Cheatography

XSLT 2.0 Cheat Sheet Cheat Sheet

by Jin Lei (Univer) via cheatography.com/2588/cs/673/

About

XSL stands for EXtensible Stylesheet Language, and is a style sheet language for XML documents.
XSLT stands for XSL Transformations. In this tutorial you will learn how to use XSLT to transform XML documents into other formats, like XHTML.

Elements

apply-imports	Applies a template rule from an imported style sheet
apply-templates	Applies a template rule to the current element or to the current element's child nodes
attribute	Adds an attribute
attribute-set	Defines a named set of attributes
call-template	Calls a named template
choose	Used in conjunction with <when> and <otherwise> to express multiple conditional tests
comment	Creates a comment node in the result tree
copy	Creates a copy of the current node (without child nodes and attributes)
copy-of	Creates a copy of the current node (with child nodes and attributes)
decimal-format	Defines the characters and symbols to be used when converting numbers into strings, with the format-number() function
element	Creates an element node in the output document

Elements (cont)

fallback	Specifies an alternate code to run if the processor does not support an XSLT element
for-each	Loops through each node in a specified node set
if	Contains a template that will be applied only if a specified condition is true
import	Imports the contents of one style sheet into another. Note: An imported style sheet has lower precedence than the importing style sheet
include	Includes the contents of one style sheet into another. Note: An included style sheet has the same precedence as the including style sheet
key	Declares a named key that can be used in the style sheet with the key() function
message	Writes a message to the output (used to report errors)
namespace-alias	Replaces a namespace in the style sheet to a different namespace in the output
number	Determines the integer position of the current node and formats a number

Elements (cont)

otherwise	Specifies a default action for the <choose> element
output	Defines the format of the output document
param	Declares a local or global parameter
preserve-space	Defines the elements for which white space should be preserved
processing-instruction	Writes a processing instruction to the output
sort	Sorts the output
strip-space	Defines the elements for which white space should be removed
stylesheet	Defines the root element of a style sheet
template	Rules to apply when a specified node is matched
text	Writes literal text to the output
transform	Defines the root element of a style sheet
value-of	Extracts the value of a selected node
variable	Declares a local or global variable
when	Specifies an action for the <choose> element
with-param	Defines the value of a parameter to be passed into a template

By **Jin Lei** (Univer)
cheatography.com/univer/
jinlei.me

Published 21st November, 2012.
Last updated 5th June, 2014.

Cheatography

XSLT 2.0 Cheat Sheet Cheat Sheet

by Jin Lei (Univer) via cheatography.com/2588/cs/673/

Selectors

nodename

Selects all nodes with the name "nodename"

/

Selects from the root node

//

Selects nodes in the whole document

.

Selects the current node

..

Selects the parent of the current node

@

Selects attributes

*

Matches any element node

@*

Matches any attribute node

//node[1]

Selects the first element that is the child of the element.

//node[@class and @id]

select the node with both "class" and "id"

//node[count(child)=2]

select the node with two "child" elements

//node[contains(@title,"text")]

select the node with "text" in the title attribute

//node[child/child1]

select the node with "child/child1" child nodes

//node[position() mode 2 ==0]

select the odd children elements

//node/text()[2]

return the second text element of node

//node[not(@class)]

the node without "class" attribute

Accessor Functions

node-name(node)

Returns the node-name of the argument node

nilled(node)

Returns a Boolean value indicating whether the argument node is nilled

data(item.item,...)

