



CALLABLE UNITS

XSLT



NAMED TEMPLATES

NAMED TEMPLATE

The `xsl:call-template` element is used to invoke a template by name. By invoke, we mean that the named template is called and applied to the source document. If a template does not have a name, it cannot be called by this element.

The `xsl:template` element is used to create a template. You can name a template by using the `name` attribute of the `xsl:template` element. Further, the name called by the mandatory `name` attribute of the `xsl:call-template` element must match the name specified by the `name` attribute of the `xsl:template` element. Complicating matters is the fact that a template is not required to have a name. A template is only required to have either a `name` or `match` attribute. (It can have both.) Ideally, each template will have a unique name. However, if a name is repeated, then the two templates must have a different import precedence (refer to the `xsl:import` element), otherwise it is an error.



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

<xsl:variable name="hoo">
<html>
<body>
<xsl:call-template name="boo">
<xsl:with-param name="name" />
<xsl:with-param name="dob" />
</xsl:call-template>
</body>
</html>
</xsl:variable>

<xsl:template name="boo" match="/">
<xsl:param name="name" />
<xsl:param name="dob" />
<xsl:for-each select="devguru_staff/programmer">
<div>
NAME: <xsl:value-of select="name" />
<br />
DOB: <xsl:value-of select="dob" />
<hr />
</div>
</xsl:for-each>
</xsl:template>

</xsl:stylesheet>
```

OUTPUT

NAME: Bugs Bunny
DOB: 03/21/1970

NAME: Daisy Duck
DOB: 08/09/1949

NAME: Minnie Mouse
DOB: 04/13/1977

NAME: Pluto
DOB: 07/04/1979

NAME: Porky Pig
DOB: 11/30/1956

NAME: Road Runner
DOB: 01/19/1953

XSLT FUNCTIONS



Introduction to XSLT function

Many built-in functions in XSLT allow for the manipulation of strings, numbers, dates, and many other types of data. They are defined by the prefix namespace fn:, which provides the appropriate category's useful functions. Standard XSLT 1.0 or 2.0 is used to write the templates where the logic of the functions is implemented, which are the main components of building an XSLT function. Each function needs at least one signature.



List of XSLT Functions with Syntax

Here, we'll examine the various XSLT functions. From the site where simple XPATH expressions are offered, functions can be called. Functions may be both built-in and user-defined. A properly-designed function should include parameters and return types, as well as names that are the same throughout all of the functions.

CURRENT()

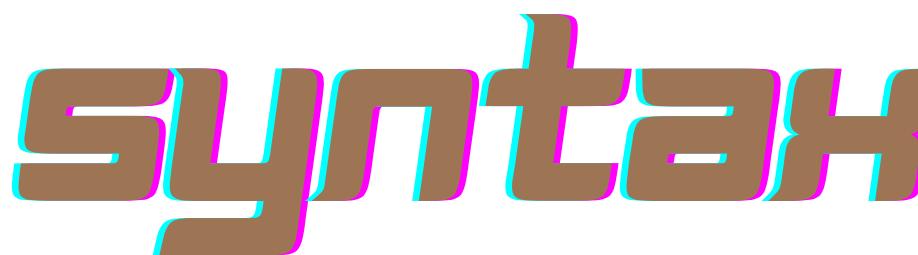
The XSLT function returns the element that contains the current node. This function is necessary for accessing the current node when it is not the same as the context node provided in the code because both the nodes are the same.

syntax

```
<xsl:value-of>=Current()  
// This function doesn't take any argument
```

FORMAT_NUMBER()

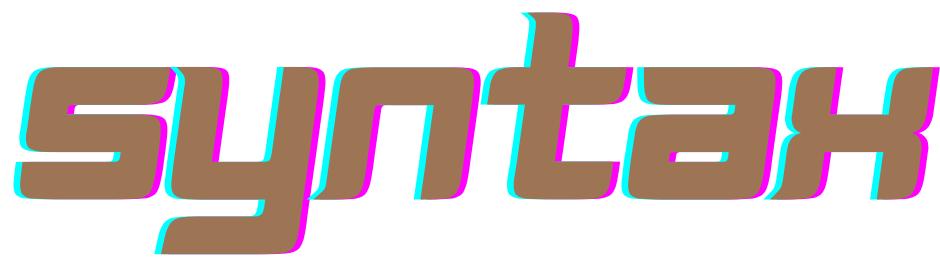
This function converts a given input number into a string. This acts together with the decimal format. For instance, let's say the number 89067 is an input value, and the second argument is the pattern.



string format-number(number,format,[decimalformat])

function_available()

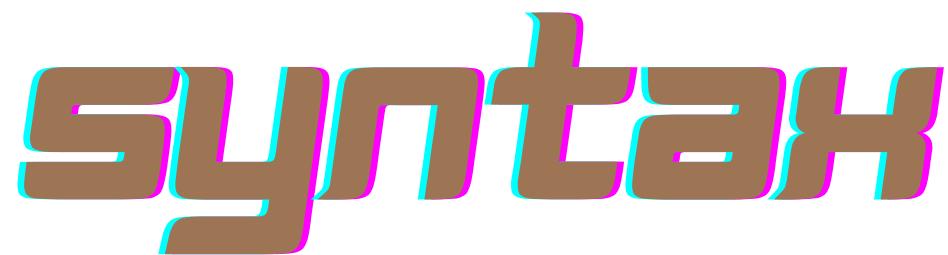
This boolean function checks whether the processor supports the function specified inside this function.



Boolean function_available(sum/add... a string);

element-available()

This function checks the available of the element that supports it. This function uses few elements inside the templates:
xsl:choose,xsl:copy,xsl:attributes, xsl:value-of and so on.



Element-available(function string;)

KEY()

This function returns specific nodes from the document using the element <xsl:key>.



<xsl:for-each> key(string , value)

FORMAT-DATE()

Format date is used much in web development frequently for manipulation. The format date on XSLT Technique uses the Time Standard format. The format-date function uses picture strings with the Specifiers like Y for Year, M-month in a year, D for Day in a week.



```
<xsl:attribute name="feed">
<xsl:value-of select =" format-date(function name) (inpvvalue, 'MMM dd, yyyy')"/>
<xsl:attribute>
```

SUBSTRING-BEFORE()

This function provides a section of a string to be printed in the result. The substring -before returns the value before the subpart.



```
Substring-before('free - dom' ,'-')  
// this returns a value before the hyphen .so it is 'free'.
```





THANKS FOR WATCHING

Thynk Unlimited

Presentations are communication tools
that can be used as demonstrations.