">李拆</span>
    </li>
    ▼<li class="info-item">
      <span class="info-item-key">主演:</span>
      <span class="info-item-val">李拆 / 陈晓楠</span>
    </li>
    ▼<li class="info-item">
      <span class="info-item-key">类型:</span>
      <span class="info-item-val">历史 / 灾难 / 纪录片</span>
    </li>
    ▶<li class="info-item">...</li>
    ▶<li class="info-item">...</li>
  </ul>
</div>
</div>
▶<div id="dale_movie_review_bottom_right" ad-status="appended" data-
sell-type="CPM" data-type="DoubanRender" data-version="3.2.19" class>
...</div>

```



其他依次类推

QUERY 选电影 电视剧 排行榜 分类 影评 2019年度榜单 2019-RESULTS (1) 年度电影榜单

//ul[@class="info-list"]/li[2]/span[1]

主演:

比病毒更凶猛的

李拆 / 陈晓楠

类型:

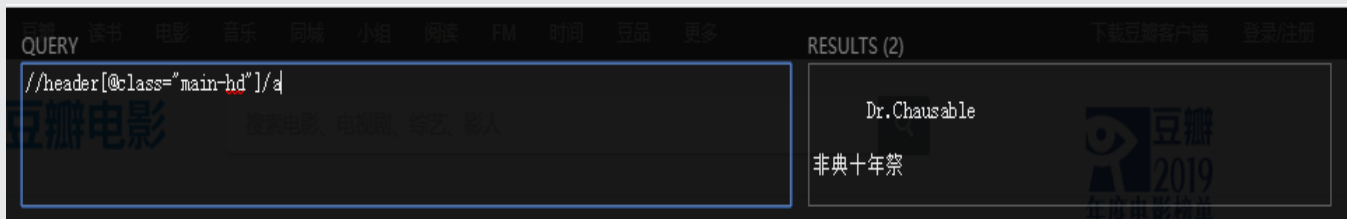
历史 / 灾难 / 纪录片

比病毒更凶猛的

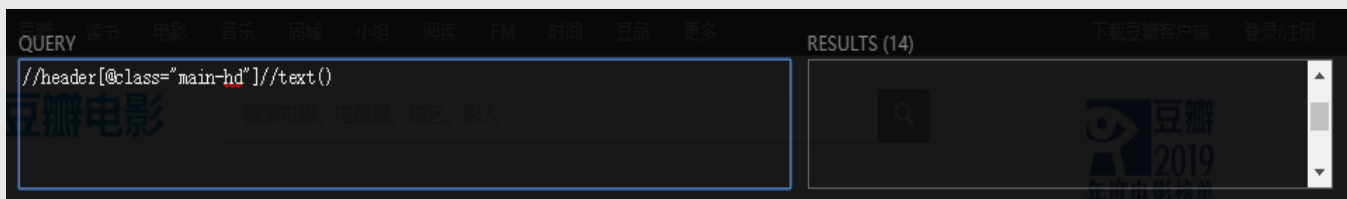
比病毒更凶猛的

比病毒更凶猛的

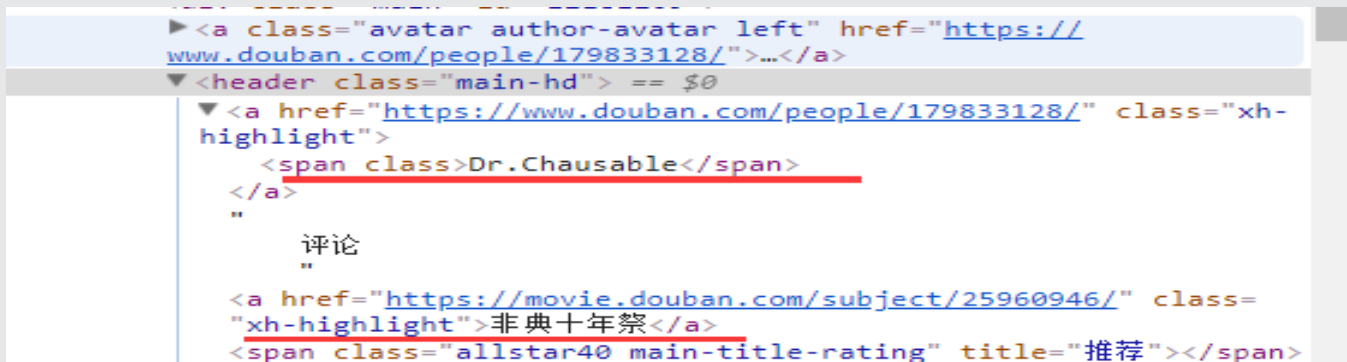
获取评论者和评分



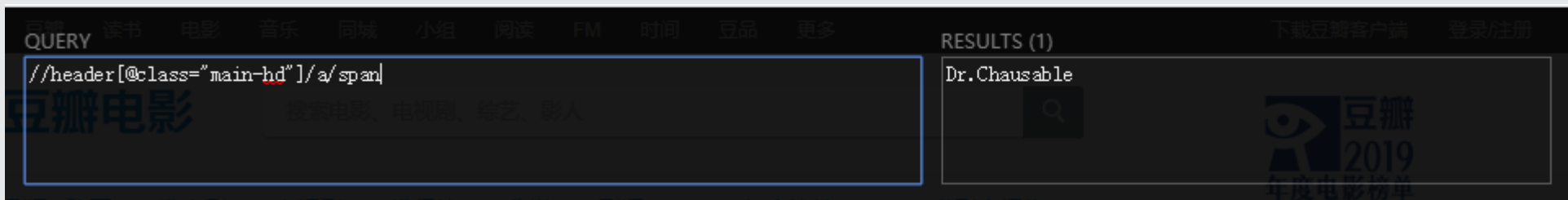
问题和刚才一样



发现都是该标签下的
a标签下的内容，修
改XPath代码



获取的两个内容中，第
一个含在span标签中，
第二个则在a标签中，因
此直接再加个子标签即可



比病毒更凶猛的



Dr.Chausable

评论 非典十年祭



2020-01-22 19:14:30

span.allstar40.main-title-rating | 55x11

码了一个多小时的文字，死活发不出来，修改拼音也发不出来，甚至改成截图之后还是修改了几次才能发出来，不知道触及了哪些mingan词。你瓣就是这个尿性。

正文内容，按顺序一张接一张

中国联通 VPN

下午 7:17

90%

< 所有 iCloud



非典，SARS事件的通俗叫法，SARS事件

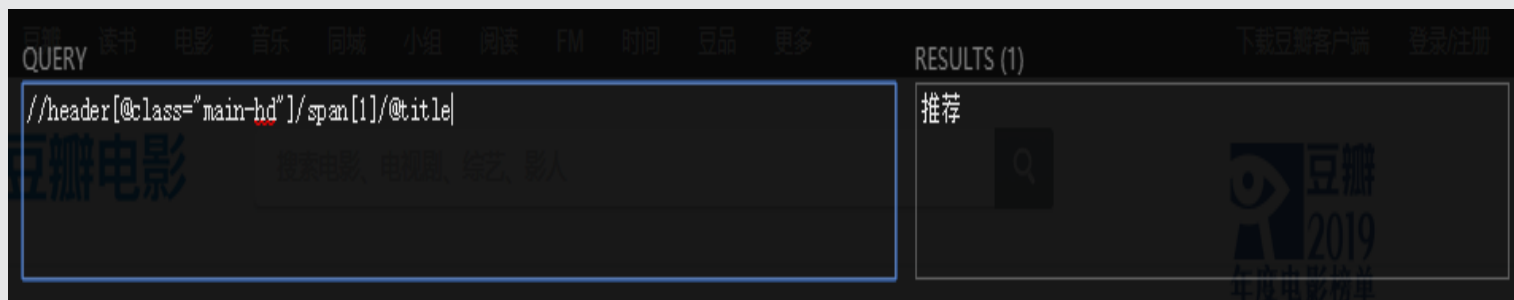


> 非典十年祭

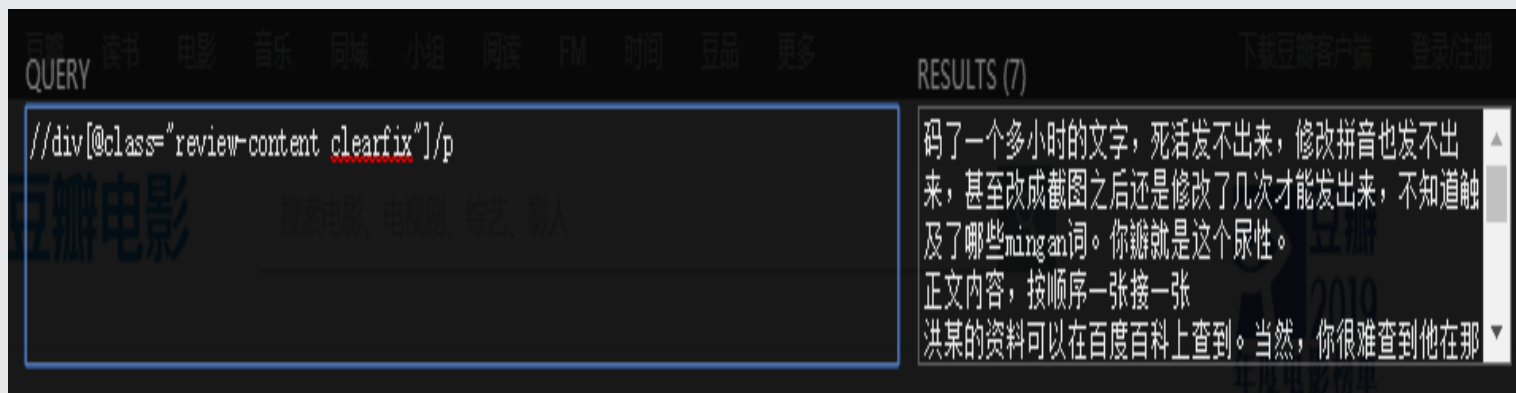
```
<script id="suggesult" type="text/x-jquery-templ"></script>
