

Google Sheets All Functions List

1. Date Functions

Type	Name	Syntax	Description
Date	DATE	DATE(year, month, day)	Converts a provided year, month, and day into a date.
Date	DATEDIF	DATEDIF(start_date, end_date, unit)	Calculates the number of days, months, or years between two dates.
Date	DATEVALUE	DATEVALUE(date_string)	Converts a provided date string in a known format to a date value.
Date	DAY	DAY(date)	Returns the day of the month that a specific date falls on, in numeric format.
Date	DAYS	DAYS(end_date, start_date)	Returns the number of days between two dates.
Date	DAYS360	DAYS360(start_date, end_date, [method])	Returns the difference between two days based on the 360 day year used in some financial interest calculations.
Date	EDATE	EDATE(start_date, months)	Returns a date a specified number of months before or after another date.
Date	EOMONTH	EOMONTH(start_date, months)	Returns a date representing the last day of a month which falls a specified number of months before or after another date.

Type	Name	Syntax	Description
Date	EPOCHTODATE	<code>EPOCHTODATE(timestamp, [unit])</code>	Converts a Unix epoch timestamp in seconds, milliseconds, or microseconds to a datetime in UTC.
Date	HOUR	<code>HOUR(time)</code>	Returns the hour component of a specific time, in numeric format.
Date	ISOWEEKNUM	<code>ISOWEEKNUM(date)</code>	Returns the number of the ISO week of the year where the provided date falls.
Date	MINUTE	<code>MINUTE(time)</code>	Returns the minute component of a specific time, in numeric format.
Date	MONTH	<code>MONTH(date)</code>	Returns the month of the year a specific date falls in, in numeric format.
Date	NETWORKDAYS	<code>NETWORKDAYS(start_date, end_date, [holidays])</code>	Returns the number of net working days between two provided days.
Date	NETWORKDAYS.NTL	<code>NETWORKDAYS.INTL(start_date, end_date, [weekend], [holidays])</code>	Returns the number of net working days between two provided days excluding specified weekend days and holidays.
Date	NOW	<code>NOW()</code>	Returns the current date and time as a date value.
Date	SECOND	<code>SECOND(time)</code>	Returns the second component of a specific time, in numeric format.
Date	TIME	<code>TIME(hour, minute, second)</code>	Converts a provided hour, minute, and

Type	Name	Syntax	Description
			second into a time.
Date	TIMEVALUE	TIMEVALUE(time_string)	Returns the fraction of a 24-hour day the time represents.
Date	TODAY	TODAY()	Returns the current date as a date value.
Date	WEEKDAY	WEEKDAY(date, [type])	Returns a number representing the day of the week of the date provided.
Date	WEEKNUM	WEEKNUM(date, [type])	Returns a number representing the week of the year where the provided date falls.
Date	WORKDAY	WORKDAY(start_date, num_days, [holidays])	Calculates the end date after a specified number of working days.
Date	WORKDAY.INTL	WORKDAY.INTL(start_date, num_days, [weekend], [holidays])	Calculates the date after a specified number of workdays excluding specified weekend days and holidays.
Date	YEAR	YEAR(date)	Returns the year specified by a given date.
Date	YEARFRAC	YEARFRAC(start_date, end_date, [day_count_convention])	Returns the number of years, including fractional years, between two dates using a specified day count convention.

2. Engineering

Type	Name	Syntax	Description
Engineering	BIN2DEC	BIN2DEC(signed_binary_number)	Converts a signed binary

Type	Name	Syntax	Description
			number to decimal format.
Engineering	BIN2HEX	<code>BIN2HEX(signed_binary_number, [significant_digits])</code>	Converts a signed binary number to signed hexadecimal format.
Engineering	BIN2OCT	<code>BIN2OCT(signed_binary_number, [significant_digits])</code>	Converts a signed binary number to signed octal format.
Engineering	BITAND	<code>BITAND(value1, value2)</code>	Bitwise boolean AND of two numbers. ↗
Engineering	BITLSHIFT	<code>BITLSHIFT(value, shift_amount)</code>	Shifts the bits of the input a certain number of places to the left. ↗
Engineering	BITOR	<code>BITOR(value1, value2)</code>	Bitwise boolean OR of 2 numbers. ↗
Engineering	BITRSHIFT	<code>BITRSHIFT(value, shift_amount)</code>	Shifts the bits of the input a certain number of places to the right. ↗
Engineering	BITXOR	<code>BITXOR(value1, value2)</code>	Bitwise XOR (exclusive OR) of 2 numbers. ↗
Engineering	COMPLEX	<code>COMPLEX(real_part, imaginary_part, [suffix])</code>	Creates a complex number given real and imaginary coefficients.
Engineering	DEC2BIN	<code>DEC2BIN(decimal_number, [significant_digits])</code>	Converts a decimal number to signed binary format.
Engineering	DEC2HEX	<code>DEC2HEX(decimal_number, [significant_digits])</code>	Converts a decimal number

Type	Name	Syntax	Description
			to signed hexadecimal format.
Engineering	DEC2OCT	DEC2OCT(decimal_number, [significant_digits])	Converts a decimal number to signed octal format.
Engineering	DELTA	DELTA(number1, [number2])	Compare two numeric values, returning 1 if they're equal.
Engineering		(lower_bound, [upper_bound])	The function returns the integral of the Gauss error function over an interval of values. .
Engineering	.PRECISE	.PRECISE(lower_bound, [upper_bound])	See
Engineering	GESTEP	GESTEP(value, [step])	Returns 1 if the rate is strictly greater than or equal to the provided step value or 0 otherwise. If no step value is provided then the default value of 0 will be used. ...
Engineering	HEX2BIN	HEX2BIN(signed_hexadecimal_number, [significant_digits])	Converts a signed hexadecimal number to signed binary format.
Engineering	HEX2DEC	HEX2DEC(signed_hexadecimal_number)	Converts a signed hexadecimal number to decimal format.

