

The Eval Engine has over 100 built in functions which are syntactically similar to Microsoft Excel's® so that it's easy and intuitive for users to use. The follow are the functions which can be included in any Eval-O-Matic or Calc-O-Matic expression. Most parameters are flexible, so the term *number* is used throughout to mean than any number, integer or real, can be used.

ABS

ABS(number) as number

Returns the absolute value of a number.

ACOS

ACOS(number) as number

Returns the arccosine of a number. The argument must be between or including -1 and 1. The result is an angle expressed in the units set in the AngleUnit property.

ACOSH

ACOSH(number) as number

Returns the arc-hyperbolic cosine of a number. The number taken by ACOSH is a hyperbolic cosine of another number, and must be a real number greater than or equal to 1. The return value is the number whose hyperbolic cosine is the number you supply.

AND

AND(logical expression1..n) as boolean

Takes each logical expression separated by commas and performs the AND operator on them. It returns True only if every argument is true..

ASIN

ASIN(number) as number

Returns the arcsine of a number. The argument must be between or including -1 and 1. The result is an angle expressed in the units set in the AngleUnit property.

ASINH

ASINH(number) as number

Returns the arc-hyperbolic sine of a number. The number taken by ASINH is a hyperbolic sine of another number, and can be any real number. The return value is the number whose hyperbolic sine is the number you supply.

ATAN

ATAN(number) as number

Returns the arctangent of a number. The result is an angle expressed in the units set in the AngleUnit property.

ATAN2

ATAN2(X, Y as number) as number

Returns the arctangent of a triangle with sides X and Y. The result is an angle expressed in the units set in the AngleUnit property.

ATANH

ATANH(number) as number

Returns the arc-hyperbolic tangent of a number. The number taken by ATANH is a hyperbolic tangent of another number, and must be a real between, but not including -1 and 1. The return value is the number whose hyperbolic cosine is the number you supply.

AVEDEV

AVEDEV(Series of numbers) as number

Returns the average deviation of data points from their mean.

AVERAGE

AVERAGE(series of numbers) as number

Returns the average of a series of numbers.

BITSLLEFT

BITSLLEFT(Bits, Shift) as integer

Treats an integer as a series of 32 bits and shifts them to the left.

BITSOFF

BITSOFF(Bits, Position as Integer) as Integer

Treats an integer as a series of bits, and turns a specific bit off.

BITSON

BITSON(Bits, Position as Integer) as Integer

Treats an integer as a series of bits, and turns a specific bit on.

BITSRIGHT

BITSRIGHT(Bits, Shift as Integer) as Integer

Treats an integer as a series of 32 bits and shifts them to the right.

CEILING

CEILING(Value, Sig as number) as number

Rounds to the next highest absolute value *multiple* of Sig. Positive Sig means to the right of the decimal, negative to the left. *example*: Ceiling(1.03, 0.05) = 1.05.

CHAR

CHAR(number) as string

Returns the ASCII character for a number 0-255.

CLEAN

CLEAN(String) as String

Removes all non-printable characters from text (ASCII 1-31 and 128-159).

CMFEET

CMFEET(number) as number

Translates CM to Feet.

CMINCH

CMINCH(number) as number

Translates CM to Inches.

CODE

CODE(Character) as integer

Takes a single ASCII character and returns the code number 0-255.

COMBIN

COMBIN(ItemsTotal, ItemsInGroup as Integer) as Integer

Returns the number of combinations of groups you can form. Combin differs from Permut in that the order does not matter.

CONCATENATE

CONCATENATE(series of strings) as string

Joins several strings together into one larger string.

COS

COS(number) as number

Returns the cosine of an angle. The Angle units are set by the AngleUnit property.

COSH

COSH(number) as number

Returns the hyperbolic cosine of a number.

CREDITCARDTYPE

CreditCardType(S as String) as integer

Takes a series of digits as a string, and calculates a checksum that determines if it is a potentially valid credit card number. It returns the first character as a number, and 0 if it is invalid.

CTOF

CtoF(R as double) as double

Translates Centigrade to Fahrenheit.

CTOK

CtoK(R as double) as double

Translates Centigrade to Kelvin.

DB

DB(Cost, Salvage as double, Life, Period, Months as integer) as double

Returns depreciation using the fixed-declining balance method. Cost is the initial cost of an item, Salvage the end value, Life the number of depreciable periods(months), Period the specific period you want the depreciation for, and Months is the number of months in the first year.