<script src="//img3.doubanio.com/dae/accounts/resources/f5f3d66/movie/
bundle.js" defer="defer"></script>
<div id="wrapper" class="movie-content review-wrapper">
  <div id="content" class="
    <div class="grid-16-8 clearfix">
      <div class="article">
        <h1></h1>
        <div class="
          <div class="main" id="12181209">
            <a class="avatar author-avatar left" href="https://
            www.douban.com/people/179833128/"></a>
            <header class="main-hd">
              <a href="https://www.douban.com/people/179833128/" class="
              <span class="xh-highlight">Dr.Chausable</span>
            </a>
            "
            评论
            "
            <a href="https://movie.douban.com/subject/25960946/" class="非
            典十年祭"></a>
            <span class="allstar40 main-title-rating" title="推荐"></span>
            <span class="main-title-hide">4</span>
            <span content="2020-01-22" class="main-meta">2020-01-22
            19:14:30</span>
            <script type="application/ld+json"></script>
          </header>
```

这里的评分我们看到的是星星，但是在html中显示的是“推荐”。这个推荐并不是标签中的值，而是标签中的属性

如何提取标签中的属性的内容？



我们利用【/@属性】来获取，这是未来获取内容的重要手段
获取影评



作业：
换一个豆瓣电影的页面你来试试抓取。

<https://movie.douban.com/subject/27119724/?tag=%E7%83%AD%E9%97%A8&from=gaia>

小丑 Joker (2019)



导演: 托德·菲利普斯
编剧: 托德·菲利普斯 / 斯科特·西尔弗 / 鲍勃·凯恩 / 比尔·芬格 / 杰瑞·罗宾逊
主演: 华金·菲尼克斯 / 罗伯特·德尼罗 / 马克·马龙 / 莎姬·贝兹 / 谢伊·惠格姆 / 更多...
类型: 剧情 / 惊悚 / 犯罪
