数据提取之xpath

知识点：

- 掌握 xpath获取节点属性的方法
- 掌握 xpath获取文本的方法
- 掌握 xpath查找特定节点的方法

1. 为什么要学习xpath和lxml

lxml是一款高性能的 Python HTML/XML 解析器，我们可以利用XPath，来快速的定位特定元素以及获取节点信息

2. 什么是xpath

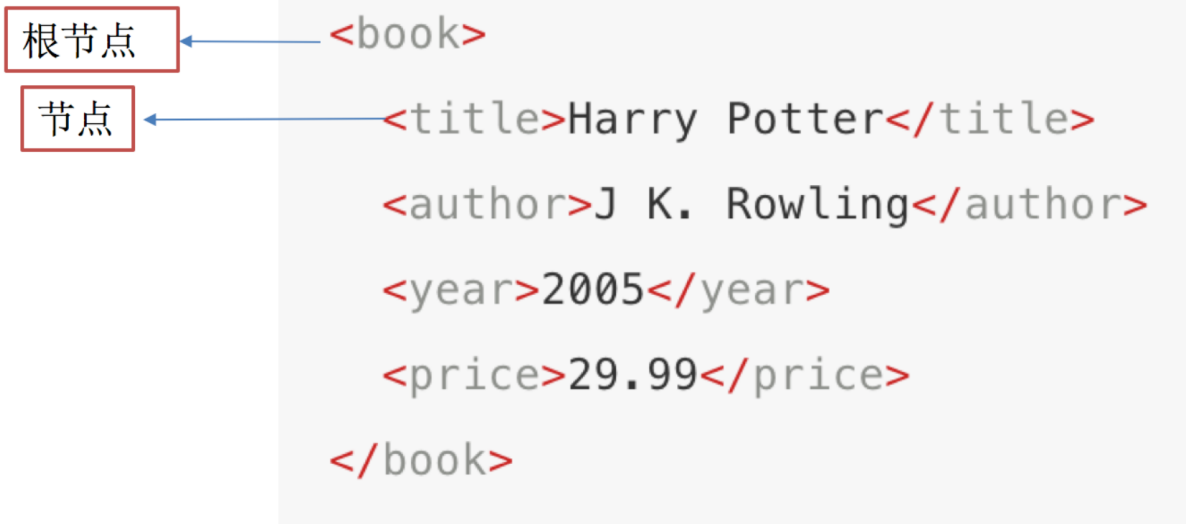
XPath (XML Path Language) 是一门在 HTML\XML 文档中查找信息的**语言**，可用来在 HTML\XML 文档中对**元素和属性进行遍历**。

