

XSLT

XSL

XSL (llenguatge de fulls d'estil extensible o *extensible stylesheet language*) és un llenguatge XML per a expressar fulls d'estil (com s'ha de representar un llenguatge XML concret). Consisteix en tres components principals: XSLT (*XSL transformations*), XPath i XSL-FO (*XSL-formatting objects*).

XSL

A diferència d'HTML, on el significat de cada etiqueta està clarament definit (salt de paràgraf, salt de línia, capçalera, negreta) i on resulta simple assignar estils (fonts, mides, colors, etc.) a aquestes etiquetes, en XML les etiquetes no estan definides, sinó que l'usuari les pot definir. En XML l'etiqueta `taula` pot representar una taula d'HTML en un cas i les mesures d'un tauler de fusta en l'altre, amb la qual cosa els navegadors no saben com representar les etiquetes. Consegüentment, el llenguatge de fulls d'estil per a la presentació ha de descriure com mostrar un document XML més clarament.

XSL

El llenguatge d'estils d'XML, XSL, consisteix en tres components principals:

- XSLT, un llenguatge de transformació de documents.
- XPath, un llenguatge per a referenciar parts de documents XML.
- XSL-FO, un llenguatge de format de documents XML.

XSL

Amb aquests tres components XSL és capaç de fer el següent:

- Transformar XML en XML, com per exemple XHTML o WML.
- Filtrar i/o ordenar dades XML.
- Definir parts d'un document XML.
- Formatar un document XML sobre la base dels valors de les dades emmagatzemades.
- Extreure les dades XML a XSL-FO que podem usar per a generar fitxers com a PDF.

XSL i XSLT

XSL és un llenguatge estàndard del W3C. Va ser estandarditzat en dues etapes: la primera, el novembre del 1999, incloïa XSLT i XPath, mentre que la segona, completada l'octubre del 2000, incloïa l'XSL-FO.

XSLT és la part de l'estàndard XML que serveix per a transformar documents XML en altres documents XML (com per exemple XHTML, WML, etc.).

XSLT

Normalment XSLT ho fa transformant cada element XML en un altre element XML. XSLT també pot afegir altres elements XML a la sortida, o bé pot eliminar elements. A més a més pot reordenar o recol·locar elements i fer comprovacions i decisions sobre quins elements mostrar.

XSLT

Durant la transformació, XSLT utilitza XPath per a especificar o referenciar parts del document que compleixen un o més patrons definits. Quan troba una coincidència de patrons, XSLT transformarà la part coincident del document origen en el document destinació. Les parts no coincidents no es transformen, sinó que queden al document destinació sense cap canvi.

Una simple transformació

Com gairebé tots els estàndards del W3C, XSLT és en si mateix un llenguatge XML i ha de començar amb un element arrel. Aquest element arrel és del tipus `xsl:stylesheet` o `xsl:transform` (ambdues etiquetes són totalment equivalents). La forma correcta d'ús és:

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

Aquesta declaració identifica l'espai de noms recomanat pel W3C. Si usem aquest espai de noms, hem d'afegir també l'atribut `version` amb el valor **1.0**.

El fitxer XML d'exemple que transformarem és el següent:

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

```
<expedient alumne="Linus Torvalds">
```

```
  <assignatura id="1">
```

```
    <nom>
```

```
    Programació bàsica
```

```
  </nom>
```

```
  <nota>
```

```
  Notable
```

```
  </nota>
```

```
</assignatura>
```

```
<assignatura id="2">
```

```
  <nom>
```

```
  Sistemes operatius
```

```
  </nom>
```

```
  <nota>
```

```
  Excel·lent
```

```
  </nota>
```

```
</assignatura>
```

```
<assignatura>
```

```
  ...
```

Una transformació simple

Aquest document XML correspon a un expedient d'un estudiant d'una universitat qualsevol. Es tracta d'un document molt simple, però totalment vàlid per a les nostres necessitats.

El document XSL per a convertir aquest document XML en un altre XHTML és el següent:

Una transformació simple

Aquest document XML correspon a un expedient d'un estudiant d'una universitat qualsevol. Es tracta d'un document molt simple, però totalment vàlid per a les nostres necessitats.

