

**XPath 是一门在 XML 文档中查找信息的语言。XPath 用于在 XML 文档中通过元素和属性进行导航。**

**在学习之前应该具备的知识：**

- HTML / XHTML
- XML / XML 命名空间

**什么是 XPath?**

- XPath 使用路径表达式在 XML 文档中进行导航
- XPath 包含一个标准函数库
- XPath 是 XSLT 中的主要元素
- XPath 是一个 W3C 标准

**XPath 路径表达式**

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。

**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。**

**XML 实例文档**

我们将在下面的例子中使用这个 XML 文档。

```
<?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>
```

## 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。
//	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。

## 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	选取 bookstore 元素的所有子节点。
/bookstore	选取根元素 <b>bookstore</b> 。  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。

## 谓语 ( Predicates )

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。

## 实例

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。

## 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。

## 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
/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。

## 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/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。