XPath 简介

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 节点

XPath 术语

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。

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) 或者步 (steps) 来选取的。

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。

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。

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。

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。

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。

XPath Axes（轴）

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：

```
1  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。

XPath 运算符

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

XPath、XQuery 以及 XSLT 函数函数参考手册

有关数值的函数

名称	说明
fn:number(arg)	返回参数的数值。参数可以是布尔值、字符串或节点集。例子：number('100')结果：100
fn:abs(num)	返回参数的绝对值。例子：abs(3.14)结果：3.14例子：abs(-3.14)结果：3.14
fn:ceiling(num)	返回大于 num 参数的最小整数。例子：ceiling(3.14)结果：4
fn:floor(num)	返回不大于 num 参数的最大整数。例子：floor(3.14)结果：3
fn:round(num)	把 num 参数舍入为最接近的整数。例子：round(3.14)结果：3
fn:round-half-to-even()	例子：round-half-to-even(0.5)结果：0例子：round-half-to-even(1.5)结果：2例子：round-half-to-even(2.5)结果：2

有关字符串的函数

名称	说明
fn:string(arg)	返回参数的字符串值。参数可以是数字、逻辑值或节点集。例子：string(314)结果："314"
fn:concat(string,string,...)	返回字符串的拼接。例子：concat('XPath ','is ','FUN!')结果：'XPath is FUN!'
fn:string-join((string,string,...),sep)	使用 sep 参数作为分隔符，来返回 string 参数拼接后的字符串。例子：string-join(('We', 'are', 'having', 'fun!'), ' ')结果：' We are having fun! '例子：string-join(('We', 'are', 'having', 'fun!'))结果：'Wearehavingfun!'例子：string-join((), 'sep')结果：''
fn:substring(string,start,len)fn:substring(string,start)	返回从 start 位置开始的指定长度的子字符串。第一个字符的下标是 1。如果省略 len 参数，则返回从位置 start 到字符串末尾的子字符串。例子：substring('Beatles',1,4)结果：'Beat'例子：substring('Beatles',2)结果：'eatles'
fn:string-length(string)fn:string-length()	返回指定字符串的长度。如果没有 string 参数，则返回当前节点的字符串值的长度。例子：string-length('Beatles')结果：7
fn:upper-case(string)	把 string 参数转换为大写。例子：upper-case('The XML')结果：'THE XML'
fn:lower-case(string)	把 string 参数转换为小写。例子：lower-case('The XML')结果：'the xml'
fn:translate(string1,string2,string3)	把 string1 中的 string2 替换为 string3。例子：translate('12:30','30','45')结果：'12:45'例子：translate('12:30','03','54')结果：'12:45'例子：translate('12:30','0123','abcd')结果：'bc:da'
fn:contains(string1,string2)	如果 string1 包含 string2，则返回 true，否则返回 false。例子：contains('XML','XM')结果：true
fn:starts-with(string1,string2)	如果 string1 以 string2 开始，则返回 true，否则返回 false。例子：starts-with('XML','X')结果：true
fn:ends-with(string1,string2)	如果 string1 以 string2 结尾，则返回 true，否则返回 false。例子：ends-with('XML','X')结果：false
fn:substring-before(string1,string2)	返回 string2 在 string1 中出现之前的子字符串。例子：substring-before('12/10','/')结果：'12'

名称	说明
fn:substring-after(string1,string2)	返回 string2 在 string1 中出现之后的子字符串。例子：substring-after('12/10','/')结果：'10'
fn:matches(string,pattern)	如果 string 参数匹配指定的模式，则返回 true，否则返回 false。例子：matches("Merano", "ran")结果：true
fn:replace(string,pattern,replace)	把指定的模式替换为 replace 参数，并返回结果。例子：replace("Bella Italia", "l", "")*结果：'Be**a Ita*ia'例子：replace("Bella Italia", "l", "")结果：'Bea Itaia'

关于布尔值的函数

名称	说明
fn:boolean(arg)	返回数字、字符串或节点集的布尔值。
fn:not(arg)	首先通过 boolean() 函数把参数还原为一个布尔值。如果该布尔值为 false，则返回 true，如果该布尔值为 true，则返回 false。例子：not(true())结果：false
fn:true()	返回布尔值 true。例子：true()结果：true
fn:false()	返回布尔值 false。例子：false()结果：false

合计函数

名称	说明
fn:count((item,item,...))	返回节点的数量。
fn:avg((arg,arg,...))	返回参数值的平均数。 例子：avg((1,2,3)) 结果：2
fn:max((arg,arg,...))	返回大于其它参数的参数。 例子：max((1,2,3)) 结果：3 例子：max(('a', 'k')) 结果：'k'
fn:min((arg,arg,...))	返回小于其它参数的参数。 例子：min((1,2,3)) 结果：1 例子：min(('a', 'k')) 结果：'a'
fn:sum(arg,arg,...)	返回指定节点集中每个节点的数值的总和。

上下文函数

名称	说明
fn:position()	返回当前正在被处理的节点的 index 位置。例子：//book[position()<=3]结果：选择前三个 book 元素
fn:last()	返回在被处理的节点列表中的项目数目。例子：//book[last()]结果：选择最后一个 book 元素
fn:current-dateTime()	返回当前的 dateTime（带有时区）。
fn:current-date()	返回当前的日期（带有时区）。
fn:current-time()	返回当前的时间（带有时区）。

xpath解析

- 匹配演示 - 猫眼电影top100