El document XSL per a convertir aquest document XML en un altre XHTML és el següent:

```
<?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>Expedient académic</h2>
```

```
        <table border="1">
```

```
            <tr bgcolor="#9acd32">
```

```
                <th align="left">Assignatura</th>
```

```
                <th align="left">Nota</th>
```

```
            </tr>
```

```
            <xsl:for-each select="expedient/assignatura">
```

```
                <tr>
```

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

```
                    <td><xsl:value-of select="nota"/></td>
```

```
                </tr>
```

```
            </xsl:for-each>
```

```
        </table>
```

```
    </body>
```

```
    </html>
```

```
</xsl:template>
```

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

Si anomenem el document XSL `expedient.xsl`, podem vincular-lo amb el nostre XML afegint una referència al full d'estil al principi de l'XML, de la manera següent:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="expedient.xsl"?>
<expedient alumne="Linus Torvalds">
    <assignatura id="1">
        <nombre>
            Programació bàsica
        </nombre>
        <nota>
            Notable
        </nota>
    </assignatura>
    ...

```

Si disposem d'un navegador amb suport XSL (com Mozilla en versions superiors a la 1.2) per a obrir el document XML, el navegador utilitzarà el document XSL per a transformar-lo en XHTML.

L'element `xsl:template`

Un full d'estil d'XSL consisteix en una sèrie de plantilles (*templates*) de transformació. Cada element `xsl:template` conté les transformacions que XSL ha d'aplicar si el patró especificat en l'element coincideix amb el trobat al document XML.

Per a especificar l'element XML a què hem d'aplicar la plantilla utilitzarem l'atribut `match` (també podem aplicar la plantilla a tot el document XML, per a la qual cosa podem especificar `match="/"`). Els valors que podem assignar a l'atribut `match` són els especificats per l'estàndard XPath.

Per exemple, la transformació XSL següent torna un codi XHTML concret en processar el document amb l'expedient de l'alumne.


```
<?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>Expedient acadèmic</h2>
    <table border="1">
      <tr bgcolor="#9acd32">
        <th align="left">Assignatura</th>
        <th align="left">Nota</th>
      </tr>
    </table>
  </body>
</html>
</xsl:template>

</xsl:stylesheet>
```

L'element xsl: template

Com podem veure, si provem aquest document XSL, el resultat és tan sols una capçalera de pàgina. Si analitzem el document XSL, veurem que disposa d'una plantilla que s'aplica quan coincideix amb l'element arrel del document (`match="/"`) i que imprimeix al resultat el que hi ha a l'etiqueta.

L'element `value-of`

L'element `value-of` serveix per seleccionar i afegir a la sortida el valor de l'element XML seleccionat.

Per exemple, si afegim el codi següent al nostre exemple anterior:

```

<?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>Expedient acadèmic</h2>
    <table border="1">
      <tr bgcolor="#9acd32">
        <th align="left">Assignatura</th>
        <th align="left">Nota</th>
      </tr>
      <tr>
        <td><xsl:value-of
          select="expedient/assignatura/nom"/></td>
        <td><xsl:value-of
          select="expedient/assignatura/nota"/></td>
      </tr>
    </table>
  </body>
</html>
</xsl:template>
</xsl:stylesheet>

```

L'element value-of

Veurem en el resultat que apareix la primera nota de l'expedient. Això es deu al fet que les etiquetes `value-of` seleccionen el valor del primer element que compleix el patró especificat.

L'element `xsl:for-each`

L'element `xsl:for-each` d'XSL es pot utilitzar per a seleccionar cada un dels elements del document XML que pertanyin a un conjunt determinat.

Si a l'exemple anterior, on només apareixia la primera nota de l'expedient, hi afegim un `xsl:for-each` que faci el recorregut per tot l'expedient de la manera següent:

Obtindrem ara una llista de totes les notes de les assignatures.

```
<?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>Expedient Acadèmic</h2>
        <table border="1">
            <tr bgcolor="#9acd32">
                <th align="left">Assignatura</th>
                <th align="left">Nota</th>
            </tr>
            <xsl:for-each select="expedient/assignatura">
                <tr>
                    <td><xsl:value-of select="nom"/></td>
                    <td><xsl:value-of select="nota"/></td>
                </tr>
            </xsl:for-each>
        </table>
    </body>
    </html>
</xsl:template>

</xsl:stylesheet>
```

Ordenació: xsl: sort

Per a obtenir una sortida ordenada, simplement hem d'afegir un element `xsl:sort` a l'element `xsl:for-each` al nostre fitxer 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>Expedient Acadèmic</h2>
    <table border="1">
      <tr bgcolor="#9acd32">
        <th align="left">Assignatura</th>
        <th align="left">Nota</th>
      </tr>
      <xsl:for-each select="expedient/assignatura">
        <xsl:sort select="nom"/>
```


Ordenació: xsl: sort

```
<tr>
  <td><xsl:value-of select="nom"/></td>
  <td><xsl:value-of select="nota"/></td>
</tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>

</xsl:stylesheet>
```

L'atribut `select` ens permet indicar l'element pel qual farem l'ordenació; en aquest cas per nom d'assignatura.