Type	Name	Syntax	Description
Engineering	HEX2OCT	HEX2OCT(signed_hexadecimal_number, significant_digits)	Converts a signed hexadecimal number to signed octal format.
Engineering	IMABS	IMABS(number)	Returns absolute value of a complex number.
Engineering	IMAGINARY	IMAGINARY(complex_number)	Returns the imaginary coefficient of a complex number.
Engineering	IMARGUMENT	IMARGUMENT(number)	The IMARGUMENT function returns the angle (also known as the argument or θ) of the given complex number in radians. .
Engineering	IMCONJUGATE	IMCONJUGATE(number)	Returns the complex conjugate of a number.
Engineering	IMCOS	IMCOS(number)	The IMCOS function returns the cosine of the given complex number. .
Engineering	IMCOSH	IMCOSH(number)	Returns the hyperbolic cosine of the given complex number. For example, a given complex number "x+yi" returns "cosh(x+yi)." .
Engineering	IMCOT	IMCOT(number)	Returns the cotangent of the

Type	Name	Syntax	Description
			given complex number. For example, a given complex number "x+yi" returns "cot(x+yi)." .
Engineering	IMCOTH	IMCOTH(number)	Returns the hyperbolic cotangent of the given complex number. For example, a given complex number "x+yi" returns "coth(x+yi)." .
Engineering	IMCSC	IMCSC(number)	Returns the cosecant of the given complex number. .
Engineering	IMCSCH	IMCSCH(number)	Returns the hyperbolic cosecant of the given complex number. For example, a given complex number "x+yi" returns "csch(x+yi)." .
Engineering	IMDIV	IMDIV(dividend, divisor)	Returns one complex number divided by another.
Engineering	IMEXP	IMEXP(exponent)	Returns Euler's number, e (~2.718) raised to a complex power. .
Engineering	IMLOG	IMLOG(value, base)	Returns the logarithm of a complex number for a specified base. .

Type	Name	Syntax	Description
Engineering	IMLOG10	IMLOG10(value)	Returns the logarithm of a complex number with base 10. .
Engineering	IMLOG2	IMLOG2(value)	Returns the logarithm of a complex number with base 2. .
Engineering	IMPRODUCT	IMPRODUCT(factor1, [factor2, ...])	Returns the result of multiplying a series of complex numbers together.
Engineering	IMREAL	IMREAL(complex_number)	Returns the real coefficient of a complex number.
Engineering	IMSEC	IMSEC(number)	Returns the secant of the given complex number. For example, a given complex number "x+yi" returns "sec(x+yi)." .
Engineering	IMSECH	IMSECH(number)	Returns the hyperbolic secant of the given complex number. For example, a given complex number "x+yi" returns "sech(x+yi)." .
Engineering	IMSIN	IMSIN (number)	Returns the sine of the given complex number. .
Engineering	IMSINH	IMSINH(number)	Returns the hyperbolic sine of the given complex number.

Type	Name	Syntax	Description
			For example, a given complex number "x+yi" returns "sinh(x+yi)." .
Engineering	IMSUB	IMSUB(first_number, second_number)	Returns the difference between two complex numbers.
Engineering	IMSUM	IMSUM(value1, [value2, ...])	Returns the sum of a series of complex numbers.
Engineering	IMTAN	IMTAN(number)	Returns the tangent of the given complex number. .
Engineering	IMTANH	IMTANH(number)	Returns the hyperbolic tangent of the given complex number. For example, a given complex number "x+yi" returns "tanh(x+yi)." .
Engineering	OCT2BIN	OCT2BIN(signed_octal_number, [significant_digits])	Converts a signed octal number to signed binary format.
Engineering	OCT2DEC	OCT2DEC(signed_octal_number)	Converts a signed octal number to decimal format.
Engineering	OCT2HEX	OCT2HEX(signed_octal_number, [significant_digits])	Converts a signed octal number to signed hexadecimal format.

3.Filter

Type	Name	Syntax	Description
Filter	FILTER	<code>FILTER(range, condition1, [condition2])</code>	Returns a filtered version of the source range, returning only rows or columns which meet the specified conditions.
Filter	SORT	<code>SORT(range, sort_column, is_ascending, [sort_column2], [is_ascending2])</code>	Sorts the rows of a given array or range by the values in one or more columns.
Filter	SORTN	<code>SORTN(range, [n], [display_ties_mode], [sort_column1, is_ascending1], ...)</code>	Returns the first n items in a data set after performing a sort.
Filter	UNIQUE	<code>UNIQUE(range)</code>	Returns unique rows in the provided source range, discarding duplicates. Rows are returned in the order in which they first appear in the source range.

4.Financial

Type	Name	Syntax	Description
Financial	ACCRINT	<code>ACCRINT(issue, first_payment, settlement, rate, redemption, frequency, [day_count_convention])</code>	Calculates the accrued interest of a security that has periodic payments.
Financial	ACCRINTM	<code>ACCRINTM(issue, maturity, rate, [redemption], [day_count_convention])</code>	Calculates the accrued interest of a security that pays interest at maturity.
Financial	AMORLINC	<code>AMORLINC(cost, purchase_date, first_period_end, salvage, period, rate, [basis])</code>	Returns the depreciation for an accounting period, or the prorated depreciation if the asset was purchased in the middle of a period. ↗
Financial	COUPDAYBS	<code>COUPDAYBS(settlement, maturity, frequency, [day_count_convention])</code>	Calculates the number of days from the first coupon, or interest payment, until settlement.
Financial	COUPDAYS	<code>COUPDAYS(settlement, maturity, frequency, [day_count_convention])</code>	Calculates the number of days in the coupon, or interest payment, period

