

XSLT elements

From Wikipedia, the free encyclopedia

This article lists some of the elements used in XSLT.

Contents

- 1 XSLT logic elements
 - 1.1 Example XSLT stylesheet using logic elements
- 2 XSLT file I/O elements
- 3 Other XSLT semantics
- 4 Functions defined by XSLT
- 5 External links

XSLT logic elements

Node	Description/ attributes	Container/ children	Example
xsl:apply-templates	Specifies that other matches may exist within that node; if this is not specified any matches will be ignored If “select” is specified, only the templates that specify a “match” that fits the selected node or attribute type will be applied ,i.e. the matching elements by "select" attribute in apply-templates corresponding to the template that match the same elements . If “mode” is specified, only the templates that have the same “mode” and have an appropriate “match” will be applied	Any parent Can contain any number of xsl:sort and xsl:with-param children	<xsl:apply-templates/>
xsl:choose	Multiple choices No attributes	Any parent Contains xsl:when blocks and up to one xsl:otherwise block	<xsl:choose> ... </xsl:choose>
xsl:for-each	Creates a loop which repeats for every match “select” designates the match criteria	Any parent Can contain any XML	<xsl:for-each select="input[@name=\$name]"> ... </xsl:for-each>
Example XSLT stylesheet using logic elements	Yes, No, containing	Any parent	<xsl:if test="\$type='text' or \$type='password'">
xsl:if	“test” specifies criteria for entering the if	Can contain any XML	...</xsl:if>

```

<xsl:stylesheet>
<xsl:template match="//input">
<xsl:variable name="type" select="@type"/>
<xsl:variable name="name" select="@name"/>
<xsl:if test="$type='text' or $type='password' or $type='radio' or $type='checkbox'">
  <xsl:choose>
    <xsl:when test="$type='radio'">
      <xsl:if test="not(preceding-sibling::input[@type='radio'])">
        <select name="{@name}">
          <xsl:for-each select="../input[@name=$name]">
            <option value="{@value}">
              <xsl:apply-templates/>
            </option>
          </xsl:for-each>
        </select>
      </xsl:if>
    </xsl:when>
    <xsl:when test="$type='text'">
      <input name="{@name}" type="{@type}">
        <xsl:apply-templates/>
      </input>
    </xsl:when>
    <xsl:when test="$type='password'">
      <input name="{@name}" type="{@type}">
        <xsl:apply-templates/>
      </input>
    </xsl:when>
  </xsl:choose>
</xsl:if>
</xsl:template>
</xsl:stylesheet>

```

XSLT file I/O elements

Node	Description/ attributes	Container/ children	Example
xsl:call-template	Calls the template whose name is specified. <i>name</i> matches the <i>name</i> in xsl-template	Any parent children are xsl:with-param	<xsl:call-template name="lib:make-elem">
xsl:import	Retrieves another XSLT file <i>href</i> is the URI of the file	xsl:stylesheet No children	<xsl:import href="..\Library\FuncLib.xslt"/>
xsl:output	Describes how data will be returned <i>method</i> designates what kind of data is returned <i>omit-xml-declaration</i> indicates if the initial xml tag should be included. <i>encoding</i> designates how the data is returned.	xsl:stylesheet No children	<xsl:output method="xml" omit-xml-declaration="yes" encoding="UTF-8"/>

xsl:param	Designates a parameter which may be passed into the template with xsl:call-template.	xsl:template No children	<xsl:param name="elem-name"/>
xsl:text	Outputs the tag content	Any parent No children	<xsl:text>ClaimNumber ClaimSeqNumber</xsl:text>
xsl:value-of	Outputs a variable <i>select</i> specifies the variable	Any parent No children	<xsl:value-of select="\$s"/>
xsl:with-param	Designates a parameter passed to xsl:call-template. It must be matched by an xsl:param in the template. <i>name</i> specifies the parameter name. <i>select</i> specifies the parameter value	xsl:call-template No children	<xsl:with-param name="elem-name" select="'hma:ClaimNumber'"/>

Client-side XSLT can be implemented in a browser by adding a line like the following to the source XML file, right after the root XML tag. <?xml-stylesheet type="text/xsl" href="family.xsl"?>

This is described on the page <http://www.xml.com/pub/a/2000/10/25/msie/index.html>

Other XSLT semantics

symbol	meaning	where used
	union operator. used to group nodesets in XPath expressions.	select attributes of elements such as xsl:when, xsl:if, xsl:for-each
&	and	any conditional or match criterion, for example xsl:if.test, xsl:when.test, xsl:template.select and xsl:for-each.select
<!--	begin comment	anywhere not in a tag
-->	end comment	anywhere not in a tag
\$	start of a variable name	anywhere in a tag, for example xsl:value-of.select and xsl:variable.name
name()	the name of the tag being processed. Useful if the matching criteria contains s (pipe symbols)	any conditional or match criterion, for example xsl:if.test, xsl:when.test, xsl:template.select and xsl:for-each.select
@	an attribute within the XML	any conditional or match criterion, for example xsl:if.test, xsl:when.test, xsl:template.select and xsl:for-each.select

Functions defined by XSLT

The following functions can occur in many XSLT attributes, such as xsl:value-of.select and xsl:for-each.select.

Function	Description/ Syntax	Example
ceiling	The ceiling function returns the smallest integer that is equal to or is larger than the numeric value of the number argument.	ceiling(3.57)

concat	Concatenates two or more strings, which are listed in the parentheses	concat(\$fname,' ', \$lname)
count	The count function counts and returns the number of nodes in a node-set.	count(elements)
floor	The floor function returns the largest integer that is equal to or is smaller than the numeric value of the number argument.	floor(3.57)
normalize-space	Removes white-space from the beginning and end of the string	normalize-space(\$fname)
position	The position function returns the position number in the current node list of the node that is currently being processed by an xsl:for-each or xsl:apply-templates element. There are no arguments.	position()
round	The round function rounds a number to its closest integer.	round(3.57)
string	The string function converts the value argument to a string.	string()
string-length	The string-length function returns the number of characters in a string. The string argument is optional. If omitted, the default is to use the string value of the context node.	string-length('hello')
substring	A segment within a variable value. Substring takes three parameters: the input variable, the first character to be selected, and the length of the resulting string	substring(\$dob,4,2)
substring-after	The substring-after function returns a portion out of the string specified in the string argument that occurs after the substring specified in the substring argument.	substring-after('In 1814 we took a little trip', 'we')
substring-before	The substring-before function returns a portion out of the string specified in the string argument that occurs before the substring specified in the substring argument.	substring-before('In 1814 we took a little trip', 'we')
sum	The sum function adds and returns the total value of a set of numeric values in a node-set or list of values.	sum(1,3,7,12)
translate	Takes the string in the value argument, replaces all occurrences of a string specified in the string1 argument with substitute characters specified in string2 argument and returns the modified string.	translate('colored armor', 'or', 'our')

External links

- W3C XSLT 1.0 recommendation (<http://www.w3.org/TR/xslt>) - Describes the whole syntax and semantics of XSLT 1.0
- W3C XSLT 2.0 recommendation (<http://www.w3.org/TR/xslt20>)
- XSLT Elements List (http://www.w3schools.com/xsl/xsl_w3celementref.asp) - by W3Schools

Retrieved from "http://en.wikipedia.org/wiki/XSLT_elements"

Categories: XML-based standards

- This page was last modified on 14 September 2010 at 15:44.
- Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. See Terms of Use for details.
Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.