Condicions en XSL

Disposem de dos elements XSL que ens permeten implementar condicions en les nostres transformacions. Es tracta d'`xsl:if` i `xsl:choose`.

Element xsl: if

L'element `xsl:if` ens permet aplicar una plantilla només en cas que la condició especificada es compleixi (sigui certa).

Un exemple de format d'`xsl:if` és el següent:

```
<xsl:if test="nota < 5">
```

```
..... només apareixerà amb nota més petita que 5 .....
```

```
</xsl:if>
```

Element xsl: if

Per exemple, podem modificar el codi anterior perquè només mostri les notes superiors a 5.

```
<?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>Expedient Acadèmic</h2>
    <table border="1">
      <tr bgcolor="#9acd32">
        <th align="left">Assignatura</th>
```

Element xsl: if

```
    <th align="left">Nota</th>
</tr>
  <xsl:for-each select="expedient/assignatura">
    <xsl:if test="nota > 5">
      <tr>
        <td><xsl:value-of select="nom"/></td>
        <td><xsl:value-of select="nota"/></td>
      </tr>
    </xsl:if>
  </xsl:for-each>
</table>
</body>
</html>
</xsl:template>

</xsl:stylesheet>
```

L'element `xsl:choose`

L'element `xsl:choose` (juntament amb `xsl:when` i `xsl:otherwise`) ens permet modelar tests condicionals múltiples. Això és, podem, en funció d'una condició múltiple (amb múltiples valors possibles), obtenir resultats diversos.

Un exemple de format d'`xsl:choose` és el següent:

L'element xsl:choose

```
<xsl:choose>
  <xsl:when test="nota &lt; 5">
    ... codi (suspès) ....
  </xsl:when>
  <xsl:when test="nota &lt; 9">
    ... codi (normal) ....
  </xsl:when>
  <xsl:otherwise>
    ... codi (excel·lent) ....
  </xsl:otherwise>
</xsl:choose>
```

Modifiquem l'exemple anterior perquè les notes inferiors a cinc apareguin en color vermell.

```
<?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>Expedient Acadèmic</h2>
    <table border="1">
      <tr bgcolor="#9acd32">
        <th align="left">Assignatura</th>
        <th align="left">Nota</th>
      </tr>
      <xsl:for-each select="expedient/assignatura">
        <tr>
          <td><xsl:value-of select="nom"/></td>
```



```
        <td>
          <xsl:choose>
            <xsl:when test="nota &lt; 5">
              <font color="#FF0000">
                <xsl:value-of select="nota"/>
              </font>
            </xsl:when>
            <xsl:otherwise>
              <xsl:value-of select="nota"/>
            </xsl:otherwise>
          </xsl:choose>
        </td>
      </tr>
    </xsl:if>
  </xsl:for-each>
</table>
</body>
</html>
</xsl:template>

</xsl:stylesheet>
```

L'element xsl: apply-templates

`apply-templates` aplica una plantilla a l'element actual o als elements **fill** de l'element actual. Amb l'atribut `select` podem processar només aquells elements fill que especifiquem i l'ordre en què s'han de processar.

Per exemple:

```
<?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>Expedient académic</h2>
    <xsl:apply-templates />
  </body>
</html>
</xsl:template>
```

```
<xsl:template match="expedient">
  <xsl:apply-templates select="assignatura"/>
</xsl:template>
```

```
<xsl:template match="assignatura">
<p>
  <xsl:apply-templates select="nom"/>
  <xsl:apply-templates select="nota"/>
</p>
</xsl:template>
```

```
<xsl:template match="nombre">
Nombre: <xsl:value-of select="."/></span>
<br />
</xsl:template>
```

```
<xsl:template match="nota">  
Nota: <xsl:value-of select="." /></span>  
<br />  
</xsl:template>  
  
</xsl:stylesheet>
```

Com podem veure, aquesta organització és molt més modular i ens permet un millor manteniment i revisió del full d'estil.