Type	Name	Syntax	Description
			that contains the specified settlement date.
Financial	COUPDAYSNC	COUPDAYSNC(settlement, maturity, frequency, [day_count_convention])	Calculates the number of days from the settlement date until the next coupon, or interest payment.
Financial	COUPNCD	COUPNCD(settlement, maturity, frequency, [day_count_convention])	Calculates next coupon, or interest payment, date after the settlement date.
Financial	COUPNUM	COUPNUM(settlement, maturity, frequency, [day_count_convention])	Calculates the number of coupons, or interest payments, between the settlement date and the maturity date of the investment.
Financial	COUPPCD	COUPPCD(settlement, maturity, frequency, [day_count_convention])	Calculates last coupon, or interest payment, date before the settlement date.
Financial	CUMIPMT	CUMIPMT(rate, number_of_periods, present_value, first_period, last_period, end_or_beginning)	Calculates the cumulative interest over a range of payment periods for an investment based on constant-amount periodic payments and a constant interest rate.
Financial	CUMPRINC	CUMPRINC(rate, number_of_periods, present_value, first_period, last_period, end_or_beginning)	Calculates the cumulative principal paid over a range of payment periods for an investment based on constant-amount periodic payments and a constant interest rate.
Financial	DB	DB(cost, salvage, life, period, [month])	Calculates the depreciation of an asset for a specified period using the arithmetic

Type	Name	Syntax	Description
			declining balance method.
Financial	DDB	DDB(cost, salvage, life, period, [factor])	Calculates the depreciation of an asset for a specified period using the double-declining balance method.
Financial	DISC	DISC(settlement, maturity, price, redemption, [day_count_convention])	Calculates the discount rate of a security based on price.
Financial	DOLLARDE	DOLLARDE(fractional_price, unit)	Converts a price quotation given as a decimal fraction into a decimal value.
Financial	DOLLARFR	DOLLARFR(decimal_price, unit)	Converts a price quotation given as a decimal value into a decimal fraction.
Financial	DURATION	DURATION(settlement, maturity, rate, yield, frequency, [day_count_convention]) .	Calculates the number of compounding periods required for an investment of a specified present value appreciating at a given rate to reach a target value.
Financial	EFFECT	EFFECT(nominal_rate, periods_per_year)	Calculates the annual effective interest rate given the nominal rate and number of compounding periods per year.
Financial	FV	FV(rate, number_of_periods, payment_amount, [present_value], [end_or_beginning])	Calculates the future value of an annuity investment based on constant-amount periodic payments and a constant interest rate.
Financial	FVSCHEDULE	FVSCHEDULE(principal, rate_schedule)	Calculates the future value of some principal based on a specified

Type	Name	Syntax	Description
			series of potentially varying interest rates.
Financial	INTRATE	INTRATE(buy_date, sell_date, buy_price, sell_price, [day_count_convention])	Calculates the effective interest rate generated when an investment is purchased at one price and sold at another with no interest or dividends generated by the investment itself.
Financial	IPMT	IPMT(rate, period, number_of_periods, present_value, [future_value], [end_or_beginning])	Calculates the payment on interest for an investment based on constant-amount periodic payments and a constant interest rate.
Financial	IRR	IRR(cashflow_amounts, [rate_guess])	Calculates the internal rate of return on an investment based on a series of periodic cash flows.
Financial	ISPMT	ISPMT(rate, period, number_of_periods, present_value)	The ISPMT function calculates the interest paid during a particular period of an investment. .
Financial	MDURATION	MDURATION(settlement, maturity, rate, yield, frequency, [day_count_convention])	Calculates the modified Macaulay duration of a security paying periodic interest, such as a US Treasury Bond, based on expected yield.
Financial	MIRR	MIRR(cashflow_amounts, financing_rate, reinvestment_return_rate)	Calculates the modified internal rate of return on an investment based on a series of periodic cash flows and the difference between the interest rate paid on financing versus the return received on reinvested income.
Financial	NOMINAL	NOMINAL(effective_rate, periods_per_year)	Calculates the annual nominal interest rate

Type	Name	Syntax	Description
			given the effective rate and number of compounding periods per year.
Financial	NPER	NPER(rate, payment_amount, present_value, [future_value], [end_or_beginning])	Calculates the number of payment periods for an investment based on constant-amount periodic payments and a constant interest rate.
Financial	NPV	NPV(discount, cashflow1, [cashflow2, ...])	Calculates the net present value of an investment based on a series of periodic cash flows and a discount rate.
Financial	PDURATION	PDURATION(rate, present_value, future_value)	Returns the number of periods for an investment to reach a specific value at a given rate. .
Financial	PMT	PMT(rate, number_of_periods, present_value, [future_value], [end_or_beginning])	Calculates the periodic payment for an annuity investment based on constant-amount periodic payments and a constant interest rate.
Financial	PPMT	PPMT(rate, period, number_of_periods, present_value, [future_value], [end_or_beginning])	Calculates the payment on the principal of an investment based on constant-amount periodic payments and a constant interest rate.
Financial	PRICE	PRICE(settlement, maturity, rate, yield, redemption, frequency, [day_count_convention])	Calculates the price of a security paying periodic interest, such as a US Treasury Bond, based on expected yield.
Financial	PRICEDISC	PRICEDISC(settlement, maturity, discount, redemption, [day_count_convention])	Calculates the price of a discount (non-interest-bearing) security, based on expected yield.

