

[XPath][<https://www.w3school.com.cn/xpath/index.asp>]

XPath是一门在XML文档中查找信息的语言, 可用在XML文档中对元素和属性进行遍历.

XPath中, 有7种类型的节点: 元素, 属性, 文本, 命名空间, 处理指令, 注释以及文档节点(或称为根节点)

XPath术语

1. 节点(Node)

XML文档时被作为节点树来对待的. 树的根被称为文档节点或根节点:

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<bookstore>

<book>
  <title lang="en">Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>

</bookstore>
```

`<bookstore>`: 文档节点

`<author>J K. Rowling</author>`: 元素节点

`lang="en"`: 属性节点

2. 基本值(或称原子值, Atomic value)

基本值是无父或无子的节点:

`J K. Rowling`

`"en"`

3. 项目(Item)

项目是基本值或着节点

节点关系

1. 节点关系-父(Parent)

每个元素以及属性都有一个父

下例中, book元素是title, author, year以及price元素的父:

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

2. 节点关系-子(Children)

元素节点可有0个, 1个或多个子, 上例中, title, author, year以及price元素都是book元素的子.

3. 同胞(Sibling)

拥有相同的父节点, 上例中, title, author, year以及price元素都是同胞

4. 先辈(Ancestor)/后代(Descendant)

某节点的父, 父的父, 等等; 某个节点的子, 子的子, 等等.

XPath语法

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)或着步(steps)来选取的.

1. 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(path)或着步(steps)来选取的.

`nodename`: 选取此节点的所有子节点

`/`: 从根节点选取

`//`: 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

`.`: 选取当前节点

`..`: 选取当前节点的父节点

`@`: 选取属性

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<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>
```

例如:

`bookstore`: 选取bookstore元素的所有子节点

`/bookstore`: 选取根元素bookstore(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, 而不管它们位于bookstore之下的什么位置

`//@lang`: 选取名为lang的所有属性

2. 谓语句(Predicates)

谓语句用来查找某个特定的节点或者包含了指定的值的节点, 谓语句被嵌在方括号中.

`/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元素的值必须大于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元素的值必须大于35.00

3. 选取未知节点

XPath通配符可用来选取未知的XML元素

`*`: 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

例如:

`/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. 选取若干路径

通过路径表达式中使用"`|`"运算符, 可选取若干路径

`//book/title //book/price`: 选取book元素的所有title和price元素

`//title | //price`: 选取文档中的所有title和price元素

`/bookstore/book/tile | //price`: 选取属于bookstore元素的book元素的所有tile元素, 以及文档中所有的price元素

XPath Axes(轴)

1. XPath 轴

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。

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. 在这两种情况中, 位置路径均包含一个或多个步, 每个步均被斜杠分隔:

绝对路径: `/step/step/...`

相对路径: `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

零个或着更多谓词(predicate): 更深入地提炼所选的节点集

轴名称::节点测试[谓词]

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。

XPath 运算符

XPath表达式可返回节点集, 字符串, 逻辑值以及数字

1. XPath 运算符

运算符	描述	实例	返回值
	计算两个节点集	//book //cd	返回所有拥有 book 和 cd 元素的节点集
+	加法	6 + 4	10
-	减法	6 - 4	2
*	乘法	6 * 4	24
div	除法	8 div 4	2
=	等于	price=9.80	如果 price 是 9.80, 则返回 true。 如果 price 是 9.90, 则返回 false。
!=	不等于	price!=9.80	如果 price 是 9.90, 则返回 true。 如果 price 是 9.80, 则返回 false。
<	小于	price<9.80	如果 price 是 9.00, 则返回 true。 如果 price 是 9.90, 则返回 false。
<=	小于或等于	price<=9.80	如果 price 是 9.00, 则返回 true。 如果 price 是 9.90, 则返回 false。
>	大于	price>9.80	如果 price 是 9.90, 则返回 true。 如果 price 是 9.80, 则返回 false。
>=	大于或等于	price>=9.80	如果 price 是 9.90, 则返回 true。 如果 price 是 9.70, 则返回 false。
or	或	price=9.80 or price=9.70	如果 price 是 9.80, 则返回 true。 如果 price 是 9.50, 则返回 false。
and	与	price>9.00 and price<9.90	如果 price 是 9.80, 则返回 true。 如果 price 是 8.50, 则返回 false。
mod	计算除法的余数	5 mod 2	1

XML 实例文档

"books.xml":<https://www.w3school.com.cn/example/xml/books.xml>

Mischellaneous

matchNamespaces

namespaces: a named character vector giving the namespace prefix and URI pairs that are to be used in the XPath expression and matching of nodes. The prefix is just a simple string that acts as a short-hand or alias for the URI that is the unique identifier for the namespace. The URI is the element in this vector and the prefix is the corresponding element name. One only needs to specify the namespaces in the XPath expression and for the nodes of interest rather than requiring all the namespaces for the entire document. Also note that the prefix used in this vector is local only to the path. It does not have to be the same as the prefix used in the document to

identify the namespace.