

## M04 - UF2 - NF2 (Conversió i formatació de documents)

### Continguts:

1. XSL - Introducció.
2. XSLT.
  - 2.1. For-each (bucles)
  - 2.2. Sort (ordenació)
  - 2.3. IF (condicional)
  - 2.4. Funcions
    - 2.4.1. count()
    - 2.4.2. sum()
    - 2.4.3. Mitjana
    - 2.4.4. Mínim i màxim
  - 2.5. Creació d'elements i atributs
3. XPath.

## 1. XSL - Introducció

XSL fa referència a les sigles eXtensible Stylesheet Language, que ve a representar els fulls d'estil de documents xml.

XSL és una recomanació de l'organisme W3C i està recolzada per les següents tecnologies/llenguatges:

- XSLT: Llenguatge que descriu com transformar un document xml. Per exemple, convertir-lo a XHTML.
- Xpath: Llenguatge que ens permet accedir a diferents elements dins d'un document xml per realitzar accions amb ell. És també el llenguatge base per a realitzar consultes Xquery.
- XSL-FO: Conjunt d'objectes i propietats per donar format al document transformat prèviament amb XSLT. Habitualment s'utilitza per a la transformació en documents pdf.

## 2. XSLT

Per poder transformar un document XML en un altre, com per exemple HTML, s'han de seguir unes regles (Rules) que s'aplicaran als elements del document.

El primer que hem de definir correctament, és la declaració del document XSL. Una declaració correcta del document XSL es pot fer de dues maneres:

```
<?xml version="1.0" encoding="UTF-8" ?>  
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

o també:

```
<?xml version="1.0" encoding="UTF-8" ?>  
<xsl:transform version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

El següent pas per poder realitzar la transformació és aplicar les regles anomenades “Templates” corresponents als nodes o elements del document XML.

La sintaxi d'una regla és **<xsl:template>** seguit de l'element al que es fa referència. Per poder indicar a quin element es fa referència, afegim un atribut a la regla de la següent forma **<xsl:template match="agenda">**, que aplica la regla a l'element “agenda” del document XML. També es pot indicar per exemple “/” que indicaria l'element arrel del document. Aquestes referències formen part del llenguatge Xpath.

Per poder realitzar la transformació, hem de poder extreure el contingut dels elements del document XML i aplicar-los al nou document transformat. Disposem de l'element **<value-of>** per extreure aquesta informació, per exemple:

```
<xsl:value-of select="agenda/nom"/>
```

Exemple d'aplicació d'un full d'estil XSL sobre un XML:

Document XML sobre el que aplicarem el fitxer XSL:

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

Document XSL que aplicarem sobre el 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>
        <tr>
            <td><xsl:value-of select="catalog/cd/title"/></td>
            <td><xsl:value-of select="catalog/cd/artist"/></td>
        </tr>
    </table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

ALERTA!!!: LES COMILLES HAN DE SER RECTES (") I NO LES QUE TENEN FORMA CURSIVA.

Ara si obrim el fitxer cdcatalog.xml amb un navegador web, el resultat seria:

## My CD Collection

Title	Artist
Empire Burlesque	Bob Dylan

### 2.1. For-each (bucles)

Si proveu a afegir més d'un CD al document XML, se n'adonareu que no els mostra, tan sols mostra el primer. Per aconseguir que el procés s'aplique a TOTS els cd's del document hem d'utilitzar l'element **<xsl:for-each>** que realitza un bucle fins que no troba més elements.

```

<?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>
```

Podem afinar un poc més afegint un criteri de filtre, per seleccionar aquells de tots que ens interessin:

```
<xsl:for-each select="catalog/cd[artist='Bob Dylan']">
```

L'exemple anterior selecciona tots els cd's de "catalog" que contenen un element "artist" amb el valor "Bob Dylan".