官方网站: www.jokermovie.net
制片国家/地区: 美国 / 加拿大
语言: 英语
上映日期: 2019-08-31(威尼斯电影节) / 2019-10-04(美国)
片长: 122分钟 / 118分钟(威尼斯电影节)
又名: 小丑起源电影: 罗密欧 / Romeo / Joker Origin Movie
IMDb链接: [tt7286456](https://www.imdb.com/title/tt7286456)

想看 看过 评价: ☆☆☆☆☆
写短评 写影评 分享到

小丑的剧情简介 · · · · · ·

亚瑟·弗兰克是一名以小丑职业为生的普通人，患有精神疾病的他和母亲一同住在哥谭市的一座公寓里，幻想成为脱口秀演员的亚瑟为了这个目标而努力的生活着，但是现实却屡次击败他的梦想，亚瑟渐渐地变得越来越癫狂，某天在地铁上，亚瑟为了自保杀害了几名嘲笑他的人，同时，一个疯狂的想法在亚瑟心灵萌发.....在看似和平的哥谭市，即将发生翻天覆地的巨变。

豆瓣评分

8.7  489088人评价

5星	50.6%
4星	37.1%
3星	10.3%
2星	1.3%
1星	0.8%

好于 97% 犯罪片
好于 98% 惊悚片

推荐



抓取上面的主要信息



示范几个:

QUERY 电影 音乐 同城 小组 阅读 FM 时间 豆品 更多

RESULTS (2)

小丑 Joker (2019)

豆瓣 2019 年度电影榜单

电影名

QUERY 电影 音乐 同城 小组 阅读 FM 时间 豆品 更多

RESULTS (1)

托德·菲利普斯

豆瓣 2019 年度电影榜单

导演

QUERY 电影 音乐 同城 小组 阅读 FM 时间 豆品 更多

RESULTS (3)

小丑 Joker (2019)

剧情
惊悚
犯罪

豆瓣 2019 年度电影榜单

类型

QUERY 电影 音乐 同城 小组 阅读 FM 时间 豆品 更多

RESULTS (1)

小丑 Joker (2019)

亚瑟·弗兰克是一名以小丑职业为生的普通人，患有精神疾病的他和母亲一同住在哥谭市的一座公寓里，幻想成为脱口秀演员的亚瑟为了这个目标而努力的生活着，但是现实却屡次击败他的梦想，亚瑟渐渐

豆瓣 2019 年度电影榜单

剧情简介

lxml库的安装

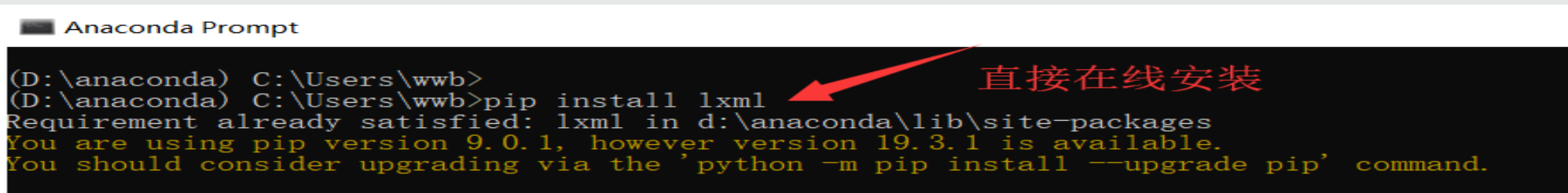
我们在前面课程中将html页面获取后存放在content中，获取的内容仅仅只是一个包含所有内容的html字符串。网页中的html可以用Xpath语法获取数据，但是在content中显然就无法做到了。

即：Xpath语法是无法直接作用于这样的一个字符串进行数据提取的。

因此，我们使用lxml库对html这样的字符串进行解析，将它还原为一个HTML页面，换句话说，Python里面的lxml库只做了这样一件事：将html字符串进行解析，供Xpath语法进行数据提取。

由于lxml是用C语言编写的【这个就是为什么使用xpath语法解析起来速度比较快的原因】，是一款高性能的HTML/XML解析器，我们可以利用之前学习的XPath语法，来快速的定位特定元素以及节点信息。

通过pip install lxml直接在anaconda prompt进行在线安装。



