### Matrix Manipulations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function** | | **tokenize**  **tokenize convert** | | | | |
| Description | | Breaks a string into individual token (pieces)  **tokenize** (…) generates a parameter set of literals, regardless of the contents.  **tokenize convert** (…) will convert data to other Beyond4P types, e.g. YYYY-MM-DD dates into dates, 'true' into Boolean, etc.  The tokenizer supports full UNICODE character set. | | | | |
| Call as | |  | Procedure call | | **X** | Function call, providing a return value |
| Parameter count | | Min 1 | | | | |
| Parameters | 1. | Literal (Numbers converted to literal) | | Input string | | |
|  | Opt. 2 | Parameter set or  Literal | | Separator strings  Specify at least 1 separator string (e.g. blank, comma, tab, slash, etc). The string may contain multiple characters. In this case, the sequence of these multiple represent the separation  Example: { ",", "//" } Commas and 2 consecutive slashes | | |
|  | Opt. 3 | Parameter set or Literal | | Quotation Marks  If one literal used: considered for opening and closing- Example: "Hello World"  If 2 literals used: Separate for opening and closing. Example: «Hello World»  If number of quotation marks provided is odd, then the last quotation mark applies both as opening and closing quotation marks.  Multi-character quotation marks, e.g. <TEXT>Hello World</TEXT> are also allowed. | | |
|  | Opt. 4 | Parameter set or Literal | | Additional Tokens  Collection of token symbols to be parsed separately and suitable for categorization. This is specifically important if these tokens follow without separators (e.g. white spaces) in-between. You can also assign multi-character tokens, e.g. "<=", "<>", "while", "for", … etc. | | |
|  | Opt. 5 | Parameter set of Literal | | Comment Symbols  Parameter handling like with quotation marks (opening and closing comment symbols)  Example: { "/\*", "\*/", "<--", "-->" } | | |
|  | Opt. 6 | Parameter set or Literal | | Line comment Symbols  List of literals indicating begin of comment till end of line  Example: { "#!", "//", "Comment:" } | | |
|  | Opt. 7 | Parameter set or Literal | | Options  numbers with decimal separators (Next character: comma, point)  numbers with thousand separators (Next character: point, comma, apostrophe, space, local)  numbers with exponents (Accepts nnnE+nnn, nnnNE-nnn, nnnEnnn)  ignore multiple separators  ignore case  allow new lines in quotations  include quotations in text  include quotations as tokens | | |
| Return value | | Parameter set | | Tokenized result | | |
| Examples | | **tokenize** ( "The \_quick\_ 'brown fox' jumps,over the black\_dog", {" ", ","}, "'", "\_" );  Return: { "The", "quick", "brown fox", "jumps", "over", "the", "blackdog" } | | | | |
| See also | | **parameter set (…)** | | | | |

a[] = **tokenize**( "a + 5 \* (b – 1)", {" "}, {"+","-","\*","/","(",")" } );

b[] = **parse and run**( a[ ], { … rules }, statement 1, statement 2, statement 3, .. );

**clean if numeral** ( number, dec point, option )

strict Clean if only containing digits, blanks, decimal and thousand separators  
 Minus sign must precede the numbers.

negations Also converts brackets or parentheses to minus sign

standard Strips out text and symbols (e.g. currency names)