List of functions in Google Sheet

List of functions in Google Sheet. Google Sheet is a sheet of Google made to help you use Excel online. Like Excel, Google Sheet can also create different functions for statistical calculations.

Google Sheet is a sheet of Google made with the purpose to help you use "Excel online". Like Excel, Google Sheet can also create different functions for statistical calculations. In this article, TipsMake.com will list the function lists included in Google Sheet.

The list of functions below will be listed by us based on the type of function used.

1. Database function

DAVERAGE function

DAVERAGE (database_data; field; criteria)

Returns the average value of a selected set of values ??from an array or database table-like range by using SQL-like queries.

DCOUNT function

DCOUNT (database_data; field; criteria)

Counts numeric values ??selected from an array or range similar to a database table using SQL-like queries.

DCOUNTA function

DCOUNTA (database_data, school, criteria)

Counts values, including text, selected from an array or range like a database table using SQL-like queries.

DGET function

DGET (database_data; field; criteria)

Returns a unique value from an array or range similar to a database table using SQL-like queries.

DMAX function

DMAX (data base; field; criteria)

Returns the selected maximum value from an array or database table-like range by using SQL-like queries.

DMIN function

DMIN (database_data; school; criteria)

Returns the smallest value selected from an array or range similar to a database table using SQL-like queries.

DPRODUCT function

DPRODUCT (data base; school; criteria)

Returns the product of selected values ??from an array or range similar to a database table using SQL-like queries.

DSTDEV function

DSTDEV (database_data; field; criteria)

Returns the standard deviation of a population sample selected from an array or range similar to a database table using SQL-like queries.

DSUM function

DSUM (facilities_data, schools, criteria)

Returns the sum of selected values ??from an array or range similar to a database table using SQL-like queries.

DVARP function

DVARP (facilities_data; schools; criteria)

Returns the variance of the entire selected population from an array or range similar to a database table using SQL-like queries.

2. Google function

ARRAYFORMULA function

ARRAYFORMULA (formula_range)

Enables displaying values ??returned from an array formula into multiple rows and / or columns and using non-array functions with arrays

DETECTLANGUAGE function

DETECTLANGUAGE (text_or_range)

Specifies the language used in the text within a specified range of cells.

GOOGLEFINANCE function

GOOGLEFINANCE (certificate_key; [attributes]; [start_date]; [end_date | num_dates]; [approximately])

Fetching current or historical securities information from Google Finance.

GOOGLETRANSLATE function

GOOGLETRANSLATE (text; [source language]; [target language])

Translate text from one language to another

IMAGE function

IMAGE (url; [mode]; [height]; [width])

Insert image into cell.

QUERY function

QUERY (data; query; [title])

Run a query in the Google Visualization API Query Language on a variety of data.

SPARKLINE function

SPARKLINE (data; [optional])

Create small charts in a cell.

3. Logic function

IFS

IFS (condition1; value1; [condition2; value2]; .)

Evaluates multiple conditions and returns a value corresponding to the first true condition.

AND FUNCTION

AND (expression_logic1, [expression_logic2, .])

Returns true if all of the provided arguments are logically true, and returns false if any of the provided arguments are logically false.

FALSE function

FALSE ()

Returns the logical value `FALSE`.

IF function

IF (log_logic, value_if_true, value_if_)

Returns a value if a logical expression is `TRUE` and another value if the logical expression is` FALSE`.

IFERROR function

IFERROR function (value, [value_if_error])

Returns the first argument if it is not an error value, otherwise returns the second argument if this argument is present or left blank if this argument is absent.

IFNA function

IFNA (value; value_if_error)

Evaluate a value. If the value is an # N / A error, the function will return the specified value. .

FUNNY NOT

NOT (expression_logic)

Returns the opposite of the logical value - `NOT (TRUE)` returns `FALSE`; `NOT (FALSE)` returns `TRUE`.

OR function

OR (expression_logic1, [expression_logic2, .])

Returns true if any of the provided arguments are logically true, and returns false if all of the provided arguments are logically false.

TRUE function

TRUE ()

Returns the logical value `TRUE`.

XOR function

XOR (expression_logic1; [expression_logic2; .])

The XOR function executes the privilege or of 2 arguments to return 1 if the arguments are different and returns 0 otherwise. .