Type	Name	Syntax	Description
Financial	PRICEMAT	PRICEMAT(settlement, maturity, issue, rate, yield, [day_count_convention])	Calculates the price of a security paying interest at maturity, based on expected yield.
Financial	PV	PV(rate, number_of_periods, payment_amount, [future_value], [end_or_beginning])	Calculates the present value of an annuity investment based on constant-amount periodic payments and a constant interest rate.
Financial	RATE	RATE(number_of_periods, payment_per_period, present_value, [future_value], [end_or_beginning], [rate_guess])	Calculates the interest rate of an annuity investment based on constant-amount periodic payments and the assumption of a constant interest rate.
Financial	RECEIVED	RECEIVED(settlement, maturity, investment, discount, [day_count_convention])	Calculates the amount received at maturity for an investment in fixed-income securities purchased on a given date.
Financial	RRI	RRI(number_of_periods, present_value, future_value)	Returns the interest rate needed for an investment to reach a specific value within a given number of periods. .
Financial	SLN	SLN(cost, salvage, life)	Calculates the depreciation of an asset for one period using the straight-line method.
Financial	SYD	SYD(cost, salvage, life, period)	Calculates the depreciation of an asset for a specified period using the sum of years digits method.
Financial	TBILLEQ	TBILLEQ(settlement, maturity, discount)	Calculates the equivalent annualized rate of return of a US

Type	Name	Syntax	Description
			Treasury Bill based on discount rate.
Financial	TBILLPRICE	TBILLPRICE(settlement, maturity, discount)	Calculates the price of a US Treasury Bill based on discount rate.
Financial	TBILLYIELD	TBILLYIELD(settlement, maturity, price)	Calculates the yield of a US Treasury Bill based on price.
Financial	VDB	VDB(cost, salvage, life, start_period, end_period, [factor], [no_switch])	Returns the depreciation of an asset for a particular period (or partial period). .
Financial	XIRR	XIRR(cashflow_amounts, cashflow_dates, [rate_guess])	Calculates the internal rate of return of an investment based on a specified series of potentially irregularly spaced cash flows.
Financial	XNPV	XNPV(discount, cashflow_amounts, cashflow_dates)	Calculates the net present value of an investment based on a specified series of potentially irregularly spaced cash flows and a discount rate.
Financial	YIELD	YIELD(settlement, maturity, rate, price, redemption, frequency, [day_count_convention])	Calculates the annual yield of a security paying periodic interest, such as a US Treasury Bond, based on price.
Financial	YIELDDISC	YIELDDISC(settlement, maturity, price, redemption, [day_count_convention])	Calculates the annual yield of a discount (non-interest-bearing) security, based on price.
Financial	YIELDMAT	YIELDMAT(settlement, maturity, issue, rate, price, [day_count_convention])	Calculates the annual yield of a security paying interest at maturity, based on price.

TypeNameSyntaxDescription

Google	ARRAYFORMULA	ARRAYFORMULA(array_formula)	Enables the display of values returned from an array formula into multiple rows and/or columns and the use of non-array functions with arrays.
Google	DETECTLANGUAGE	DETECTLANGUAGE(text_or_range)	Identifies the language used in text within the specified range.
Google	GOOGLEFINANCE	GOOGLEFINANCE(ticker, [attribute], [start_date], [end_date\ num_days], [interval])	Fetches current or historical securities information from Google Finance.
Google	GOOGLETRANSLATE	GOOGLETRANSLATE(text, [source_language], [target_language])	Translates text from one language into another
Google	IMAGE	IMAGE(url, [mode], [height], [width])	Inserts an image into a cell.
Google	QUERY	QUERY(data, query, [headers])	Runs a Google Visualization API Query Language query across data.
Google	SPARKLINE	SPARKLINE(data, [options])	Creates a miniature chart contained within a single cell.

5.Google

Type	Name	Syntax	Description
Google	ARRAYFORMULA	ARRAYFORMULA(array_formula)	Enables the display of values returned from an array formula into multiple rows and/or columns and the use of non-

Type	Name	Syntax	Description
			array functions with arrays.
Google	DETECTLANGUAGE	DETECTLANGUAGE(text_or_range)	Identifies the language used in text within the specified range.
Google	GOOGLEFINANCE	GOOGLEFINANCE(ticker, [attribute], [start_date], [end_date\ num_days], [interval])	Fetches current or historical securities information from Google Finance.
Google	GOOGLETRANSLATE	GOOGLETRANSLATE(text, [source_language], [target_language])	Translates text from one language into another
Google	IMAGE	IMAGE(url, [mode], [height], [width])	Inserts an image into a cell.
Google	QUERY	QUERY(data, query, [headers])	Runs a Google Visualization API Query Language query across data.
Google	SPARKLINE	SPARKLINE(data, [options])	Creates a miniature chart contained within a single cell.

6. Info

Type	Name	Syntax	Description
Info	ERROR.TYPE	ERROR.TYPE(reference)	Returns a number corresponding to the error value in a different cell.
Info	ISBLANK	ISBLANK(value)	Checks whether the referenced cell is empty.
Info	ISDATE	ISDATE(value)	Returns whether a value is a date. _
Info	ISEMAIL	ISEMAIL(value)	Checks whether a value is a valid email address.
Info	ISERR	ISERR(value)	Checks whether a value is an error other than #N/A.
Info	ISERROR	ISERROR(value)	Checks whether a value is an error.
Info	ISFORMULA	ISFORMULA(cell)	Checks whether a formula is in the referenced cell.

Type	Name	Syntax	Description
Info	ISLOGICAL	ISLOGICAL(value)	Checks whether a value is <code>TRUE</code> or <code>FALSE</code> .
Info	ISNA	ISNA(value)	Checks whether a value is the error <code>#N/A</code> .
Info	ISNONTEXT	ISNONTEXT(value)	Checks whether a value is non-textual.
Info	ISNUMBER	ISNUMBER(value)	Checks whether a value is a number.
Info	ISREF	ISREF(value)	Checks whether a value is a valid cell reference.
Info	ISTEXT	ISTEXT(value)	Checks whether a value is text.
Info	N	N(value)	Returns the argument provided as a number.
Info	NA	NA()	Returns the "value not available" error, <code>#N/A</code> .
Info	TYPE	TYPE(value)	Returns a number associated with the type of data passed into the function.
Info	CELL	CELL(info_type, reference)	Returns the requested information about the specified cell.

7.Logical

Type	Name	Syntax	Description
Logical	AND	AND(logical_expression1, [logical_expression2, ...])	Returns true if all of the provided arguments are logically true, and false if any of the provided arguments are logically false.
Logical	FALSE	FALSE()	Returns the logical value <code>FALSE</code> .
Logical	IF	IF(logical_expression, value_if_true, value_if_false)	Returns one value if a logical expression is <code>TRUE</code> and another if it is <code>FALSE</code> .
Logical	IFERROR	IFERROR(value, [value_if_error])	Returns the first argument if it is not an error value, otherwise returns the second argument if present, or a blank if the second argument is absent.
Logical	IFNA	IFNA(value, value_if_na)	Evaluates a value. If the value is an <code>#N/A</code> error, returns the specified value. .

