XQuery 教程

XQuery 之于 XML 作用就类似于 SQL 之于数据库的作用。

XQuery 被设计用来查询 XML 数据。

现在开始学习 XQuery！

XQuery 实例

```
for $x in doc("books.xml")/bookstore/book
where $x/price>30
order by $x/title
return $x/title
```

XQuery 参考手册

在菜鸟教程中，我们提供 XQuery 1.0 中所有运算符、内置函数、数据类型的完整参考手册。

XQuery 参考手册

点我分享笔记

反馈/建议

Copyright © 2013-2018 菜鸟教程 runoob.com All Rights Reserved. 备案号：闽ICP备15012807号-1

首页 HTML CSS JS 本地书签

XQuery 简介

解释 XQuery 最佳方式是这样讲：XQuery 相对于 XML 的关系，等同于 SQL 相对于数据库表的关系。

XQuery 被设计用来查询 XML 数据 - 不仅仅限于 XML 文件，还包括任何可以 XML 形态呈现的数据，包括数据库。

您应该具备的基础知识：

在您继续学习之前，需要对下面的知识有基本的了解：

- HTML / XHTML
- XML / XML 命名空间

XPath

如果您希望首先学习这些项目，请在我们的 首页 访问这些教程。

什么是 XQuery?

XQuery

XQuery 是用于 XML 数据查询的语言

XQuery 对 XML 的作用类似 SQL 对数据库的作用

XQuery 建立在 XPath 表达式之上

XQuery 被所有主要的数据库引擎支持（IBM、Oracle、Microsoft等等）

XQuery 是 W3C 标准

XQuery 和 XML 查询有关

XQuery 是用来从 XML 文档查找和提取元素及属性的语言。

这是一个 XQuery 解决实际问题的例子：

"从存储在名为 cd_catalog.xml 的 XML 文档中的 CD 集那里选取所有价格低于 10 美元的 CD 记录。"

XQuery 与 XPath

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。假如您已经学习了 XPath，那么学习 XQuery 也不会有问题。

您可以在我们的《XPath 教程》中阅读更多有关 XPath 的知识。

XQuery - 应用举例

XQuery 可用于：

提取信息以便在网络服务中使用

生成摘要报告

把 XML 数据转换为 XHTML

为获得相关信息而搜索网络文档

XQuery 是一个 W3C 推荐标准

XQuery 与多种 W3C 标准相兼容，比如 XML、Namespaces、XSLT、XPath 以及 XML Schema。

XQuery 1.0 在 2007年1月23日 被确立为 W3C 推荐标准。

如需获得更多有关 W3C 的 XQuery 活动的信息，请阅读我们的《W3C 教程》。

□ XQuery 教程 XQuery 实例 □

□点我分享笔记

反馈/建议

Copyright © 2013-2018 菜鸟教程 runoob.com All Rights Reserved. 备案号：闽ICP备15012807号-1

□

XQuery 实例

在本节，让我们通过研究一个例子来学习一些基础的 XQuery 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 "books.xml" 文件。

如何从 "books.xml" 选取节点？

函数

XQuery 使用函数来提取 XML 文档中的数据。

doc() 用于打开 "books.xml" 文件：

```
doc("books.xml")
```

路径表达式

XQuery 使用路径表达式在 XML 文档中通过元素进行导航。

下面的路径表达式用于在 "books.xml" 文件中选取所有的 title 元素：

```
doc("books.xml")/bookstore/book/title
```

(/bookstore 选取 bookstore 元素，/book 选取 bookstore 元素下的所有 book 元素，而 /title 选取每个 book 元素下的所有 title 元素)

上面的 XQuery 可提取以下数据：

```
<title lang="en">Everyday Italian</title>
<title lang="en">Harry Potter</title>
<title lang="en">XQuery Kick Start</title>
<title lang="en">Learning XML</title>
```

谓语

XQuery 使用谓语来限定从 XML 文档所提取的数据。

下面的谓语用于选取 bookstore 元素下的所有 book 元素，并且所选取的 book 元素下的 price 元素的值必须小于 30：

```
doc("books.xml")/bookstore/book[price<30]
```

