**XPath 教程**

**XPath 是一门在 XML 文档中查找信息的语言。XPath 可用来在 XML 文档中对元素和属性进行遍历。**

**XPath 是 W3C XSLT 标准的主要元素，并且 XQuery 和 XPointer 同时被构建于 XPath 表达之上。**

**因此，对 XPath 的理解是很多高级 XML 应用的基础。**

[开始学习 XPath](http://www.w3school.com.cn/xpath/xpath_intro.asp) ！

**XPath 参考手册**

在 W3School，我们提供完整的 XPath 2.0、XQuery 1.0 和 XSLT 2.0 的内置函数参考手册。

[XPath 函数](http://www.w3school.com.cn/xpath/xpath_functions.asp)

**内容目录**

[XPath 简介](http://www.w3school.com.cn/xpath/xpath_intro.asp)

本章讲解 XPath 的概念。

[XPath 节点](http://www.w3school.com.cn/xpath/xpath_nodes.asp)

本章详细介绍 XPath 中不同类型的节点，以及节点之间的关系。

[XPath 语法](http://www.w3school.com.cn/xpath/xpath_syntax.asp)

本章讲解 XPath 的语法。

[XPath 轴](http://www.w3school.com.cn/xpath/xpath_axes.asp)

本章讲解 XPath axes（轴）。

[XPath 运算符](http://www.w3school.com.cn/xpath/xpath_operators.asp)

本章列出了可以用于 XPath 表达式的运算符。

[XPath 实例](http://www.w3school.com.cn/xpath/xpath_examples.asp)

本章使用 "books.xml" 文档来演示一些 XPath 实例。

[XPath 摘要](http://www.w3school.com.cn/xpath/xpath_summary.asp)

本文内容包括在本教程所学知识的一个总结，以及我们向你推荐的下一步应该学习的内容。

**XPath 参考手册**

[XPath 函数](http://www.w3school.com.cn/xpath/xpath_functions.asp)

XPath 2.0、XQuery 1.0 和 XSLT 2.0 的内置函数。

# XPath 简介

* [Previous Page](http://www.w3school.com.cn/xpath/index.asp)
* [Next Page](http://www.w3school.com.cn/xpath/xpath_nodes.asp)

**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：**

在您继续学习之前，应该对下面的知识有基本的了解：

* HTML / XHTML
* XML / XML 命名空间

如果您希望首先学习这些项目，请在我们的 [首页](http://www.w3school.com.cn/index.html) 访问这些教程。

**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 标准函数**

XPath 含有超过 100 个内建的函数。这些函数用于字符串值、数值，日期和时间比较、节点和 QName 处理、序列处理、逻辑值等等。

**XPath 在 XSLT 中使用**

XPath 是 XSLT 标准中的主要元素。如果没有 XPath 方面的知识，您就无法创建 XSLT 文档。

您可以在我们的《[XSLT 教程](http://www.w3school.com.cn/xsl/index.asp)》中阅读更多的内容。

XQuery 和 XPointer 均构建于 XPath 表达式之上。XQuery 1.0 和 XPath 2.0 共享相同的数据模型，并支持相同的函数和运算符。

您可以在我们的《[XQuery 教程](http://www.w3school.com.cn/xquery/index.asp)》中阅读更多有关 XQuery 的知识。

**XPath 是 W3C 标准**

XPath 于 1999年11月16日 成为 W3C 标准。

XPath 被设计供 XSLT、XPointer 以及其他 XML 解析软件使用。

您可以在我们的《[W3C 教程](http://www.w3school.com.cn/w3c/index.asp)》中阅读更多有关 XPath 标准的信息。

# 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、注释以及文档节点（或成为根节点）。**

**XPath 术语**

**节点（Node）**

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。

请看下面这个 XML 文档：

<?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>

上面的XML文档中的节点例子：

<bookstore> （文档节点）

<author>J K. Rowling</author> （元素节点）

lang="en" （属性节点）

**基本值（或称原子值，Atomic value）**

基本值是无父或无子的节点。

基本值的例子：

J K. Rowling

"en"

