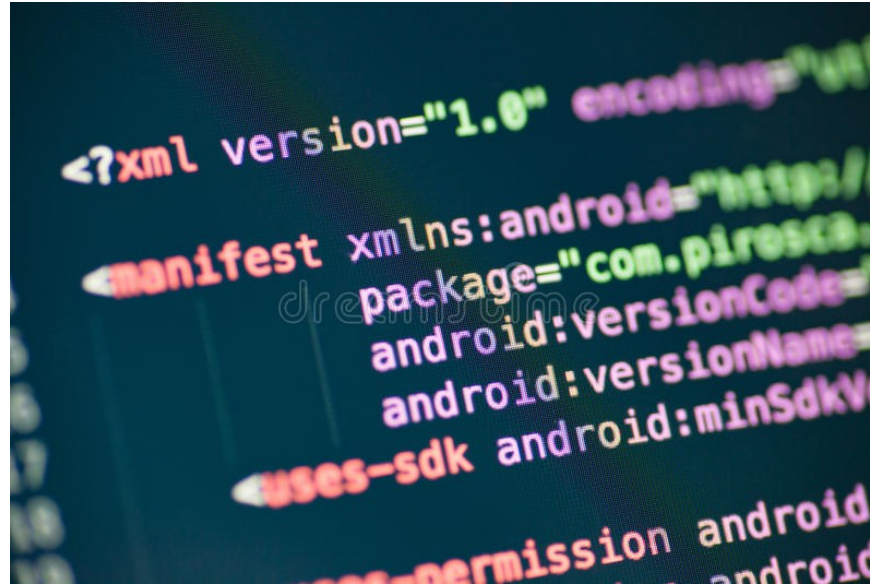


# Lenguajes de marcas y sistemas de gestión de la información



## UT06 – Conversión de documentos

### 3 – XSLT – Funciones – Variables – Include

# Funciones en XPath y XSLT

XSLT, XPath y XQuery comparten la misma biblioteca de funciones.

La URI del namespace de funciones XSLT es <http://www.w3.org/2005/02/xpath-functions>, y su prefijo habitual "fn".

Cuando procesamos XSLT no es necesario declarar el namespace (xmlns:fn="..."). XSLT interpretará adecuadamente las funciones.

Esto quiere decir que cuando usemos una función XSLT basta con que la usemos directamente en la expresión.

Hay más de 100 funciones en XSL, para una referencia más completa, ver, por ejemplo: [https://www.w3schools.com/xml/xsl\\_functions.asp](https://www.w3schools.com/xml/xsl_functions.asp)

# Funciones - Cadenas

Función	Observaciones
string(arg)	Convierte un valor a String. Ejemplo: convertir un número a String.
compare(c1,c2) compare(c1,c2,collation)	Compara dos cadenas. Devuelve -1, 0 o 1. Se puede especificar una intercalación (collation) concreta.
concat(s1, s2, s3, ...)	Concatena cadenas.
string-join((s1,s2,...), sep)	Une cadenas separándolas con la cadena indicada.
substring(s,inicio,longitud) substring(s,inicio)	Obtiene una subcadena de cierta longitud. Si no se especifica longitud, devuelve hasta el final.
string-length(string) string-length()	Obtiene la longitud de la cadena. Si no hay parámetro, usa el valor del nodo actual
normalize-space(string) normalize-space()	Elimina espacios antes y después, y secuencias interiores de espacios, que cambia por uno.

# Funciones - Cadenas

Función	Observaciones
upper-case(s) lower-case(s)	Convierte a mayúsculas y minúsculas.
translate(s, c1, c2)	Reemplaza caracteres en s según la equivalencia entre c1 y c2. El primer carácter de c1 se reemplaza por el primero de c2, y así sucesivamente.
contains(s1,s2)	Devuelve true si s1 contiene s2
starts-with(s1, s2) ends-with(s1, s2)	Devuelve true si s1 empieza o termina (respectivamente) por s2.
replace(s, s1, s2)	Reemplaza todas las apariciones de s1 en la cadena s por s2.
tokenize(s, patron)	Equivalente a un split de Java. Pueden usarse expresiones regulares

# Funciones - Números

Función	Observaciones
number(arg)	Convierte a número el valor recibido.
abs(num)	Valor absoluto
ceiling(num)	Redondea hacia arriba.
floor(num)	Redondea hacia abajo
round(num)	Redondea de la forma habitual
round-half-to-even()	Redondea al número par más cercano

Los resultados de estas funciones se pueden operar con los operadores habituales: +, -, \*, div (no se puede usar la barra "/" porque es parte de XML y de XPath).