4. Statistical functions

AVERAGE.WEIGHTED

AVERAGE.WEIGHTED (value; weight; [value added]; [value_sung])

Find the weighted average of a set of values ??when knowing the corresponding values ??and weights. .

CHISQ.DIST

CHISQ.DIST (x; degrees_freedom; cumulative)

Calculates the left chi square distribution function, often used in hypothesis testing.

MAXIFS

MAXIFS (range; criteria_range1; criteria1; [criteria_range2; criterion2; .])

Returns the maximum value in a range filtered by a group of criteria.

AVEDEV function

AVEDEV (value1; [value2; .])

Calculate the average of the magnitudes of data deviations from the midpoint of a dataset.

AVERAGE function

AVERAGE (value_1; [value_2; .])

Returns the arithmetic mean value in a dataset, ignoring text.

AVERAGEA function

AVERAGEA (value1; [value2, .])

Returns the arithmetic mean value in a dataset.

AVERAGEIF function

AVERAGEIF (criteria_range; criteria; [average_range])

Returns the average value of a range based on multiple criteria.

AVERAGEIFS function

AVERAGEIFS (midrange_range; criteria_range1; criterion1; [criteria_range2; criterion2; .])

Returns the average value of a range based on multiple criteria.

BETA.DIST

BETA.DIST (value; alpha; beta; cumulative; lower_; upper_value)

Returns the probability of a given value as determined by the beta distribution function. .

BETA.INV function

BETA.INV (probability; alpha; beta; lower_value; upper_limit)

Returns the inverse value of the beta distribution function for a given probability.

CHIINV function

CHIINV (probability; degrees_freedom)

Calculate the inverse of the right chi square distribution function.

CONFIDENCE.NORM function

CONFIDENCE.NORM (alpha; standard_value; compound_size)

Calculate the width of half confidence interval for a standard distribution function. .

CONFIDENCE.T function

CONFIDENCE.T (alpha; standard_label; size_)

Calculate the width of half the confidence interval for a Student's t-distribution function. .

CORREL function

CORREL (data_y; data_x)

Calculate r, the Pearson product moment correlation coefficient of a data set.

COUNT function

COUNT (value1; [value2, .])

Returns the total number of values ??in a data set.

COVARIANCE.S function

COVARIANCE.S (data_y; data_x)

Calculate the covariance of a data set, in which the dataset is a sample of the entire population. .

EXPON.DIST

EXPON.DIST (x; lambda; cumulative)

Returns the value of the exponential distribution function with a given lambda at a specified value. .

FORECAST function

FORECAST (x, data_y; data_x)

Predict the y value for a specified x based on linear regression of a data set.

GAMMA function

GAMMA (number)

Returns the Gamma function evaluated at the specified value. .

GAMMA.INV function

GAMMA.INV (probability; alpha; beta)

The GAMMA.INV function returns the inverse of the gamma cumulative distribution function for the specified probability as well as the beta and alpha parameters. .

GAUSS function

GAUSS (z)

The GAUSS function returns the probability that a random variable, derived from a normal distribution function, will lie between the mean standard deviation and z above (or below) the mean. .

HYPGEOMDIST function

HYPGEOMDIST (ordinal_success; number_traction_sunition; frequency_construct_conventions; merge_size)

Calculate the probability of withdrawing a certain number of successes in a given number of tests, given a given set of dimensions, containing a certain number of successes, without redeploying the draws .

LARGE function

LARGE (data; n)

Returns the nth largest element of a data set, where n is user defined.

MAX function

MAX (value1; [value2, .])

Returns the maximum value in a numeric data set.

MEDIAN function

SCOPE (value1; [value2; .])

Returns the average value in a numeric data set.

MIN function

MIN (value_1; [value_2; .])

Returns the smallest value in a numeric data set.

MODE function

MODE (value1; [value2; .])

Returns the most frequent value in a dataset.

MODE.MULT function

MODE.MULT (value1; value2)

Returns the most frequent value in a dataset. .

NORMDIST function

NORMDIST (x; mean_value; standard_deviation; cumulative)

Returns the value of the standard distribution function (or standard cumulative distribution function) for a specified value, mean, and standard deviation.

PERCENTILE function

PERCENTILE (data; percentile)

Returns the value at a given percentile of a data set.