```
Anaconda Prompt
(D:\anaconda) C:\Users\wwb>
(D:\anaconda) C:\Users\wwb>pip install lxml
Requirement already satisfied: lxml in d:\anaconda\lib\site-packages
You are using pip version 9.0.1, however version 19.3.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

直接在线安装

使用lxml中的etree对html进行处理

学习中我们要完成：

1、读取html字符串，这里我们新建一个字符串

```
text = \
"""
<tr class = "hots">
    <td class = "1">hot1</td>
    <td class = "2">hot2</td>
    <td class = "3">hot3</td>
    <td class = "4">hot4</td>
    <td class = "5">hot5</td>
    <td class = "6">爬虫</td>
</tr>
"""
```

```
In [7]: print(result)
b'<html><body><tr class="hots">\n    <td class="1">hot1</td>
\n    <td class="2">hot2</td>\n    <td class="3">hot3</td>\n
<td class="4">hot4</td>\n    <td class="5">hot5</td>\n    <td
class="6">&#29228;&#34411;</td>\n</tr>\n</body></html>'
```

```
1 from lxml import etree
2
3 """
4 1、读取html字符串
5 """
6
7 text = \
8 """
9 <tr class = "hots">
10     <td class = "1">hot1</td>
11     <td class = "2">hot2</td>
12     <td class = "3">hot3</td>
13     <td class = "4">hot4</td>
14     <td class = "5">hot5</td>
15     <td class = "6">爬虫</td>
16 </tr>
17 """
18 #利用etree.HTML将字符串解析为HTML文档
19 #结果是个Element类型的文档，无法直接打印
20 html = etree.HTML(text)
21 #打印解析后的html文档
22 result = etree.tostring(html)
23 print(result)
```

使用etree下的HTML函数对字符串进行解析


```
In [7]: print(result)
b'<html><body><tr class="hots">\n      <td class="1">hot1</td>\n
\n      <td class="2">hot2</td>\n      <td class="3">hot3</td>\n
<td class="4">hot4</td>\n      <td class="5">hot5</td>\n      <td
class="6">&#29228;&#34411;</td>\n</tr>\n</body></html>'
```

结果中并未出现我们构造的“爬虫”字样的数据，原因是前面‘b’，表示为bytes流数据，因此我们在使用代码的过程中，对我们的字符串进行了编码的转换和解析。

因此，这里我们需要进行定义编码，再解码的工作，修改代码。

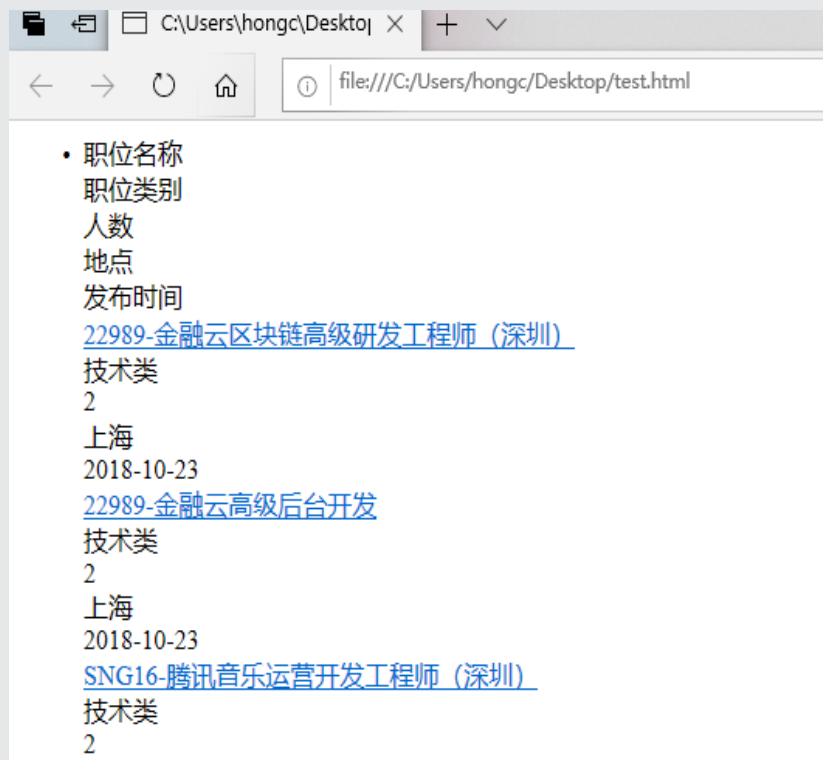
```
1 from lxml import etree
2
3 """
4 1、读取html字符串
5 """
6
7 text = \
8 """
9 <tr class = "hots">
10     <td class = "1">hot1</td>
11     <td class = "2">hot2</td>
12     <td class = "3">hot3</td>
13     <td class = "4">hot4</td>
14     <td class = "5">hot5</td>
15     <td class = "6">爬虫</td>
16 </tr>
17 """
18 #利用etree.HTML将字符串解析为HTML文档
19 #结果是个Element类型的文档，无法直接打印
20 html = etree.HTML(text)
21 #打印解析后的html文档
22 #解析后的文档是一个bytes流数据，因此需要先编码，再解码
23 result = etree.tostring(html,encoding='utf8').decode('utf8')
24 print(result)
```

```
In [8]: result =
etree.tostring(html,encoding='utf8').decode('utf8')
...: print(result)
<html><body><tr class="hots">
    <td class="1">hot1</td>
    <td class="2">hot2</td>
    <td class="3">hot3</td>
    <td class="4">hot4</td>
    <td class="5">hot5</td>
    <td class="6">爬虫</td>
</tr>
</body></html>
```