上面的 XQuery 可提取到下面的数据：

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

□ XQuery 简介　　XQuery FLWOR 表达式 □

□点我分享笔记

反馈/建议

Copyright © 2013-2018 菜鸟教程 runoob.com All Rights Reserved. 备案号：闽ICP备15012807号-1

□ XQuery 实例　　XQuery FLWOR + HTML □

XQuery FLWOR 表达式

XML 实例文档

我们将在下面的例子中继续使用这个 "books.xml" 文档（与上一节中的 XML 文件相同）。

在您的浏览器中查看 "books.xml" 文件。

如果使用 FLWOR 从 "books.xml" 选取节点

请看下面这个路径表达式：

```
doc("books.xml")/bookstore/book[price>30]/title
```

上面这个表达式可选取 bookstore 元素下的 book 元素下所有的 title 元素，并且其中的 price 元素的值必须大于 30。

下面这个 FLWOR 表达式所选取的数据和上面的路径表达式是相同的：

```
for $x in doc("books.xml")/bookstore/book
where $x/price>30
return $x/title
```

输出结果：

```
<title lang="en">XQuery Kick Start</title>
<title lang="en">Learning XML</title>
```

通过 FLWOR，您可以对结果进行排序：

```
for $x in doc("books.xml")/bookstore/book
where $x/price>30
order by $x/title
return $x/title
```

FLWOR 是 "For, Let, Where, Order by, Return" 的只取首字母缩写。

for 语句把 bookstore 元素下的所有 book 元素提取到名为 $x 的变量中。

where 语句选取了 price 元素值大于 30 的 book 元素。

order by 语句定义了排序次序。将根据 title 元素进行排序。

return 语句规定返回什么内容。在此返回的是 title 元素。

上面的 XQuery 表达式的结果：

```
<title lang="en">Learning XML</title>
<title lang="en">XQuery Kick Start</title>
```

XQuery 实例 XQuery FLWOR + HTML

点我分享笔记

反馈/建议

Copyright © 2013-2018 菜鸟教程 runoob.com All Rights Reserved. 备案号：闽ICP备15012807号-1

XQuery FLWOR 表达式 XQuery 术语

XQuery FLWOR + HTML

XML 实例文档

我们将在下面的例子中继续使用这个 "books.xml" 文档（与上一节中的文件相同）。

在您的浏览器中查看 "books.xml" 文件。

在一个 HTML 列表中提交结果

请看下面的 XQuery FLWOR 表达式：

```
for $x in doc("books.xml")/bookstore/book/title
order by $x
return $x
```

上面的表达式会选取 bookstore 元素下的 book 元素下的所有 title 元素，并以字母顺序返回 title 元素。

现在，我们希望使用 HTML 列表列出我们的书店中所有的书目。我们向 FLWOR 表达式添加 <ul> 和 <li> 标签：

```
<ul>
{
for $x in doc("books.xml")/bookstore/book/title
order by $x
return <li>{$x}</li>
}
</ul>
```

以上代码输出结果：

```
<ul>
<li><title lang="en">Everyday Italian</title></li>
<li><title lang="en">Harry Potter</title></li>
<li><title lang="en">Learning XML</title></li>
<li><title lang="en">XQuery Kick Start</title></li>
</ul>
```

现在我们希望去除 title 元素，而仅仅显示 title 元素内的数据。

```
<ul>
{
for $x in doc("books.xml")/bookstore/book/title
order by $x
return <li>{data($x)}</li>
}
</ul>
```

结果将是一个 HTML 列表：

```
<ul>
<li>Everyday Italian</li>
<li>Harry Potter</li>
<li>Learning XML</li>
<li>XQuery Kick Start</li>
</ul>
```

□ XQuery FLWOR 表达式　　XQuery 术语 □

□点我分享笔记

反馈/建议

Copyright © 2013-2018 菜鸟教程 runoob.com All Rights Reserved. 备案号：闽ICP备15012807号-1

□ XQuery FLWOR + HTML　　XQuery 语法 □

XQuery 术语

在 XQuery 中，有七种节点：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）。

XQuery 术语

节点

在 XQuery 中，有七种节点：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。

请看下面的 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)
```

基本值是无父或无子的节点。

基本值的例子：