Takes a sequence of items and returns a sequence of atomic values

base-uri() fn:base-uri(node)

Returns the value of the base-uri property of the current or specified node

document-uri(node)

Returns the value of the document-uri property for the specified node

Functions on Nodes

name()	Returns the node-name of the argument node
local-name()	Returns a Boolean value indicating whether the argument node is nilled
namespace-uri()	Takes a sequence of items and returns a sequence of atomic values
lang(lang)	Returns the value of the base-uri property of the current or specified node
root()	Returns the value of the document-uri property for the specified node

Functions on Numeric Values

number(arg)	Returns the numeric value of the argument. The argument could be a boolean, string, or node-set
abs(num)	Returns the absolute value of the argument
ceiling(num)	Returns the smallest integer that is greater than the number argument
floor(num)	Returns the largest integer that is not greater than the number argument
round(num)	Rounds the number argument to the nearest integer

Aggregate Functions

count((item,item,...))

Returns the count of nodes

avg((arg,arg,...))

Returns the average of the argument values

max((arg,arg,...))

Returns the argument that is greater than the others

min((arg,arg,...))

Returns the argument that is less than the others

sum(arg,arg,...)

Returns the sum of the numeric value of each node in the specified node-set

Context Functions

position()	Returns the index position of the node that is currently being processed

By **Jin Lei** (Univer)
cheatography.com/univer/
jinlei.me

Published 21st November, 2012.
Last updated 5th June, 2014.

Cheatography

XSLT 2.0 Cheat Sheet Cheat Sheet

by Jin Lei (Univer) via cheatography.com/2588/cs/673/

Context Functions (cont)

last()	Returns the number of items in the processed node list
current-dateTime()	Returns the current dateTime (with timezone)
current-date()	Returns the current date (with timezone)
current-time()	Returns the current time (with timezone)

Functions on Strings

string(arg)

> Returns the string value of the argument. The argument could be a number, boolean, or node-set

codepoints-to-string(int,int,...)

> Returns a string from a sequence of code points

string-to-codepoints(string)

> Returns a sequence of code points from a string

codepoint-equal(comp1,comp2)

> Returns true if the value of comp1 is equal to the value of comp2, according to the Unicode code point collation, otherwise it returns false

compare(comp1,comp2)

> Returns -1 if comp1 is less than comp2, 0 if comp1 is equal to comp2, or 1 if comp1 is greater than comp2 (according to the rules of the collation that is used)

string-join((string,string,...),sep)

> Returns a string created by concatenating the string arguments and using the sep argument as the separator

substring(string,start,len)

> Returns the substring from the start position to the specified length. Index of the first character is 1. If length is omitted it returns the substring from the start position to the end

Functions on Strings (cont)

string-length(string)

> Returns the length of the specified string. If there is no string argument it returns the length of the string value of the current node

normalize-space(string)

> Removes leading and trailing spaces from the specified string, and replaces all internal sequences of white space with one and returns the result. If there is no string argument it does the same on the current node

normalize-unicode()

upper-case(string)

> Converts the string argument to upper-case

lower-case(string)

> Converts the string argument to lower-case

translate(string1,string2,string3)

> Converts string1 by replacing the characters in string2 with the characters in string3

escape-uri(stringURI,esc-res)

contains(string1,string2)

> Returns true if string1 contains string2, otherwise it returns false

starts-with(string1,string2)

> Returns true if string1 starts with string2, otherwise it returns false

ends-with(string1,string2)

> Returns true if string1 ends with string2, otherwise it returns false

substring-before(str1,str2)

> Returns the start of string1 before string2 occurs in it

substring-after(str1,str2)

> Returns the remainder of string1 after string2 occurs in it

Functions on Strings (cont)

matches(string,pattern)

> Returns true if the string argument matches the pattern, otherwise, it returns false

replace(string,pattern,replace)

> Returns a string that is created by replacing the given pattern with the replace argument

tokenize(string,pattern)

Functions on Boolean Values

boolean (arg)	Returns a boolean value for a number, string, or node-set
not(arg)	The argument is first reduced to a boolean value by applying the boolean() function. Returns true if the boolean value is false, and false if the boolean value is true
true()	Returns the boolean value true
false()	Returns the boolean value false

By **Jin Lei** (Univer)
cheatography.com/univer/
jinlei.me

Published 21st November, 2012.
Last updated 5th June, 2014.