



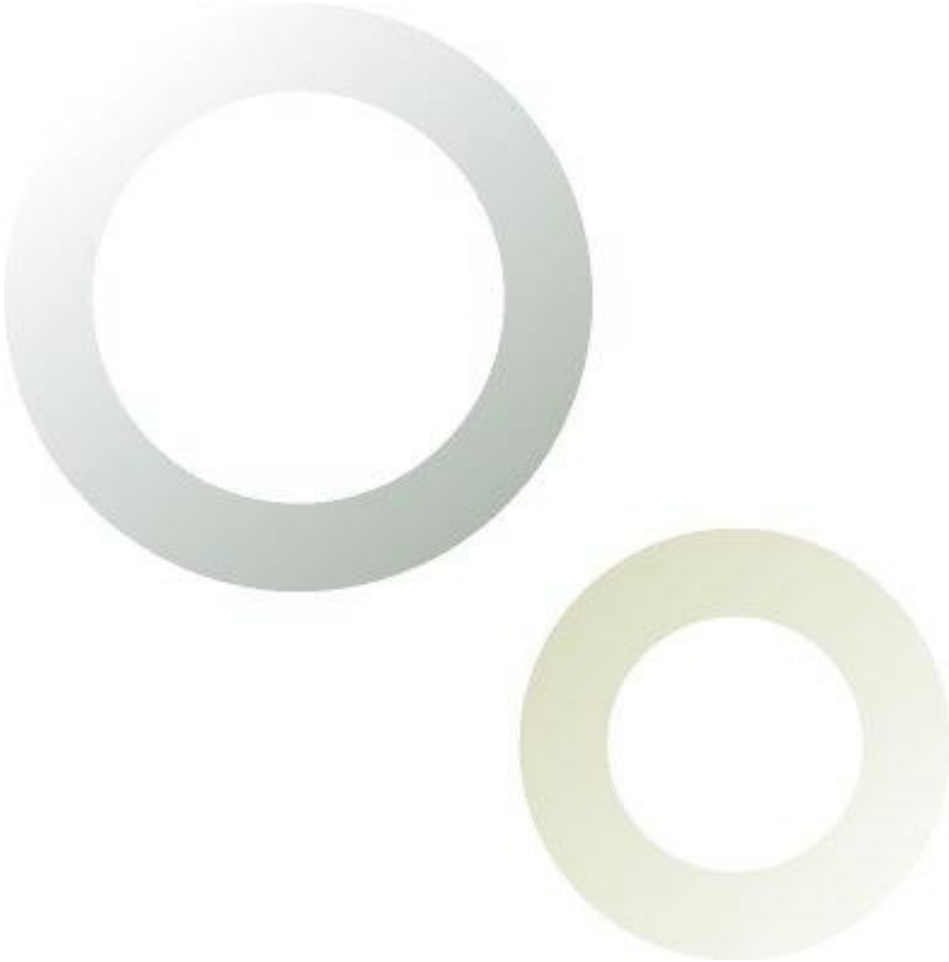
LENGUAJES DE MARCAS

UD7. Transformaciones con XSLT - Parte 2

Desarrollo de Aplicaciones Web

Profesores: Diana Bautista / Óscar Villar

Índice de contenido

- 
1. Repaso de lo selectores vistos
 2. Elemento for-each
 3. Elemento sort
 4. Elemento if
 5. Elemento choose
 6. Elemento attribute

1. Repaso de los selectores

Repaso de los selectores

El elemento <xsl:template>

- El elemento <xsl:template> se utiliza para construir las plantillas.
- El atributo match se utiliza para asociar una plantilla con un elemento XML.
- El atributo match también se puede utilizar para definir una plantilla para todo el documento XML.
- El valor del atributo match es una expresión XPath (por ejemplo match="/" define todo el documento) .

```
<?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>
  <cd>
    <title>Hide your heart</title>
    <artist>Bonnie Tyler</artist>
    <country>UK</country>
    <company>CBS Records</company>
    <price>9.90</price>
    <year>1988</year>
  </cd>
</catalog>
```

XML

```
<?xml version="1.0"?>

<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>Title</th>
            <th>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>
```

XSLT

Repaso de los selectores

El elemento <xsl:template>

- El elemento <xsl:template> se utiliza para construir las plantillas.
- El atributo match se utiliza para asociar una plantilla con un elemento XML.
- El atributo match también se puede utilizar para definir una plantilla para todo el documento XML.
- El valor del atributo match es una expresión XPath (por ejemplo match="/" define todo el documento) .