## 2.2. Sort (ordenació)

També podem aplicar un criteri d'ordenació dels elements que es mostren amb la següent sentència:

```
<xsl:sort select="year"/>
```

que mostra els elements seleccionats, ordenats per l'element "year" del document xml. Ficariem la sentència anterior dins de la sentència del for-each.

```
<?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>Year</th>
        </tr>
        <xsl:for-each select="catalog/cd[artist='Bob Dylan']">
          <xsl:sort select="year"/>
          <tr>
            <td><xsl:value-of select="title"/></td>
            <td><xsl:value-of select="artist"/></td>
```

```

        <td><xsl:value-of select="year"/></td>
    </tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

## 2.3. IF (condicional)

Encara que dins de la sentència for-each es poden posar filtres, també hi ha la sentència "if" per a fer-ho:

```

<?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>Year</th>
                </tr>
                <xsl:for-each select="catalog/cd[artist='Bob Dylan']">
                    <xsl:sort select="year"/>
                    <xsl:if test="year >= 1985">
                        <tr>
                            <td><xsl:value-of select="title"/></td>
                            <td><xsl:value-of select="artist"/></td>
                            <td><xsl:value-of select="year"/></td>
                        </tr>
                    </xsl:if>
                </xsl:for-each>
            </table>
        </body>
    </html>
</xsl:template>
</xsl:stylesheet>

```

En algunes ocasions, pot ser haurem de substituir el signe "<" o ">" per l'equivalent "&lt;" o "&gt;," respectivament.

## 2.3. Choose + when-otherwise

Per personalitzar encara més la forma com es mostra la informació, utilitzem CHOOSE juntament amb WHEN/OTHERWISE, de manera que definirem una vista en funció d'una condició i un altra en cas contrari. Per entendre millor el que s'intenta explicar, realitzarem un exemple que mostre els cd's de l'any 1985 amb fons "cyan" i la resta amb fons "groc"

```
<?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>Year</th>
        </tr>
        <xsl:for-each select="catalog/cd[artist='Bob Dylan']">
          <xsl:sort select="year"/>
          <xsl:if test="year >= 1985">
            <tr>
              <xsl:choose>
                <xsl:when test="year=1985">
                  <td bgcolor="cyan"><xsl:value-of select="title"/></td>
                  <td bgcolor="cyan"><xsl:value-of select="artist"/></td>
                  <td bgcolor="cyan"><xsl:value-of select="year"/></td>
                </xsl:when>
                <xsl:otherwise>
                  <td bgcolor="yellow"><xsl:value-of select="title"/></td>
                  <td bgcolor="yellow"><xsl:value-of select="artist"/></td>
                  <td bgcolor="yellow"><xsl:value-of select="year"/></td>
                </xsl:otherwise>
              </xsl:choose>
            </tr>
          </xsl:if>
        </xsl:for-each>
      </table>
    </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```

En la web <http://www.w3schools.com/xsl/default.asp> hi ha un curs de XSL on podreu trobar més informació.

## 2.4. Funcions

### 2.4.1. count()

Exemple:

```
<h4><xsl:value-of select="count(//Producto)" /></h4>
```

### 2.4.2. sum()

Exemple:

```
<xsl:value-of select="sum(//Precio)" />
```

### 2.4.3. Mitjana

No es pot fer directament. Exemple:

```
<xsl:value-of select="sum(//Precio) div count(//Producto)" />
```

### 2.4.4. Mínim i màxim

Exemple:

```
<!--  
| assign variable based on picked the first item in  
| the numerically-sorted-descending list of items.  
+>  
<xsl:variable name="the_max">  
  <xsl:for-each select="//list/item">  
    <xsl:sort data-type="number" order="descending"/>  
    <xsl:if test="position()=1"><xsl:value-of select="."/></xsl:if>  
  </xsl:for-each>  
</xsl:variable>
```

for the min, you just reverse the sort order:

```
<!--  
| assign variable based on picked the first item in  
| the numerically-sorted-descending list of items.
```

```

+>
<xsl:variable name="the_min">
  <xsl:for-each select="/list/item">
    <xsl:sort data-type="number" order="ascending"/>
    <xsl:if test="position()=1"><xsl:value-of select="."/></xsl:if>
  </xsl:for-each>
</xsl:variable>

```

També es pot fer amb la llibreria math (<http://www.exslt.org/math/>)

Exemple:

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="numeros.xsl"?>
<numeros>
  <numero>3</numero>
  <numero>5</numero>
  <numero>10</numero>
  <numero>1</numero>
</numeros>

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:math="http://exslt.org/math">
  <xsl:template match="/numeros">
    <xsl:value-of select="math:max(numero)"/>
  </xsl:template>
</xsl:stylesheet>

```

## 2.5. Creació d'elements i atributs

El següent codi equivaldria a <a href="adreça">anar a Google</a>