Type	Name	Syntax	Description
Logical	IFS	IFS(condition1, value1, [condition2, value2], ...)	Evaluates multiple conditions and returns a value that corresponds to the first true condition. ...
Logical	LAMBDA	LAMBDA(name, formula_expression)	Creates and returns a custom function with a set of names and a formula_expression that uses them. To calculate the formula_expression, you can call the returned function with as many values as the name declares.
Logical	LET	LET(name1, value_expression1, [name2, ...], [value_expression2, ...], formula_expression)	Assigns name with the value_expression results and returns the result of the formula_expression. The formula_expression can use the names defined in the scope of the LET function. The value_expressions are evaluated only once in the LET function even if the following value_expressions or the formula_expression use them multiple times.
Logical	NOT	NOT(logical_expression)	Returns the opposite of a logical value - NOT(TRUE) returns FALSE ; NOT(FALSE) returns TRUE .
Logical	OR	OR(logical_expression1, [logical_expression2, ...])	Returns true if any of the provided arguments are logically true, and false if all of the provided arguments are logically false.
Logical	SWITCH	SWITCH(expression, case1, value1, [default or case2, value2], ...)	Tests an expression against a list of cases and returns the corresponding value of the first matching case, with an optional default value if nothing else is met.
Logical	TRUE	TRUE()	Returns the logical value TRUE .
Logical	XOR	XOR(logical_expression1, [logical_expression2, ...])	The XOR function forms an exclusive or of 2 numbers that returns a 1 if the numbers are different, and a 0 otherwise. .

7.Lookup

Type	Name	Syntax	Description
Lookup	ADDRESS	ADDRESS(row, column, [absolute_relative_mode], [use_a1_notation], [sheet])	Returns a cell reference as a string.
Lookup	CHOOSE	CHOOSE(index, choice1, [choice2, ...])	Returns an element from a list of choices based on index.
Lookup	COLUMN	COLUMN([cell_reference])	Returns the column number of a specified cell, with A=1 .
Lookup	COLUMNS	COLUMNS(range)	Returns the number of columns in a specified array or range.
Lookup	FORMULATEXT	FORMULATEXT(cell)	Returns the formula as a string. .
Lookup	GETPIVOTDATA	GETPIVOTDATA(value_name, any_pivot_table_cell, [original_column, ...], [pivot_item, ...])	Extracts an aggregated value from a pivot table that corresponds to the specified row and column headings.
Lookup	HLOOKUP	HLOOKUP(search_key, range, index, [is_sorted])	Horizontal lookup. Searches across the first row of a range for a key and returns the value of a specified cell in the column found.
Lookup	INDEX	INDEX(reference, [row], [column])	Returns the content of a cell, specified by row and column offset.
Lookup	INDIRECT	INDIRECT(cell_reference_as_string, [is_A1_notation])	Returns a cell reference

Type	Name	Syntax	Description
			specified by a string.
Lookup	LOOKUP	LOOKUP(search_key, search_range\ search_result_array, [result_range])	Looks through a row or column for a key and returns the value of the cell in a result range located in the same position as the search row or column.
Lookup	MATCH	MATCH(search_key, range, [search_type])	Returns the relative position of an item in a range that matches a specified value.
Lookup	OFFSET	OFFSET(cell_reference, offset_rows, offset_columns, [height], [width])	Returns a range reference shifted a specified number of rows and columns from a starting cell reference.
Lookup	ROW	ROW([cell_reference])	Returns the row number of a specified cell.
Lookup	ROWS	ROWS(range)	Returns the number of rows in a specified array or range.
Lookup	VLOOKUP	VLOOKUP(search_key, range, index, [is_sorted])	Vertical lookup. Searches down the first column of a range for a key and returns the value of a specified cell in the row found.
Lookup	XLOOKUP	XLOOKUP(search_key, lookup_range, result_range, missing_value, [match_mode], [search_mode])	Returns the values in the result range based on the position where a

Type	Name	Syntax	Description
			match was found in the lookup range. If no match is found, it returns the closest match.

8. Math

Type	Name	Syntax	Description
Math	ABS	<code>ABS(value)</code>	Returns the absolute value of a number.
Math	ACOS	<code>ACOS(value)</code>	Returns the inverse cosine of a value, in radians.
Math	ACOSH	<code>ACOSH(value)</code>	Returns the inverse hyperbolic cosine of a number.
Math	ACOT	<code>ACOT(value)</code>	Returns the inverse cotangent of a value, in radians. .
Math	ACOTH	<code>ACOTH(value)</code>	Returns the inverse hyperbolic cotangent of a value, in radians. Must not be between -1 and 1, inclusive. .
Math	ASIN	<code>ASIN(value)</code>	Returns the inverse sine of a value, in radians.
Math	ASINH	<code>ASINH(value)</code>	Returns the inverse hyperbolic sine of a number.
Math	ATAN	<code>ATAN(value)</code>	Returns the inverse tangent of a value, in radians.
Math	ATAN2	<code>ATAN2(x, y)</code>	Returns the angle between the x-axis and a line segment from the origin (0,0) to specified coordinate pair (<code>x</code> , <code>y</code>), in radians.