My CD Collection

Title	Artist
Empire Burlesque	Bob Dylan
Hide your heart	Bonnie Tyler

Resultado

Repaso de los selectores

El elemento **<xsl:value-of>** se utiliza para extraer el valor de un nodo seleccionado

El elemento **<xsl:apply-templates>** se aplica una plantilla al elemento actual o para nodos hijo del elemento actual.

```
<?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>
  <cd>
    <title>Hide your heart</title>
    <artist>Bonnie Tyler</artist>
    <country>UK</country>
    <company>CBS Records</company>
    <price>9.90</price>
    <year>1988</year>
  </cd>
</catalog>
```

XML

```
<?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>
      <xsl:apply-templates/>
    </body>
  </html>
</xsl:template>
```

```
<xsl:template match="cd">
  <p>
    <xsl:apply-templates select="title"/>
    <xsl:apply-templates select="artist"/>
  </p>
</xsl:template>
```

```
<xsl:template match="title">
  Title: <span style="color:#ff0000">
    <xsl:value-of select="."/></span>
  <br />
</xsl:template>
```

```
<xsl:template match="artist">
  Artist: <span style="color:#00ff00">
    <xsl:value-of select="."/></span>
  <br />
</xsl:template>
```

```
</xsl:stylesheet>
```

XSLT

Repaso de los selectores

El elemento `<xsl:value-of>`
se utiliza para extraer el valor
de un nodo seleccionado

**El elemento
`<xsl:apply-templates>`** se
aplica una plantilla al
elemento actual o para nodos
hijo del elemento actual.

My CD Collection

Title: **Empire Burlesque**
Artist: **Bob Dylan**

Title: **Hide your heart**
Artist: **Bonnie Tyler**

Resultado

2. Elemento for-each

Elementos for-each

El elemento <xsl:for-each>

- Permite hacer un bucle en XSLT.
- Puede ser usado para seleccionar cada elemento XML de un conjunto de nodos especificado

```
<?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>
  <cd>
    <title>Hide your heart</title>
    <artist>Bonnie Tyler</artist>
    <country>UK</country>
    <company>CBS Records</company>
    <price>9.90</price>
    <year>1988</year>
  </cd>
</catalog>
```

XML

```
<?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>Title</th>
        <th>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>
```

XSLT

Elementos for-each

El elemento <xsl:for-each>

- Permite hacer un bucle en XSLT.
- Puede ser usado para seleccionar cada elemento XML de un conjunto de nodos especificado

My CD Collection

Title	Artist
Empire Burlesque	Bob Dylan
Hide your heart	Bonnie Tyler

Resultado

NOTA: usando apply-templates podemos obtener el mismo resultado sin usar bucles

Elementos for-each

Filtrado de salida

- También podemos filtrar la salida del archivo XML mediante la adición de un criterio para el atributo select en el elemento <xsl:for-each>.
- Por ejemplo: **<xsl:for-each select="catalog/cd[artist='Bob Dylan']">**
- De esta forma de todos los discos que hay en el catálogo sólo nos presentará los del artista Bob Dylan
- **Operadores de filtro posibles:**
 - = (Igual)
 - != (notequal)
 - < (menor que)
 - > (mayor que)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:template match="/">
    <html>
      <body>
        <h2>My CD Collection</h2>
        <table border="1">
          <tr bgcolor="#9acd32">
            <th>Title</th>
            <th>Artist</th>
          </tr>
          <xsl:for-each select="catalog/cd[artist='Bob Dylan']">
            <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>
```

My CD Collection

Title	Artist
Empire Burlesque	Bob Dylan

XSLT

3. Elemento sort

Elemento sort

Elemento <xsl:sort>

- Se utiliza para ordenar la salida
- **Parámetros de sort:**
 - lang: "language-code"
 - data-type: "text | number | QName".
 - order: "ascending | descending".
 - case-order: "upper-first | lower-first"

ORDEN POR DEFECTO ASCENDENTE

