

# XML DOM 节点列表

节点列表由 `getElementsByTagName()` 方法和 `childNodes` 属性返回。



## 尝试一下 - 实例

下面的实例使用 XML 文件 [books.xml](#)。

函数 `loadXMLDoc()`，位于外部 JavaScript 中，用于加载 XML 文件。

[从第一个 <title> 元素获取文本](#)

本例使用 `getElementsByTagName()` 方法从 "books.xml" 中的第一个 `<title>` 元素获取文本。

[使用 length 属性遍历节点](#)

本例使用节点列表和 `length` 属性来遍历 "books.xml" 中所有的 `<title>` 元素。

[获取元素的属性](#)

本例使用属性列表从 "books.xml" 中的第一个 `<book>` 元素获取属性。

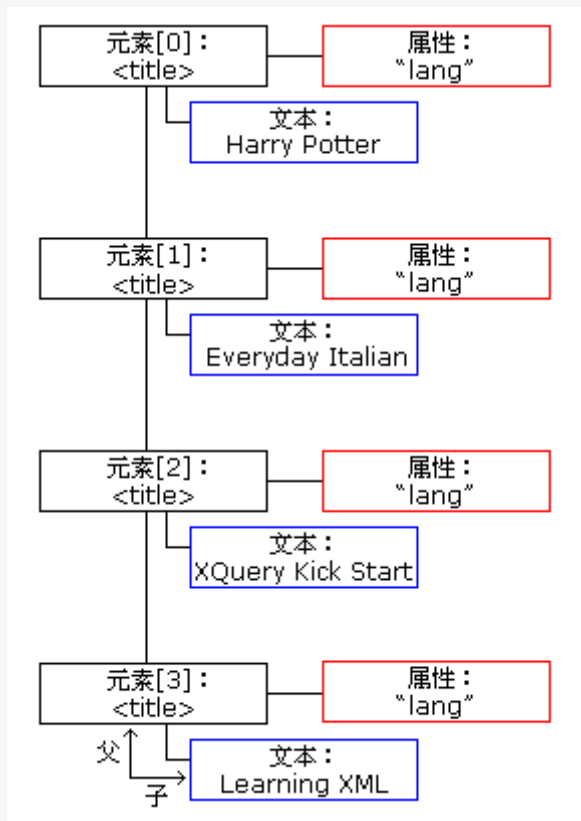
## DOM 节点列表 ( Node List )

当使用诸如 `childNodes` 或 `getElementsByTagName()` 的属性或方法时，会返回节点列表对象。

节点列表对象表示节点的列表，与 XML 中的顺序相同。

节点列表中的节点使用从 0 开始的索引号进行访问。

下面的图像表示 "[books.xml](#)" 中 `<title>` 元素的节点列表：



下面的代码片段通过使用 `loadXMLDoc()` 把 `"books.xml"` 载入 `xmlDoc` 中，并返回 `"books.xml"` 中 `title` 元素的节点列表：

```
xmlDoc=loadXMLDoc("books.xml");  
  
x=xmlDoc.getElementsByTagName("title");
```

在上面的语句执行之后，`x` 是节点列表对象。

下面的代码片段从节点列表 (`x`) 中的第一个 `<title>` 元素返回文本：

### 实例

```
txt=x[0].childNodes[0].nodeValue;
```

尝试一下 »

在上面的语句执行之后，`txt = "Everyday Italian"`。

## 节点列表长度 (Node List Length)

节点列表对象会保持自身的更新。如果删除或添加了元素，列表会自动更新。

节点列表的 `length` 属性是列表中节点的数量。

下面的代码片段通过使用 `loadXMLDoc()` 把 `"books.xml"` 载入 `xmlDoc` 中，并返回 `"books.xml"` 中 `<title>` 元素的数量：

```
xmlDoc=loadXMLDoc("books.xml");  
  
x=xmlDoc.getElementsByTagName('title').length;
```

在上面的语句执行之后，`x = 4`。

节点列表的长度可用于遍历列表中所有的元素。

下面的代码片段使用 `length` 属性来遍历 `<title>` 元素的列表：

### 实例

```
xmlDoc=loadXMLDoc("books.xml");  
// x 变量获取指定节点列表  
x=xmlDoc.getElementsByTagName('title');  
for (i=0;i<x.length;i++)  
{  
    document.write(x[i].childNodes[0].nodeValue);  
    document.write("");  
}
```

输出：

```
Everyday Italian  
Harry Potter  
XQuery Kick Start  
Learning XML
```

尝试一下 »

实例解释：

1. 使用 `loadXMLDoc()` 把 `"books.xml"` 载入 `xmlDoc` 中

2. 设置 x 变量来保存所有 title 元素的节点列表
3. 从所有 <title> 元素的文本节点输出值

## DOM 属性列表（命名节点图 Named Node Map）

元素节点的 attributes 属性返回属性节点的列表。

这被称为命名节点图（Named Node Map），除了方法和属性上的一些差别以外，它与节点列表相似。

属性列表会保持自身的更新。如果删除或添加属性，这个列表会自动更新。

下面的代码片段通过使用 [loadXMLDoc\(\)](#) 把 "books.xml" 载入 xmlDoc 中，并返回 "books.xml" 中第一个 <book> 元素的属性节点列表：

```
xmlDoc=loadXMLDoc("books.xml");  
  
x=xmlDoc.getElementsByTagName('book')[0].attributes;
```

在上面的代码执行之后，x.length 等于属性的数量，可使用 x.getNamedItem() 返回属性节点。

下面的代码片段显示一个 book 的 "category" 属性的值，以及其属性的数量：

### 实例

```
xmlDoc=loadXMLDoc("books.xml");  
x=xmlDoc.getElementsByTagName("book")[0].attributes;  
document.write(x.getNamedItem("category").nodeValue);  
document.write(" " + x.length);
```

输出：

```
cooking  
1
```

[尝试一下 »](#)

实例解释：

1. 使用 loadXMLDoc() 把 "books.xml" 载入 xmlDoc 中
2. 设置 x 变量来保存第一个 <book> 元素的所有属性的一个列表
3. 从 "category" 属性输出值
4. 输出属性列表的长度

[← XML DOM 节点信息](#)

[XML DOM – 遍历节点树 →](#)

[点我分享笔记](#)