```
J K. Rowling

"en"
```

项目

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）

节点元素可有零个、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>
```

反馈/建议

Copyright © 2013-2018 菜鸟教程 runoob.com All Rights Reserved. 备案号：闽ICP备15012807号-1

XQuery 语法

XQuery 对大小写敏感，XQuery 的元素、属性以及变量必须是合法的 XML 名称。

XQuery 的基础语法规则：

一些基本的语法规则：

XQuery 对大小写敏感

XQuery 的元素、属性以及变量必须是合法的 XML 名称。

XQuery 字符串值可使用单引号或双引号。

XQuery 变量由 "$" 并跟随一个名称来进行定义，举例，$bookstore

XQuery 注释被 (: 和 :) 分割，例如，(: XQuery 注释 :)

XQuery 条件表达式

"If-Then-Else" 可以在 XQuery 中使用。

请看下面的例子：

```
for $x in doc("books.xml")/bookstore/book
return if ($x/@category="CHILDREN")
then <child>{data($x/title)}</child>
else <adult>{data($x/title)}</adult>
```

请注意 "If-Then-Else" 的语法：if 表达式后的圆括号是必需的。else 也是必需的，不过只写 "else ()" 也可以。

上面的例子的结果：

```
<adult>Everyday Italian</adult>
<child>Harry Potter</child>
<adult>Learning XML</adult>
<adult>XQuery Kick Start</adult>
```

XQuery 比较

在 XQuery 中，有两种方法来比较值。

1. 通用比较：=, !=, <, <=, >, >=
2. 值的比较：eq、ne、lt、le、gt、ge

这两种比较方法的差异如下：

请看下面的 XQuery 表达式：

```
$bookstore//book/@q > 10
```

如果 q 属性的值大于 10，上面的表达式的返回值为 true。

如下实例，如果仅返回一个 q，且它的值大于 10，那么表达式返回 true。如果不止一个 q 被返回，则会发生错误：

```
$bookstore//book/@q gt 10
```

Copyright © 2013-2018 菜鸟教程 runoob.com All Rights Reserved. 备案号：闽ICP备15012807号-1

XQuery 添加元素 和属性

XML 实例文档

我们将在下面的例子中继续使用这个 "books.xml" 文档（和上面的章节所使用的 XML 文件相同）。

在您的浏览器中查看 "books.xml" 文件。

向结果添加元素和属性

正如在前面一节看到的，我们可以在结果中引用输入文件中的元素和属性：

```
for $x in doc("books.xml")/bookstore/book/title
order by $x
return $x
```

上面的 XQuery 表达式会在结果中引用 title 元素和 lang 属性，就像这样：

```
<title lang="en">Everyday Italian</title>
<title lang="en">Harry Potter</title>
<title lang="en">Learning XML</title>
<title lang="en">XQuery Kick Start</title>
```

以上 XQuery 表达式返回 title 元素的方式和它们在输入文档中被描述的方式的相同的。

现在我们要向结果添加我们自己的元素和属性！

添加 HTML 元素和文本

现在，我们要向结果添加 HTML 元素。我们会把结果放在一个 HTML 列表中：

```
<html>
<body>

<h1>Bookstore</h1>

<ul>
{
for $x in doc("books.xml")/bookstore/book
order by $x/title
return <li>{data($x/title)}. Category: {data($x/@category)}</li>
}
</ul>

</body>
</html>
```

以上 XQuery 表达式会生成下面的结果：

```
<html>
<body>
```

```
<h1>Bookstore</h1>

<ul>
<li>Everyday Italian. Category: COOKING</li>
<li>Harry Potter. Category: CHILDREN</li>
<li>Learning XML. Category: WEB</li>
<li>XQuery Kick Start. Category: WEB</li>
</ul>

</body>
</html>
```