**项目（Item）**

项目是基本值或者节点。

**节点关系**

**父（Parent）**

每个元素以及属性都有一个父。

在下面的例子中，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>

**子（Children）**

元素节点可有零个、一个或多个子。

在下面的例子中，title、author、year 以及 price 元素都是 book 元素的子：

<book>

<title>Harry Potter</title>

<author>J K. Rowling</author>

<year>2005</year>

<price>29.99</price>

</book>

**同胞（Sibling）**

拥有相同的父的节点

在下面的例子中，title、author、year 以及 price 元素都是同胞：

<book>

<title>Harry Potter</title>

<author>J K. Rowling</author>

<year>2005</year>

<price>29.99</price>

</book>

**先辈（Ancestor）**

某节点的父、父的父，等等。

在下面的例子中，title 元素的先辈是 book 元素和 bookstore 元素：

<bookstore>

<book>

<title>Harry Potter</title>

<author>J K. Rowling</author>

<year>2005</year>

<price>29.99</price>

</book>

</bookstore>

**后代（Descendant）**

某个节点的子，子的子，等等。

在下面的例子中，bookstore的后代是 book、title、author、year 以及 price 元素：

<bookstore>

<book>

<title>Harry Potter</title>

<author>J K. Rowling</author>

<year>2005</year>

<price>29.99</price>

</book>

</bookstore>

# XPath 语法

* [Previous Page](http://www.w3school.com.cn/xpath/xpath_nodes.asp)
* [Next Page](http://www.w3school.com.cn/xpath/xpath_axes.asp)

**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 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 | 选取所有属于 bookstore 的子元素的 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 | 选择所有属于 bookstore 元素的后代的 book 元素，而不管它们位于 bookstore 之下的什么位置。 |
| //@lang | 选取所有名为 lang 的属性。 |

**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] | 选取所有 bookstore 元素的 book 元素，且其中的 price 元素的值须大于 35.00。 |
| /bookstore/book[price>35.00]/title | 选取所有 bookstore 元素中的 book 元素的 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 元素的所有子节点 |
| //\* | 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 | 选取所有 book 元素的 title 和 price 元素。 |
| //title | //price | 选取所有文档中的 title 和 price 元素。 |
| /bookstore/book/title | //price | 选取所有属于 bookstore 元素的 book 元素的 title 元素，以及文档中所有的 price 元素。 |

窗体底端

[**XPath, XQuery, 以及XSLT函数**](http://www.w3school.com.cn/xpath/xpath_functions.asp)

# XPath 运算符

* [Previous Page](http://www.w3school.com.cn/xpath/xpath_axes.asp)
* [Next Page](http://www.w3school.com.cn/xpath/xpath_examples.asp)

**XPath 表达式可返回节点集、字符串、逻辑值以及数字。**

**XPath 运算符**

下面列出了可用在 XPath 表达式中的运算符：

|  |  |  |  |
| --- | --- | --- | --- |
| **运算符** | **描述** | **实例** | **返回值** |
| | | 计算两个节点集 | //book | //cd | 返回所有带有 book 和 ck 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 fasle。 |
| != | 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 fasle。 |
| < | 小于 | price<9.80 | 如果 price 是 9.00，则返回 true。  如果 price 是 9.90，则返回 fasle。 |
| <= | 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 fasle。 |
| > | 大于 | price>9.80 | 如果 price 是 9.90，则返回 true。  如果 price 是 9.80，则返回 fasle。 |
| >= | 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 fasle。 |
| 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 fasle。 |
| 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 fasle。 |
| mod | 计算除法的余数 | 5 mod 2 | 1 |

# XPath 实例

* [Previous Page](http://www.w3school.com.cn/xpath/xpath_operators.asp)
* [Next Page](http://www.w3school.com.cn/xpath/xpath_summary.asp)

