Table of Contents

Foswiki Spreadsheet Plugin **1**
Syntax Rules 1
Built-in Functions 1
ABOVE( ) -- address range of cells above the current cell 2
ABS(num) -- absolute value of a number 2
AND(list) -- logical AND of a list 2
AVERAGE(list) -- average of a list or a range of cells 2
CHAR(number) -- ASCII character represented by number 2
CODE(text) -- ASCII numeric value of character 2
COLUMN(offset) -- current column number 2
COUNTITEMS(list) -- count individual items in a list 3
COUNTSTR(list, str) -- count the number of cells in a list equal to a given string 3
DEF(list) -- find first non-empty list item or cell 3
EMPTY(text) -- test for empty text 3
EVAL(formula) -- evaluate a simple mathematical formula 3
EVEN(num) -- test for even number 4
EXACT(text1, text2) -- compare two text strings 4
EXEC(formula) -- execute a spreadsheet formula 4
EXISTS(topic) -- check if topic exists 4
EXP(num) -- exponent (e) raised to the power of a number 4
FIND(string, text, start) -- find one string within another string 4
FORMAT(type, precision, number) -- format a number to a certain type and precision 5
FORMATGMTIME(serial, text) -- convert a serialized date into a GMT date string 5
FORMATTIME(serial, text) -- convert a serialized date into a date string 5
FORMATTIMEDIFF(unit, precision, time) -- convert elapsed time to a string 6
GET(name) -- get the value of a previously set variable 6
IF(condition, value if true, value if 0) -- return a value based on a condition 6
INSERTSTRING(text, start, new) -- insert a string into a text string 7
INT(formula) -- evaluate formula and round down to nearest integer 7
LEFT( ) -- address range of cells to the left of the current cell 7
LEFTSTRING(text, num) -- extract characters at the beginning of a text string 7
LENGTH(text) -- length of text in bytes 8
LIST(range) -- convert content of a cell range into a list 8
LISTIF(condition, list) -- remove elements from a list that do not meet a condition 8
LISTITEM(index, list) -- get one element of a list 8
LISTJOIN(separator, list) -- convert a list into a string 8
LISTMAP(formula, list) -- evaluate and update each element of a list 9
LISTRAND(list) -- get one random element of a list 9
LISTREVERSE(list) -- opposite order of a list 9
LISTSIZE(list) -- number of elements in a list 9
LISTSHUFFLE(list) -- shuffle element of a list in random order 9
LISTSORT(list) -- sort a list 10
LISTTRUNCATE(size, list) -- truncate list to size 10
LISTUNIQUE(list) -- remove all duplicates from a list 10
LN(num) -- natural logarithm of a number 10
LOG(num, base) -- logarithm of a number to a given base 10
LOWER(text) -- lower case string of a text 10
MAX(list) - biggest value of a list or range of cells 11
MEDIAN(list) -- median of a list or range of cells 11
MIN(list) -- smallest value of a list or range of cells 11
MOD(num, divisor) -- reminder after dividing num by divisor 11
NOEXEC(formula) -- do not execute a spreadsheet formula 11
NOP(text) -- no-operation 11
NOT(num) -- reverse logic of a number 11

Table of Contents

Foswiki Spreadsheet Plugin

ODD(num) -- test for odd number....12
OR(list) -- logical OR of a list....12
PERCENTILE(num, list) -- percentile of a list or range of cells....12
PI( ) -- mathematical constant Pi, 3.14159265358979....12
PRODUCT(list) -- product of a list or range of cells....12
PROPER(text) -- properly capitalize text....12
PROPERSPACE(text) -- properly space out WikiWords....13
RAND(max) -- random number....13
REPEAT(text, num) -- repeat text a number of times....13
REPLACE(text, start, num, new) -- replace part of a text string....13
RIGHT( ) -- address range of cells to the right of the current cell....13
RIGHTSTRING(text, num) -- extract characters at the end of a text string. num must be a positive number. Negative values of num are interpreted as zero. If num is larger than the length of the text the entire text is returned with no additional spaces....13
ROUND(formula, digits) -- round a number....14
ROW(offset) -- current row number....14
SEARCH(string, text, start) -- search a string within a text....14
SET(name, value) -- set a variable for later use....14
SETIFEMPTY(name, value) -- set a variable only if empty....15
SETM(name, formula) -- update an existing variable based on a formula....15
SIGN(num) -- sign of a number....15
SQRT(num) -- square root of a number....15
SUBSTITUTE(text, old, new, instance, option) -- substitute text....15
SUBSTRING(text, start, num) -- extract a substring out of a text string....16
SUM(list) -- sum of a list or range of cells....16
SUMDAYS(list) -- sum the days in a list or range of cells....16
SUMPRODUCT(list, list) -- scalar product on ranges of cells....16
T(address) -- content of a cell....16
TRANSLATE(text, from, to) -- translate text from one set of characters to another....17
TIME(text) -- convert a date string into a serialized date number....17
TIMEADD(serial, value, unit) -- add a value to a serialized date....17
TIMEDIFF(serial_1, serial_2, unit) -- time difference between two serialized dates....17
TODAY( ) -- serialized date of today at midnight GMT....18
TRIM(text) -- trim spaces from text....18
UPPER(text) -- upper case string of a text....18
VALUE(text) -- convert text to number....18
WORKINGDAYS(serial_1, serial_2) -- working days between two serialized dates....18
FAQ....18
Can I use CALC in a formatted search?....18
How can I easily repeat a formula in a table?....19
Bug Tracking Example....19
Plugin Settings....19
Plugin Installation Instructions....20
Plugin Info....20

Foswiki Spreadsheet Plugin

This Plugin adds spreadsheet capabilities to Wiki topics. Formulae like **`%CALC{"$INT(7/3)"}%`** are evaluated at page view time. They can be placed in table cells and outside of tables. In other words, this Plugin provides general formula evaluation capability, not just classic spreadsheet functions.

Example:

Region:	Sales:
Northeast	320
Northwest	580
South	240
Europe	610
Asia	220
Total:	1970

The formula next to "Total" is **`%CALC{"$SUM( $ABOVE() )"}%`**.
(you see the formula instead of the sum in case the Plugin is not installed or not enabled.)

Interactive example:

Formula: `%CALC{""}%`
Result: Wiki Guest

Syntax Rules

The action of this Plugin is triggered by the **`%CALC{"..."}%`** macro, which gets rendered according to the built-in function(s) found between the quotes.

- Built-in function are of format **`$FUNCNAME(parameter)`**
- Functions may be nested, e.g. **`%CALC{"$SUM( R2:C$COLUMN(0)..R$ROW(-1):C$COLUMN(0) )"}%`**
- Functions are evaluated from left to right, and from inside to outside if nested
- The function parameter can be text; a mathematical formula; a cell address; or a range of cell addresses
- Multiple parameters form a list; they are separated by a comma, followed by optional space, e.g. **`%CALC{"$SUM( 3, 5, 7 )"}%`**
- A table cell can be addressed as **`R1:C1`**. Table address matrix:

`R1:C1`	**`R1:C2`**	**`R1:C3`**	**`R1:C4`**
`R2:C1`	**`R2:C2`**	**`R2:C3`**	**`R2:C4`**

- A table cell range is defined by two cell addresses separated by **`".."`**, e.g. "row 1 through 20, column 3" is: **`R1:C3..R20:C3`**
- Lists can refer to values and/or table cell ranges, e.g. **`%CALC{"$SUM( 3, 5, $T(R1:C7), R1:C11..R1:C15 )"}%`**
- Formulae can only reference cells in the current or preceeding row of the current table; they may not reference cells below the current table row
- Formulae can also be placed outside of tables; they can reference cells in the preceeding table
- Formulae can be placed in a FormattedSearch, but the CALC needs to be escaped. Learn how to use a CALC in a formatted search

Built-in Functions

Conventions for Syntax:

- Required parameters are indicated in **`( bold )`**

- Optional parameters are indicated in **(*bold italic*)**

ABOVE( ) -- address range of cells above the current cell

- Syntax: **$ABOVE( )**
- Example: **%CALC{"$SUM($ABOVE())"}%** returns the sum of cells above the current cell
- Related: $LEFT(), $RIGHT()