W3School官方文档： http://www.w3school.com.cn/xpath/index.asp

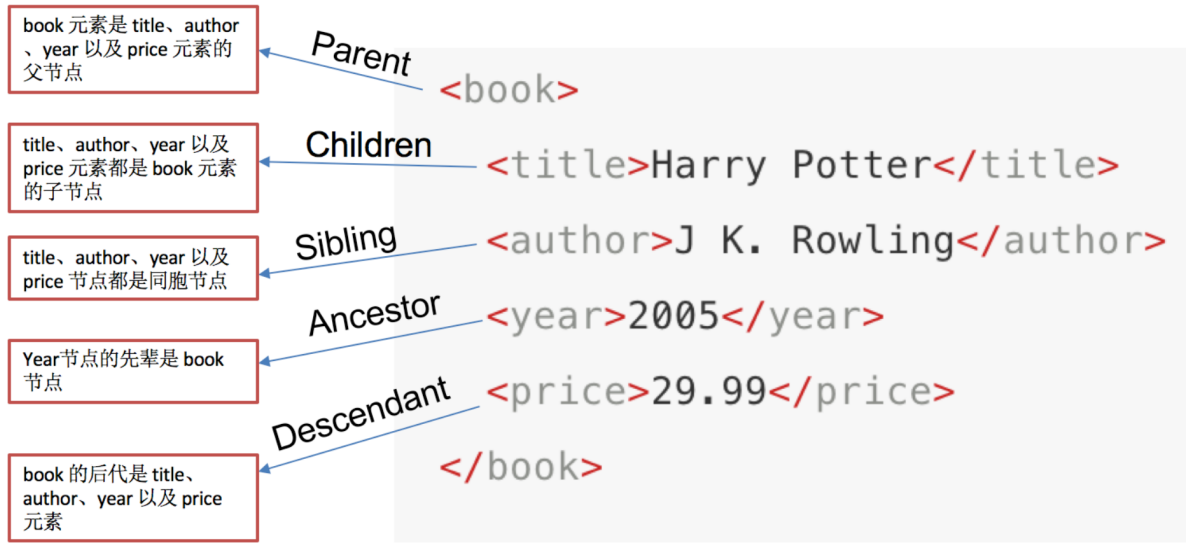
3. xpath的节点关系

3.1 xpath中的节点是什么

每个XML的标签我们都称之为节点，其中最顶层的节点称为根节点。



3.2 xpath中节点的关系



4. 谷歌浏览器xpath helper插件的安装和使用

> 要想利用lxml模块提取数据，需要我们掌握xpath语法规则。接下来我们就来了解一下xpath helper插件，它可以帮助我们练习xpath语法

4.1 谷歌浏览器xpath helper插件的作用

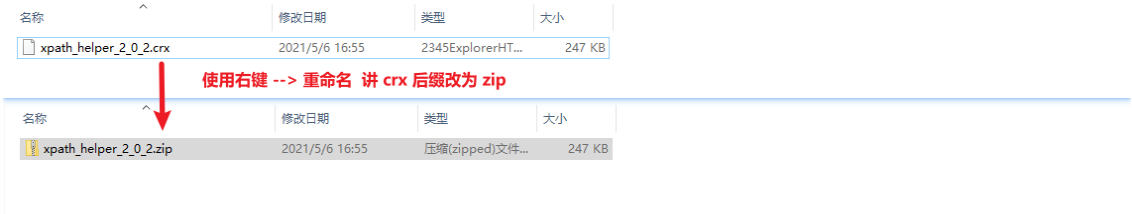
> 在谷歌浏览器中对当前页面测试xpath语法规则

4.2 谷歌浏览器xpath helper插件的安装和使用

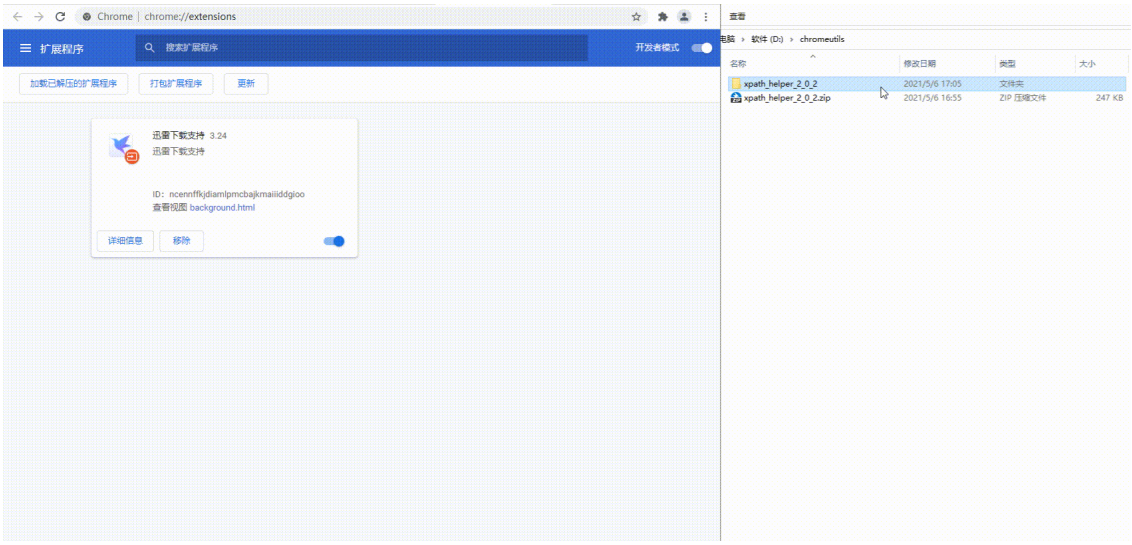
我们以windos为例进行xpath helper的安装

4.2.1xpath helper插件的安装

- 下载Chrome插件 XPath Helper
  - 可以在chrome应用商城进行下载，如果无法下载，也可以从下面的链接进行下载
  - 下载地址： https://pan.baidu.com/s/1-pKX0KL7jCMin4A6zBFHLg 密码:cvyh
- 把文件的后缀名crx改为zip，然后解压到同名文件夹中



- 把解压后的文件夹拖入到已经开启开发者模式的chrome浏览器扩展程序界面



- 重启浏览器后，访问url之后在页面中点击xpath图标，就可以使用了

如果是**linux或macOS操作系统**，无需操作上述的步骤2，直接将crx文件拖入已经开启开发者模式的chrome浏览器扩展程序界面

5. xpath语法

我们将在下面的例子中使用这个 XML 文档。

```
<bookstore>

<book>
  <title lang="eng">Harry Potter</title>
  <price>29.99</price>
</book>

<book>
  <title lang="eng">Learning XML</title>
  <price>39.95</price>
</book>

</bookstore>
```

5.1 选取节点

XPath 使用路径表达式来选取 XML 文档中的节点或者节点集。这些路径表达式和我们在常规的**电脑文件系统中看到的表达式**非常相似。

使用chrome插件选择标签时候，选中时，选中的标签会添加属性class="xh-highlight"

下面列出了最有用的表达式：

表达式	描述
nodename	选中该元素
/	从根节点选取、或者是元素和元素间的过度
//	从匹配选择的当前节点选择文档中的节点，而不考虑他们的位置。
.	选取当前节点
..	选择当前节点的父节点
@	选取属性
text()	选取文本