Type	Name	Syntax	Description
Math	ATANH	ATANH(value)	Returns the inverse hyperbolic tangent of a number.
Math	BASE	BASE(value, base, [min_length])	Converts a number into a text representation in another base, for example, base 2 for binary. .
Math	CEILING	CEILING(value, [factor])	Rounds a number up to the nearest integer multiple of specified significance.
Math	CEILING.MATH	CEILING.MATH(number, [significance], [mode])	Rounds a number up to the nearest integer multiple of specified significance, with negative numbers rounding toward or away from 0 depending on the mode. .
Math	CEILING.PRECISE	CEILING.PRECISE(number, [significance])	Rounds a number up to the nearest integer multiple of specified significance. If the number is positive or negative, it is rounded up. .
Math	COMBIN	COMBIN(n, k)	Returns the number of ways to choose some number of objects from a pool of a given size of objects.
Math	COMBINA	COMBINA(n, k)	Returns the number of ways to choose some number of objects from a pool of a given size of objects, including ways that choose the same object multiple times. .
Math	COS	COS(angle)	Returns the cosine of an angle provided in radians.

Type	Name	Syntax	Description
Math	COSH	<code>COSH(value)</code>	Returns the hyperbolic cosine of any real number.
Math	COT	<code>COT(angle)</code>	Cotangent of an angle provided in radians. .
Math	COTH	<code>COTH(value)</code>	Returns the hyperbolic cotangent of any real number. .
Math	COUNTBLANK	<code>COUNTBLANK(range)</code>	Returns the number of empty cells in a given range.
Math	COUNTIF	<code>COUNTIF(range, criterion)</code>	Returns a conditional count across a range.
Math	COUNTIFS	<code>COUNTIFS(criteria_range1, criterion1, [criteria_range2, criterion2, ...])</code>	Returns the count of a range depending on multiple criteria.
Math	COUNTUNIQUE	<code>COUNTUNIQUE(value1, [value2, ...])</code>	Counts the number of unique values in a list of specified values and ranges.
Math	CSC	<code>CSC(angle)</code>	Returns the cosecant of an angle provided in radians. .
Math	CSCH	<code>CSCH(value)</code>	The CSCH function returns the hyperbolic cosecant of any real number. .
Math	DECIMAL	<code>DECIMAL(value, base)</code>	The DECIMAL function converts the text representation of a number in another base, to base 10 (decimal). .
Math	DEGREES	<code>DEGREES(angle)</code>	Converts an angle value in radians to degrees.
Math	C	<code>C(z)</code>	Returns the complementary Gauss error function of a value.
Math	C.PRECISE	<code>C.PRECISE(z)</code>	See C

Type	Name	Syntax	Description
Math	EVEN	<code>EVEN(value)</code>	Rounds a number up to the nearest even integer.
Math	EXP	<code>EXP(exponent)</code>	Returns Euler's number, e (~2.718) raised to a power.
Math	FACT	<code>FACT(value)</code>	Returns the factorial of a number.
Math	FACTDOUBLE	<code>FACTDOUBLE(value)</code>	Returns the "double factorial" of a number.
Math	FLOOR	<code>FLOOR(value, [factor])</code>	Rounds a number down to the nearest integer multiple of specified significance.
Math	FLOOR.MATH	<code>FLOOR.MATH(number, [significance], [mode])</code>	Rounds a number down to the nearest integer multiple of specified significance, with negative numbers rounding toward or away from 0 depending on the mode. ↗
Math	FLOOR.PRECISE	<code>FLOOR.PRECISE(number, [significance])</code>	The FLOOR.PRECISE function rounds a number down to the nearest integer or multiple of specified significance. ↗
Math	GAMMALN	<code>GAMMALN(value)</code>	Returns the the logarithm of a specified Gamma function, base e (Euler's number).
Math	GAMMALN.PRECISE	<code>GAMMALN.PRECISE(value)</code>	See GAMMALN
Math	GCD	<code>GCD(value1, value2)</code>	Returns the greatest common divisor of one or more integers.
Math	IMLN	<code>IMLN(complex_value)</code>	Returns the logarithm of a complex number, base e (Euler's number).
Math	IMPOWER	<code>IMPOWER(complex_base, exponent)</code>	Returns a complex number raised to a power.

Type	Name	Syntax	Description
Math	IMSQRT	IMSQRT(<i>complex_number</i>)	Computes the square root of a complex number.
Math	INT	INT(<i>value</i>)	Rounds a number down to the nearest integer that is less than or equal to it.
Math	ISEVEN	ISEVEN(<i>value</i>)	Checks whether the provided value is even.
Math	ISO.CEILING	ISO.CEILING(<i>number</i> , [<i>significance</i>])	See CEILING.PRECISE
Math	ISODD	ISODD(<i>value</i>)	Checks whether the provided value is odd.
Math	LCM	LCM(<i>value1</i> , <i>value2</i>)	Returns the least common multiple of one or more integers.
Math	LN	LN(<i>value</i>)	Returns the the logarithm of a number, base e (Euler's number).
Math	LOG	LOG(<i>value</i> , <i>base</i>)	Returns the the logarithm of a number given a base.
Math	LOG10	LOG10(<i>value</i>)	Returns the the logarithm of a number, base 10.
Math	MOD	MOD(<i>dividend</i> , <i>divisor</i>)	Returns the result of the modulo operator, the remainder after a division operation.
Math	MROUND	MROUND(<i>value</i> , <i>factor</i>)	Rounds one number to the nearest integer multiple of another.
Math	MULTINOMIAL	MULTINOMIAL(<i>value1</i> , <i>value2</i>)	Returns the factorial of the sum of values divided by the product of the values' factorials.
Math	MUNIT	MUNIT(<i>dimension</i>)	Returns a unit matrix of size dimension x dimension. .
Math	ODD	ODD(<i>value</i>)	Rounds a number up to the nearest odd integer.