ABS(num) -- absolute value of a number

- Syntax: **$ABS(num)**
- Example: **%CALC{"$ABS(-12.5)"}%** returns **12.5**
- Related: $SIGN(), $EVEN(), $ODD()

AND(list) -- logical AND of a list

- Syntax: **$AND(list)**
- Example: **%CALC{"$AND(1, 0, 1)"}%** returns **0**
- Related: $NOT(), $IF(), $OR()

AVERAGE(list) -- average of a list or a range of cells

- Syntax: **$AVERAGE(list)**
- Example: **%CALC{"$AVERAGE(R2:C5..R$ROW(-1):C5)"}%** returns the average of column 5, excluding the title row
- Related: $LIST(), $MAX(), $MEDIAN(), $MIN()

CHAR(number) -- ASCII character represented by number

- Syntax: **$CHAR(number)**
- Example: Example: **%CALC{"$CHAR(97)"}%** returns **a**
- Related: $CODE()

CODE(text) -- ASCII numeric value of character

- The ASCII numeric value of the first character in text
- Syntax: **$CODE(text)**
- Example: **%CALC{"$CODE(abc)"}%** returns **97**
- Related: $CHAR()

COLUMN(offset) -- current column number

- The current table column number with an optional offset
- Syntax: **$COLUMN(*offset*)**
- Example: **%CALC{"$COLUMN()"}%** returns **2** for the second column
- Related: $ROW(), $T()

COUNTITEMS(list) -- count individual items in a list

- Syntax: **$COUNTITEMS(list)**
- Example: **%CALC{"$COUNTITEMS($ABOVE())"}%** returns **Closed: 1, Open: 2** assuming one cell above the current cell contains **Closed** and two cells contain **Open**
- Related: $COUNTSTR(), $LIST()

COUNTSTR(list, str) -- count the number of cells in a list equal to a given string

- Count the number of cells in a list equal to a given string (if str is specified), or counts the number of non empty cells in a list
- Syntax: **$COUNTSTR(list, *str*)**
- Example: **%CALC{"$COUNTSTR($ABOVE())"}%** counts the number of non empty cells above the current cell
- Example: **%CALC{"$COUNTSTR($ABOVE(), DONE)"}%** counts the number of cells equal to **DONE**
- Related: $COUNTITEMS(), $LIST()

DEF(list) -- find first non-empty list item or cell

- Returns the first list item or cell reference that is not empty
- Syntax: **$DEF(list)**
- Example: **%CALC{"$DEF(R1:C1..R1:C3)"}%**
- Related: $COUNTSTR(), $LISTIF(), $LIST()

EMPTY(text) -- test for empty text

- Returns **1** if text is empty, or **0** if not
- Syntax: **$EMPTY(*text*)**
- Example: **%CALC{"$EMPTY(foo)"}%** returns **0**
- Example: **%CALC{"$EMPTY()"}%** returns **1**
- Example: **%CALC{"$EMPTY($TRIM( ))"}%** returns **1**
- Related: $EXACT(), $IF(), $TRIM()

EVAL(formula) -- evaluate a simple mathematical formula

- Addition, substraction, multiplication, division and modulus of numbers are supported. Any nesting is permitted
- Numbers may be decimal integers (1234), binary integers (0b1110011), octal integers (01234), hexadecimal integers (0x1234) or of exponential notation (12.34e-56)
- Syntax: **$EVAL(formula)**
- Example: **%CALC{"$EVAL((5 * 3) / 2 + 1.1)"}%** returns **8.6**
- Related: $EXEC(), $INT(), $MOD(), $ROUND(), $VALUE()

EVEN(num) -- test for even number

- Syntax: **`$EVEN( num )`**
- Example: **`%CALC{"$EVEN(2)"}%`** returns **`1`**
- Related: `$ABS()`, `$MOD()`, `$ODD()`, `$SIGN()`

EXACT(text1, text2) -- compare two text strings

- Compares two text strings and returns **`1`** if they are exactly the same, or **`0`** if not
- Syntax: **`$EXACT( text1, text2 )`**
- Example: **`%CALC{"$EXACT(foo, Foo)"}%`** returns **`0`**
- Example: **`%CALC{"$EXACT(foo, $LOWER(Foo))"}%`** returns **`1`**
- Related: `$EMPTY()`, `$IF()`, `$TRIM()`

EXEC(formula) -- execute a spreadsheet formula

- Execute a spreadsheet formula, typically retrieved from a variable. This can be used to store a formula in a variable once and execute it many times using different parameters.
- Syntax: **`$EXEC( formula )`**
- Example: **`%CALC{"$SET(msg, $NOEXEC(Hi $GET(name)))"}%`** sets the `msg` variable with raw formula `Hi $GET(name)`
- Example: **`%CALC{"$SET(name, Tom) $EXEC($GET(msg))"}%`** executes content of `msg` variable and returns `Hi Tom`
- Example: **`%CALC{"$SET(name, Jerry) $EXEC($GET(msg))"}%`** returns `Hi Jerry`
- Related: `$EVAL()`, `$GET()`, `$NOEXEC()`, `$SET()`

EXISTS(topic) -- check if topic exists

- Topic can be `TopicName` or a `Web.TopicName`. Current web is used if web is not specified.
- Syntax: **`$EXISTS( topic )`**
- Example: **`%CALC{"$EXISTS(WebHome)"}%`** returns **`1`**
- Example: **`%CALC{"$EXISTS(ThisDoesNotExist)"}%`** returns **`0`**
- Related: `$EXACT()`, `$IF()`, `$TRIM()`

EXP(num) -- exponent (e) raised to the power of a number

- EXP is the inverse of the LN function
- Syntax: **`$EXP( num )`**
- Example: **`%CALC{"$EXP(1)"}%`** returns **`2.71828182845905`**
- Related: `$LN()`, `$LOG()`

FIND(string, text, start) -- find one string within another string

- Finds one text `string`, within another `text`, and returns the number of the starting position of `string`, from the first character of `text`. This search is case sensitive and is not a regular expression search; use `$SEARCH()` for regular expression searching. Starting position is 1; a 0 is returned if nothing is matched.
- Syntax: **`$FIND( string, text, start )`**

- Example: **%CALC{"$FIND(f, fluffy)"}%** returns **1**
- Example: **%CALC{"$FIND(f, fluffy, 2)"}%** returns **4**
- Example: **%CALC{"$FIND(@, fluffy, 1)"}%** returns **0**
- Related: $INSERTSTRING(), $LEFTSTRING(), $REPLACE(), $RIGHTSTRING(), $SUBSTRING(), $SEARCH()

FORMAT(type, precision, number) -- format a number to a certain type and precision

- Supported type:
  - COMMA for comma format, such as 12,345.68
  - DOLLAR for Dollar format, such as $12,345.68
  - KB for Kilo Byte format, such as 1205.63 KB
  - MB for Mega Byte format, such as 1.18 MB
  - KBMB for Kilo/Mega/Giga/Tera Byte auto-adjust format
  - NUMBER for number, such as 12345.7
  - PERCENT for percent format, such as 12.3%
- The precision indicates the the number of digits after the dot
- Syntax: **$FORMAT(type, prec, number)**
- Example: **%CALC{"$FORMAT(COMMA, 2, 12345.6789)"}%** returns **12,345.68**
- Example: **%CALC{"$FORMAT(DOLLAR, 2, 12345.67)"}%** returns **$12,345.67**
- Example: **%CALC{"$FORMAT(KB, 2, 1234567)"}%** returns **1205.63 KB**
- Example: **%CALC{"$FORMAT(MB, 2, 1234567)"}%** returns **1.18 MB**
- Example: **%CALC{"$FORMAT(KBMB, 2, 1234567)"}%** returns **1.18 MB**
- Example: **%CALC{"$FORMAT(KBMB, 2, 1234567890)"}%** returns **1.15 GB**
- Example: **%CALC{"$FORMAT(NUMBER, 1, 12345.67)"}%** returns **12345.7**
- Example: **%CALC{"$FORMAT(PERCENT, 1, 0.1234567)"}%** returns **12.3%**
- Related: $FORMATTIME(), $FORMATTIMEDIFF(), $ROUND()

