

<https://www.mclibre.org/consultar/xml/lecciones/xml-xslt.html>

Etiquetas de uso en XSL:

- `<?xml-stylesheet type="text/xsl" href="ejemplo.xsl"?>`
Se usa para enlazar un XML con un XSL.
- `<xsl:template match="/__"> </xsl:template>`
Creamos una plantilla para el nodo que elijamos, solo se puede usar una plantilla a la vez, a no ser que lo especifiquemos con `<xsl:apply-templates select="nombre_de_plantilla_hijo" />` dentro del template padre.
Ej:
`<xsl:template match="//producto[@cod='LACT012']">`
 `<xsl:text> Atopado!! </xsl:text>`
 `</xsl:template>`
- `<xsl:apply-templates select="__" /> // <xsl:apply-templates />`
En caso de que haya varias plantillas se tendria que usar esta etiqueta con el nombre de la ruta.
Ej:
`<xsl:template match="/equipos">`
 `<xsl:element name="discos">`
 `<xsl:apply-templates select="/equipos/máquina/hardware/disco"/>`
 `</xsl:element>`
 `</xsl:template>`

 `<xsl:template match="/equipos/máquina/hardware/disco">`
 `<xsl:element name="disco">`
 `...contenido, elements, attribute etc...`
 `</xsl:element>`
 `</xsl:template>`
- `<xsl:output/>`
Defines las características del documento de salida.
Ej:
`<xsl:output`
 `method="xml|html|text"`
 `indent="yes|no"`
 `version="string"`
 `encoding="string"`
 `omit-xml-declaration="yes|no"`
 `standalone="yes|no"`
 `doctype-public="string"`
 `doctype-system="string"`
 `cdata-section-elements="namelist"`
 `media-type="string"`
 `/>`

- `<xsl:strip-space elements="*" />`
Quita los espacios y mete sangría a la hora de transformar el XML, mejora la legibilidad (en notepad++ no funciona, usar el de "text
").
- `<xsl:text> </xsl:text>`
Escribes texto sin formato.
Para hacer un salto de línea: `<xsl:text>
 </xsl:text>`
- `<xsl:element name="___"> </xsl:element>`
Sirve para crear un elemento nuevo (importante cerrar la etiqueta despues de añadir el contenido dentro del elemento).
Ej: `<xsl:element name="{name()}">`
- `<xsl:attribute name="___"> </xsl:attribute>`
Se usa dentro de la etiqueta `<xsl:element>` para añadirle un atributo a este.
- `<xsl:attribute-set name="___"> <xsl:attribute-set>`
Se usa para guardar una configuracion de atributos, ej: dentro puede ir un attribute name, y luego dentro de este un value of.
Ej:
`<xsl:attribute-set name="atr_módulo">`
 `<xsl:attribute name="nome">`
 `<xsl:value-of select="."/>`
 `</xsl:attribute>`
 `<xsl:attribute name="sesións_anuais">`
 `<xsl:value-of select="@horas * 1.2"/>`
 `</xsl:attribute>`
`</xsl:attribute-set>`

Ej (uso): `<xsl:element name="___" use-attribute-sets="___">`
- `<xsl:value-of select="___"/>`
Se usa para obtener el nombre del elemento o del atributo y utilizarlo en la salida.
Ej:
`<xsl:attribute name="sesións_anuais">`
 `<xsl:value-of select="ciclo/módulo/@horas * 1.2"/>`
 `<xsl:value-of select="."/>` <!--Para mostrar el contenido del mismo-->
`</xsl:attribute>`
- `<xsl:comment> ___</xsl:comment>`
Añadir un comentario en el documento de salida

Funciones para XSL:

- `Name()`
Para copiar nombre de un elemento

- Text()

Se obtiene el texto dentro de la etiqueta.

Ej:

```
<xsl:template match="/módulo/profesor">
  <xsl:element name="{name()}">
    <xsl:attribute name="nome">
      <xsl:value-of select="text()" />
    </xsl:attribute>
  </xsl:element>
</xsl:template>
```

- Count(//ruta_a_contar)

Cuenta el numero de etiquetas que existe en el documento.

Ej:

```
<xsl:element name="{cliente/@cod}">
  <xsl:attribute name="num_produtos">
    <xsl:value-of select="count(produtos/produto)" />
  </xsl:attribute>
</xsl:element>
```

- substring(@cod, 1, 3) <!-- Ejemplo -->

Muestra del elemento o atributo que seleccionamos, las posiciones que le indicamos.

Ej:

```
<xsl:template match="/venda/produtos/produto">
  <xsl:text>Produto da familia </xsl:text>
  <xsl:value-of select="substring(@cod,1,4)" />
  <xsl:text> con código </xsl:text>
  <xsl:value-of select="substring(@cod,5,7)" />
  <xsl:text> para o cliente de código </xsl:text>
  <xsl:value-of select="../../cliente/@cod" />
</xsl:template>
```