

# MAPPING FUNCTIONS

## Arithmetic

add : adds two numerical inputs

sub : subtract two numerical inputs

multiply : multiply two numerical inputs

div : divide two numerical input

1/x : returns the reciprocal of an input

abs : returns the absolute value of an input number

equalsA : compare two numbers if both were the same returns true  
otherwise returns false.

sqr : square value of a number

sqrt : square root value of a number

neg : returns the negative value of an input

less : compare two inputs and if first one is less than second one returns  
true otherwise returns false

greater : compares two inputs and if first one is greater than second one  
returns true otherwise returns false

power : calculates the first input using the second input as power.

max : compare two inputs and returns the maximum value

min : compares two inputs and returns the minimum value

ceil : returns the next whole number of an float value

floor : returns the previous whole number of an float value

round : round a number using its floating value

counter : generates the number by the initial and increment value given in properties of functions

FormatNum : converts a number using the format given in the function properties.

## **Boolean**

And : if both inputs are true returns true otherwise returns false

Or : if one input is true returns true otherwise false

Not : returns the negation of input

Equals : if both inputs are same returns true otherwise returns false

NotEquals : if both inputs are not same returns true otherwise return false

If : if a condition is true returns the then part otherwise returns the else part

IfWithoutElse : if a condition is true returns the then part otherwise returns nothing

Ifs and IfsWithoutElse are same like above which is used for string parameters only

isNill : always returns false

## **Text**

Substring : returns a string using start position and number of characters given in function properties

concat : returns a merged string of two input texts

equalsS : compare two texts and if both were same returns true otherwise it will be false

indexOf (2) : finds a index of an character in a string

indexOf(3) : finds an index of a character in a string from the specific position

lastindexOf(2) : do the same indexof(2) but starts from the end of the text

LastindexOf(3) : do the same lastindexOf(2) but starts from a specific position

Compare : compares two texts and both are same returns 0 not same returns positive or negative integer as per the size of texts

replaceString : replace a specific text as specific text from a text

Length : count a characters in a string

endsWith(2) : check a string using a string that is the end of that string or not and provides true or false

endsWith(3) : same as endsWith(2) starts with specific position

startsWith(2) : search a string in a string if it is the starting of the string returns true otherwise returns false

startsWith(3) : same as startsWith(2) but starts with specific position

toUpperCase : change all the characters in a string to Uppercase

toLowerCase : change all the characters in a string to Lowercase

Trim : remove the spaces at the beginning and ending of a string but never removes spaces between the string

## **Date**

currentDate : returns the current date at the execution time

DateTrans : converts the date format into specific format

DateBefore : compare a date using a date if first is before of second returns true otherwise returns false

DateAfter : compare a date using a date if first is after second date  
returns true otherwise returns false

CompareDates : doubt

## **Constants**

Constant : to insert a constant value

CopyValue : to copy a value from one field to another.

Sender : returns Test\_SenderSystem

Receiver :returns Test\_Receiver\_System

Xsl:nil : parse an empty value to the target field

## **Statistics**

sum : returns the sum value of subtrees

average: returns the average value of subtrees

count: returns the count of subtrees

index: generates numbers using initial and incremental numbers given in  
function properties and repeat or restart for the context changes

## **Node Functions**

Createif : if input is true parse field but no value , if input is false  
suppress the field.

removeContexts : suppress the fields in subtrees and move it to the  
source node

Collapse contexts : same as remove contexts but move only first fields to  
the source node

Exists : if a node exists return true otherwise returns false

SplitbyValue : split the context using a value given in properties and provides it to the subtree in upcoming contexts

sort : sort subtrees by properties given in function.. To sort float values type of field should be xsd:float

mapwithDefault : map a field using a default value

## **Doubts :**

Counter (arithmetic)

useOneasMany (Node)

formatByExample (Node)

SortByKey (Node)

Compare (text)