FORMATGMTIME(serial, text) -- convert a serialized date into a GMT date string

- The date string represents the time in Greenwich time zone. Same variable expansion as in $FORMATTIME().
- Syntax: **$FORMATGMTIME(serial, text)**
- Example: **%CALC{"$FORMATGMTIME(1041379200, $day $mon $year)"}%** returns **01 Jan 2003**
- Related: $FORMATTIME(), $FORMATTIMEDIFF(), $TIME(), $TIMEADD(), $TIMEDIFF(), $TODAY()

FORMATTIME(serial, text) -- convert a serialized date into a date string

- The following formatting tokens in text are expanded: $second (seconds, 00..59); $minute (minutes, 00..59); $hour (hours, 00..23); $day (day of month, 01..31); $month (month, 01..12); $mon (month in text format, Jan..Dec); $year (4 digit year, 1999); $ye (2 digit year, 99), $wd (day number of the week, 1 for Sunday, 2 for Monday, etc), $wday (day of the week, Sun..Sat), $weekday (day of the week, Sunday..Saturday), $yearday (day of the year, 1..365, or 1..366 in leap years). Date is assumed to be server time; add GMT to indicate Greenwich time zone.
- Syntax: **$FORMATTIME(serial, text)**

- Example: **%CALC{"$FORMATTIME(0, $year/$month/$day GMT)"}%** returns **1970/01/01 GMT**
- Related: $FORMATGMTIME(), $TIME(), $FORMATTIMEDIFF(), $TIMEADD(), $TIMEDIFF(), $TODAY()

FORMATTIMEDIFF(unit, precision, time) -- convert elapsed time to a string

- Convert elapsed time to a human readable format, such as: 12 hours and 3 minutes
- The input unit can be second, minute, hour, day, month, year. Note: An approximation is used for month and year calculations.
- The precision indicates the number of output units to use
- Syntax: **$FORMATTIMEDIFF(unit, precision, time)**
- Example: **%CALC{"$FORMATTIMEDIFF(min, 1, 200)"}%** returns **3 hours**
- Example: **%CALC{"$FORMATTIMEDIFF(min, 2, 200)"}%** returns **3 hours and 20 minutes**
- Example: **%CALC{"$FORMATTIMEDIFF(min, 1, 1640)"}%** returns **1 day**
- Example: **%CALC{"$FORMATTIMEDIFF(min, 2, 1640)"}%** returns **1 day and 3 hours**
- Example: **%CALC{"$FORMATTIMEDIFF(min, 3, 1640)"}%** returns **1 day, 3 hours and 20 minutes**
- Related: $FORMATTIME(), $TIME(), $TIMEADD(), $TIMEDIFF()

GET(name) -- get the value of a previously set variable

- Specify the variable name (alphanumeric characters and underscores). An empty string is returned if the variable does not exist. Use $SET() to set a variable first. Unlike table ranges, variables live for the time of the page view and persist across tables, i.e. you can use it to summarize results across several tables.
- Syntax: **$GET(name)**
- Example: **%CALC{"$GET(my_total)"}%** returns the value of the my_total variable
- Related: $EXEC(), $NOEXEC(), $SET(), $SETIFEMPTY(), $SETM()

IF(condition, value if true, value if 0) -- return a value based on a condition

- The condition can be a number (where **0** means condition not met), or two numbers with a comparison operator **<** (less than), **<=** (less than or equal), **==** (equal), **!=** (not equal), **>=** (greater than or equal), **>** (greater than).
- Syntax: **$IF(condition, value if true, value if 0)**
- Example: **%CALC{"$IF($T(R1:C5) > 1000, Over Budget, OK)"}%** returns **Over Budget** if value in R1:C5 is over 1000, **OK** if not
- Example: **%CALC{"$IF($EXACT($T(R1:C2),), empty, $T(R1:C2))"}%** returns the content of R1:C2 or **empty** if empty
- Example: **%CALC{"$SET(val, $IF($T(R1:C2) == 0, zero, $T(R1:C2)))"}%** sets a variable conditionally
- Related: $AND(), $EMPTY(), $EXACT(), $LISTIF(), $NOT(), $OR()

INSERTSTRING(text, start, new) -- insert a string into a text string

- Insert `new` string into text string `text` to the right of `start` position. Position starts at 0 (insert before first character). Use a negative `start` to count from the end of the text. If `start` is greater than the length of the `text` the original text is returned.
- Syntax: **`$INSERTSTRING( text, start, new )`**
- Example: **`%CALC{"$INSERTSTRING(abcdefg, 2, XYZ)"}%`** returns **`abXYZcdefg`**
- Example: **`%CALC{"$INSERTSTRING(abcdefg, -2, XYZ)"}%`** returns **`abcdeXYZfg`**
- Related: `$FIND()`, `$LEFTSTRING()`, `$REPLACE()`, `$RIGHTSTRING()`, `$SEARCH()`, `$SUBSTITUTE()`, `$SUBSTRING()`, `$TRANSLATE()`

INT(formula) -- evaluate formula and round down to nearest integer

- Addition, substraction, multiplication, division and modulus of numbers are supported. Any nesting is permitted
- Numbers may be decimal integers (`1234`), binary integers (`0b1110011`), octal integers (`01234`), hexadecimal integers (`0x1234`) or of exponential notation (`12.34e-56`)
- If you expect a single decimal integer value with leading zeros, use **`$INT( $VALUE( number ) )`**
- Syntax: **`$INT( formula )`**
- Example: **`%CALC{"$INT(10 / 4)"}%`** returns **`2`**
- Example: **`%CALC{"$INT($VALUE(09))"}%`** returns **`9`**
- Related: `$EVAL()`, `$ROUND()`, `$VALUE()`

LEFT( ) -- address range of cells to the left of the current cell

- Syntax: **`$LEFT( )`**
- Example: **`%CALC{"$SUM($LEFT())"}%`** returns the sum of cells to the left of the current cell
- Related: `$ABOVE()`, `$RIGHT()`

LEFTSTRING(text, num) -- extract characters at the beginning of a text string

- Retrieve the `num` of characters from the left end of `text`. The leftmost character is returned if `num` is missing. If `num` is greater than the length of `text` the entire text string is returned with no additional spaces added. If `num` is negative `num` characters are removed from the end of the string. If `num` is zero or `num` is a negative value with the number greater than the length of `text` an empty string is returned.
- Syntax: **`$LEFTSTRING( text, num )`**
- Example: **`%CALC{"$LEFTSTRING(abcdefg)"}%`** returns **`a`**
- Example: **`%CALC{"$LEFTSTRING(abcdefg, 0)"}%`** returns an empty string.
- Example: **`%CALC{"$LEFTSTRING(abcdefg, 5)"}%`** returns **`abcde`**
- Example: **`%CALC{"$LEFTSTRING(abcdefg, 12)"}%`** returns **`abcdefg`**
- Example: **`%CALC{"$LEFTSTRING(abcdefg, -3)"}%`** returns **`abcd`**
- Example: **`%CALC{"$LEFTSTRING(abcdefg, -12)"}%`** returns an empty string.
- Related: `$FIND()`, `$INSERTSTRING()`, `$REPLACE()`, `$RIGHTSTRING()`, `$SEARCH()`, `$SUBSTITUTE()`, `$SUBSTRING()`, `$TRANSLATE()`

LENGTH(text) -- length of text in bytes

- Syntax: **$LENGTH(text)**
- Example: **%CALC{"$LENGTH(abcd)"}%** returns **4**
- Related: $LISTSIZE()

LIST(range) -- convert content of a cell range into a list

- Convert the content of a range of cells into a flat list, delimited by comma. Cells containing commas are merged into the list
- Syntax: **$LIST(range)**
- Example: **%CALC{"$LIST($LEFT())"}%** returns **Apples, Lemons, Oranges, Kiwis** assuming the cells to the left contain **| Apples | Lemons, Oranges | Kiwis |**
- Related: $AVERAGE(), $COUNTITEMS(), $COUNTSTR(), $DEF(), $LISTIF(), $LISTITEM(), $LISTJOIN(), $LISTMAP(), $LISTRAND(), $LISTREVERSE(), $LISTSHUFFLE(), $LISTSIZE(), $LISTSORT(), $LISTTRUNCATE(), $LISTUNIQUE(), $MAX(), $MEDIAN(), $MIN(), $PRODUCT(), $SUM(), $SUMDAYS(), $SUMPRODUCT()