实例

在下面的表格中，我们已列出了一些路径表达式以及表达式的结果：

路径表达式	结果
bookstore	选择bookstore元素
/bookstore	选择bookstore元素，假如路径起始于正斜杠(/) 则此路径始终代表到某元素的绝对路径。
bookstore/book	选择bookstored的子元素的所有book元素
//book	选择所有book子元素，而不管他们在文档中的位置
bookstore//book	选择属于bookstore元素的后代的所有book元素，而不管他们位于bookstore之下的什么位置。
//book/title/@lang	选择所有的book下面的title中的lang属性的值
//book/title/text()	选择所有的book下面的title的文本。

xpath基础语法练习：

接下来我们听过豆瓣电影top250的页面来练习上述语法： https://movie.douban.com/top250

- 选择所有的h1下的文本
  - `//h1/text()`
  - 

    豆瓣电影 Top 250
- 获取所有的a标签的href
  - `//a/@href`
  - 

- 获取html下的head下的title的文本
  - `/html/head/title/text()`

- 

- 获取html下的head下的link标签的href
  - `/html/head/link/@href`
  - 

但是当我们需要选择所有的电影名称的时候会特别费力，通过下一小节的学习，就能够解决这个问题

5.2 查找特定的节点

路径表达式	结果
//title[@lang='eng']	选择lang属性值为eng的所有title元素
/bookstore/book[1]	选择属于bookstore子元素的第一个book元素
/bookstore/book[last()]	选择属于bookstore子元素最后一个book元素
/bookstore/book[last()-1]	选择属于bookstore子元素的倒数第二个book元素
/bookstore/book[position()>1]	选择bookstore下面的book元素，从第二个开始选择
//book/title[text()='Harry Potter']	选择所有book下的title元素，仅选择文本为Harry Potter的title元素
/bookstore/book[price>35.00]/title	选择bookstore元素中book元素的所有title元素，且其中的price元素的值须大于35

注意点: **在xpath中，第一个元素的位置是1，最后一个元素的位置是last()，倒数第二个是last()-1**

xpath基础语法练习2：

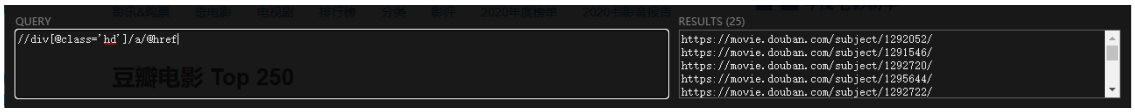
从豆瓣电影top250的页面中：选择所有的电影的名称，href，评分，评价人数

- 选取所有的电影的名字：

```
# 先定位到 class=hd的div标签，再取下面的a标签下面的第一个span标签
//div[@class='hd']/a/span[1]
```

- 选取所有的href：

```
# 先定位到 class=hd的div标签，再取下面的a标签中href属性的值
//div[@class='hd']/a/@href
```



- 选取所有的评分：

```
# 评分的标签是一个span标签，并且对应的有class属性，先定位到 class=rating_num 的span标签，再取标签下的文本即可
//span[@class='rating_num']/text()
```

- 选取所有的评价人数：

```
# 评分标签是一个span标签，有class属性，但是没有值，我们可以先定位到他的父级div，
# 因为评分是最后一个span标签，所以使用last() 取最后一个span标签即可
//div[@class='star']/span[last()]
```

5.3 选取未知节点

XPath 通配符可用来选取未知的 XML 元素。

xpath通配符	描述
*	匹配任何元素节点
@*	匹配任何属性节点
node()	匹配任何类型的节点

实例

在下面的表格中，我们列出了一些路径表达式，以及这些表达式的结果：

路径表达式	描述
/bookstore/*	选取bookstore元素的所有子元素
//*	选取文档中所有元素
//title[@*]	选取所有带有属性的title元素

5.4 选取若干路径

通过在路径表达式中使用“|”运算符，您可以选取若干个路径。

实例

在下面的表格中，我们列出了一些路径表达式，以及这些表达式的结果：

路径表达式	结果
//book/title | //book/price	选取book元素所有的title和price元素
//title | //price	选取文档中所有的title和price元素
/bookstore/book/title | //price	选取属于bookstore元素的book元素的所有title元素，以及文档中所有的price元素

总结

- xpath中常用的获取节点的表达式
  - `/` 常用于元素和元素之间的过度
  - `//` 选取节点，不考虑其位置，常用
  - `.` 表示选取当前节点
  - `..` 表示选取当前节点的上一级节点(父节点)
  - `@属性名` 根据属性名，获取标签中属性的值。
  - `text()` 获取标签下的文本内容(字符串内容)
- xpath中常用的获取特定节点的表达式:
  - `//标签名[@属性名=值]` 根据标签的属性以及属性的值获取特定的标签
  - `//标签名[num]` 获取选取到的所有标签的第几个标签，num从 1 开始。
  - `//标签名[last()]` 获取选取到的所有的标签的最后一个标签，注意：最后一个不是 -1。
  - `//标签名[text()=值]` 根据标签中的文本内容的值，获取到某个标签。
  - `//标签名[position()>num]` 表示从第num个标签获取
  - `//标签名[position()<num]` 表示从第一个获取到第num个标签