PERCENTILE.EXC function

PERCENTILE.EXC (data; percentile)

Returns the value at a given percentile of a data set between 0 and 1.

PERCENTRANK function

PERCENTRANK (data; value; [numerals_value])

Returns the percentage rank (percentile) of a specified value in a dataset.

PERCENTRANK.EXC function

PERCENTRANK.EXC (data; value; [numerals_value])

Returns the percentage rank (percentile) from 0 to 1 regardless of a specific value in a data set.

PERCENTRANK.INC function

PERCENTRANK.INC (data; value; [numerals_value])

Returns the percentage rank (percentile) from 0 to 1 including a specified value in a dataset.

PERMUTATIONA function

PERMUTATIONA (number_selected)

Returns the number of permutations to select a group of objects (with substitution) in the total number of objects.
.

PHI function

PHI (x)

The PHI function returns the value of the standard distribution function whose mean value is 0 and the standard deviation is 1.

POISSON.DIST

POISSON.DIST (x; average_value; [cumulative])

Returns the value of the Poisson distribution function (or Poisson cumulative distribution function) for a specified value and mean. .

QUARTILE function

QUARTILE (data; decimal_name)

Returns the closest value to a specific quartile of a data set.

QUARTILE.EXC function

QUARTILE.EXC (data; decimal_name)

Returns the nearest value for a specific quartile of a data set between 0 and 4.

RANK function

RANK (value; data; [increasing])

Returns the rank of a specified value in a dataset.

RANK.AVG function

RANK.AVG function (value; data; [increasing])

Returns the rank of a specified value in a dataset. If the data set has multiple values ??with the same rank, the function will return the average rank of the entries.

RANK.EQ function

RANK.EQ (value; data; [increasing])

Returns the rank of a specified value in a dataset. If there is more than one entry of the same value in the data set, the function will return the highest rank of the entries.

SKEW.P function

SKEW.P (value1; value2)

Calculate the deviation of a data set that represents the entire population. .

SLOPE function

SLOPE (data_y; data_x)

Calculate the slope of a linear regression line for a data set.

STDEV function

STDEV (value_1; [value_2; .])

Calculate the standard deviation based on a sample.

T.DIST

T.DIST (x; degrees_freedom; cumulative)

Returns the right-tailed Student distribution function for x value. .

T.DIST.2T

T.DIST.2T (x; degrees_freedom)

Returns the two-tailed Student distribution function for x values. .

T.DIST.RT function

T.DIST.RT (x; degrees_freedom)

Returns the right-tailed Student distribution for the value x. .

T.TEST function

T.TEST (range_1; range_2; side; type)

Returns the probability associated with Student's t-test. Determine whether the two samples are likely to originate from the two original sets having the same mean. .

TDIST function

TDIST (x; degrees_freedom; tail)

Calculate the probability for Student's t-distribution with the given input (x).

VAR function

VAR (value1, [value2, .])

Calculate variance based on a sample.

Z.TEST function

Z.TEST (data; value; [standard_deviation])

Returns the one-sided P value of the Z-test with standard distribution function. .

COUNTA function

COUNTA (value_1; [value_2; .])

Returns the total number of values ??in a dataset.

5. Technical functions

BITAND function

BITAND (value1; value2)

Boolean reverses the AND bit of two numbers.

BITLSHIFT function

BITLSHIFT (value; transfer_value)

Moves the bits of the entered number to the left of the positions.

BITOR function

BITOR (value1; value2)

Boolean reverses the OR bit of two numbers.

BITRSHIFT function

BITRSHIFT (value; transfer_value)

Moves the bits of the entered number into the right positions.

BITXOR function

BITXOR (value1; value2)

Inverts the XOR bit (excluding the OR function) of the two numbers.

COMPLEX function

COMPLEX (real_part; part_;; [suffix])

Create a complex number that is given to the real and virtual coefficients

DELTA function

DELTA (number 1; [number 2])

Comparing 2 numeric values, returns 1 if they are equal.

ERF function

ERF (lower_limit; [upper_limit])

The ERF function returns the integral of the Gauss error function in a range of values. .

GESTEP function

GESTEP (value, [steps])

Returns 1 if the ratio is greater than or equal to the step value provided or 0 otherwise. If no step value is provided, the default value is 0.

