

# 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](#)