```
【1】查找所有的dd节点
    //dd
【2】获取所有电影的名称的a节点：所有class属性值为name的a节点
    //p[@class="name"]/a
【3】获取dl节点下第2个dd节点的电影节点
    //dl[@class="board-wrapper"]/dd[2]
【4】获取所有电影详情页链接：获取每个电影的a节点的href的属性值
    //p[@class="name"]/a/@href

【注意】
    1> 只要涉及到条件，加 [] ：//dl[@class="xxx"]    //dl/dd[2]
    2> 只要获取属性值，加 @  ：//dl[@class="xxx"]    //p/a/@href
```

- 选取节点

```
【1】// ：从所有节点中查找（包括子节点和后代节点）
【2】@  ：获取属性值
  2.1> 使用场景1（属性值作为条件）
       //div[@class="movie-item-info"]
  2.2> 使用场景2（直接获取属性值）
       //div[@class="movie-item-info"]/a/img/@src

【3】练习 - 猫眼电影top100
  3.1> 匹配电影名称

      //div[@class="movie-item-info"]/p[1]/a/@title
```

```
3.2> 匹配电影主演
    //div[@class="movie-item-info"]/p[2]/text()
3.3> 匹配上映时间
    //div[@class="movie-item-info"]/p[3]/text()
3.4> 匹配电影链接
    //div[@class="movie-item-info"]/p[1]/a/@href
```

- 匹配多路径（或）

```
xpath表达式1 | xpath表达式2 | xpath表达式3
```

- 常用函数

```
【1】text() ：获取节点的文本内容
    xpath表达式末尾不加 /text() :则得到的结果为节点对象
    xpath表达式末尾加 /text() 或者 /@href ： 则得到结果为字符串

【2】contains() ： 匹配属性值中包含某些字符串节点
    匹配class属性值中包含 'movie-item' 这个字符串的 div 节点
     //div[contains(@class,"movie-item")]
```

- 终极总结

```
【1】字符串：xpath表达式的末尾为：/text() 、/@href  得到的列表中为'字符串'

【2】节点对象：其他剩余所有情况得到的列表中均为'节点对象'
    [<element dd at xxxa>,<element dd at xxxb>,<element dd at xxxc>]
    [<element div at xxxa>,<element div at xxxb>]
    [<element p at xxxa>,<element p at xxxb>,<element p at xxxc>]

【1】基准xpath：匹配所有电影信息的节点对象列表
  //dl[@class="board-wrapper"]/dd
  [<element dd at xxx>,<element dd at xxx>,...]

【2】遍历对象列表，依次获取每个电影信息
  item = {}
  for dd in dd_list:
      item['name'] = dd.xpath('.//p[@class="name"]/a/text()').strip()
      item['star'] = dd.xpath('.//p[@class="star"]/text()').strip()
      item['time'] = dd.xpath('.//p[@class="releasetime"]/text()').strip()
```

- 课堂练习

```
【1】匹配汽车之家-二手车,所有汽车的链接 ：
   //li[@class="cards-li list-photo-li"]/a[1]/@href
   //a[@class="carinfo"]/@href
【2】匹配汽车之家-汽车详情页中,汽车的
    2.1)名称：  //div[@class="car-box"]/h3/text()
    2.2)里程：  //ul/li[1]/h4/text()
    2.3)时间：  //ul/li[2]/h4/text()
    2.4)挡位+排量：//ul/li[3]/h4/text()
    2.5)所在地：//ul/li[4]/h4/text()
    2.6)价格：   //div[@class="brand-price-item"]/span[@class="price"]/text()
```