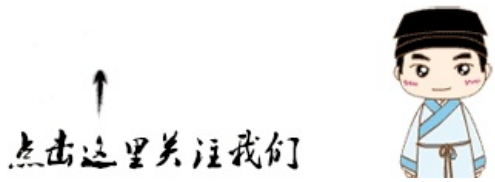
【零基础学爬虫】xpath解析篇

一文读懂XPath基本语法_XPath语法详解_XPath教程



因为最近在想整理有关python爬虫的文章，连带遇到XPath的使用，就顺便一起整理出来。

XPath与自动化的关系

- XPath是一门在XML文档中查找信息的语言，可用来在XML文档中对元素和属性进行遍历。
- XPath是用来选择“节点”的一种基于表达式的语言；
- 表达式的格式类似于文件系统，eg. C:\user\admin；
- XPath最常用的场景之一就是在自动化中用来选择HTML DOM 节点；
- XPath是Selenium自动化测试中作为选择web元素的主要方法之一

01 Xpath简介

XPath 使用路径表达式来选取 XML 文档中的节点或节点集。节点是通过沿着路径 (path) 或者步 (steps) 来选取的。

02 Xpath节点

在 XPath 中，有七种类型的节点：元素、属性、文本、命名空间、处理指令、注释以及文档（根）节点。XML 文档是被作为节点树来对待的。树的根被称为文档节点或者根节点。
下面一个xml示例（一）：

```
<?xml version="1.0" encoding="UTF-8"?>

<bookstore>
  <book>
    <title>Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
</bookstore>
```

上面的XML文档中的节点例子：

```
<bookstore> (文档（根）节点)

<book>...</book> <author>J K. Rowling</author> (元素节点)

lang="en" (属性节点)
```

2.1、基本值=原子值

```
J K. Rowling

"en"
```

03 节点关系

- 父（Parent）：每个元素以及属性都有一个父。
- 子（Children）：元素节点可有零个、一个或多个子。
- 同胞（Sibling）：拥有相同的父的节点
- 先辈（Ancestor）：某节点的父、父的父，等等。
- 后代（Descendant）：某个节点的子，子的子，等

等。

咱们再回到上面XML示例（一），他们各个节点之间的关系如下：

> book 元素是 title、author、year 以及 price 元素的**父**
>
> title、author、year 以及 price 元素都是 book 元素的**子**
>
> title、author、year 以及 price 元素都是**同胞**
>
> title 元素的**先辈**是 book 元素和 bookstore 元素
>
> bookstore 的**后代**是 book、title、author、year 以及 price 元素

04 基本语法

4.1、参考示例

下面所有的示例的原XML都是这个示例：

```
<?xml version="1.0" encoding="UTF-8"?><title>Harry Potter</title><title>Learning XML</title>
```

4.2、 选取节点

XPath 使用路径表达式在 XML 文档中选取节点。节点是通过沿着路径或者 step 来选取的。 下面列出了最有用的路径表达式：

表达式	描述
nodename	选取此节点的所有子节点。
/	从根节点选取（取子节点）。
//	从匹配选择的当前节点选择文档中的节点，而不考虑它们的位置（取子孙节点）。
.	选取当前节点。
..	选取当前节点的父节点。
@	选取属性。

下面我就实际来看看效果：

路径表达式	结果
bookstore	选取 bookstore 元素的所有子节点。
/bookstore	选取根元素 bookstore。 注释：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	选取属于 bookstore 的子元素的所有 book 元素。
//book	选取所有 book 子元素，而不管它们在文档中的位置。
bookstore//book	选择属于 bookstore 元素的后代的所有 book 元素，而不管它们位于 bookstore 之下的什么位置。
//@lang	选取名为 lang 的所有属性。

4.3、 谓语

谓语用来查找某个特定的节点或者包含某个指定的值的节点。
谓语被嵌在方括号中。
在下面的表格中，我们列出了带有谓语的一些路径表达式，以及表达式的结果：

路径表达式	结果
/bookstore/book[1]	选取属于 bookstore 子元素的第一个 book 元素。
/bookstore/book[last()]	选取属于bookstore子元素的最后一个book元素。
/bookstore/book[last()-1]	选取属于bookstore子元素的倒数第二个book元素。
/bookstore/book[position()<3]	选取最前面的两个属于 bookstore 元素的子元素的 book 元素。
//title[@lang]	选取所有拥有名为 lang 的属性的 title 元素。
//title[@lang='eng']	选取所有 title 元素，且这些元素拥有值为 eng 的 lang 属性。
/bookstore/book[price>35.00]	选取 bookstore 元素的所有 book 元素，且其中的 price 元素的值须大于 35.00。
/bookstore/book[price>35.00]//title	选取 bookstore 元素中的 book 元素的所有 title 元素，且其中的 price 元素的值须大于 35.00。

4.4、 选取未知节点

XPath 通配符可用来选取未知的 XML 元素。

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。

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
/bookstore/*	选取 bookstore 元素的所有子元素。
//*	选取文档中的所有元素。
//title[@*]	选取所有带有属性的 title 元素。

4.5、 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。
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：

05 Xpath轴（Axes）

5.1、轴的表达式

轴可定义相对于当前节点的节点集。

轴名称	结果
ancestor	选取当前节点的所有先辈（父、祖父等）。
ancestor-or-self	选取当前节点的所有先辈（父、祖父等）以及当前节点本身。
attribute	选取当前节点的所有属性。
child	选取当前节点的所有子元素。
descendant	选取当前节点的所有后代元素（子、孙等）。
descendant-or-self	选取当前节点的所有后代元素（子、孙等）以及当前节点本身。

following	选取文档中当前节点的结束标签之后的所有节点。
namespace	选取当前节点的所有命名空间节点。
parent	选取当前节点的父节点。
preceding	选取文档中当前节点的开始标签之前的所有节点。
preceding-sibling	选取当前节点之前的所有同级节点。
self	选取当前节点。

5.2、位置路径表达式

位置路径可以是绝对的，也可以是相对的。
绝对路径起始于正斜杠(/)，而相对路径不会这样。在两种情况中，位置路径均包括一个或多个步，每个步均被斜杠分割：
绝对位置路径：

> /step/step/...

相对位置路径：

> step/step/...

每个步均根据当前节点集之中的节点来进行计算。
步（step）包括：
轴（axis）
定义所选节点与当前节点之间的树关系
节点测试（node-test）
识别某个轴内部的节点
零个或者更多谓语（predicate）
更深入地提炼所选的节点集
步的语法：

> 轴名称::节点测试[谓语]

5.3、轴的示例

例子	结果
child::book	选取所有属于当前节点的子元素的 book 节点。
attribute::lang	选取当前节点的 lang 属性。
child::*	选取当前节点的所有子元素。
attribute::*	选取当前节点的所有属性。
child::text()	选取当前节点的所有文本子节点。
child::node()	选取当前节点的所有子节点。
descendant::book	选取当前节点的所有 book 后代。
ancestor::book	选择当前节点的所有 book 先辈。
ancestor-or-self::book	选取当前节点的所有 book 先辈以及当前节点（如果此节点是 book 节点）
child::*/child::price	选取当前节点的所有 price 孙节点。

06 Xpath运算符

XPath 表达式可返回节点集、字符串、逻辑值以及数字。
下面列出了可用在 XPath 表达式中的运算符：

07 总结

想要了解更多的XPath用法可以参考手册，里面还有更多本文未写出的内容：XPath参考手册

左右滑动查看更多