LISTIF(condition, list) -- remove elements from a list that do not meet a condition

- In addition to the condition described in $IF(), you can use **$item** to indicate the current element, and **$index** for the list index, starting at 1
- Syntax: **$LISTIF(condition, list)**
- Example: **%CALC{"$LISTIF($item > 12, 14, 7, 25)"}%** returns **14, 25**
- Example: **%CALC{"$LISTIF($NOT($EXACT($item,)), A, B, , E)"}%** returns non-empty elements **A, B, E**
- Example: **%CALC{"$LISTIF($index > 2, A, B, C, D)"}%** returns **C, D**
- Related: $EMPTY(), $EXACT(), $IF(), $LIST(), $LISTITEM(), $LISTMAP(), $LISTREVERSE(), $LISTSIZE(), $LISTSORT(), $LISTUNIQUE(), $SUM()

LISTITEM(index, list) -- get one element of a list

- Index is 1 to size of list; use a negative number to count from the end of the list
- Syntax: **$LISTITEM(index, list)**
- Example: **%CALC{"$LISTITEM(2, Apple, Orange, Apple, Kiwi)"}%** returns **Orange**
- Example: **%CALC{"$LISTITEM(-1, Apple, Orange, Apple, Kiwi)"}%** returns **Kiwi**
- Related: $COUNTITEMS(), $COUNTSTR(), $LIST(), $LISTIF(), $LISTMAP(), $LISTRAND(), $LISTREVERSE(), $LISTSIZE(), $LISTSORT(), $LISTUNIQUE(), $SUM()

LISTJOIN(separator, list) -- convert a list into a string

- By default, list items are separated by a comma and a space. Use this function to indicate a specific separator string, which may include $comma for comma, $n for newline, $sp for space, and $nop for no separator between list items.
- Syntax: **$LISTJOIN(separator, list)**
- Example: **%CALC{"$LISTJOIN($n, Apple, Orange, Apple, Kiwi)"}%** returns the four items separated by new lines

- Related: `$LIST()`, `$LISTSIZE()`

LISTMAP(formula, list) -- evaluate and update each element of a list

- In the formula you can use **`$item`** to indicate the element; **`$index`** to show the index of the list, starting at 1. If **`$item`** is omitted, the item is appended to the formula.
- Syntax: **`$LISTMAP( formula, list )`**
- Example: **`%CALC{"$LISTMAP($index: $EVAL(2 * $item), 3, 5, 7, 11)"}%`** returns **`1: 6, 2: 10, 3: 14, 4: 22`**
- Related: `$COUNTITEMS()`, `$COUNTSTR()`, `$LIST()`, `$LISTIF()`, `$LISTITEM()`, `$LISTREVERSE()`, `$LISTSIZE()`, `$LISTSORT()`, `$LISTUNIQUE()`, `$SUM()`

LISTRAND(list) -- get one random element of a list

- Syntax: **`$LISTRAND( list )`**
- Example: **`%CALC{"$LISTRAND(Apple, Orange, Apple, Kiwi)"}%`** returns one of the four elements
- Related: `$COUNTITEMS()`, `$COUNTSTR()`, `$LIST()`, `$LISTIF()`, `$LISTITEM()`, `$LISTMAP()`, `$LISTSHUFFLE()`, `$LISTSIZE()`, `$LISTSORT()`, `$LISTUNIQUE()`, `$RAND()`, `$SUM()`

LISTREVERSE(list) -- opposite order of a list

- Syntax: **`$LISTREVERSE( list )`**
- Example: **`%CALC{"$LISTREVERSE(Apple, Orange, Apple, Kiwi)"}%`** returns **`Kiwi, Apple, Orange, Apple`**
- Related: `$COUNTITEMS()`, `$COUNTSTR()`, `$LIST()`, `$LISTIF()`, `$LISTITEM()`, `$LISTMAP()`, `$LISTSIZE()`, `$LISTSORT()`, `$LISTUNIQUE()`, `$SUM()`

LISTSIZE(list) -- number of elements in a list

- Syntax: **`$LISTSIZE( list )`**
- Example: **`%CALC{"$LISTSIZE(Apple, Orange, Apple, Kiwi)"}%`** returns **`4`**
- Related: `$COUNTITEMS()`, `$COUNTSTR()`, `$LIST()`, `$LISTIF()`, `$LISTITEM()`, `$LISTJOIN()`, `$LISTMAP()`, `$LISTREVERSE()`, `$LISTSORT()`, `$LISTTRUNCATE()`, `$LISTUNIQUE()`, `$SUM()`

LISTSHUFFLE(list) -- shuffle element of a list in random order

- Syntax: **`$LISTSHUFFLE( list )`**
- Example: **`%CALC{"$LISTSHUFFLE(Apple, Orange, Apple, Kiwi)"}%`** returns the four elements in random order
- Related: `$COUNTITEMS()`, `$COUNTSTR()`, `$LIST()`, `$LISTIF()`, `$LISTITEM()`, `$LISTMAP()`, `$LISTRAND()`, `$LISTSIZE()`, `$LISTSORT()`, `$LISTUNIQUE()`, `$RAND()`, `$SUM()`

LISTSORT(list) -- sort a list

- Sorts a list in ASCII order, or numerically if all elements are numeric
- Syntax: **$LISTSORT(list)**
- Example: **%CALC{"$LISTSORT(Apple, Orange, Apple, Kiwi)"}%** returns **Apple, Apple, Kiwi, Orange**
- Related: $COUNTITEMS(), $COUNTSTR(), $LIST(), $LISTIF(), $LISTITEM(), $LISTMAP(), $LISTREVERSE(), $LISTSHUFFLE(), $LISTSIZE(), $LISTUNIQUE(), $SUM()

LISTTRUNCATE(size, list) -- truncate list to size

- Specify the desired size of the list; use a negative number to count from the end of the list
- Syntax: **$LISTTRUNCATE(size, list)**
- Example: **%CALC{"$LISTTRUNCATE(2, Apple, Orange, Kiwi)"}%** returns **Apple, Orange**
- Related: $COUNTITEMS(), $COUNTSTR(), $LIST(), $LISTIF(), $LISTITEM(), $LISTMAP(), $LISTSIZE(), $LISTSORT(), $LISTUNIQUE(), $SUM()

LISTUNIQUE(list) -- remove all duplicates from a list

- Syntax: **$LISTUNIQUE(list)**
- Example: **%CALC{"$LISTUNIQUE(Apple, Orange, Apple, Kiwi)"}%** returns **Apple, Orange, Kiwi**
- Related: $COUNTITEMS(), $COUNTSTR(), $LIST(), $LISTIF(), $LISTITEM(), $LISTMAP(), $LISTREVERSE(), $LISTSIZE(), $LISTSORT(), $SUM()

LN(num) -- natural logarithm of a number

- LN is the inverse of the EXP function
- Syntax: **$LN(num)**
- Example: **%CALC{"$LN(10)"}%** returns **2.30258509299405**
- Related: $EXP(), $LOG()

LOG(num, base) -- logarithm of a number to a given base

- base-10 logarithm of a number (if base is 0 or not specified), else logarithm of a number to the given base
- Syntax: **$LOG(num, *base*)**
- Example: **%CALC{"$LOG(1000)"}%** returns **3**
- Example: **%CALC{"$LOG(16, 2)"}%** returns **4**
- Related: $EXP(), $LN()

LOWER(text) -- lower case string of a text

- Syntax: **$LOWER(text)**
- Example: **%CALC{"$LOWER($T(R1:C5))"}%** returns the lower case string of the text in cell **R1:C5**

- Related: $PROPER(), $PROPERSPACE(), $TRIM(), $UPPER()

MAX(list) - biggest value of a list or range of cells

- Syntax: **$MAX(list)**
- Example: To find the biggest number to the left of the current cell, write: **%CALC{"$MAX($LEFT())"}%**
- Related: $LIST(), $MEDIAN(), $MIN(), $PERCENTILE()

MEDIAN(list) -- median of a list or range of cells

- Syntax: **$MEDIAN(list)**
- Example: **%CALC{"$MEDIAN(3, 9, 4, 5)"}%** returns **4.5**
- Related: $LIST(), $MAX(), $MIN(), $PERCENTILE()