注：这里不仅完成了编码，还把html代码进行了查漏补全。

2、直接解析html文件

打开test.html示例文件



```
<ul class="ulist" padding="1" spacing="1">
  <li>
    <div class="top">
      <div class="position" width="350">职位名称</div>
      <div>职位类别</div>
      <div>人数</div>
      <div>地点</div>
      <div>发布时间</div>
    </div>
    <div class="even">
      <div class="l square"><a target="_blank" href="position_detail.php?
id=33824&keywords=python&tid=87&lid=2218">22989-金融云区块链高级研发工
程师 (深圳) </a></div>
      <div>技术类</div>
      <div>2</div>
      <div>上海</div>
      <div>2018-10-23</div>
    </div>
    <div class="odd">
      <div class="l square"><a target="_blank" href="position_detail.php?
id=29938&keywords=python&tid=87&lid=2218">22989-金融云高级后台开发
</a></div>
```

```

26 """
27 2、直接解析html文件
28 """
29 #利用parse对测试文件进行解析
30 html = etree.parse(r"C:\Users\hongc\Desktop\test.html")
31 result = etree.tostring(html,encoding='utf8').decode('utf8')
32 print(result)
33
34

```

```

....: result =
etree.tostring(html,encoding='utf8').decode('utf8')
....: print(result)
<ul class="ullist" padding="1" spacing="1">
  <li>
    <div class="top">
      <div class="position" width="350">职位名称</div>
      <div>职位类别</div>
      <div>人数</div>
      <div>地点</div>
      <div>发布时间</div>
    </div>
    <div class="even">
      <div class="l square"><a target="_blank"

```



那么我们直接将网页另存为html页面，是否也可以用etree进行解析呢？

答案是否定的，直接使用并解析会报错！！！！

```
33
34 # 保存的网页直接用etree解析
35 html = etree.parse(r"C:\Users\hongc\Desktop\baidu.html")
36 result = etree.tostring(html, encoding='utf8').decode('utf8')
37 print(result)
38 |
```

```
File "src/lxml/parser.py", line 640, in
lxml.etree._raiseParseError

File "file:/C:/Users/hongc/Desktop/baidu.html", line 22
XMLSyntaxError: xmlParseEntityRef: no name, line 22, column
76
```

显示的内容是在部分位置，标签缺失等等的问题。

【原因】

- 1、百度的页面有些标签缺失了，不够规整
- 2、我们默认使用的是xml解析器，当它解析html页面时，会造成一定错误，需要我们自定义解析器

【解决方法】

- 1、自定义一个解析器
- 2、将自定义的解析器作为参数传递给parse

```
34 #保存的网页直接用etree解析
35 """
36 默认使用xml解析器
37 """
38 #自定义一个解析器
39 parser = etree.HTMLParser(encoding='utf8')
40 #将定义的解析器作为参数，再次传递给parse
41 html = etree.parse(r"C:\Users\hongc\Desktop\baidu.html", parser=parser)
42 result = etree.tostring(html, encoding='utf8').decode('utf8')
43 print(result)
44
```

```
Text editor - result

http-equiv="Content-Type" content="text/html; charset=UTF-8"/><meta
http-equiv="X-UA-Compatible" content="IE=edge,chrome=1"/><meta
content="always" name="referrer"/><meta name="theme-color"
content="#2932e1"/><link rel="shortcut icon" href="https://
www.baidu.com/favicon.ico" type="image/x-icon"/><link rel="search"
type="application/opensearchdescription+xml" href="https://
www.baidu.com/content-search.xml" title="百度搜索"/><link rel="icon"
sizes="any" mask="" href="https://www.baidu.com/img/
baidu_85beaf5496f291521eb75ba38eacbd87.svg"/><title>百度一下，你就知道</
title>

<style id="style_super_inline" type="text/
css">blockquote,body,button,dd,dl,dt,fieldset,form,h1,h2,h3,h4,h5,h6,hr
,input,legend,li,ol,p,pre,td,textarea,th,ul{margin:0;padding:0}

Save and Close Close
```