HEX2DEC function

HEX2DEC (decimal_number_range)

Convert signed hexadecimal numbers to decimal format.

IMABS function

IMABS (number)

Returns the absolute value of a complex number.

IMAGINARY function

IMAGINARY (number_company)

Returns the imaginary coefficient of a complex number.

IMARGUMENT function

IMARGUMENT (number)

The IMARGUMENT function returns the angle (also called the argument or theta) of the given complex number in radians. .

IMCONJUGATE

IMCONJUGATE (number)

Returns the complex conjugate of a number.

IMCOS function

IMCOS (number)

The IMCOS function returns the cosine of a given complex number. .

IMCOSH function

IMCOSH (number)

Returns the hyperbolic cosine of a given complex number. For example, a complex number that gave "x + yi" returns "cosh (x + yi)". .

IMCOT function

IMCOT (number)

Returns the cotang of a given complex number. For example, a given complex number "x + yi" returns "cot (x + yi)". .

IMCOTH function

IMCOTH (number)

Returns the hyperbolic cotangent of a given complex number. For example, a complex number that gave "x + yi" returns "coth (x + yi)". .

IMCSC

IMCSC (number)

Returns the cosec of the complex number given. .

IMCSCH function

IMCSCH (number)

Returns the hyperbolic cosec of a given complex number. For example, a complex number that gave "x + yi" returns "csch (x + yi)". .

IMDIV function

IMDIV (number_number; number_chia)

Returns a complex number divided by another complex number.

IMEXP

IMEXP (number_guards)

Returns the Euler number, e (~ 2.718) raised to a complex power. .

IMLOG

IMLOG (value; basis)

Returns the logarithm of a complex number with a specified base number. .

IMLOG10 function

IMLOG10 (value)

Returns the logarithm of a complex number with base 10.

IMLOG2 function

IMLOG2 (value)

Returns the logarithm of a complex number with base 2.

IMPRODUCT function

IMPRODUCT (excess_1; [excess__2; .])

Returns the result when multiplying a series of complex numbers together.

IMREAL function

IMREAL (number_company)

Returns the real coefficient of a complex number.

IMSEC function

IMSEC (number)

Returns the sec of a given complex number. For example, a complex number that gave "x + yi" returns "sec (x + yi)". .

IMSECH function

IMSECH (number)

Returns the hyperbolic sec of a given complex number. For example, a complex number that gave "x + yi" returns "sec (x + yi)". .

IMSIN

IMSIN (number)

Returns the sine of a given complex number. .

IMSINH function

IMSINH (number)

Returns the hyperbolic sine of a given complex number. For example, a complex number that gave "x + yi" returns "sinh (x + yi)". .

IMSUB function

IMSUB (first_name; second_name)

Returns the difference of two complex numbers.

IMSUM function

IMSUM (value1; [value2; .])

Returns the sum of a sequence of complex numbers.

IMTAN

IMTAN (number)

Returns the tangent of the given complex number. .

IMTANH function

IMTANH (number)

Returns the hyperbolic tangent of the given complex number. For example, a given complex number "x + yi" returns "tanh (x + yi)". .

6. Filter function

FILTER function

FILTER (range; condition1; [condition2])

Returns a filtered version of the source range, returning only rows or columns that meet the specified conditions.

SORT function

SORT (range; sorted_range; ascending; [sorted2 column]; [ascending2])

Sorts rows of a given array or range of values ??by one or more columns.

UNIQUE function

UNIQUE (range_)

Returns unique rows in a given source range, discarding duplicate rows. Rows are returned in the order they first appeared in the source range.

7. Array function

ARRAY_CONSTRAIN function

ARRAY_CONSTRAIN (first range; number_ row; number_column)

Constraints an array value to a specified size.

FREQUENCY function

FREQUENCY (data, layer)

Calculates the frequency distribution of values ??in a column, based on given classes.

GROWTH function

GROWTH (known_data_y; data_data_x]; [new_data_x]; [b])

Give a portion of the data about an exponential growth trend, find the appropriate exponential growth trend and / or predict further value.

LINEST function

LINEST (known_data_y; data_data_x], [calculated_b], [details])

Given a piece of data about a linear trend, calculate the parameters for the ideal linear trend using the least square method.

MMULT function

MMULT (match1; match2)

Calculates the matrix product of 2 specified matrices as an array or range.

