

XSLT - 在客户端

如果您的浏览器支持 XSLT，那么在浏览器中它可被用来将文档转换为 XHTML。

JavaScript 解决方案

在前面的章节，我们已向您讲解如何使用 XSLT 将某个 XML 文档转换为 XHTML。我们是通过以下途径完成这个工作的：向 XML 文件添加 XSL 样式表，并通过浏览器完成转换。

即使这种方法的效果很好，在 XML 文件中包含样式表引用也不总是令人满意的（例如，在无法识别 XSLT 的浏览器这种方法就无法奏效）。

更通用的方法是使用 JavaScript 来完成转换。

通过使用 JavaScript，我们可以：

- 进行浏览器确认测试
- 根据浏览器和用户需求来使用不同的样式表

这就是 XSLT 的魅力所在！XSLT 的设计目的之一就是使数据从一种格式转换到另一种格式成为可能，同时支持不同类型的浏览器以及不同的用户需求。

客户端的 XSLT 转换一定会成为未来浏览器所执行的主要任务之一，同时我们也会看到其在特定的浏览器市场的增长（盲文、听觉浏览器、网络打印机，手持设备，等等）。

XML 文件和 XSL 文件

请看这个在前面的章节已展示过的 XML 文档：

```
<?xml version="1.0" encoding="UTF-8"?>
<catalog>
  <cd>
    <title>Empire Burlesque</title>
    <artist>Bob Dylan</artist>
    <country>USA</country>
    <company>Columbia</company>
    <price>10.90</price>
    <year>1985</year>
  </cd>
  .
  .
</catalog>
```

[查看 XML 文件。](#)

以及附随的 XSL 样式表：

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<body>
  <h2>My CD Collection</h2>
  <table border="1">
    <tr bgcolor="#9acd32">
      <th style="text-align:left">Title</th>
      <th style="text-align:left">Artist</th>
    </tr>
    <xsl:for-each select="catalog/cd">
      <tr>
        <td><xsl:value-of select="title"/></td>
        <td><xsl:value-of select="artist"/></td>
      </tr>
    </xsl:for-each>
  </table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

[查看 XSL 文件。](#)

请注意，这个 XML 文件没有包含对 XSL 文件的引用。

重要事项：上面这句话意味着，XML 文件可使用多个不同的 XSL 样式表来进行转换。

在浏览器中把 XML 转换为 XHTML

这是用于在客户端把 XML 文件转换为 XHTML 的源代码：

实例

```
<!DOCTYPE html>
<html>
<head>
<script>
function loadXMLDoc(filename)
{
  if (window.ActiveXObject)
  {
    xhttp = new ActiveXObject("Msxml2.XMLHTTP");
  }
  else
  {
    xhttp = new XMLHttpRequest();
  }
  xhttp.open("GET", filename, false);
  try {xhttp.responseType = "msxml-document"} catch(err) {} // Helping IE11
  xhttp.send("");
```

```
return xhttp.responseXML;
}
function displayResult()
{
xml = loadXMLDoc("cdcatalog.xml");
xsl = loadXMLDoc("cdcatalog.xsl");
// code for IE
if (window.ActiveXObject || xhttp.responseType == "msxml-document")
{
ex = xml.transformNode(xsl);
document.getElementById("example").innerHTML = ex;
}
// code for Chrome, Firefox, Opera, etc.
else if (document.implementation && document.implementation.createDocument)
{
xsltProcessor = new XSLTProcessor();
xsltProcessor.importStylesheet(xsl);
resultDocument = xsltProcessor.transformToFragment(xml, document);
document.getElementById("example").appendChild(resultDocument);
}
}
</script>
</head>
<body onload="displayResult()">
<div id="example" />
</body>
</html>
```

[尝试一下 »](#)

提示：假如您不了解如何编写 JavaScript，请学习我们的 [JavaScript 教程](#)。

实例解释：

loadXMLDoc() 函数

loadXMLDoc() 函数是用来加载 XML 和 XSL 文件。

它检查用户拥有的和加载文件的浏览器类型。

displayResult() 函数

该函数用来显示使用 XSL 文件定义样式的 XML 文件。

- 加载 XML 和 XSL 文件
- 测试用户拥有的浏览器类型
- 如果用户浏览器支持 ActiveX 对象：
 - 使用 transformNode() 方法把 XSL 样式表应用到 XML 文档
 - 设置当前文档 (id="example") 的 body 包含已经应用样式的 XML 文档
- 如果用户的浏览器不支持 ActiveX 对象：
 - 创建一个新的 XSLTProcessor 对象并导入 XSL 文件

- 使用 transformToFragment() 方法把 XSL 样式表应用到 XML 文档
- 设置当前文档 (id="example") 的 body 包含已经应用样式的 XML 文档

[← XSLT <xsl:apply-templates> 元素](#)[XSLT 在服务器端 →](#)[📝 点我分享笔记](#)