

Procesamiento de Formatos en Aplicaciones Telemáticas: XPath y XSLT

XSLT (*XSL Transformations*) es un estándar del W3C que define un lenguaje para transformar unos documentos XML en otros. Una de sus aplicaciones típicas es la transformación de documentos XML en visualizaciones HTML (normalmente, XHTML), aunque tiene muchas otras aplicaciones. XSLT se apoya en XPath para direccionar fragmentos de documentos XML. XPath es también un lenguaje definido por el W3C. Permite construir expresiones que se evalúan en el ámbito de árboles que representan documentos XML.

Referencias básicas

Se recomienda consultar el [libro *XSLT 1.0 Pocket Reference*, Evan Lenz, O'Reilly \(2005\)](#).

La especificación de [XSLT](#) y la especificación de [XPath](#) pueden resultar también útiles, así como los manuales disponibles en W3Schools acerca de [XSLT](#) y [XPath](#).

Sesión 1: grupo magistral

En esta sesión se realiza una introducción teórica a los aspectos más relevantes de XPath.

Sesión 2: grupo magistral

En esta sesión se realiza una introducción teórica a los aspectos más relevantes de XSLT.

Sesión 3: práctica

Cómo aplicar plantillas XSLT a un documento XML

Para aplicar una plantilla XSLT a un documento XML es necesario disponer de una aplicación que procese XSLT. La API estándar de Java cuenta ya con una incluida desde el JDK 1.4. El programa [XSLTProcess.java](#) que se proporciona utiliza esta API para aplicar la plantilla XSLT cuyo nombre de fichero se le pase como primer argumento de línea de comandos al documento XML cuyo nombre de fichero se le pase como segundo argumento. El resultado se vuelca a la salida estándar. Un ejemplo de invocación sería:

```
java XSLTProcess plantilla.xsl documento.xml
```

Ejercicios de XPath

En primer lugar, trabajarás con una plantilla XSLT sencilla, y en ella tendrás que ir modificando expresiones XPath. Descarga los ficheros iniciales: [letras.xml](#) y [plantilla-letras.xsl](#). Se reproducen sus contenidos a continuación:

```
<?xml version="1.0"?>
<a id="a1">
  <b id="b1">Letra <c id="c1">B</c></b>
  <d id="d1">
```

```

    <b id="b2">Otra vez</b>
    <e id="e1">
      <d id="d2">
        <e id="e2">
          <f id="f1" tipo="terminal" />
        </e>
      </d>
    </e>
  </d>

  <d id="d3">
    <b id="b3"><c id="c2">...</c></b>
    <e id="e3">
      <f id="f2" tipo="elemental">
        <b id="b4">Letra F</b>
      </f>
    </e>
  </d>

  <e id="e4">
    <f id="f3" tipo="terminal" />
  </e>

</a>

<?xml version="1.0"?>

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

  <xsl:output method="xml" indent="yes" encoding="ISO-8859-1" />

  <xsl:template match="/a">
    <xsl:element name="a" >
      <xsl:apply-templates select="descendant::e" />
    </xsl:element>
  </xsl:template>

  <xsl:template match="*">
    <xsl:copy />
  </xsl:template>

</xsl:stylesheet>

```

Apartado 1

Abre con un editor la plantilla *plantilla-letras.xsl*, e intenta comprender qué es lo que hace. Observa también el documento *letras.xml*. Aplícale la plantilla a este fichero y comprueba que el resultado coincide con lo que esperabas.

Apartado 2

Modifica la plantilla anterior para que seleccione todos los elementos **e** y **d** hijos de **a**.

Apartado 3

Como puedes comprobar, **xsl:copy** sólo copia el nodo actual, pero no sus atributos. Modifica el documento XSLT original para que también copie los atributos de los elementos. **Pista:** utiliza **xsl:apply-templates**, dentro del elemento **xsl:copy**, que seleccione los atributos del elemento. Crea una plantilla (**xsl:template**) que, aplicada a un atributo, lo copie.