SUMPRODUCT function

SUMPRODUCT (array1; [array2; .])

Calculates the sum of the products of corresponding elements in two equal-sized arrays or ranges.

SUMX2MY2 function

SUMX2MY2 (array_x; array_y)

Calculate the sum of the difference of squares of values ??in two arrays.

SUMXMY2 function

SUMXMY2 (array_x; array_y)

Calculate the sum of squares of differences of values ??in 2 arrays.

TRANSPOSE function

TRANSPOSE (array_or_range)

Transpose the rows and columns of an array or range.

TREND function

TREND (known_data_y; data_data_x]; [new_data_x]; [b])

Given a piece of data about a linear trend, find an appropriate linear trend using the least squares method and / or add value predictions.

8. The function of the date

DATE function

DATE (year, month, day)

Convert a given year, month, and day to a date (in calendar).

DATEDIF function

DATEDIF (start_date; end_date; unit)

Calculate the number of days, months or years between two dates.

DATEVALUE function

DATEVALUE (string_day_month)

Converts a given date string in a known format into a date value.

The DAY function

DAY (days)

Returns the day number of the month, based on a given date value, in numeric format.

DAYS function

DAYS (end_date; start_date)

Returns the number of days between two dates.

DAYS360 function

DAYS360 (start_date; end_date; [method])

Returns the number of days between two dates on the basis of a 360-day year used in some financial interest calculations.

EDATE function

EDATE (start_date; months)

Returns a date (in calendar) before or after a specified number of months from another ((calendar)) date.

EOMONTH function

EOMONTH (start_date; months)

Returns the date representing the last day of a month before or after a number of months from another date.

HOUR function

HOUR (hours)

Returns the hour component of a specific time, in numeric format.

MINUTE function

MINUTE (time_gian)

Returns the minute component of a specific time, in numeric format.

MONTH function

MONTH (day_month)

Returns the month of the year based on a given date, in numeric format.

NETWORKDAYS function

NETWORKDAYS (start_date; end_date; [date_date])

Returns the number of actual business days between two given days.

NETWORKDAYS.INTL function

NETWORKDAYS.INTL (start_date; end_month; end_day_day;; [date_date])

Returns the number of actual working days between two given days, excluding specific holidays and weekends.

NOW function

NOW ()

Returns the current date and time as a date value.

SECOND function

SECOND (time)

Returns the second component of a specific time, in numeric format.

TIME

TIME (hours; minutes; seconds)

Converts a given hour, minute, and second to hours.

TIMEVALUE function

TIMEVALUE (string_hours)

Returns the fraction of a 24-hour day corresponding to a given time.

TODAY function

TODAY ()

Returns the current date as a date value.

WEEKDAY function

WEEKDAY (date, [type])

Returns a number representing the day of the week of the given date.

WEEKNUM function

WEEKNUM (day_month; [type])

Returns the week number of the year based on a given date.

WORKDAY function

WORKDAY (start_date_date; days_date; [date_date])

Calculates the end date after the specified number of working days.

WORKDAY.INTL function

WORKDAY.INTL (start_date; start_date; [last_day_day]; [date_date])

Calculate dates after a specific number of working days, excluding specific holidays and weekends.

YEAR function

YEAR (days)

Returns the year specified with a given date.

YEARFRAC function

YEARFRAC (start_date; end_date; [date_date])

Returns the number of years, including the fraction of the year, between two dates using a specified date counting convention.

9. Financial functions

AMORLINEC function

AMORLINC (cost; purchase_date; first_implementation; return_value; rate; [basis])

Returns the depreciation for the accounting period or pro-rated depreciation if the asset is purchased in the middle of the period.

COUPDAYSNC function

COUPDAYSNC (settlement_date; maturity_date; frequency; [daily_date]]

Calculate the number of days from the settlement date until the next interest payment or coupon payment.

DURATION function

DURATION (settlement_date; maturity_date; interest_interest; interest; frequency; [daily_date]).

Calculates the number of periods of compound interest required for an investment of the present value determined when judging at a given rate to reach the target value.

FV function

FV (interest_date, term_payment, payment_value, [current value], [end_or_date])

Calculate the future value of an annuity investment based on periodic equal payments and a constant interest rate.