```

<!--
| Opció 1
+>
<xsl:element name="a">
  <xsl:attribute name="href">
    <xsl:value-of select="adreça" />
  </xsl:attribute>
  Anar a Google
</xsl:element>

```



<!--

| Opció 2

+>

<a>

<xsl:attribute name="href">

<xsl:value-of select="adreça" />

</xsl:attribute>

Anar a Google

</a>

<!--

| Opció 3

+>

<a href="{adreça}">Anar a Google </a>

### 3. Xpath

Si ens fixem en els exemples anteriors, a l'hora de seleccionar un element de entre els que tenim a un document XML hem utilitzat una sintaxi particular (per exemple `catalog/cd[artist='Bob Dylan']`). Aquestes expressions que ens han ajudat a seleccionar elements d'un document XML formen part del llenguatge Xpath.

XPath és una recomanació de W3C i l'utilitzarem en XSLT i XQuery entre altres.

Xpath estableix una ordenació de tipus jeràrquic de manera que els elements d'un document XML es relacionen entre ells a l'estil d'un arbre genealògic, es a dir, els elements poden tenir avis, pares, fills, etc.

Exemple:

<?xml version="1.0" encoding="UTF-8"?>

<catalog>

<cd code="001">

<title>Empire Burlesque</title>

<artist>Bob Dylan</artist>

<price>10.90</price>

<year>1985</year>

</cd>

<cd code="002">

<title>Bringing It All Back Home</title>

<artist>Bob Dylan</artist>

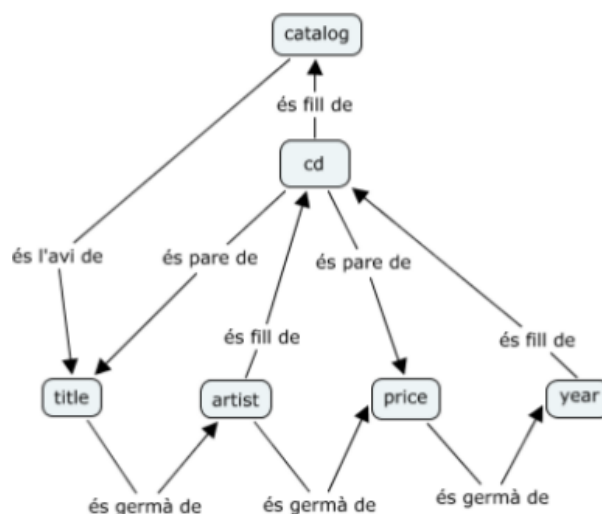
<price>12</price>

<year>1965</year>

</cd>

<cd code="003">

<title>Blonde on Blonde</title>



```

    <artist>Bob Dylan</artist>
    <price>14.50</price>
    <year>1966</year>
  </cd>
  <cd code="004">
    <title>Modern Times</title>
    <artist>Bob Dylan</artist>
    <price>14.50</price>
    <year>2006</year>
  </cd>
</catalog>

```

Taula de selecció:

| Expressió           | Descripció   | Exemple  |
|---------------------|--|--|
| <b>Nom_del_node</b> | Selecciona els nodes fills de "Nom_del_node"       | <b>catalog</b>   |
| <b>/</b>            | Arrel del document                                 | <b>/</b><br><b>/catalog</b> (tots les elements descendents, que són a dins de, de catalog)<br><b>cd/title</b> (el primer títol descendent de cd) |
| <b>//</b>           | Selecciona elements en qualsevol part del document | <b>//cd</b> (tots els elements cd)<br><b>cd//title</b> (tots els títols descendents, que són a dins de, de cd)                                   |
| <b>.</b>            | Selecciona el node actual                          |  |
| <b>..</b>           | Selecciona el pare de l'element actual             |  |
| <b>@</b>            | Selecciona els atributs.                           | <b>catalog/cd/@code</b> (l'atribut "code" del primer cd de catalog)  |

En la web [http://www.w3schools.com/xsl/xpath\\_intro.asp](http://www.w3schools.com/xsl/xpath_intro.asp) hi ha un curs de XPath on podreu trobar més informació.