DDB

DDB(Cost, Salvage as double, Life, Period, Factor as integer) as double

Returns depreciation using the double-declining balance method, or declining by another factor. Cost is the initial cost of an item, Salvage the end value, Life the number of depreciable periods(months), Period the specific period you want the depreciation for, and Factor is the depreciation factor - use 2 for double.

DEGREES

DEGREES(number) as number

Translates Radians to Degrees.

DOLLAR

DOLLAR(number) as string

Takes a number and formats it as money using windows currency settings.

EVEN

EVEN(number) as number

Rounds to the next highest absolute even number.

EXACT

EXACT(S1, S2 as string) as boolean

Returns true if S1 and S2 are the same. Exact is case-sensitive.

EXP

EXP(number) as number

Returns the mathematical constant "e" raised to a given power.

FACT

FACT(integer) as integer

Returns the factorial of a number.

FALSE

FALSE()

Returns the value FALSE. Same as False, and kept for compatibility reasons.

FEETCM

FEETCM(number) as number

Translates Feet into Centimeters.

FEETM

FEETM(number) as number

Translates Feet into Meters.

FIND

FIND(S1, S2 as String, Start as Integer) as integer

Returns the position of S1 within S2 starting at position Start. Find is case-sensitive.

FLOOR

FLOOR(Value, Sig as number) as number

Rounds to the next lowest absolute value *multiple* of Sig. Positive Sig means to the right of the decimal, negative to the left. *example*: Floor(1.08, 0.05) = 1.05.

FTOC

FTOC(number) as number

Translates Fahrenheit to Centigrade.

FV

FV(Rate as double, Periods as Integer, Payment, InitialVal as double, PaymentType as integer) as double

Returns the future value of an investment. Rate is an interest rate (if periods is months, be sure to divide annual rates by 12). Periods is the number of payment periods. Payment is the amount of fixed payment made each period. InitialVal is the opening balance. PaymentType is 0 if payments are applied at the start of a period, and 1 if applied at the end.

GALLTR

GallTr(number) as number

Translates Gallons to Liters.

IF

IF(Conditional Expression, Value1, Value2) as variant

If the conditional expression is true, then IF returns the value of Value1 which can be numerical or string. If it is false, it returns Value2.

INCM

INCM(number) as number

Translates Inches to Centimeters.

INT

INT(number) as integer

Rounds down to the next lowest whole number.

IPMT

IPMT(Rate as Double, Period, Periods as Integer, InitialVal, FinalVal as Double, PaymentType as Integer) as double

Returns the amount of interest paid for a given period of a loan. Rate is an interest rate (if periods is months, be sure to divide annual rates by 12). Period is the specific period you're calculating for. Periods is the number of payment periods. InitialVal is the opening balance. FinalVal is the amount of the final balance (often zero). PaymentType is 0 if payments are applied at the start of a period, and 1 if applied at the end.

KGPOUND

KGPOUND(number) as number

Translates Kilograms into pounds.

KMMILE

KMMILE(number) as number

Translates Kilometers into Miles.

KURT

KURT(Series of numbers) as number

Returns the Kurtosis of a series of numbers. Maximum number of items is 1024.

LEFT

LEFT(S as string, Len as Integer) as string

Returns the left part of a string.

LEN

LEN(string) as integer

Returns the length of a string.

LN

LN(number) as number

Returns the natural log of a number. The argument must be greater than zero.

LOG

LOG(Num, Base as number) as number

Returns the logarithm of a number. Num and Base must both be greater than zero.

LOG10

Log(number) as number

Returns the logarithm of a number of base 10. Num must be greater than zero.

LOWER

LOWER(string) as string

Returns the string in all lower case letters.

LTRGAL

LTRGAL(number) as number

Translates Liters into Gallons.

MAX

MAX(Series of numbers) as number

Returns the Maximum value from a series of numbers.

MEDIAN

MEDIAN(Series of numbers) as number

Returns the Median value from a series of numbers. Maximum number of items is 1024.

MFEET

MFEET(number) as number

Translates Meters into Feet.

MID

MID(S as string, Start, Len as integer) as string

Returns a subsection of a string starting at Start, of Len characters.

