



Lenguaje de Marcas

UT-10 - Conversiones de documentos y transformaciones. XSLT

2021 - 2022



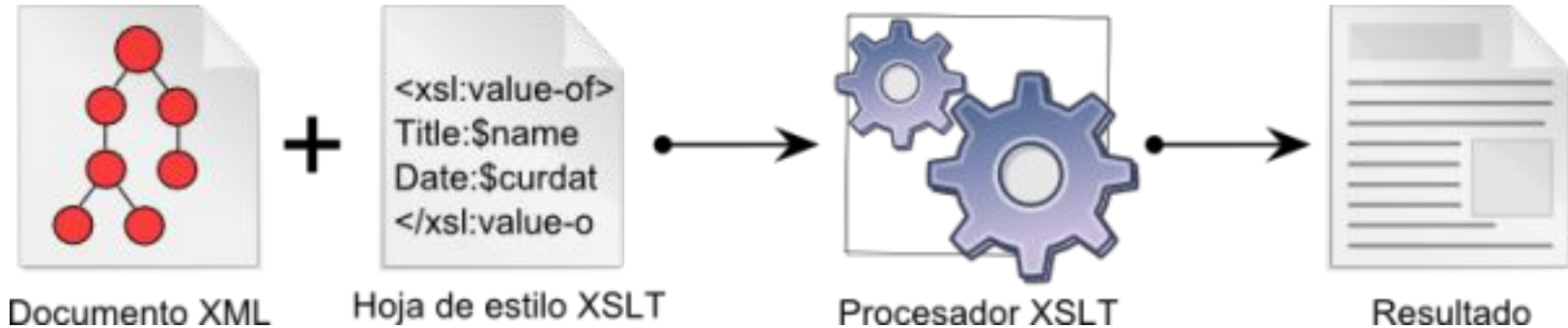
Introducción a XSLT

XSLT (eXtensible Stylesheet Language for Transformations, Lenguaje de hoja de estilo extensible para transformaciones) es un lenguaje que permite aplicar una transformación a un documento XML para obtener otro documento XML, un documento HTML o un documento de texto plano.

La hoja de estilos XSLT con las reglas de transformación es también un documento de texto XML en sí, generalmente con extensión .xsl, por lo que se podrá comprobar si está bien formado o no.

El funcionamiento lo podemos observar en la siguiente imagen:

Introducción a XSLT



A un documento XML se le pueden aplicar una o varias transformaciones XSLT e incluso una transformación CSS.

- Permiten cambiar el orden los elementos.
- Permiten realizar operaciones con sus valores.
- Permiten agrupar elementos.



Introducción a XSLT

XSLT se utiliza para transformar un documento XML en otro documento XML, u otro tipo de documento que es reconocido por un navegador, como HTML y XHTML.

Con XSLT puede agregar / quitar elementos y atributos hacia o desde el archivo de salida. También puede reorganizar y ordenar elementos, realizar pruebas y tomar decisiones acerca de cuáles son los elementos a ocultar y mostrar, y mucho más.

XSLT utiliza XPath para encontrar información en un documento XML. XPath se utiliza para navegar por los elementos y atributos en los documentos XML.



Introducción a XSLT. ¿Cómo funciona?

En el proceso de transformación, XSLT utiliza XPath para definir las partes del documento de origen que debe coincidir con una o varias plantillas predefinidas. Cuando se encuentra una coincidencia, XSLT transformará la parte correspondiente del documento de origen en el documento de resultado.

XSLT es una Recomendación del W3C.



XSL. Vinculamos la hoja de estilo XSLT al documento XML

Agregamos la referencia de hoja de estilo XSLT a un documento XML ("catalogocds.xml"):

```
<?xml version="1.0" encoding="UTF-8"?>  
<?xml-stylesheet type="text/xsl" href="catalogocds.xslt"?>  
<catalogo>
```



XSL. Enlace de documentos Xml con hojas de estilo

El elemento raíz que declara que el documento es una hoja de estilo XSL es `<xsl:stylesheet>` o `<xsl:transform>`.

Nota: `<xsl:stylesheet>` y `<xsl:transform>` son completamente sinónimos

La forma correcta de declarar una hoja de estilo XSL de acuerdo con la Recomendación del W3C XSLT es:

```
<xsl:stylesheet version="1.0"  
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

o

```
<xsl:transform version="1.0"  
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```



XSL. Plantillas: Elemento Xsl:Template

Una hoja de estilo XSL consta de uno o varios conjuntos de reglas que se llaman plantillas. Una plantilla contiene las reglas a aplicar cuando se empareja un nodo especificado.

El elemento **<xsl:template>** se utiliza para construir las plantillas.

