String Functions

In the functions below, the parameters are expressions where *s* should be interpreted as a string.

`ord( s )`

ASCII number of first character of string *s*. The result is an integer.

Example:

`ord('A')` returns the number *65*.

`chr( n )`

The character corresponding to number *n* in ASCII (1-127), ANSI (128-255) or Unicode (256 and above). The result is a string.

Example:

`chr(65)` returns the string *'A'*.

`len( s )`

Length of string *s*. The result is an integer.

Example:

`len(Name)` where *Name* = 'Peter' returns *5*.

`left( s , n )`

Substring of the string *s*. The result is a string consisting of the first *n* characters of *s*.

Example:

`left('abcdef',3)` returns *'abc'*.

`left(Date, 4)` where *Date* = 1997-07-14 returns *1997*.

For a more complicated example, see the **index** function below.

`right( s , n )`

Substring of the string *s*. The result is a string consisting of the last *n* characters of *s*.

Example:

`right('abcdef',3 )` returns *'def'*.

`right( Date,2 )` where *Date* = 1997-07-14 returns *14*.

`mid(s, n1[, n2 ])`

Substring of the string *s*. The result is the string starting at character *n1* with the length of *n2* characters. If *n2* is omitted, the function returns the rightmost part of the string starting at character *n1*. The positions in the string are numbered from *1* and up.

Example:

`mid('abcdef',3 )` returns *'cdef'*.

`mid('abcdef',3, 2 )` returns *'cd'*.

`mid( Date,3 )` where *Date* = 970714 returns *0714*.

`mid( Date,3,2 )` where *Date* = 970714 returns *07*.

For a more complicated example, see the index function below.

index(*s1 , s2[, n]*)

Position of a substring. This function gives the starting position of the *n*:th occurrence of substring *s2* in string *s1*. If *n* is omitted, the first occurrence is assumed. If *n* is negative, the search is made starting from the end of string *s1*. The result is an integer. The positions in the string are numbered from *1* and up.

Examples:

`index( 'abcdefg', 'cd' )` returns *3*

`index( 'abcdabcd', 'b', 2 )` returns *6*

`index( 'abcdabcd', 'b', -2 )` returns *2*

`left( Date, index( Date,'-') -1 )` where *Date* = 1997-07-14 returns *1997*

`mid( Date, index( Date, '-', 2 ) -2, 2 )` where *Date* = 1997-07-14 returns *07*

upper(*textexpression*)

Forces upper case for all data in the expression.

Example:

`upper( 'abcD' )` returns *'ABCD'*.

lower(*textexpression*)

Forces lower case for all data in the expression.

Example:

`lower( 'abcD' )` returns *'abcd'*.

repeat(*s, n*)

Forms a string consisting of the string *s* repeated *n* times.

Example:

`repeat( ' * ', rating )` when *rating* = 4 returns '****'.

ltrim(*s*)

Returns the string *s* trimmed of any leading spaces.

Example:

`ltrim ( ' abc' )` returns '*abc*'.

`ltrim ( 'abc ' )` returns '*abc* '

rtrim(*s*)

Returns the string s trimmed of any trailing spaces.

Example:

`rtrim ( ' abc' )` returns ' *abc*'

`rtrim ( 'abc ' )` returns '*abc*'

trim(*s*)

Returns the string *s* trimmed of any leading and trailing spaces.

Example:

`trim ( ' abc' )` returns '*abc*'

`trim ( 'abc ' )` returns '*abc*'

`trim ( ' abc ' )` returns '*abc*'

subfield(*s, 'delimiter' [, index]*)

In its three-parameter version, this script function returns a given substring from a larger string *s* with delimiter *'delimiter'*. *index* is an optional integer denoting which of the substrings should be returned. If *index* is omitted when **subfield** is used in a field expression in a **load** statement, the **subfield** function will cause the **load** statement to automatically generate one full record of input data for each substring that can be found in *s*.

In its two-parameter version, the **subfield** function generates one record for each substring that can be taken from a larger string *s* with the delimiter *'delimiter'*. If several **subfield** functions are used in the same **load** statement, the Cartesian product of all combinations will be generated.

Examples:

(For three parameters)

`subfield(S, ';' ,2)` returns '*cde*' if S is '*abc*;*cde*;*efg*'

`subfield(S, ';' ,1)` returns NULL if S is an empty string

`subfield(S, ';' ,1)` returns an empty string if S is ';'

KeepChar(*s1 , s2*)

Returns the string *s1* less all characters not contained in string *s2*.

Example:

`keepchar ( 'a1b2c3','123' )` returns '*123'*

PurgeChar(*s1, s2*)

Returns the string *s1* less all characters contained in string *s2*.

Example:

`purgechar ( 'a1b2c3','123' )` returns '*abc*'

capitalize(*s*)

Returns the string *s* with all words capitalized.

Example:

`capitalize ('my little pony')` returns '*My Little Pony*'

`capitalize ( 'AA bb cC Dd')` returns '*Aa Bb Cc Dd*'

evaluate(*s*)

If the text string *s* can be evaluated as a valid QlikView expression, the evaluated result of the expression will be returned. If *s* is not a valid expression, NULL is returned.

Note!
This string function can not be used in chart expressions.

Example:

`evaluate ( 5 * 8 )` returns *40*

TextBetween(*s , beforetext , aftertext [, n]*)

Returns the text between the *n*:th occurrence of *beforetext* and the immediately following occurrence of *aftertext* within the string *s*.

Examples:

`TextBetween('<abc>', '<', '>')` returns '*abc*

> `TextBetween('<abc><de>', '<', '>',2)` returns '*de*'

Replace(*s, fromstring ,tostring*)

Returns a string after replacing all occurrences of a given substring within the string *s* with another substring. The function is non-recursive and works from left to right.

s is the original string.

fromstring is a string which may occur one or more times within string.

tostring is the string which will replace all occurrences of *fromstring* within the string.

Example:

`replace('abccde','cc','xyz')` returns '*abxyzde*'

FindOneOf(*text , characterset [, n]*)

Returns the position of the *n*:th occurrence in the string *text* of any of the characters found in the string *characterset*. If *n* is omitted, the position of the first occurrence is returned. If there no matching string is found, *0* is returned.

Example:

`findoneof( 'my example text string', 'et%s')` returns '*4*'

`findoneof( 'my example text string', 'et%s', 3)` returns '*12*'

`findoneof( 'my example text string', '¤%&')` returns '*0*'

hash128(*expression {, expression}*)

Returns a 128-bit hash of the combined input expression values. The result is a string.

Examples:

```
hash128 ( 'abc', 'xyz', '123' )
hash128 ( Region, Year, Month )
```

hash160(*expression {, expression}***)**

Returns a 160-bit hash of the combined input expression values. The result is a string.

Example:

```
hash160 ( Region, Year, Month )
```

hash256(*expression {, expression}* **)**

Returns a 256-bit hash of the combined input expression values. The result is a string.

Example:

```
hash256 ( Region, Year, Month )
```

substringcount(*text , substring***)**

Returns the number of times the string substring appears within the string text. The result is an integer. If there is no match, 0 is returned

Example:

`substringcount ( 'abcdefgcdxyz', 'cd' )` returns *2*

applycodepage(*text, codepage***)**

Applies a different codepage to the field or text stated in the expression. The codepage must be in number format.

QlikView 11.20 SR7