注：这种方法了解即可，主要掌握第一种方法，因为我们在content中获取的html页面是正常的页面。

lxml库与Xpath结合使用提取数据



使用示例中的内容，新建一个text

```
1 text = \
2 """
3 <ul class="ullist" padding="1" spacing="1">
4   <li>
5     <div id="top">
6       <span class="position" width="350">职位名称</span>
7       <span>职位类别</span>
8       <span>人数</span>
9       <span>地点</span>
10      <span>发布时间</span>
11    </div>
12    <div id="even">
13      <span class="l square">
14        <a target="_blank" href="position_detail.php?id=33824&";
15      </span>
16      <span>技术类</span>
17      <span>2</span>
18      <span>上海</span>
19      <span>2018-10-23</span>
20    </div>
```

注：我们先观察下该字符串。最外层是ul标签，下一层是li标签，再下一层是众多的div标签，div标签中存放了很多span标签，这些标签中存放了关于职位的相关信息。

```
106 from lxml import etree
107
108 #将html 字符串解析为HTML 文档
109 html = etree.HTML(text)
110
```

下面我们例举一些需求，使用Xpath语法完成这些需求。

任务：

- 1.获取所有的div标签 【节点选取】
- 2.获取指定的某个div标签 【谓语的使用】
- 3.获取所有的id='even'的div标签
- 4.获取标签的某个属性值
- 5.获取div里面所有的职位信息



使用xpath函数，函数()内填写相应的Xpath语法，完成提取。

```
112 """
113 1. 获取所有的div标签【节点选取】
114 """
115 divs = html.xpath('//div')
116 print(divs)
```

```
[<Element div at 0x25ab23ef048>, <Element div at 0x25ab24c4908>, <Element div at 0x25ab1fcaa48>, <Element div at 0x25ab1877708>, <Element div at 0x25ab1877a08>, <Element div at 0x25ab1877648>, <Element div at 0x25ab1877a88>, <Element div at 0x25ab1877108>, <Element div at 0x25ab1877308>, <Element div at 0x25ab1877808>, <Element div at 0x25ab1877188>]
```

返回的结果为element，老问题，解析的有问题。因此我们需要进行解码。

```
119 for div in divs:
120     d = etree.tostring(div,encoding='utf8').decode('utf8')
121     print(d)
122     break #这里的break 仅仅为了打印出一个div结果
```

```
<div id="top">
    <span class="position" width="350">职位名称</span>
    <span>职位类别</span>
    <span>人数</span>
    <span>地点</span>
    <span>发布时间</span>
</div>
```

注：去掉break，完成全部div标签的打印。

```
119 for div in divs:
120     d = etree.tostring(div,encoding='utf8').decode('utf8')
121     print(d)
122     print("*"*50)
```

```
*****
<div id="even">
    <span class="l square">
        <a target="_blank" href="position_detail.php?id=31648&keywords=python&tid=87&lid=2218">高级AI开发工程师</a>
    </span>
    <span>技术类</span>
    <span>4</span>
    <span>上海</span>
    <span>2018-10-23</span>
</div>
*****
```

注：为了更好的分隔，每个div之间以*****作为分割线。

注：Xpath提取数据返回的结果是列表，后续操作需要使用列表操作。


```

124
125 """
126 2. 获取指定的某个div标签【谓语的使用】
127 """
128 div = html.xpath('//div[1]')
129 print(etree.tostring(div,encoding='utf8').decode('utf8'))
130
131

```

TypeError: Type 'list' cannot be serialized.

【报错原因】

xpath返回的必定是列表，列表无法使用etree中的函数进行操作

【解决方案】

需要将div这个列表的内容取出来，再进行解码等其它操作

```

125 """
126 2. 获取指定的某个div标签【谓语的使用】
127 """
128 div = html.xpath('//div[1]')[0]
129 print(etree.tostring(div,encoding='utf8').decode('utf8'))
130
131

```

xpath语法中的//div[1]，表示的是所有div标签中的第一个。
 这里的结果只有一个，存放在列表中。

[0]表示的是把上面这个列表中的元素取出来，
 python中第一个即第“0”个。

```

In [20]: div = html.xpath('//div[1]')[0]
...:
print(etree.tostring(div,encoding='utf8').decode('utf8'))
<div id="top">
  <span class="position" width="350">职位名称</span>
  <span>职位类别</span>
  <span>人数</span>
  <span>地点</span>
  <span>发布时间</span>
</div>

```

```

133 """
134 3. 获取所有的id='even'的div标签
135 """
136 divs = html.xpath('//div[@id="even"]')
137 #print(divs)
138 for div in divs:
139     d = etree.tostring(div,encoding='utf8').decode('utf8')
140     print(d)
141     print("*"*60)