MILEKM

MILEKM(number) as number

Translates Miles into Kilometers.

MIN

MIN(Series of numbers) as number

Returns the Maximum value from a series of numbers.

MLOZ

MLOZ(number) as number

Translates Milliliters to Ounces.

MOD

MOD(number, divisor as number) as number

Divides two numbers and returns only the remainder.

MODE

MODE(Series of numbers) as number

Returns the Mode (most frequently appearing) value from a series of numbers.
Maximum number of items is 1024.

NOT

NOT(boolean) as boolean

Returns the opposite of a boolean expression.

NPER

NPER(Rate, Payment, PresentVal, FutureVal as double, PaymentType as integer) as double

Returns the number of periods of an investment. Rate is an interest rate (if periods is months, be sure to divide annual rates by 12). Payment is the amount of fixed payment made each period. PresentVal is the present value of the investment. FutureVal is the final value of the investment. PaymentType is 0 if payments are applied at the start of a period, and 1 if applied at the end.

NumberToWords

NumberToWords(R as double) as String

Takes a number ranging from n,nnn,nnn,nnn.nn and converts it to words. *example:* `ccNumberToWords(121)` = "One Hundred Twenty One".

ODD

ODD(number) as number

Rounds to the next highest absolute value odd number.

OR

OR(logical expression1..n) as boolean

Takes each logical expression separated by commas and performs the OR operator on them. It returns True if any argument is true..

OZML

OZML(number) as number

Translates Ounces to Milliliters.

PERMUT

Permut(ItemsTotal, ItemsInGroup as Integer) as Integer

Returns the number of permutations of objects you can form. Permut differs from Combin in that the order matters.

PI

PI() as number

Returns the constant pi, approximately 3.14159265358979.

PMT

PMT(Rate as Double, Periods as Integer, PresentVal, FutureVal as double, PaymentType as integer) as double

Returns the periodic payment amount. Rate is an interest rate (if periods is months, be sure to divide annual rates by 12). Periods is the number of payments to be made. PresentVal is the present value of the investment. FutureVal is the final value of the investment. PaymentType is 0 if payments are applied at the start of a period, and 1 if applied at the end.

POUNDKG

PoundKG(number) as number

Translates Pounds to Kilograms.

POWER

POWER(Base, Exponent as number) as number

Raises the Base to the Exponent.

PPMT

PPMT(Rate as Double, Period, Periods as Integer, InitialVal, FinalVal as Double, PaymentType as Integer) as double

Returns the amount of principle paid for a given period of a loan. Rate in an interest rate (if periods is months, be sure to divide annual rates by 12). Period is the specific period you're calculating for. Periods is the number of payment periods. InitialVal is the opening balance. FinalVal is the amount of the final balance (often zero). PaymentType is 0 if payments are applied at the start of a period, and 1 if applied at the end.

PRODUCT

PRODUCT(Series of numbers) as number

Returns the product of a series of numbers. $N1*N2*N3\dots n$

PROPER

Proper(String) as String

Takes a string and changes the capitalization so that the first letter of each word is capitalized and the rest are lowercase.

PV

PV(Rate as double, Periods as Integer, Payment, FutureVal as double, PaymentType as integer) as double

Returns the present value of an investment. Rate in an interest rate (if periods is months, be sure to divide annual rates by 12). Periods is the number of payment periods. Payment is the amount of fixed payment made each period. FutureVal is the final balance. PaymentType is 0 if payments are applied at the start of a period, and 1 if applied at the end.

RADIANS

RADIANS(number) as number

Translates Degrees to Radians.

RAND

RAND([bottom], [top] as number) as number

Returns a random number. It can take from 0 to 2 arguments. With no arguments, the range will be from $0 \leq n < 1$. With a single argument, it will be from $0 \leq n < \text{Top}$. With two arguments, the result will be $\text{bottom} \leq n < \text{top}$.

RATE

Rate(Periods as Integer, Payment, PresentVal, FutureVal as double, PaymentType as integer) as double

Returns the interest rate of an investment. Periods is the number of periods(months) payments are made. Payment is the amount of fixed payment made each period. PresentVal is the present value of the investment. FutureVal is the final value of the investment. PaymentType is 0 if payments are applied at the start of a period, and 1 if applied at the end.

