

1 Uso de ferramentas de procesamento

Para facer o procesamento das follas XSL pódese facer con:

1.1 Os navegadores web

Si o noso navegador o soporta, poderemos empregar XSLT para transformar o documento XML nun novo XHTML. O noso navegador fará de xeito automático a conversión ó XHTML empregando o arquivo XSL que se atopará na mesma ruta ou carpeta que o arquivo orixinal XML.

1.2 A linguaxe Javascript

Outra forma de facer a conversión é empregando Javascript. O uso de Javascript permitirá:

- facer probas específicas de cada navegador.
- empregar diferentes follas de estilo de acordo ó navegador ou necesidades do usuario.

Un dos obxetivos do XSLT foi o facer posible a transformacións de datos dun formato a outro, dando soporte a diferentes navegadores e diferentes necesidades do usuario.

Imos a por a continuación un exemplo de cómo se podería converter un arquivo XML nun XHTML empregando a linguaxe Javascript.

Arquivo XML:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<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>
```

Folla de estilos XSL:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<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 align="left">Title</th>
        <th 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>
```

Nota importante: Fixarse que na folla XML non facemos ninguna referencia á folla de estilos XSL.

Código de Javascript que faría a transformación do xml ó xhtml empregando XSL::

```
<html>
<head>
<script>
function loadXMLDoc(dname)
{
if (window.XMLHttpRequest)
{
xhttp=new XMLHttpRequest();
}
else
{
xhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
xhttp.open("GET",dname,false);
xhttp.send("");
return xhttp.responseXML;
}

function displayResult()
{
xml=loadXMLDoc("cdcatalog.xml");
xsl=loadXMLDoc("cdcatalog.xsl");
// code for IE
if (window.ActiveXObject)
{
ex=xml.transformNode(xsl);
document.getElementById("example").innerHTML=ex;
}
// code for Mozilla, 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>
```

A función loadXMLDoc()

A función loadXMLDoc() emprégase para cargar o arquivo XML e XSL. Comproba qué tipo de navegador estamos usando e carga o arquivo.

A función displayResult()

Esta función emprégase para amosar o arquivo XML seguindo a estrutura indicada no arquivo XSL.

- Cargar o arquivo XML e XSL
- Comprobar qué tipo de navegador estamos usando
- Si o navegador soporta o obxeto ActiveX:
 - ◆ Usar o método transformNode() para aplicar a folla XSL ó documento XML
 - ◆ Carga no corpo do documento (id="example") a folla xml con estilos.
- Si o usuario ten un navegador que non soporta o obxeto ActiveX:
 - ◆ Crear un obxeto XSLTProcessor e importar o arquivo XSL
 - ◆ Usar o método transformToFragment() para aplicar a folla de estilo XSL ó documento XML
 - ◆ Cargar no corpo de documento (id="example") o resultado final

1.3 No servidor

Debido a que non todos os navegadores soportan as transformacións XSLT, unha solución consiste en facer as transformacións de XML ó XHTML no servidor.

Empregando os mesmos arquivos XML e XSL anteriores, un exemplo de cómo se podería facer a transformación no servidor empregando a linguaxe ASP, podería ser:

```
<%  
'Load XML  
set xml = Server.CreateObject("Microsoft.XMLDOM")  
xml.async = false  
xml.load(Server.MapPath("cdcatalog.xml"))  
  
'Load XSL  
set xsl = Server.CreateObject("Microsoft.XMLDOM")  
xsl.async = false  
xsl.load(Server.MapPath("cdcatalog.xsl"))  
  
'Transform file  
Response.Write(xml.transformNode(xsl))  
%>
```