Type	Name	Syntax	Description
Math	PI	PI()	Returns the value of Pi to 14 decimal places.
Math	POWER	POWER(base, exponent)	Returns a number raised to a power.
Math	PRODUCT	PRODUCT(factor1, [factor2, ...])	Returns the result of multiplying a series of numbers together.
Math	QUOTIENT	QUOTIENT(dividend, divisor)	Returns one number divided by another.
Math	RADIANS	RADIANS(angle)	Converts an angle value in degrees to radians.
Math	RAND	RAND()	Returns a random number between 0 inclusive and 1 exclusive.
Math	RANDARRAY	RANDARRAY(rows, columns)	Generates an array of random numbers between 0 and 1. .
Math	RANDBETWEEN	RANDBETWEEN(low, high)	Returns a uniformly random integer between two values, inclusive.
Math	ROUND	ROUND(value, [places])	Rounds a number to a certain number of decimal places according to standard rules.
Math	ROUNDDOWN	ROUNDDOWN(value, [places])	Rounds a number to a certain number of decimal places, always rounding down to the next valid increment.
Math	ROUNDUP	ROUNDUP(value, [places])	Rounds a number to a certain number of decimal places, always rounding up to the next valid increment.
Math	SEC	SEC(angle)	The SEC function returns the secant of an angle, measured in radians. .
Math	SECH	SECH(value)	The SECH function returns the hyperbolic secant of an angle.

Type	Name	Syntax	Description
Math	SEQUENCE	SEQUENCE(rows, columns, start, step)	Returns an array of sequential numbers, such as 1, 2, 3, 4. .
Math	SERIESSUM	SERIESSUM(x, n, m, a)	Given parameters x , n , m , and a , returns the power series sum $a_1x^n + a_2x^{n+m} + \dots + a_ix^{n+(i-1)m}$, where i is the number of entries in range a .
Math	SIGN	SIGN(value)	Given an input number, returns -1 if it is negative, 1 if positive, and 0 if it is zero.
Math	SIN	SIN(angle)	Returns the sine of an angle provided in radians.
Math	SINH	SINH(value)	Returns the hyperbolic sine of any real number.
Math	SQRT	SQRT(value)	Returns the positive square root of a positive number.
Math	SQRTPI	SQRTPI(value)	Returns the positive square root of the product of Pi and the given positive number.
Math	SUBTOTAL	SUBTOTAL(function_code, range1, [range2, ...])	Returns a subtotal for a vertical range of cells using a specified aggregation function.
Math	SUM	SUM(value1, [value2, ...])	Returns the sum of a series of numbers and/or cells.
Math	SUMIF	SUMIF(range, criterion, [sum_range])	Returns a conditional sum across a range.
Math	SUMIFS	SUMIFS(sum_range, criteria_range1, criterion1, [criteria_range2, criterion2, ...])	Returns the sum of a range depending on multiple criteria.
Math	SUMSQ	SUMSQ(value1, [value2, ...])	Returns the sum of the squares of a series of

Type	Name	Syntax	Description
			numbers and/or cells.
Math	TAN	TAN(angle)	Returns the tangent of an angle provided in radians.
Math	TANH	TANH(value)	Returns the hyperbolic tangent of any real number.
Math	TRUNC	TRUNC(value, [places])	Truncates a number to a certain number of significant digits by omitting less significant digits.

9. Operator

Type	Name	Syntax	Description
Operator	ADD	ADD(value1, value2)	Returns the sum of two numbers. Equivalent to the + operator.
Operator	CONCAT	CONCAT(value1, value2)	Returns the concatenation of two values. Equivalent to the & operator.
Operator	DIVIDE	DIVIDE(dividend, divisor)	Returns one number divided by another. Equivalent to the / operator.
Operator	EQ	EQ(value1, value2)	Returns TRUE if two specified values are equal and FALSE otherwise. Equivalent to the = operator.
Operator	GT	GT(value1, value2)	Returns TRUE if the first argument is strictly greater than the second, and FALSE otherwise. Equivalent to the > operator.
Operator	GTE	GTE(value1, value2)	Returns TRUE if the first argument is greater than or equal to the second, and

Type	Name	Syntax	Description
			<code>FALSE</code> otherwise. Equivalent to the <code>>=</code> operator.
Operator	ISBETWEEN	<code>ISBETWEEN(value_to_compare, lower_value, upper_value, lower_value_is_inclusive, upper_value_is_inclusive)</code>	Checks whether a provided number is between two other numbers either inclusively or exclusively.
Operator	LT	<code>LT(value1, value2)</code>	Returns <code>TRUE</code> if the first argument is strictly less than the second, and <code>FALSE</code> otherwise. Equivalent to the <code><</code> operator.
Operator	LTE	<code>LTE(value1, value2)</code>	Returns <code>TRUE</code> if the first argument is less than or equal to the second, and <code>FALSE</code> otherwise. Equivalent to the <code><=</code> operator.
Operator	MINUS	<code>MINUS(value1, value2)</code>	Returns the difference of two numbers. Equivalent to the <code>-</code> operator.
Operator	MULTIPLY	<code>MULTIPLY(factor1, factor2)</code>	Returns the product of two numbers. Equivalent to the <code>*</code> operator.
Operator	NE	<code>NE(value1, value2)</code>	Returns <code>TRUE</code> if two specified values are not equal and <code>FALSE</code> otherwise. Equivalent to the <code><></code> operator.
Operator	POW	<code>POW(base, exponent)</code>	Returns a number raised to a power.
Operator	UMINUS	<code>UMINUS(value)</code>	Returns a number with the sign reversed.
Operator	UNARY_PERCENT	<code>UNARY_PERCENT(percentage)</code>	Returns a value interpreted as a percentage; that is,

Type	Name	Syntax	Description
			UNARY_PERCENT(100) equals 1 .
Operator	UNIQUE	UNIQUE(range, by_column, exactly_once)	Returns unique rows in the provided source range, discarding duplicates. Rows are returned in the order in which they first appear in the source range.
Operator	UPLUS	UPLUS(value)	Returns a specified number, unchanged.