INTRATE function

INTRATE (purchase_date; sell_date; purchase_date; sale_date; [daily_date]]

Calculates the actual interest rate that arises when buying an investment at one price and selling at another price without the interest or dividends generated by the investment itself.

IPMT function

IPST (yield_period; period_value; current value; [value_value]; [end_or_date])

Calculate interest payment for an investment based on periodic equal payments and constant interest rates.

IRR function

IRR (payout_payment; [rate of return]

Calculate the internal rate of return on an investment based on a series of periodic cash flows.

ISPM function

ISPMT (interest_period; period_value; current value)

The ISPMT function calculates the interest payable for a specific period of time for an investment. .

NPER function

NPER (yield_payment; payment_value; current value; [value_value]; [ending_or_development])

Calculates the number of payment periods for an investment based on periodic equal payments and a constant interest rate.

NPV function

NPV (discount_top; money1 stream; [cashflow2; .])

Calculates the net present value of an investment based on a series of periodic cash flows and discount rates.

PDURATION function

PDURATION (interest rate; current value; future_value)

Returns the number of periods for an investment to reach a specific value at a given rate. .

PMT function

PMT (interest_value; period_value; current_value; [future_value]; [end_or_development])

Calculation of periodic payments for annuity investments based on periodic equal payments and constant interest rates.

PPMT function

PPMT (interest_period; term; period; current value; [fair_value]; [ending or starting_)]

Calculate the principal payment of an investment based on periodic equal payments and a constant interest rate.

PRICEMAT function

PRICEMAT (settlement_date; maturity_date ;_date_date; interest_interest; interest; [daily_date]]

Calculate the price of security paying interest at maturity, based on expected yield.

PV function

PV (interest_due; period_value; current_value; payment_value; [future_value]; [ending_or_development])

Calculates the present value of an annuity investment based on periodic equal payments and a constant interest rate.

RECEIVED function

RECEIVED (settlement_date; maturity_date; investment_date; discount; [daily_date])

Calculates the amount received at maturity for an investment in fixed-income securities purchased on a date.

RRI function

RRI (period_value; current_value; future_value)

Returns the interest rate required for an investment to reach a specific value in a given period. .

VDB function

VDB (cost; return_value; period_activity; start_start period; end_value; [factor]; [non-conversion])

Returns the depreciation of an asset for a specific period (or a fraction of a period). .

XIRR function

XIRR (payoff_payment; pay_date_date_value; [Deluxeguess])

Calculates the internal rate of return of an investment based on a series of potentially non-recurring cash flows.

YIELDDISC function

YIELDDISC (settlement_date; maturity_date; price; refund_value; [daily_date]]

Calculates the annual yield of a discounted stock (without interest), based on price.

10. Information function

ISBLANK function

ISBLANK (value)

Checks whether the referenced cell is blank.

ISDATE function

ISDATE (value)

Returns whether the value is a date.

ISEMAIL function

ISEMAIL function (value)

Check if the value is a valid email address.

ISERROR function

ISERROR (value)

Checks whether the value is an error or not.

ISLOGICAL function

ISLOGICAL (value)

Checks whether the value is `TRUE` or` FALSE`.

ISNA function

ISNA (value)

Checks whether the value is a `# N / A` error.

ISNUMBER function

ISNUMBER (value)

Checks whether the value is a number or not.

ISTEXT function

ISTEXT (value)

Checks whether the value is text.

N function

N (value)

Returns the argument provided as a number.

NA function

NA ()

Returns the "invalid" error, `# N / A`.

TYPE function

TYPE (value)

Returns a number corresponding to the type of data included in the function.

CELL function

CELL (info_type; reference)

Returns the required information for a specific cell.

11. Search function

ADDRESS function

ADDRESS (row, column, [absolute_mode], [use_range1], [worksheet])

Returns a cell reference as a string.

CHOOSE function

CHOOSE (index, choice1, [choice2, .])

Returns the element from the list of options based on the index.

COLUMN function

COLUMN ([reference]]

Returns the column number of a specified cell, with `A = 1`.

COLUMNS function

COLUMNS (range_)

Returns the number of columns in a specified array or range.

FORMULATEXT function

FORMULATEXT (cell)

Returns the formula as a string. .

HLOOKUP function

HLOOKUP (search_key, range, index, [sorted]]

Search horizontally. Searches across the first row of the range for a key and returns the value of a specified cell in the column found.