向 HTML 元素添加属性

接下来，我们要把 category 属性作为 HTML 列表中的 class 属性来使用：

```
<html>
<body>

<h1>Bookstore</h1>

<ul>
{
for $x in doc("books.xml")/bookstore/book
order by $x/title
return <li class="{data($x/@category)}">{data($x/title)}</li>
}
</ul>

</body>
</html>
```

上面的 XQuery 表达式可生成以下结果：

```
<html>
<body>
<h1>Bookstore</h1>

<ul>
<li class="COOKING">Everyday Italian</li>
<li class="CHILDREN">Harry Potter</li>
<li class="WEB">Learning XML</li>
<li class="WEB">XQuery Kick Start</li>
</ul>

</body>
</html>
```

□ XQuery 语法　　XQuery 选择和过滤 □

□点我分享笔记

反馈/建议

Copyright © 2013-2018 菜鸟教程 runoob.com All Rights Reserved. 备案号：闽ICP备15012807号-1

□ XQuery 添加元素和属性　　XQuery 函数 □

XQuery 选择 和 过滤

XML实例文档

我们将在下面的例子中继续使用这个 "books.xml" 文档（和上面的章节所使用的 XML 文件相同）。

在您的浏览器中查看 "books.xml" 文件。

选择和过滤元素

正如在前面的章节所看到的，我们使用路径表达式或 FLWOR 表达式来选取和过滤元素。

请看下面的 FLWOR 表达式：

```
for $x in doc("books.xml")/bookstore/book
where $x/price>30
order by $x/title
return $x/title
```

for - （可选） 向每个由 in 表达式返回的项目捆绑一个变量

let - （可选）

where - （可选） 设定一个条件

order by - （可选） 设定结果的排列顺序

return - 规定在结果中返回的内容

for 语句

for 语句可将变量捆绑到由 in 表达式返回的每个项目。for 语句可产生迭代。在同一个 FLWOR 表达式中可存在多重 for 语句。

如需在一个 for 语句中进行指定次数地循环，您可使用关键词 to ：

```
for $x in (1 to 5)
return <test>{$x}</test>
```

结果：

```
<test>1</test>
<test>2</test>
<test>3</test>
<test>4</test>
<test>5</test>
```

关键词 **at** 可用于计算迭代：

```
for $x at $i in doc("books.xml")/bookstore/book/title
return <book>{$i}. {data($x)}</book>
```

结果：

```
<book>1. Everyday Italian</book>
<book>2. Harry Potter</book>
<book>3. XQuery Kick Start</book>
<book>4. Learning XML</book>
```

在 for 语句中同样允许多个 in 表达式。请使用逗号来分割每一个 in 表达式：

```
for $x in (10,20), $y in (100,200)
return <test>x={$x} and y={$y}</test>
```

结果：

```
<test>x=10 and y=100</test>
<test>x=10 and y=200</test>
<test>x=20 and y=100</test>
<test>x=20 and y=200</test>
```

let 语句

let 语句可完成变量分配，并可避免多次重复相同的表达式。let 语句不会导致迭代。

```
let $x := (1 to 5)
return <test>{$x}</test>
```

结果：

```
<test>1 2 3 4 5</test>
```

where 语句

where 语句用于为结果设定一个或多个条件（criteria）。

```
where $x/price>30 and $x/price<100
```

order by 语句

order by 语句用于规定结果的排序次序。在这里，我们要根据 category 和 title 来对结果进行排序：

```
for $x in doc("books.xml")/bookstore/book
order by $x/@category, $x/title
return $x/title
```

结果：

```
<title lang="en">Harry Potter</title>
<title lang="en">Everyday Italian</title>
<title lang="en">Learning XML</title>
<title lang="en">XQuery Kick Start</title>
```

return 语句：

return 语句规定要返回的内容。

```
for $x in doc("books.xml")/bookstore/book
return $x/title
```

结果：

```
<title lang="en">Everyday Italian</title>
<title lang="en">Harry Potter</title>
<title lang="en">XQuery Kick Start</title>
<title lang="en">Learning XML</title>
```