# Funciones - Nodos y secuencias

Función	Observaciones
count(nodeset)	Cuenta el número de nodos
name(), name(nodeset)	Nombre de nodo actual, o del primero de la secuencia.
local-name() local-name(nodeset)	Igual, pero sin prefijo de namespace, si es que los nodos lo tenían.
index-of(sq, item)	Posición del elemento en la secuencia
remove(sq, position)	Elimina de la secuencia el elemento en esa posición
empty(sq)	True si la secuencia está vacía
exists(sq)	True si la secuencia no está vacía
distinct-values(sq, coll)	Devuelve no duplicados. Opcional: intercalación
reverse(sq)	Invierte la secuencia
subsequence(sq, inicio, longitud)	Devuelve una subsecuencia de la secuencia. Empieza a contar en 1, no en 0 como en Java.

# Funciones - Contexto

Función	Observaciones
position()	Índice del nodo que estamos procesando
last()	Devuelve la posición del último elemento del conjunto de nodos seleccionado / procesado.
current-dateTime()	Obtiene la fecha y hora actual
current-date()	Obtiene la fecha (sin hora) actual
current-time()	Obtiene la hora (sin fecha) actual

# Funciones - Específicas XSLT

Función	Observaciones
current()	El nodo actual del XML que se está procesando
element-available	Permite comprobar si el procesador XSLT admite cierto elemento XSLT
format-number	Permite dar formato a un número. Para formato de la cadena de formato ver <a href="https://www.w3schools.com/xml/func_formatnumber.asp">https://www.w3schools.com/xml/func_formatnumber.asp</a>
function-available	Permite comprobar si el procesador XSLT admite cierta función

Estas funciones sólo son para XSLT, no para XQuery o XPath.



# XSLT - Variables

En XSLT se pueden definir variables para almacenar valores de forma temporal, para que sea más fácil el procesamiento de la información.

Pueden ser globales, si se declaran en el nivel superior de la XSLT:

```
<xsl:transform version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">  
  
  <xsl:output method="html" omit-xml-declaration="yes" media-type="text/html" />  
  
  <!-- Declaración de variable global -->  
  <xsl:variable name="variableGlobal"></xsl:variable>
```

Tendrán cierto ámbito si se declaran en template / for-each / choose / etc.

```
<xsl:for-each select="biblioteca/libro">  
  <xsl:sort select="anio" order="ascending" />  
  <xsl:variable name="numSubGeneros" select="count(subgeneros/subgenero)"/>
```

# XSLT - Variables

Al declarar la variable se le da un nombre con el atributo “name”.

Para acceder a las variables, se hace lo mismo que en los parámetros de plantillas, se usa el nombre de la variable precedido por “\$”.

Se pueden usar en value-of, atributos con {}, funciones y

```
<xsl:if test="$numSubGeneros > 0">  
  <p><xsl:value-of select="$numSubGeneros"/> subgéneros:</p>
```

```
<xsl:variable name="variableGlobal1">Número de libros: </xsl:variable>
```

```
/  
<xsl:variable name="variableGlobal2" select="count(/biblioteca/libro)"/>
```

# XSLT - Include

Desde una XSLT se pueden referenciar, incluir, otras XSLT.

```
<xsl:include href="comun.xslt"/>
```

Dentro del fichero incluido se definen plantillas:

```
<xsl:transform version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template name="cabecera">
    <xsl:param name="titulo" />
    <header>
      <h1>
        <xsl:value-of select="$titulo" />
      </h1>
    </header>
  </xsl:template>
```

Que se pueden invocar con call-template o con apply-templates.

```
<xsl:call-template name="cabecera">
  <xsl:with-param name="titulo">Biblioteca (02)</xsl:with-param>
</xsl:call-template>
```