MIN(list) -- smallest value of a list or range of cells

- Syntax: **$MIN(list)**
- Example: **%CALC{"$MIN(15, 3, 28)"}%** returns **3**
- Related: $LIST(), $MAX(), $MEDIAN(), $PERCENTILE()

MOD(num, divisor) -- reminder after dividing num by divisor

- Syntax: **$MOD(num, divisor)**
- Example: **%CALC{"$MOD(7, 3)"}%** returns **1**
- Related: $EVAL()

NOEXEC(formula) -- do not execute a spreadsheet formula

- Prevent a formula from getting executed. This is typically used to store a raw formula in a variable for later use as described in $EXEC().
- Syntax: **$NOEXEC(formula)**
- Example: **%CALC{"$SET(msg, $NOEXEC(Hi $GET(name)))"}%** sets the msg variable with the formula Hi $GET(name) without executing it
- Related: $EVAL(), $EXEC(), $GET(), $SET()

NOP(text) -- no-operation

- Useful to change the order of Plugin execution. For example, it allows preprossing to be done before %SEARCH{}% is evaluated. The percent character '%' can be escaped with $per
- Syntax: **$NOP(text)**

NOT(num) -- reverse logic of a number

- Returns 0 if **num** is not zero, 1 if zero
- Syntax: **$NOT(num)**

- Example: **%CALC{"$NOT(0)"}%** returns **1**
- Related: $AND(), $EMPTY(), $IF(), $OR()

ODD(num) -- test for odd number

- Syntax: **$ODD(num)**
- Example: **%CALC{"$ODD(2)"}%** returns **0**
- Related: $ABS(), $EVEN(), $MOD(), $SIGN()

OR(list) -- logical OR of a list

- Syntax: **$OR(list)**
- Example: **%CALC{"$OR(1, 0, 1)"}%** returns **1**
- Related: $AND(), $IF(), $NOT()

PERCENTILE(num, list) -- percentile of a list or range of cells

- Calculates the num-th percentile, useful to establish a threshold of acceptance. num is the percentile value, range 0..100
- Syntax: **$PERCENTILE(num, list)**
- Example: **%CALC{"$PERCENTILE(75, 400, 200, 500, 100, 300)"}%** returns **450**
- Related: $LIST(), $MAX(), $MEDIAN(), $MIN()

PI( ) -- mathematical constant Pi, 3.14159265358979

- Syntax: **$PI( )**
- Example: **%CALC{"$PI()"}%** returns **3.14159265358979**

PRODUCT(list) -- product of a list or range of cells

- Syntax: **$PRODUCT(list)**
- Example: To calculate the product of the cells to the left of the current one use **%CALC{"$PRODUCT($LEFT())"}%**
- Related: $LIST(), $PRODUCT(), $SUM(), $SUMPRODUCT()

PROPER(text) -- properly capitalize text

- Capitalize letters that follow any character other than a letter; convert all other letters to lowercase letters
- Syntax: **$PROPER(text)**
- Example: **%CALC{"$PROPER(a small STEP)"}%** returns **A Small Step**
- Example: **%CALC{"$PROPER(f1 (formula-1))"}%** returns **F1 (Formula-1)**
- Related: $LOWER(), $PROPERSPACE(), $TRIM(), $UPPER()

PROPERSPACE(text) -- properly space out WikiWords

- Properly spaces out WikiWords preceeded by white space, parenthesis, or **`][`**. Words listed in the DONTSPACE DefaultPreferences variable or DONTSPACE Plugins setting are excluded
- Syntax: **`$PROPERSPACE( text )`**
- Example: Assuming DONTSPACE contains MacDonald: **`%CALC{"$PROPERSPACE(Old MacDonald had a ServerFarm, EeEyeEeEyeOh)"}%`** returns **`Old MacDonald had a Server Farm, Ee Eye Ee Eye Oh`**
- Related: `$LOWER()`, `$PROPER()`, `$TRIM()`, `$UPPER()`

RAND(max) -- random number

- Random number, evenly distributed between 0 and **`max`**, or 0 and 1 if max is not specified
- Syntax: **`$RAND( max )`**
- Related: `$EVAL()`, `$LISTRAND()`, `$LISTSHUFFLE()`

REPEAT(text, num) -- repeat text a number of times

- Syntax: **`$REPEAT( text, num )`**
- Example: **`%CALC{"$REPEAT(/\, 5)"}%`** returns **`/\/\/\/\/\`**

REPLACE(text, start, num, new) -- replace part of a text string

- Replace `num` number of characters of text string `text`, starting at `start`, with new text `new`. Starting position is 1; use a negative `start` to count from the end of the text
- Syntax: **`$REPLACE( text, start, num, new )`**
- Example: **`%CALC{"$REPLACE(abcdefghijk, 6, 5, *)"}%`** returns **`abcde*k`**
- Related: `$FIND()`, `$INSERTSTRING()`, `$LEFTSTRING()`, `$RIGHTSTRING()`, `$SEARCH()`, `$SUBSTITUTE()`, `$SUBSTRING()`, `$TRANSLATE()`

RIGHT( ) -- address range of cells to the right of the current cell

- Syntax: **`$RIGHT( )`**
- Example: **`%CALC{"$SUM($RIGHT())"}%`** returns the sum of cells to the right of the current cell
- Related: `$ABOVE()`, `$LEFT()`

RIGHTSTRING(text, num) -- extract characters at the end of a text string. `num` must be a positive number. Negative values of `num` are interpreted as zero. If `num` is larger than the length of the text the entire text is returned with no additional spaces.

- Retrieve the `num` of characters from the right end of `text`. The rightmost character is returned if `num` is missing.
- Syntax: **`$RIGHTSTRING( text, num )`**

- Example: **`%CALC{"$RIGHTSTRING(abcdefg)"}%`** returns **`g`**
- Example: **`%CALC{"$RIGHTSTRING(abcdefg, 0)"}%`** returns an empty string

- Example: **%CALC{"$RIGHTSTRING(abcdefg, 5)"}%** returns **cdefg**
- Example: **%CALC{"$RIGHTSTRING(abcdefg, 10)"}%** returns **abcdefg**
- Example: **%CALC{"$RIGHTSTRING(abcdefg, -2)"}%** returns an empty string

- Related: $FIND(), $INSERTSTRING(), $LEFTSTRING(), $REPLACE(), $SEARCH(), $SUBSTITUTE(), $SUBSTRING(), $TRANSLATE()

ROUND(formula, digits) -- round a number

- Evaluates a simple **formula** and rounds the result up or down to the number of digits if **digits** is positive; to the nearest integer if digits is missing; or to the left of the decimal point if digits is negative
- Syntax: **$ROUND(formula, digits)**
- Example: **%CALC{"$ROUND(3.15, 1)"}%** returns **3.2**
- Example: **%CALC{"$ROUND(3.149, 1)"}%** returns **3.1**
- Example: **%CALC{"$ROUND(-2.475, 2)"}%** returns **-2.48**
- Example: **%CALC{"$ROUND(34.9, -1)"}%** returns **30**
- Related: $INT(), $FORMAT()

ROW(offset) -- current row number

- The current table row number with an optional offset
- Syntax: **$ROW(*offset*)**
- Example: To get the number of rows excluding table heading (first row) and summary row (last row you are in), write: **%CALC{"$ROW(-2)"}%**
- Related: $COLUMN(), $T()

SEARCH(string, text, start) -- search a string within a text

- Finds one text string, within another text, and returns the number of the starting position of string, from the first character of text. This search is a RegularExpression search; use $FIND() for non-regular expression searching. Starting position is 1; a 0 is returned if nothing is matched
- Syntax: **$SEARCH(string, text, *start*)**
- Example: **%CALC{"$SEARCH([uy], fluffy)"}%** returns **3**
- Example: **%CALC{"$SEARCH([uy], fluffy, 4)"}%** returns **6**
- Example: **%CALC{"$SEARCH([abc], fluffy,)"}%** returns **0**
- Related: $FIND(), $INSERTSTRING(), $LEFTSTRING(), $REPLACE(), $RIGHTSTRING(), $SUBSTRING()

SET(name, value) -- set a variable for later use

- Specify the variable name (alphanumeric characters and underscores) and the value. The value may contain a formula; formulae are evaluated before the variable assignment; see $NOEXEC() if you want to prevent that. This function returns no output. Use $GET() to retrieve variables. Unlike table ranges, variables live for the time of the page view and persist across tables, i.e. you can use it to summarize results across several tables and also across included topics
- Syntax: **$SET(name, value)**
- Example: **%CALC{"$SET(my_total, $SUM($ABOVE()))"}%** sets the my_total variable to the sum of all table cells located above the current cell and returns an empty string

RIGHTSTRING(text, num) -- extract characters at the end of a textstring. num must be a positive number.