**在本节，让我们通过实例来学习一些基础的 XPath 语法。**

**XML实例文档**

我们将在下面的例子中使用这个 XML 文档：

**"books.xml" :**

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

<bookstore>

<book category="COOKING">

<title lang="en">Everyday Italian</title>

<author>Giada De Laurentiis</author>

<year>2005</year>

<price>30.00</price>

</book>

<book category="CHILDREN">

<title lang="en">Harry Potter</title>

<author>J K. Rowling</author>

<year>2005</year>

<price>29.99</price>

</book>

<book category="WEB">

<title lang="en">XQuery Kick Start</title>

<author>James McGovern</author>

<author>Per Bothner</author>

<author>Kurt Cagle</author>

<author>James Linn</author>

<author>Vaidyanathan Nagarajan</author>

<year>2003</year>

<price>49.99</price>

</book>

<book category="WEB">

<title lang="en">Learning XML</title>

<author>Erik T. Ray</author>

<year>2003</year>

<price>39.95</price>

</book>

</bookstore>

[在您的浏览器中查看此 "books.xml" 文件](http://www.w3school.com.cn/xpath/books.xml)。

**节点选取**

我们将使用微软的 XML DOM 对象来载入 XML 文档，并使用 selectNodes() 函数从 XML 文档选取节点：

set xmlDoc=CreateObject("Microsoft.XMLDOM")

xmlDoc.async="false"

xmlDoc.load("books.xml")

xmlDoc.selectNodes(路径表达式)

**选取所有的 book 节点**

下面的这个例子选取了 bookstore 元素下所有的 book 节点：

xmlDoc.selectNodes("/bookstore/book")

假如您正在使用 IE 5 或更高的版本，可以[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=xpth_select_cdnodes)。

**选取第一个 book 节点**

下面的例子仅选取 bookstore 元素下第一个 book 节点：

xmlDoc.selectNodes("/bookstore/book[0]")

假如您正在使用 IE 5或更高的版本，可以[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=xpth_select_cdnodes_first)。

**注释：**IE 5 和 6 会把 [0] 作为第一个节点来执行，但是根据 W3C 的标准，应该使用 [1] ！！

**注释：**这个问题在 IE 6 SP2 中被纠正了！

**选取 price**

下面的例子从所有的 price 节点选取文本：

xmlDoc.selectNodes("/bookstore/book/price/text()")

假如您正在使用 IE 5 或更高的版本，可以[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=xpth_select_pricenodes_text)。

**选取价格高于 35 的 price 价格**

下面的例子会选取所有价格高于 35 的 price 节点：

xmlDoc.selectNodes("/bookstore/book[price>35]/price")

假如您正在使用 IE 5 或更高的版本，可以[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=xpth_select_pricenodes_35)。

**选取价格高于 35 的 title 节点**

下面的例子会选取所有价格高于 35 的 title 节点：

xmlDoc.selectNodes("/bookstore/book[price>35]/title")

假如您正在使用 IE 5 或更高的版本，可以[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=xpth_select_pricenodes_high)

# XPath Axes（轴）

* [XPath 语法](http://www.w3school.com.cn/xpath/xpath_syntax.asp)
* [XPath 运算符](http://www.w3school.com.cn/xpath/xpath_operators.asp)

## XML 实例文档

我们将在下面的例子中使用此 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>

## 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。 |

## 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]

### 实例

|  |  |
| --- | --- |
| **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。 |

# 您已经学习了 XPath，下一步呢？

* [Previous Page](http://www.w3school.com.cn/xpath/xpath_examples.asp)
* [Next Page](http://www.w3school.com.cn/xpath/xpath_functions.asp)

**XPath 概要**

本教程已经向您讲解了如何在 XML 文档中查找信息。

您已经学习了如何使用 XPath 通过元素和属性在 XML 文档中进行导航。

你也学习了如何使用 XPath 中内建的某些标准函数。

如需更多有关 XPath 的信息，请阅读我们的《[XPath 参考手册](http://www.w3school.com.cn/xpath/xpath_functions.asp)》。

**您已经学习了 XPath，下一步应当学习什么呢？**

您下一步应该学习 XSLT、XQuery、XLink 以及 XPointer。

**XSLT**

XSLT 是针对 XML 文件的样式表语言。

通过 XSLT，您可以把 XML 文件转换为其他的格式，比如 XHTML。

如果您希望学习更多有关 XSLT 的知识，请访问我们的《[XSLT 教程](http://www.w3school.com.cn/xsl/index.asp)》。

**XQuery**

。

。

。

。