REPLACE

Replace(S1 as string, start, len as integer, S2 as string) as string

Cuts a piece of the string *S1* from *start* of length *len* and replaces it with *S2*.

REPT

REPT(S as string, N as integer) as string

Returns a string composed of *S* repeated *N* times.

RIGHT

RIGHT(S as string, Len as Integer) as string

Returns the right part of a string.

ROMAN

Roman(X, RomanType as Integer) as String

Converts a number to roman numerals, as a string. *X* is an integer from 0 to 3999. *RomanType* can be 0,1,2,3, or 4 and determines how concise the final number is made.

ROUND

ROUND(Num as number, Sig as Integer) as number

Rounds to a specified significant digit. Positive Sig means to the right of the decimal, and negative means to the left.

ROUNDDOWN

ROUNDDOWN(Num as number, Sig as Integer) as number

Rounds down to a specified significant digit. Positive Sig means to the right of the decimal, and negative means to the left.

ROUNDUP

ROUNDUP(Num as number, Sig as Integer) as number

Rounds up to a specified significant digit. Positive Sig means to the right of the decimal, and negative means to the left.

SEARCH

SEARCH(S1, S2 as String, Start as Integer) as integer

Returns the position of S1 within S2 starting at position Start. Search is not case-sensitive.

SIGN

SIGN(number) as number

Returns -1 if the number is negative, and 1 if it is positive.

SIN

SIN(number) as number

Returns the sine of an angle. The Angle units are set by the AngleUnit property.

SINH

SINH(number) as number

Returns the hyperbolic sine of a number.

SLN

SLN(Cost, Salvage as Double, Life as Integer) as Integer

Returns straight line depreciation. Cost is the initial cost of an item. Salvage is the final value. Life is the number of periods to depreciate over (years).

SQFEETSQ

M

SqFeetSqM(number) as number

Translates Square Feet to Square Meters.

SQMSQFEE

T

SqMSqFeet(number) as number

Translates Square Meters to Square Feet.

SQRT

SQRT(number) as number

Returns the square root of a number.

STDEV

STDEV(series of numbers) as number

Returns the standard deviation of a series of numbers based on a sample of data. Maximum number of items in series is 1024.

STDEVP

STDEV(series of numbers) as number

Returns the standard deviation of a series of numbers based on entire population. Maximum number of items in series is 1024.

SUBSTITUT E

Substitute(S, OldPart, NewPart as String, Instance as Integer) as String

Replaces an OldPart of a string with a NewPart where it occurs in another string. It will replace the Instance specified only, or all instances if Instance is zero.

SUM

SUM(series of numbers) as number

Adds a series of numbers together.

SUMSQ

SUMSQ(series of numbers) as number

Squares each element as it adds a series of numbers together.

SYD

SYD(Cost, Salvage as Double, Life, Period as Integer) as Integer

Returns Sum of Years depreciation for a given period. Cost is the initial cost of an item. Salvage is the final value. Life is the number of periods to depreciate over (years). Period is the period you want to calculate for.

TAN

TAN(number) as number

Returns the tangent of an angle. The Angle units are set by the AngleUnit property.

TANH

TANH(number) as number

Returns the hyperbolic tangent of a number.

TRIM

Trim(S as String) as String

Trims all spaces except single spaces from between words.

TRUE

TRUE()

Returns the value TRUE. Same as True, and kept for compatibility reasons.

TRUNC

TRUNC(val as number, sig as integer) as number

Truncates the value to the significant digit specified. Positive Sig means to the right of the decimal, negative to the left.

UPPER

UPPER(string) as string

Returns the string in all upper case letters.

VAR

VAR(series of numbers) as number.

Returns the variance in a set of numbers based on a sample of data. Maximum number of items in series is 1024.

VARP

VARP(series of numbers) as number.

Returns the variance in a set of numbers based on a complete population of data. Maximum number of items in series is 1024.

VDB

VDB(Cost, Salvage as double, Life, StartPeriod, EndPeriod, Factor, Method as integer) as double

Returns depreciation using a variable method. Cost is the initial cost of an item. Salvage is the end value. Life is the number of depreciable periods. StartPeriod and EndPeriod are the starting and ending periods you want the depreciation for. Factor is the depreciation factor - use 2 for double. If method is 0 then it will switch to straight line depreciation when it gives a greater value, and if method is 1 then it will use the specified declining method throughout the calculation.