- Related: `$EXEC()`, `$GET()`, `$NOEXEC()`, `$SETIFEMPTY()`, `SETM()`

SETIFEMPTY(name, value) -- set a variable only if empty

- Specify the variable name (alphanumeric characters and underscores) and the value.
- Syntax: **`$SETIFEMPTY( name, value )`**
- Example: **`%CALC{"$SETIFEMPTY(result, default)"}%`** sets the `result` variable to `default` if the variable is empty or 0; in any case an empty string is returned
- Related: `$GET()`, `$SET()`

SETM(name, formula) -- update an existing variable based on a formula

- Specify the variable name (alphanumeric characters and underscores) and the formula. The formula must start with an operator to **+** (add), **-** (subtract), ***** (multiply), or **/** (divide) something to the variable. This function returns no output. Use `$GET()` to retrieve variables
- Syntax: **`$SETM( name, formula )`**
- Example: **`%CALC{"$SETM(total, + $SUM($LEFT()))"}%`** adds the sum of all table cells on the left to the `total` variable, and returns an empty string
- Related: `$GET()`, `$SET()`, `$SETIFEMPTY()`

SIGN(num) -- sign of a number

- Returns -1 if **`num`** is negative, 0 if zero, or 1 if positive
- Syntax: **`$SIGN( num )`**
- Example: **`%CALC{"$SIGN(-12.5)"}%`** returns **`-1`**
- Related: `$ABS()`, `$EVAL()`, `$EVEN()`, `$INT()`, `$NOT()`, `$ODD()`

SQRT(num) -- square root of a number

- Syntax: **`$SQRT( num )`**
- Example: **`%CALC{"$SQRT(16)"}%`** returns **`4`**

SUBSTITUTE(text, old, new, instance, option) -- substitute text

- Substitutes `new` text for `old` text in a `text` string. `instance` specifies which occurance of `old` you want to replace. If you specify `instance`, only that instance is replaced. Otherwise, every occurance is changed to the new text. A literal search is performed by default; a RegularExpression search if the `option` is set to **`r`**
- Syntax: **`$SUBSTITUTE( text, old, new, instance, option )`**
- Example: **`%CALC{"$SUBSTITUTE(Good morning, morning, day)"}%`** returns **`Good day`**
- Example: **`%CALC{"$SUBSTITUTE(Q2-2002, 2, 3)"}%`** returns **`Q3-3003`**
- Example: **`%CALC{"$SUBSTITUTE(Q2-2002,2, 3, 3)"}%`** returns **`Q2-2003`**
- Example: **`%CALC{"$SUBSTITUTE(abc123def, [0-9], 9, , r)"}%`** returns **`abc999def`**
- Related: `$INSERTSTRING()`, `$LEFTSTRING()`, `$REPLACE()`, `$RIGHTSTRING()`, `$SUBSTRING()`, `$TRANSLATE()`

SUBSTRING(text, start, num) -- extract a substring out of a text string

- Extract `num` number of characters of text string `text`, starting at `start`. Starting position is 1; use a negative `start` to count from the end of the text. If `start` or `num` is zero an empty string is returned. If `num` is greater than the length of the `text` the entire text string is returned without any extra spaces added.
- Syntax: **`$SUBSTRING( text, start, num )`**
- Example: **`%CALC{"$SUBSTRING(abcdefghijk, 3, 5)"}%`** returns **`cdefg`**
- Example: **`%CALC{"$SUBSTRING(abcdefghijk, 3, 20)"}%`** returns **`cdefghijk`**
- Example: **`%CALC{"$SUBSTRING(abcdefghijk, -5, 3)"}%`** returns **`ghi`**
- Related: `$FIND()`, `$INSERTSTRING()`, `$LEFTSTRING()`, `$REPLACE()`, `$RIGHTSTRING()`, `$SEARCH()`, `$SUBSTITUTE()`, `$TRANSLATE()`

SUM(list) -- sum of a list or range of cells

- Syntax: **`$SUM( list )`**
- Example: To sum up column 5 excluding the title row, write **`%CALC{"$SUM(R2:C5..R$ROW(-1):C5)"}%`** in the last row; or simply **`%CALC{"$SUM($ABOVE())"}%`**
- Related: `$LIST()`, `$PRODUCT()`, `$SUMPRODUCT()`, `$WORKINGDAYS()`

SUMDAYS(list) -- sum the days in a list or range of cells

- The total number of days in a list or range of cells containing numbers of hours, days or weeks. The default unit is days; units are indicated by a **`h`**, **`hours`**, **`d`**, **`days`**, **`w`**, **`weeks`** suffix. One week is assumed to have 5 working days, one day 8 hours
- Syntax: **`$SUMDAYS( list )`**
- Example: **`%CALC{"$SUMDAYS(2w, 1, 2d, 4h)"}%`** returns **`13.5`**, the evaluation of `(2*5 + 1 + 2 + 4/8)`
- Related: `$SUM()`, `$TIME()`, `$FORMATTIME()`

SUMPRODUCT(list, list) -- scalar product on ranges of cells

- Syntax: **`$SUMPRODUCT( list, list, `*`list...`*` )`**
- Example: **`%CALC{"$SUMPRODUCT(R2:C1..R4:C1, R2:C5..R4:C5)"}%`** evaluates and returns the result of **`($T(R2:C1) * $T(R2:C5) + $T(R3:C1) * $T(R3:C5) + $T(R4:C1) * $T(R4:C5))`**
- Related: `$LIST()`, `$PRODUCT()`, `$SUM()`

T(address) -- content of a cell

- Syntax: **`$T( address )`**
- Example: **`%CALC{"$T(R1:C5)"}%`** returns the text in cell **`R1:C5`**
- Related: `$COLUMN()`, `$ROW()`

TRANSLATE(text, from, to) -- translate text from one set of characters to another

- The translation is done `from` a set `to` a set, one character by one. The `text` may contain commas; all three parameters are required. In the `from` and `to` parameters you can write `$comma` to escape comma, `$sp` to escape space
- Syntax: **`$TRANSLATE( text, from, to )`**
- Example: **`%CALC{"$TRANSLATE(boom,bm,cl)"}%`** returns **`cool`**
- Example: **`%CALC{"$TRANSLATE(one, two,$comma,;)"}%`** returns **`one; two`**
- Related: `$INSERTSTRING()`, `$LEFTSTRING()`, `$REPLACE()`, `$RIGHTSTRING()`, `$SUBSTRING()`, `$SUBSTITUTE()`

TIME(text) -- convert a date string into a serialized date number

- Serialized date is seconds since the Epoch, e.g. midnight, 01 Jan 1970. Current time is taken if the date string is empty. Supported date formats: `31 Dec 2009`; `31 Dec 2009 GMT`; `31 Dec 2009 LOCAL`; `31 Dec 09`; `31-Dec-2009`; `31/Dec/2009`; `2009/12/31`; `2009-12-31`; `2009/12/31`; `2009/12/31 23:59`; `2009/12/31 - 23:59`; `2009-12-31-23-59`; `2009/12/31 - 23:59:59`; `2009.12.31.23.59.59`. Date is assumed to be GMT unless SPREADSHEETPLUGIN_TIMEISLOCAL is set (default 0). Add `GMT` to force Greenwich time zone. Add `LOCAL` to force the timezone of the server. Note that if you use LOCAL or SPREADSHEETPLUGIN_TIMEISLOCAL is set to 1, dates entered by users on servers placed to the east of Greenwich will be converted to the day before which will often be undesired. It is recommended to keep SPREADSHEETPLUGIN_TIMEISLOCAL = 0 which is the default. |
- Syntax: **`$TIME( text )`**
- Example: **`%CALC{"$TIME(2003/10/14 GMT)"}%`** returns **`1066089600`**
- Related: `$FORMATGMTIME()`, `$FORMATTIME()`, `$FORMATTIMEDIFF()`, `$TIMEADD()`, `$TIMEDIFF()`, `$TODAY()`, `$WORKINGDAYS()`