□ XQuery 添加元素和属性　　XQuery 函数 □

□点我分享笔记

反馈/建议

Copyright © 2013-2018 菜鸟教程 runoob.com All Rights Reserved. 备案号：闽ICP备15012807号-1

□ XQuery 选择和过滤　　XQuery 总结 □

XQuery 函数

XQuery 1.0、XPath 2.0 以及 XSLT 2.0 共享相同的函数库。

XQuery 函数

XQuery 含有超过 100 个内建的函数。这些函数可用于字符串值、数值、日期以及时间比较、节点和 QName 操作、序列操作、逻辑值等等。您也可在 XQuery 中定义自己的函数。

XQuery 内建函数

XQuery 函数命名空间的 URI：

http://www.w3.org/2005/02/xpath-functions

函数命名空间的默认前缀是 fn:。

提示：函数经常被通过 fn: 前缀进行调用，例如 fn:string()。不过，由于 fn: 是命名空间的默认前缀，所以函数名称不必在被调用时使用前缀。

您可以在我们的 XPath 教程中找到完整的《内建 XQuery 函数参考手册》。

函数调用实例

函数调用可与表达式一同使用。请看下面的例子：

例1：在元素中

```
<name>{upper-case($booktitle)}</name>
```

例2: 在路径表达式的谓语中

```
doc("books.xml")/bookstore/book[substring(title,1,5)='Harry']
```

例3: 在 let 语句中

```
let $name := (substring($booktitle,1,4))
```

XQuery 用户定义函数

如果找不到所需的 XQuery 函数，你可以编写自己的函数。

可在查询中或独立的库中定义用户自定义函数。

语法

```
declare function 前缀:函数名($参数 AS 数据类型)
AS 返回的数据类型
{
 ...函数代码...
}
```

关于用户自定义函数的注意事项：

请使用 declare function 关键词

函数名须使用前缀

参数的数据类型通常与在 XML Schema 中定义的数据类型一致

函数主体须被花括号包围

一个在查询中声明的用户自定义函数的例子：

```
declare function local:minPrice($p as xs:decimal?,$d as xs:decimal?)
AS xs:decimal?
{
let $disc := ($p * $d) div 100
return ($p - $disc)
}

Below is an example of how to call the function above:

<minPrice>{local:minPrice($book/price,$book/discount)}</minPrice>
```

□ XQuery 选择和过滤　　XQuery 总结 □

□点我分享笔记

反馈/建议

Copyright © 2013-2018 菜鸟教程 runoob.com All Rights Reserved. 备案号：闽ICP备15012807号-1

□

首页 HTML CSS JS 本地书签

□ XQuery 函数　　XQuery 参考手册 □

您已经学习了 XQuery，接下来该学习什么内容呢？

XQuery 概要

本教程已经向您讲解了如何查询 XML 数据。

您已经明白，XQuery 被设计来查询以 XML 形态存在的任何数据，包括数据库。

您也学习了如何使用 FLWOR 表达式来查询 XML 数据，以及如何由选定的数据构造 XHTML 输出。

如需更多有关 XQuery 的信息，请阅读我们的《XQuery 参考手册》。

您已经学习了 XQuery，接下来该学习什么内容呢？

下一步该学习 XLink 和 XPointer。

XLink 和 XPointer

XML 中的链接被分为两部分：XLink 和 XPointer。

XLink 和 XPointer 定义了一套标准的在XML文档中创建超级链接的方法。

如果您希望学习更多有关 XLink 和 XPointer 的知识，请访问我们的《XLink 和 XPointer 教程》。

XQuery 函数 XQuery 参考手册

点我分享笔记

反馈/建议

Copyright © 2013-2018 菜鸟教程 runoob.com All Rights Reserved. 备案号：闽ICP备15012807号-1

XQuery 参考手册

XQuery 1.0 和 XPath 2.0 分享相同的数据模型，并支持相同的函数和运算符。

XQuery 函数

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

XPath Operators

XPath Functions

XQuery 数据类型

XQuery 分享与 XML Schema 1.0 (XSD) 相同的数据类型。

XSD String

XSD Date

XSD Numeric

XSD Misc

点我分享笔记

反馈/建议

Copyright © 2013-2018 菜鸟教程 runoob.com All Rights Reserved. 备案号：闽ICP备15012807号-1