```

```

144 """
145 4. 获取标签的某个属性值
146 """
147 # 获取所有div的id属性的值
148 divs = html.xpath('//div/@id')
149 print(divs)
150
151 # 获取所有a标签的href属性的值
152 hrefs = html.xpath('//a/@href')
153 print(hrefs)
154

```

注：/@可以用来获取属性的值。

```

*****
<div id="even">
    <span class="l square">
        <a target="_blank" href="position_detail.php?
id=32217&keywords=python&tid=87&lid=2218">Python开
发（自动化运维方向）</a>
    </span>
    <span>技术类</span>
    <span>1</span>
    <span>上海</span>
    <span>2018-10-23</span>
</div>
*****

```

```

['top', 'even', 'odd', 'even', 'odd', 'even', 'odd', 'even',
'odd', 'even', 'odd']
['position_detail.php?
id=33824&keywords=python&tid=87&lid=2218',
'position_detail.php?id=29938&keywords=python&tid=87&lid=2218',
'position_detail.php?id=31236&keywords=python&tid=87&lid=2218',
'position_detail.php?id=31235&keywords=python&tid=87&lid=2218',
'position_detail.php?id=34531&keywords=python&tid=87&lid=2218',
'position_detail.php?id=34532&keywords=python&tid=87&lid=2218',
'position_detail.php?id=31648&keywords=python&tid=87&lid=2218',
'position_detail.php?id=32218&keywords=python&tid=87&lid=2218',
'position_detail.php?id=32217&keywords=python&tid=87&lid=2218',
'position_detail.php?id=34511&keywords=python&tid=87&lid=2218']

```

所有的职位信息都在div标签中，div标签中除了第一个以外的标签都含有我们想要的职位信息，那么它就有两种方式提取

```
156 """
157 5. 获取div里面所有的职位信息
158 """
159 divs = html.xpath('//div')[1:]
160 #divs = html.xpath('//div[position()>1]')
161
```

```
156 """
157 5. 获取div里面所有的职位信息
158 """
159 divs = html.xpath('//div')[1:]
160 #divs = html.xpath('//div[position()>1]')
161 for div in divs:
162     d = etree.tostring(div, encoding='utf8').decode('utf8')
163     print(d)
164     print("*"*60)
```

获取了数据以后，我们要提取信息

这里是一堆格式相同的div，因此我们建立一个for循环，批量处理。

```
*****
<div id="odd">
    <span class="l square">
        <a target="_blank" href="position_detail.php?
id=34511&keywords=python&tid=87&lid=2218">Python数
据挖掘讲师 </a>
    </span>
    <span>技术类</span>
    <span>1</span>
    <span>上海</span>
    <span>2018-10-23</span>
</div>
*****
```

```
<div id="even">
  <span class="l square">
    <a target="_blank" href="position_detail.php?id=33824&keywords=python&tid=87&lid=2218">python开发工程师</a>
  </span>
  <span>技术类</span>
  <span>2</span>
  <span>上海</span>
  <span>2018-10-23</span>
</div>
```

我们想获取以下信息：

a标签下的**href**属性

职位

类别

人数

地点

时间

我们分别在**for**循环中一个个的来获取。

获取a标签下的href属性

```
166 for div in divs:
167     url = div.xpath('//a/@href')
168     print(url)
169
```

```
id=31236&keywords=python&tid=87&lid=2218',
'position_detail.php?
id=31235&keywords=python&tid=87&lid=2218',
'position_detail.php?
id=34531&keywords=python&tid=87&lid=2218',
'position_detail.php?
id=34532&keywords=python&tid=87&lid=2218',
'position_detail.php?
id=31648&keywords=python&tid=87&lid=2218',
'position_detail.php?
id=32218&keywords=python&tid=87&lid=2218',
'position_detail.php?
id=32217&keywords=python&tid=87&lid=2218',
'position_detail.php?
id=34511&keywords=python&tid=87&lid=2218']
```

这里我们获取了远比数据量更多的href。显然是有问题的。

【错误原因】

//a/@href获取的是全部html下的a标签。并不是我们筛选出来的div中的a标签。

【解决方法】

应该使用 .//a/@href （注：前面加了点）

表达式	描述	用法	说明
nodename	选取此节点的所有子节点	div	选取div下的所有标签
//	从全局节点中选择节点，任意位置均可	//div	选取整个HTML页面的所有div标签
/	选取某个节点下的节点	//head/title	选取head标签下的title标签
@	选取带某个属性的节点	//div[@id]	选择带有id属性的div标签
.	当前节点下	./span	选择当前节点下的span标签【代码中威力强大】

. 它能够方便我们快速获取当前节点下的内容，而不会获取全局。



获取a标签职位的文本信息

```
166 for div in divs:
167     #获取标签href属性
168     url = div.xpath('.//a/@href')
169     #获取a标签的[文本信息]
170     position = div.xpath('.//a/text()')
171     print(position)
172     break
173 |
```

```
In [28]: for div in divs:
...:     #获取标签href属性
...:     url = div.xpath('.//a/@href')
...:     #获取a标签的[文本信息]
...:     position = div.xpath('.//a/text()')
...:     print(position)
...:     break
['python开发工程师']
```