```
<?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>
  <cd>
    <title>Hide your heart</title>
    <artist>Bonnie Tyler</artist>
    <country>UK</country>
    <company>CBS Records</company>
    <price>9.90</price>
    <year>1988</year>
  </cd>
</catalog>
```

XML

```
<?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>Title</th>
          <th>Artist</th>
        </tr>
        <xsl:for-each select="catalog/cd">
          <xsl:sort select="artist"/>
          <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>
```

XSLT

Elemento sort

Elemento <xsl:sort>

- Se utiliza para ordenar la salida
- **Parámetros de sort:**
 - lang: "language-code"
 - data-type: "text | number | qname".
 - order: "ascending | descending".
 - case-order: "upper-first | lower-first"

My CD Collection

Title	Artist
Empire Burlesque	Bob Dylan
Hide your heart	Bonnie Tyler

Resultado

ORDEN POR DEFECTO ASCENDENTE

Elemento sort

Elemento <xsl:sort>

- Se utiliza para ordenar la salida
- **Parámetros de sort:**
 - lang: "language-code"
 - data-type: "text | number | QName".
 - order: "ascending | descending".
 - case-order: "upper-first | lower-first"

ORDEN DESCENDENTE

```
<?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>
  <cd>
    <title>Hide your heart</title>
    <artist>Bonnie Tyler</artist>
    <country>UK</country>
    <company>CBS Records</company>
    <price>9.90</price>
    <year>1988</year>
  </cd>
</catalog>
```

XML

```
<?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>Title</th>
          <th>Artist</th>
        </tr>
        <xsl:for-each select="catalog/cd">
          <xsl:sort select="artist" order="descending"/>
          <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>
```

XSLT

Elemento sort

Elemento <xsl:sort>

- Se utiliza para ordenar la salida
- **Parámetros de sort:**
 - lang: "language-code"
 - data-type: "text | number | QName".
 - order: "ascending | descending".
 - case-order: "upper-first | lower-first"

My CD Collection

Title	Artist
Hide your heart	Bonnie Tyler
Empire Burlesque	Bob Dylan

Resultado

ORDEN DESCENDENTE

4. El elemento if

Elemento if

Elemento <xsl:if>

- Se utiliza como condicional a la hora de obtener contenido del XML
- Para agregar una prueba condicional hay que añadir el **<xsl:if>** dentro del bucle **<xsl:for-each>**
- El atributo **test** admite:
 - Para mayor que: >
 - test="price >10"
 - Para menor que: <
 - test="price <10"
 - Para igual que: ='cadena';
 - test="price='10'"

```
<xsl:if test="expression">
  ...some output if the expression is true...
</xsl:if>
```

```
<?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>
  <cd>
    <title>Hide your heart</title>
    <artist>Bonnie Tyler</artist>
    <country>UK</country>
    <company>CBS Records</company>
    <price>9.90</price>
    <year>1988</year>
  </cd>
</catalog>
```

XML

```
<?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>Title</th>
            <th>Artist</th>
            <th>Price</th>
          </tr>
          <xsl:for-each select="catalog/cd">
            <xsl:if test="price &gt; 10">
              <tr>
                <td><xsl:value-of select="title"/></td>
                <td><xsl:value-of select="artist"/></td>
                <td><xsl:value-of select="price"/></td>
              </tr>
            </xsl:if>
          </xsl:for-each>
        </table>
      </body>
    </html>
  </xsl:template>

</xsl:stylesheet>
```

XSLT

Elemento if

Elemento <xsl:if>

- Se utiliza como condicional a la hora de obtener contenido del XML
- Para agregar una prueba condicional hay que añadir el **<xsl:if>** dentro del bucle **<xsl:for-each>**
- **El atributo test admite:**
 - Para mayor que: >
■ test="price >10"
 - Para menor que: <
■ test="price <10"
 - Para igual que: ='cadena';
■ test="price='10'"

```
<xsl:if test="expression">  
  ...some output if the expression is true...  
</xsl:if>
```

My CD Collection

Title	Artist	Price
Empire Burlesque	Bob Dylan	10.90

Resultado

El código anterior solo generará el título y los elementos del artista de los CD que tengan un precio superior a 10.

5. El elemento choose

Elemento choose

Elemento <xsl:choose>

- Se utiliza en conjunción con **<xsl:when>** y **<xsl:otherwise>** para expresar múltiples pruebas condicionales.