- Si no hay plantillas, el procesador simplemente recorre todos los nodos y extrae el texto contenido por cada nodo.
- Si hay una plantilla vacía, el procesador no genera ningún resultado en el documento final ni recorre los nodos hijos.
- En el caso más extremo, si la plantilla vacía se aplica al nodo raíz, el procesador no genera ningún resultado en el documento final ni recorre ningún nodo hijo.



XSL. Elementos básicos: Elemento <xsl:value-of>

El elemento <xsl:value-of> se utiliza para extraer el valor de un nodo seleccionado (el carácter punto "." hace referencia al propio elemento).

Para extraer los valores de los atributos, se utiliza @.

```
<?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="cd">
    <xsl:value-of select="titulo"/>
    <xsl:value-of select="artista"/>
  </xsl:template>
</xsl:stylesheet>
```



XSL. Elementos básicos: Elemento `<xsl:for-each>`

El elemento `<xsl:for-each>` le permite hacer un bucle en XSLT.

El elemento `<xsl:for-each>` XSL puede utilizarse para seleccionar todos los elementos XML de un conjunto de nodos especificado.

El valor de la selección de atributo es una expresión XPath.

```
<table border="1">
  <tr bgcolor="#9acd32">
    <th style="text-align:left"><b>Titulo</b></th>
    <th style="text-align:left"><b>Artista</b></th>
  </tr>
  <xsl:for-each select="catalogo/cd[precio>10]">
    <tr>
      <td>
        <xsl:value-of select="titulo"/>
      </td>
      <td>
        <xsl:value-of select="artista"/>
      </td>
    </tr>
  </xsl:for-each>
</table>
```



XSL. Elementos básicos: Elemento `<xsl:sort>`

El elemento `<xsl:sort>` se utiliza para ordenar la salida.

Para ordenar la salida, sólo tiene que añadir un elemento `<xsl:sort>` dentro del elemento `<xsl:for-each>` en el archivo XSLT.

```
<table border="1">
  <tr bgcolor="#9acd32">
    <th style="text-align:left"><b>Titulo</b></th>
    <th style="text-align:left"><b>Artista</b></th>
  </tr>
  <xsl:for-each select="catalogo/cd[precio>10]">
    <xsl:sort select="artista"/>
    <tr>
      <td>
        <xsl:value-of select="titulo"/>
      </td>
      <td>
        <xsl:value-of select="artista"/>
      </td>
    </tr>
  </xsl:for-each>
</table>
```



XSL. Operadores: Elemento `<xsl:if>`

El elemento `<xsl:if>` se utiliza para poner una prueba condicional contra el contenido del archivo XML.

Sintaxis:

```
<xsl:if test="expression">  
    ... alguna salida si la expresión es verdadera...  
</xsl:if>
```

El elemento `<xsl:if>` se coloca dentro del elemento `<xsl:for-each>` en el archivo XSLT



XSL. Operadores: Elemento `<xsl:choose>`

El elemento `<xsl:choose>` se utiliza en conjunción con `<xsl:when>` y `<xsl:otherwise>` para expresar múltiples pruebas condicionales. Sintaxis:

```
<xsl:choose>  
  <xsl:when test="expression">  
    ... alguna salida...  
  </xsl:when>  
  <xsl:otherwise>  
    ... alguna salida....  
  </xsl:otherwise>  
</xsl:choose>
```



XSL. Operadores: Elemento `<xsl:choose>`

El elemento `<xsl:choose>` se utiliza en conjunción con `<xsl:when>` y `<xsl:otherwise>` para expresar múltiples pruebas condicionales. Sintaxis:

```
<xsl:choose>  
  <xsl:when test="expression">  
    ... alguna salida...  
  </xsl:when>  
  <xsl:otherwise>  
    ... alguna salida....  
  </xsl:otherwise>  
</xsl:choose>
```



XSL. Operadores: Elemento `<xsl:text>`

El elemento `<xsl:text>` permite generar texto que no se puede generar simplemente añadiéndolo (saltos de líneas y espacios en blanco, por ejemplo).

```
<h2>Mi colección de CDs</h2>
  <xsl:for-each select="catálogo/cd[precio>10]">
    <p><xsl:value-of select="título"/>
      <xsl:text> | </xsl:text>
      <xsl:value-of select="artista"/>
    </p>
  </xsl:for-each>
```

XSL. Operadores: Elemento `<xsl:attribute>`

El elemento `<xsl:attribute>` permite generar un atributo y su valor. Se utiliza cuando el valor del atributo se obtiene a su vez de algún nodo.

Por ejemplo, a partir del siguiente documento XML, se quiere generar la etiqueta ``. en la que el valor del atributo `src` sea el contenido de la etiqueta ``.

```
<img>  
  <xsl:attribute name="src">  
    <xsl:value-of select="img" />  
  </xsl:attribute>
```




XSL. Elemento `<xsl:apply-template>`

La instrucción `<xsl:apply-templates>` hace que se apliquen a los subelementos las reglas que les sean aplicables.