列表

做一些适当的修改



```
166 for div in divs:
167     #获取标签href属性
168     url = div.xpath('.//a/@href')[0]
169     #获取a标签的[文本信息]
170     position = div.xpath('.//a/text()')[0]
171     print(position)
172     break
```

```
In [29]: for div in divs:
...:     #获取标签href属性
...:     url = div.xpath('.//a/@href')[0]
...:     #获取a标签的[文本信息]
...:     position = div.xpath('.//a/text()')
[0]
...:     print(position)
...:     break
python开发工程师
```

字符串

获取工作类型

```
166 for div in divs:
167     #获取标签href属性
168     url = div.xpath('.//a/@href')[0]
169     #获取a标签的[文本信息]
170     position = div.xpath('.//a/text()')[0]
171     #获取工作类型
172     work_type = div.xpath('.//span[2]/text()')[0]
173     print(work_type)
174     break
```

```
In [32]: for div in divs:
...:     #获取标签href属性
...:     url = div.xpath('.//a/@href')[0]
...:     #获取a标签的[文本信息]
...:     position = div.xpath('.//a/text()')
[0]
...:     #获取工作类型
...:     work_type = div.xpath('.//span[2]/
text()')[0]
...:     print(work_type)
...:     break
技术类
```

获取职位人数

```
166 for div in divs:
167     #获取标签href属性
168     url = div.xpath('.//a/@href')[0]
169     #获取a标签的[文本信息]
170     position = div.xpath('.//a/text()')[0]
171     #获取工作类型
172     work_type = div.xpath('.//span[2]/text()')[0]
173     #获取职位人数
174     work_num = div.xpath('.//span[3]/text()')[0]
175     print(work_num)
176     break
```

```
...:     #获取a标签的[文本信息]
...:     position = div.xpath('.//a/text()')
[0]
...:     #获取工作类型
...:     work_type = div.xpath('.//span[2]/
text()')[0]
...:     #获取职位人数
...:     work_num = div.xpath('.//span[3]/
text()')[0]
...:     print(work_num)
...:     break
2
```


获取工作地点

```
166 for div in divs:
167     #获取标签href属性
168     url = div.xpath('.//a/@href')[0]
169     #获取a标签的[文本信息]
170     position = div.xpath('.//a/text()')[0]
171     #获取工作类型
172     work_type = div.xpath('.//span[2]/text()')[0]
173     #获取职位人数
174     work_num = div.xpath('.//span[3]/text()')[0]
175     #获取工作地点
176     area = div.xpath('.//span[4]/text()')[0]
177     print(area)
178     break
```

```
...: work_type = div.xpath('.//span[2]/
text()')[0]
...: #获取职位人数
...: work_num = div.xpath('.//span[3]/
text()')[0]
...: #获取工作地点
...: area = div.xpath('.//span[4]/
text()')[0]
...: print(area)
...: break
上海
```

获取发布时间

```
166 for div in divs:
167     #获取标签href属性
168     url = div.xpath('.//a/@href')[0]
169     #获取a标签的[文本信息]
170     position = div.xpath('.//a/text()')[0]
171     #获取工作类型
172     work_type = div.xpath('.//span[2]/text()')[0]
173     #获取职位人数
174     work_num = div.xpath('.//span[3]/text()')[0]
175     #获取工作地点
176     area = div.xpath('.//span[4]/text()')[0]
177     #获取发布时间
178     time = div.xpath('.//span[5]/text()')[0]
179     print(time)
180     break
```

```
...: area = div.xpath('.//span[4]/
text()')[0]
...: #获取发布时间
...: time = div.xpath('.//span[5]/
text()')[0]
...: print(time)
...: break
2018-10-23
```


存储全部信息

```
157 5. 获取div里面所有的职位信息
158 """
159 divs = html.xpath('//div')[1:]
160 works = []
161 for div in divs:
162     work = {}
163     #获取标签href属性
164     url = div.xpath('..//a/@href')[0]
165     #获取a标签的[文本信息]
166     position = div.xpath('..//a/text()')[0]
167     #获取工作类型
168     work_type = div.xpath('..//span[2]/text()')[0]
169     #获取职位人数
170     work_num = div.xpath('..//span[3]/text()')[0]
171     #获取工作地点
172     area = div.xpath('..//span[4]/text()')[0]
173     #获取发布时间
174     time = div.xpath('..//span[5]/text()')[0]
175     work = {
176         "url":url,
177         "position":position,
178         "work_type":work_type,
179         "work_num":work_num,
180         "area":area,
181         "time":time,
182     }
183     works.append(work)
```

新建一个空的列表，将每次的职位信息存储进去
新建一个空的字典，将每次for循环中的职位信息存储进去

新建的空字典，存储所有职位信息

For循环中的一次结果添加至列表【append()函数】

The background features a light gray central rectangle. Above and below this rectangle are decorative borders composed of overlapping triangles in various shades of blue, creating a low-poly, mountain-like effect.

谢谢聆听