```
<xsl:choose>
  <xsl:when test="expression">
    ... some output ...
  </xsl:when>
  <xsl:otherwise>
    ... some output ....
  </xsl:otherwise>
</xsl:choose>
```

```
<?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>
  <cd>
    <title>Hide your heart</title>
    <artist>Bonnie Tyler</artist>
    <country>UK</country>
    <company>CBS Records</company>
    <price>9.90</price>
    <year>1988</year>
  </cd>
</catalog>
```

XML

```
<?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>Title</th>
        <th>Artist</th>
      </tr>
      <xsl:for-each select="catalog/cd">
        <tr>
          <td><xsl:value-of select="title"/></td>
          <xsl:choose>
            <xsl:when test="price > 10">
              <td bgcolor="#ff00ff">
                <xsl:value-of select="artist"/></td>
            </xsl:when>
            <xsl:otherwise>
              <td><xsl:value-of select="artist"/></td>
            </xsl:otherwise>
          </xsl:choose>
        </tr>
      </xsl:for-each>
    </table>
  </body>
</html>
</xsl:template>

</xsl:stylesheet>
```

XSLT

Elemento choose

Elemento <xsl:if>

- Se utiliza en conjunción con **<xsl:when>** y **<xsl:otherwise>** para expresar múltiples pruebas condicionales.

```
<xsl:choose>
  <xsl:when test="expression">
    ... some output ...
  </xsl:when>
  <xsl:otherwise>
    ... some output ....
  </xsl:otherwise>
</xsl:choose>
```

My CD Collection

Title	Artist
Empire Burlesque	Bob Dylan
Hide your heart	Bonnie Tyler

Resultado

El código anterior agrega un color de fondo rosa a la columna "Artista" CUANDO el precio del CD sea superior a 10

6. El elemento attribute

Elemento attribute

Elemento <xsl:attribute>

- Permite añadir atributos a una etiqueta

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

```
<?xml version="1.0" encoding="UTF-8"?>  
<museos>  
  <museo>  
    <nombre>Museo del Prado</nombre>  
    <ciudad>Madrid</ciudad>  
    <pais>España</pais>  
  </museo>  
  <museo>  
    <nombre>British Museum</nombre>  
    <ciudad>Londres</ciudad>  
    <pais>Reino Unido</pais>  
  </museo>  
  <museo>  
    <nombre>National Gallery</nombre>  
    <ciudad>Londres</ciudad>  
    <pais>Reino Unido</pais>  
  </museo>  
</museos>
```

XML

```
<?xml version="1.0" encoding="UTF-8"?>  
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">  
  <xsl:template match="/">  
    <museos>  
      <xsl:apply-templates/>  
    </museos>  
  </xsl:template>  
  <xsl:template match="museo">  
    <museo>  
      <xsl:attribute name="nombre">  
        <xsl:value-of select="nombre"/>  
      </xsl:attribute>  
      <xsl:attribute name="ciudad">  
        <xsl:value-of select="ciudad"/>  
      </xsl:attribute>  
      <xsl:attribute name="pais">  
        <xsl:value-of select="pais"/>  
      </xsl:attribute>  
    </museo>  
  </xsl:template>  
</xsl:stylesheet>
```

XSLT

Elemento attribute

Elemento <xsl:attribute>

- Permite añadir atributos a una etiqueta

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

```
<?xml version="1.0" encoding="UTF-8"?>  
<museos>  
  <museo nombre="Museo del Prado" ciudad="Madrid" pais="España"/>  
  <museo nombre="British Museum" ciudad="Londres" pais="Reino Unido"/>  
  <museo nombre="National Gallery" ciudad="Londres" pais="Reino Unido"/>  
</museos>
```

Resultado

Enlaces de interés

1. [W3 Schools - DTD Tutorial](#)
2. [W3Schools XML Tutorial](#)
3. [XML Copy Editor](#)

Bibliografía

Prácticamente todo contenido del documento ha sido extraído de https://www.w3schools.com/xml/xsl_intro.asp