Apartado 4

Modifica la expresión XPATH del documento XSLT del apartado anterior para que seleccione sólo los elementos **d** que sean hijo directo de un elemento **e**.

Apartado 5

Modifica la expresión XPATH del documento XSLT del apartado anterior para que seleccione sólo los elementos **e** que sean padres de elementos **f** que posean un atributo **tipo** con valor *"terminal"*. **Pista:** selecciona los elementos **e** como en ejercicios anteriores, y añade entre corchetes una condición a la expresión XPATH.

Apartado 6

Modifica la expresión XPATH del documento XSLT del apartado anterior para que seleccione sólo los elementos que sean hijo único de algún elemento **Pista:** utiliza la función **count()**.

Apartado 7

Modifica la expresión XPATH del documento XSLT del apartado anterior para que seleccione sólo los elementos que sean hijo en segundo grado del elemento raíz.

Ejercicios de XSLT en los documentos de películas

En este ejercicio debes escribir hojas de estilo XSLT que transformen un documento XML de películas, de los desarrollados en prácticas anteriores.

Apartado 8

Escribe una plantilla XSLT que genere un listado de los directores de las películas del fichero de entrada, en el siguiente formato:

```
<directores>
  <director>Steven Spielberg</director>
  <director>Alfred Hitchcock</director>
</directores>
```

Apartado 9

Escribe una plantilla XSLT que genere un listado de las películas en este formato alternativo:

```
<pelis>
  <pele director="Steven Spielberg">La lista de Schindler</pele>
  <pele director="Alfred Hitchcock">Con la muerte en los talones</pele>
</pelis>
```

Apartado 10

Modifica la hoja de estilo XSLT anterior para que se muestren sólo las películas producidas en 2010.

Apartado 11

Escribe una hoja de estilo XSLT que muestre en formato XHTML los datos de películas. Dado que no es objetivo de esta asignatura conocer XHTML, se muestra a continuación un ejemplo del aspecto que podría tener el código XHTML a generar:

```
<?xml version="1.0" encoding="utf-8"?>
<html>
  <head>
    <title>Películas</title>
  </head>
  <body>
    <h1>Listado de Películas</h1>
    <ul>
      <li><i>La lista de Schindler</i>, de Steven Spielberg</li>
      <li><i>Con la muerte en los talones</i>, de Alfred Hitchcock</li>
```

```
</ul>
</body>
</html>
```

Una vez resuelto el ejercicio, aplica la plantilla, guarda el documento de salida en un fichero y ábrelo con un navegador Web.

Ejemplos

```
<?xml version="1.0" encoding="UTF-8"?>
<libros>
  <libro isbn="9788401337208">
    <titulo idioma="es">El nombre del viento</titulo>
    <autor>Patrick Rothfuss</autor>
    <agno>2009</agno>
  </libro>
  <libro isbn="9788496208377">
    <titulo idioma="es">Juego de tronos</titulo>
    <autor>George R. R. Martin</autor>
    <agno>1996</agno>
  </libro>
  <libro isbn="9788497596794">
    <titulo idioma="es">El color de la magia</titulo>
    <autor>Terry Pratchett</autor>
    <agno>1983</agno>
  </libro>
</libros>

<?xml version="1.0"?>

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

  <xsl:output method="xml" indent="yes" encoding="ISO-8859-1" />

  <xsl:template match="/libros">
    <catalogo>
      <isbns>
        <xsl:apply-templates select="libro/@isbn" />
      </isbns>
      <libros>
        <xsl:apply-templates select="libro" />
      </libros>
    </catalogo>
  </xsl:template>

  <xsl:template match="@isbn">
    <isbn>
      <xsl:value-of select="." />
    </isbn>
  </xsl:template>

  <xsl:template match="libro">
    <libro><xsl:value-of select="titulo/text()" /> (<xsl:value-of select="autor/text()" />)</libro>
  </xsl:template>

</xsl:stylesheet>
```

Sesiones 4 y 5: ejercicios

En estas sesiones se resolverán ejercicios de XSLT de entre los que se encuentran en [este documento](#).