INDEX function

INDEX (reference, [row], [column])

Returns the content of a cell, determined by the difference between rows and columns

INDIRECT function

INDIRECT (string_reference_string, [is_language_A1])

Returns a cell reference specified by a string.

LOOKUP function

LOOKUP (search_key; search_range | search_array_search results; [search_range])

Finds a key in a row or column and returns the value of the cell in a result range located in the same position as the search row or column.

MATCH function

MATCH (search_key, range, [search_type])

Returns the relative position of an item in the range that matches a specified value.

OFFSET function

OFFSET (parameter_reference, minus_true row, minus_column column, [height_size], [width]

Returns the reference of the converted range of rows and columns from a starting cell reference.

ROW function

ROW ([reference])

Returns the number of rows of a specified cell.

ROWS function

ROWS (range_)

Returns the number of rows in a specified array or range.

VLOOKUP function

VLOOKUP (search_key; range; index; [sorted]]

Search vertically. Search vertically the first column of the range for a key and returns the value of a specified cell in the found row.

12. Math functions

COTH

COTH (value)

Returns the hyperbolic cotangent of any real number. .

CSC

CSC (angle)

Returns the cosec of the given angle in radians. .

ABS function

ABS (value)

Returns the absolute value of a number.

CEILING function

CEILING (value, [coefficients])

Rounds a number up to the nearest significant integer multiple.

CEILING.PRECISE function

CEILING.PRECISE (number; [multidirectory])

Rounds a number up to the nearest integer multiple of the specified number. If positive or negative, this value is rounded up.

COMBIN function

COMBIN (n, k)

Returns the number of ways to select the number of objects in a combination of objects of a given size

COMBINA

COMBINA (n; k)

Returns the number of ways to select several objects in a combination of objects of a given size, including how to select the same object multiple times. .

COT function

COT (angle)

Cotang of the given angle, in radians. .

COUNTBLANK function

COUNTBLANK (range_)

Returns the number of empty cells in a given range.

COUNTIF function

COUNTIF (range, criteria)

Returns the conditional count on a range.

COUNTIFS function

COUNTIFS (criteria_range1; criteria1; [criteria_range2; criterion2; .])

Returns the count of a range based on multiple criteria.

COUNTUNIQUE function

COUNTUNIQUE (value1; [value2; .])

Counts the unique number of values ??in a list of specified values ??and ranges.

CSCH function

CSCH (value)

The CSCH function returns the hyperbolic cosec of any real number. .

DECIMAL function

DECIMAL (value; basis)

The DECIMAL function converts the textual representation of a number in another base, into base 10 (decimal).
.

DEGREES function

DEGREES (angle)

Convert angle values ??in radians to degrees.

FLOOR function

FLOOR (value, [coefficient])

Rounds a number down to the nearest significant integer multiple.

FLOOR.MATH

FLOOR.MATH (number; [multiples_range]; [mode])

Rounds a number down to the nearest integer multiple of the specified number, with negative numbers rounded to 0 or rounded up according to the mode.

FLOOR.PRECISE function

FLOOR.PRECISE (number; [multiples_number])

The FLOOR.PRECISE function rounds a number to the nearest integer or the nearest significant multiple. .

INT function

INT (value)

Rounds a number down to the nearest integer less than or equal to itself.

LOG function

LOG (value, base)

Returns the logarithm of a number based on a base.

MOD function

MOD (number_divided, number_chia)

Returns the result of the module operator, the remainder after a division operation.

MROUND function

MROUND (value, coefficients)

Rounds a number to the nearest integer multiple of another number.

MUNIT function

MUNIT (ordinal)

Returns a unit matrix of size ax b. .

POWER function

POWER (base, number_unit)

Returns a number raised to a power.

PRODUCT

PRODUCT (coefficient_1, [coefficient_2, .])

Returns the result when multiplying a sequence of numbers together.

QUOTIENT function

QUOTIENT (number_dated, number_chia)

Returns a number divided by another number.

RAND function

RAND ()

Returns a random number from 0 to less than 1.

RANDARRAY

RANDARRAY (row; column)

Generates a sequence of random numbers between 0 and 1.

RANDBETWEEN function

RANDBETWEEN (low, high)

Returns a constant random integer between two values, including these two values.

Jaw