TIMEADD(serial, value, unit) -- add a value to a serialized date

- The `unit` is seconds if not specified; unit can be `second`, `minute`, `hour`, `day`, `week`, `month`, `year`. Note: An approximation is used for month and year calculations
- Syntax: **`$TIMEADD( serial, value, unit )`**
- Example: **`%CALC{"$TIMEADD($TIME(), 2, week)"}%`** returns the serialized date two weeks from now
- Related: `$FORMATTIME()`, `$FORMATGMTIME()`, `$TIME()`, `$TIMEDIFF()`, `$TODAY()`

TIMEDIFF(serial_1, serial_2, unit) -- time difference between two serialized dates

- The `unit` is seconds if not specified; unit can be specified as in `$TIMEADD()`. Note: An approximation is used for month and year calculations. Use `$FORMAT()`, `$FORMATTIMEDIFF()` or `$INT()` to format real numbers
- Syntax: **`$TIMEDIFF( serial_1, serial_2, unit )`**
- Example: **`%CALC{"$TIMEDIFF($TIME(), $EVAL($TIME()+90), minute)"}%`** returns **`1.5`**
- Related: `$FORMAT()`, `$FORMATGMTIME()`, `$FORMATTIME()`, `$FORMATTIMEDIFF()`, `$INT()`, `$TIME()`, `$TIMEADD()`, `$TODAY()`, `$WORKINGDAYS()`

TODAY( ) -- serialized date of today at midnight GMT

- In contrast, the related `$TIME()` returns the serialized date of today at the current time, e.g. it includes the number of seconds since midnight GMT
- Syntax: **`$TODAY( )`**
- Example: **`%CALC{"$TODAY()"}%`** returns the number of seconds since Epoch
- Related: `$FORMATTIME()`, `$FORMATGMTIME()`, `$TIME()`, `$TIMEADD()`, `$TIMEDIFF()`

TRIM(text) -- trim spaces from text

- Removes all spaces from text except for single spaces between words
- Syntax: **`$TRIM( text )`**
- Example: **`%CALC{"$TRIM( eat  spaces  )"}%`** returns **`eat spaces`**
- Related: `$EMPTY()`, `$EXACT()`, `$PROPERSPACE()`

UPPER(text) -- upper case string of a text

- Syntax: **`$UPPER( text )`**
- Example: **`%CALC{"$UPPER($T(R1:C5))"}%`** returns the upper case string of the text in cell **`R1:C5`**
- Related: `$LOWER()`, `$PROPER()`, `$PROPERSPACE()`, `$TRIM()`

VALUE(text) -- convert text to number

- Extracts a number from **`text`**. Returns **`0`** if not found
- Syntax: **`$VALUE( text )`**
- Example: **`%CALC{"$VALUE(US$1,200)"}%`** returns **`1200`**
- Example: **`%CALC{"$VALUE(PrjNotebook1234)"}%`** returns **`1234`**
- Example: **`%CALC{"$VALUE(Total: -12.5)"}%`** returns **`-12.5`**
- Related: `$EVAL()`, `$INT()`

WORKINGDAYS(serial_1, serial_2) -- working days between two serialized dates

- Working days are Monday through Friday (sorry, Israel!). The start date is not included in the count. The end date is concluded. If you need both included simply subtract one day from the start date.
- Syntax: **`$WORKINGDAYS( serial_1, serial_2 )`**
- Example: **`%CALC{"$WORKINGDAYS($TIME(2004/07/15), $TIME(2004/08/03))"}%`** returns **`13`**
- Related: `$SUMDAYS()`, `$TIME()`, `$TIMEDIFF()`

FAQ

Can I use CALC in a formatted search?

Specifically, how can I output some conditional text in a FormattedSearch?

You need to escape the CALC so that it executes once per search hit. This can be done by escaping the `%` signs of `%CALC{...}%` with `$percnt`. For example, to execute `$IF($EXACT($formfield(Tested), Yes), %PUBURL%/%SYSTEMWEB%/DocumentGraphics/choice-yes.gif, %PUBURL%/%SYSTEMWEB%/DocumentGraphics/choice-no.gif)` in the `format=""` parameter, write this:

```
%SEARCH{ .... format="| $topic |
$percntCALC{$IF($EXACT($formfield(Tested), Yes),
%PUBURL%/%SYSTEMWEB%/DocumentGraphics/choice-yes.gif,
%PUBURL%/%SYSTEMWEB%/DocumentGraphics/choice-no.gif)}$percnt |" }%
```

How can I easily repeat a formula in a table?

To repeat the same formula in all cells of a table row define the formula once in a preferences setting and use that in the CALC. The preferences setting can be hidden in HTML comments. Example:

```
<!--
   * Set MYFORMULA = $EVAL($SUBSTITUTE(...etc...))
-->
| A | 1 | %CALC{%MYFORMULA%}% |
| B | 2 | %CALC{%MYFORMULA%}% |
| C | 3 | %CALC{%MYFORMULA%}% |
```

Bug Tracking Example

Bug#:	Priority:	Subject:	Status:	Days to fix
Total: 4	High: 2 Low: 1 Medium: 1	.	Assigned: 1 Fixed: 2 Open: 1	Total: 11
Bug:1234	High	No arrange ...	Fixed	1
Bug:1233	Medium	Usability issue ...	Assigned	5
Bug:1232	High	Memory Window ...	Fixed	2
Bug:1231	Low	File Open ...	Open	3

The last row is defined as:

```
| Total: %CALC{"$ROW(-2)"}% \
  | %CALC{"$COUNTITEMS( R2:C$COLUMN()..R$ROW(-1):C$COLUMN() )"}% | . \
  | %CALC{"$COUNTITEMS( R2:C$COLUMN()..R$ROW(-1):C$COLUMN() )"}% \
  |  Total: %CALC{"$SUM( R2:C$COLUMN()..R$ROW(-1):C$COLUMN() )"}% |
```

Above table is created manually. Another Plugin could build the table dynamically, e.g. by pulling data out of a bug tracking system. The Spreadsheet Plugin can be used to display table data statistics.

Plugin Settings

You can override some default settings in the plugin by setting the following preferences.

Preference	Meaning	Default
`SPREADSHEETPLUGIN_DEBUG`	Debug plugin: (See output in `data/debug.txt`)	0
`SPREADSHEETPLUGIN_SKIPINCLUDE`	Do not handle `%CALC{}%` variable in included topic while including topic	1
`SPREADSHEETPLUGIN_DONTSPACE`		

Preference	Meaning	Default
	Comma-delimited list of WikiWords to exclude from being spaced out by the **`$PROPERSPACE(text)`** function.	CodeWarrior, MacDonald, McIntosh, RedHat, SuSE
`SPREADSHEETPLUGIN_TIMEISLOCAL`	Makes the TIME function assume input is local time and converts the entered time to GMT unless the date has 'GMT' appended. Note that this behavior creates problems for users using servers in time zones to the east of Greenwich. The setting is present for compatibility.	0

Note that the `DONTSPACE` global preference overrides the `SPREADSHEETPLUGIN_DONTSPACE` preference for historical reasons.

Plugin Installation Instructions

You do not need to install anything in the browser to use this extension. The following instructions are for the administrator who installs the extension on the server.

Open configure, and open the "Extensions" section. Use "Find More Extensions" to get a list of available extensions. Select "Install".

If you have any problems, or if the extension isn't available in `configure`, then you can still install manually from the command-line. See http://foswiki.org/Support/ManuallyInstallingExtensions for more help.

- Test if the "Total" in the first table in this topic is correct.

Plugin Info

Copyright:	Copyright (C) 2001-2007 Peter Thoeny, peter@thoeny.org and TWiki Contributors; © 2008-2009 Foswiki Contributors
License:	GPL (GNU General Public License)
Version:	5484 (2009-11-10)
Release:	10 Nov 2009
Change History:	
29 Dec 2009:	Foswikitask:Item2301: added `$nop` to $LISTJOIN() for better empty parameter
10 Nov 2009:	Added unit tests
27 Oct 2009:	Foswikitask:Item2301: Fixed $LISTJOIN() to accept an empty separator
20 Sep 2009:	Minor documentation update. trunk and release branch code synced (mainly perltidy - all functional changes have been in sync).
17 Sep 2009:	Foswikitask:Item2087: SpreadSheetPlugin forgets about zeros being floats as well
11 May 2009:	Fixed the calculation of WORKINGDAYS. Changed the default behavior of TIME back to not converting dates to GMT as this creates surprising effects for users living to the east of Greenwich. Added SPREADSHEETPLUGIN_TIMEISLOCAL so users depending on the old behavior keep the old behavior if TIME. Added the feature 'local' to TIME so conversion behavior can be used on demand.

22 Apr 2009:	Removed support for settings in the plugin topic which is a bad idea anyway as they get overwritten at each Foswiki upgrade. Define the global settings in Main.SitePreferences instead. Foswikitask:Item5471: Fixed replacing 0 in REPLACE. Fixed FIND/SEARCH handling of empty strings and corrected documentation for SEARCH
29 Mar 2009:	Added $EMPTY(), $LEFTSTRING(), $RIGHTSTRING(), SUBSTRING(), and $INSERTSTRING()
06 Jan 2009:	Foswikitask:Item4835: Allow SUBSTITUTE and REPLACE to return values 0 and ''
16 Dec 2008:	Foswiki version - no new features
13 Oct 2007:	Added $FORMATTIMEDIFF()
09 Sep 2007:	Enhanced documentation for $EVAL() and $INT()
02 Jun 2007:	Added VarCALC to have `%CALC{}%` listed in Macros
14 Apr 2007:	Fixing bug in $EXISTS() that required full `web.topic` instead of just `topic`
11 Mar 2007:	Fixing bug in $VALUE() and $INT(), introduced by version 09 Mar 2007
09 Mar 2007:	Added $EXP(), $LN(), $LOG(), $PI(), $SQRT(); fixed $ROUND() bug, contributed by TWiki:Main/SergejZnamenskij
23 Jan 2007:	Enhanced documentation
18 Dec 2006:	Added $LISTRAND(), $LISTSHUFFLE(), $LISTTRUNCATE(); fixed spurious newline at end of topic, contributed by Foswiki:Main/MichaelDaum
10 Oct 2006:	Enhanced documentation
13 May 2006:	Added $SETIFEMPTY(); fixes in documentation
17 Jun 2005:	Added $NOEXEC(), $EXEC()
25 Mar 2005:	Fixed evaluation bug when using SpeedyCGI accelerator; code refactor to load module only when needed, contributed by Foswiki:Main/CrawfordCurrie
24 Oct 2004:	Added $EXISTS(), contributed by TWiki:Main/RodrigoChandia; added $PERCENTILE()
18 Oct 2004:	Added $LISTJOIN()
26 Sep 2004:	Added $FORMAT(KB), $FORMAT(MB), contributed by Foswiki:Main/ArthurClemens; added $FORMAT(KBMB), $EVEN(), $ODD()
17 Jul 2004:	Added $WORKINGDAYS(), contributed by Foswiki:Main/CrawfordCurrie
24 May 2004:	Refactored documentation (no code changes)
03 Apr 2004:	Added $ABS(), $LISTIF(); fixed $VALUE() to remove leading zeros; changed $FIND() and $SEARCH() to return 0 instead of empty string if no match
21 Mar 2004:	Added $LISTITEM(); fixed call to unofficial function
16 Mar 2004:	Added $LISTMAP(), $LISTREVERSE(), $LISTSIZE(), $LISTSORT(), $LISTUNIQUE(), $SETM(); retired $COUNTUNIQUE() in favor of $COUNTITEMS($LISTUNIQUE()); fixed evaluation order issue of $IF(); fixed missing eval error messages suppressed since version 06 Mar 2004; redirect stderr messages to warning
08 Mar	Added $LIST()

2004:	
06 Mar 2004:	Added $AND(), $MOD(), $NOT(), $OR(), $PRODUCT(), $PROPER(), $PROPERSPACE(), $RAND(), $REPEAT(), $SIGN(), $VALUE(); added digits parameter to $ROUND(); renamed $MULT() to $PRODUCT(); $MULT() is deprecated and undocumented
27 Feb 2004:	Added $COUNTUNIQUE()
24 Oct 2003:	Added $SET(), $GET(), $MEDIAN(); added $SUMPRODUCT(), inspired by TWiki:Main/RobertWithrow; added $SUMDAYS(), contributed by Foswiki:Main/SvenDowideit
21 Oct 2003:	Added support for lists `(1, 2, 3)` and lists of table ranges `(R1:C1..R1:C5, R3:C1..R3:C5)` for all functions that accept a table range; added $TIMEADD(); in $TIMEDIFF() added week unit; in $FORMATTIME() changed $weekday to $wd and added $wday and $weekday
14 Oct 2003:	Added $TIME(), $TODAY(), $FORMATTIME(), $FORMATGMTIME(), $TIMEDIFF()
13 Oct 2003:	Added $MULT(), contributed by TWiki:Main/GerritJanBaarda
30 Jul 2003:	Added $TRANSLATE()
19 Jul 2003:	Added $FIND(), $NOP(), $REPLACE(), $SEARCH(), $SUBSTITUTE(), contributed by TWiki:Main/PaulineCheung
19 Apr 2003:	Added $COUNTSTR(), $EXACT(), $IF(), $ROUND(), $TRIM(); added $FORMAT(), contributed by TWiki:Main/JimStraus; support `%` modulus operator in $EVAL(), $INT(), and $ROUND(); fixed bug in $DEF()
07 Jun 2002:	Added $DEF(), contributed by TWiki:Main/MartinFuzzey; allow values with HTML formatting like `<u>102</u>`, suggested by TWiki:Main/GladeDiviney; added SKIPINCLUDE setting
12 Mar 2002:	Support for multiple functions per nesting level
15 Jan 2002:	Added $CHAR(), $CODE() and $LENGTH()
12 Nov 2001:	Added $RIGHT()
12 Aug 2001:	Fixed bug of disappearing multi-column cells
19 Jul 2001:	Fixed incorrect $SUM() calculation of cell with value `0`
14 Jul 2001:	Changed to plug & play
01 Jun 2001:	Fixed insecure dependencies for $MIN() and $MAX()
16 Apr 2001:	Fixed div by 0 bug in $AVERAGE()
17 Mar 2001:	Initial version with $ABOVE(), $AVERAGE(), $COLUMN(), $COUNTITEMS(), $EVAL(), $INT(), $LEFT(), $LOWER(), $MAX(), $MIN(), $ROW(), $SUM(), $T(), $UPPER()
Plugin Home:	http://foswiki.org/Extensions/SpreadSheetPlugin
Support:	http://foswiki.org/Support/SpreadSheetPlugin

Related Topics: DefaultPreferences, SitePreferences, Plugins, VarCALC

Edit | Attach | Print version | History: %REVISIONS% | Backlinks | Raw View | More topic actions
Topic revision: r0 - 13 Oct 2007 - 03:49:57 - ProjectContributor

- ☐ System

- Log In

- **Toolbox**
- Users
- Groups
- Index
- Search
- Changes
- Notifications
- RSS Feed
- Statistics
- Preferences

- **User Reference**
- BeginnersStartHere
- TextFormattingRules
- Macros
- FormattedSearch
- QuerySearch
- DocumentGraphics
- SkinBrowser
- InstalledPlugins

- **Admin Maintenance**
- Reference Manual
- AdminToolsCategory
- InterWikis
- ManagingWebs
- SiteTools
- DefaultPreferences
- WebPreferences

- **Categories**
- Admin Documentation
- Admin Tools
- Developer Doc
- User Documentation
- User Tools

- **Webs**
- Public
- System

-
-
-

FW FOSWIKI Copyright © by the contributing authors. All material on this site is the property of the contributing authors.
Ideas, requests, problems regarding Wiki? Send feedback