10.Statistical

Type	Name	Syntax	Description
Statistical	AVEDEV	AVEDEV(value1, [value2, ...])	Calculates the average of the magnitudes of deviations of data from a dataset's mean.
Statistical	AVERAGE	AVERAGE(value1, [value2, ...])	Returns the numerical average value in a dataset, ignoring text.
Statistical	AVERAGE.WEIGHTED	AVERAGE.WEIGHTED(values, weights, [additional values], [additional weights])	Finds the weighted average of a set of values, given the values and the corresponding weights. .
Statistical	AVERAGEA	AVERAGEA(value1, [value2, ...])	Returns the numerical average value in a dataset.
Statistical	AVERAGEIF	AVERAGEIF(criteria_range, criterion, [average_range])	Returns the average of a range depending on criteria.
Statistical	AVERAGEIFS	AVERAGEIFS(average_range, criteria_range1, criterion1,	Returns the average of a range depending on multiple criteria.

Type	Name	Syntax	Description
		[criteria_range2, criterion2, ...])	
Statistical	BETA.DIST	BETA.DIST(value, alpha, beta, cumulative, lower_bound, upper_bound)	Returns the probability of a given value as defined by the beta distribution function. .
Statistical	BETA.INV	BETA.INV(probability, alpha, beta, lower_bound, upper_bound)	Returns the value of the inverse beta distribution function for a given probability. .
Statistical	BETADIST	BETADIST(value, alpha, beta, lower_bound, upper_bound)	See BETA.DIST .
Statistical	BETAINV	BETAINV(probability, alpha, beta, lower_bound, upper_bound)	See BETA.INV
Statistical	BINOM.DIST	BINOM.DIST(num_successes, num_trials, prob_success, cumulative)	See BINOMDIST
Statistical	BINOM.INV	BINOM.INV(num_trials, prob_success, target_prob)	See CRITBINOM
Statistical	BINOMDIST	BINOMDIST(num_successes, num_trials, prob_success, cumulative)	Calculates the probability of drawing a certain number of successes (or a maximum number of successes) in a certain number of tries given a population of a certain size containing a certain number of successes, with replacement of draws.
Statistical	CHIDIST	CHIDIST(x, degrees_freedom)	Calculates the right-tailed chi-squared distribution, often

Type	Name	Syntax	Description
			used in hypothesis testing.
Statistical	CHIINV	CHIINV(probability, degrees_freedom)	Calculates the inverse of the right-tailed chi-squared distribution.
Statistical	CHISQ.DIST	CHISQ.DIST(x, degrees_freedom, cumulative)	Calculates the left-tailed chi-squared distribution, often used in hypothesis testing.
Statistical	CHISQ.DIST.RT	CHISQ.DIST.RT(x, degrees_freedom)	Calculates the right-tailed chi-squared distribution, which is commonly used in hypothesis testing.
Statistical	CHISQ.INV	CHISQ.INV(probability, degrees_freedom)	Calculates the inverse of the left-tailed chi-squared distribution.
Statistical	CHISQ.INV.RT	CHISQ.INV.RT(probability, degrees_freedom)	Calculates the inverse of the right-tailed chi-squared distribution.
Statistical	CHISQ.TEST	CHISQ.TEST(observed_range, expected_range)	See CHITEST
Statistical	CHITEST	CHITEST(observed_range, expected_range)	Returns the probability associated with a Pearson's chi-squared test on the two ranges of data. Determines the likelihood that the observed categorical data is drawn from an expected distribution.
Statistical	CONFIDENCE	CONFIDENCE(alpha, standard_deviation, pop_size)	See CONFIDENCE.NORM
Statistical	CONFIDENCE.NORM	CONFIDENCE.NORM(alpha, standard_deviation,	Calculates the width of half the confidence

Type	Name	Syntax	Description
		pop_size)	interval for a normal distribution. .
Statistical	CONFIDENCE.T	CONFIDENCE.T(alpha, standard_deviation, size)	Calculates the width of half the confidence interval for a Student's t-distribution. .
Statistical	CORREL	CORREL(data_y, data_x)	Calculates r, the Pearson product-moment correlation coefficient of a dataset.
Statistical	COUNT	COUNT(value1, [value2, ...])	Returns a count of the number of numeric values in a dataset.
Statistical	COUNTA	COUNTA(value1, [value2, ...])	Returns a count of the number of values in a dataset.
Statistical	COVAR	COVAR(data_y, data_x)	Calculates the covariance of a dataset.
Statistical	COVARIANCE.P	COVARIANCE.P(data_y, data_x)	See COVAR
Statistical	COVARIANCE.S	COVARIANCE.S(data_y, data_x)	Calculates the covariance of a dataset, where the dataset is a sample of the total population. .
Statistical	CRITBINOM	CRITBINOM(num_trials, prob_success, target_prob)	Calculates the smallest value for which the cumulative binomial distribution is greater than or equal to a specified criteria.
Statistical	DEVSQ	DEVSQ(value1, value2)	Calculates the sum of squares of deviations based on a sample.
Statistical	EXPON.DIST	EXPON.DIST(x, LAMBDA, cumulative)	Returns the value of the exponential

Type	Name	Syntax	Description
			distribution function with a specified LAMBDA at a specified value. .
Statistical	EXPONDIST	EXPONDIST(x, LAMBDA, cumulative)	See EXPON.DIST
Statistical	F.DIST	F.DIST(x, degrees_freedom1, degrees_freedom2, cumulative)	Calculates the left-tailed F probability distribution (degree of diversity) for two data sets with given input x. Alternately called Fisher-Snedecor distribution or Snedecor's F distribution.
Statistical	F.DIST.RT	F.DIST.RT(x, degrees_freedom1, degrees_freedom2)	Calculates the right-tailed F probability distribution (degree of diversity) for two data sets with given input x. Alternately called Fisher-Snedecor distribution or Snedecor's F distribution.
Statistical	F.INV	F.INV(probability, degrees_freedom1, degrees_freedom2)	Calculates the inverse of the left-tailed F probability distribution. Also called the Fisher-Snedecor distribution or Snedecor's F distribution.
Statistical	F.INV.RT	F.INV.RT(probability, degrees_freedom1, degrees_freedom2)	Calculates the inverse of the right-tailed F probability distribution. Also called the Fisher-Snedecor distribution or Snedecor's F distribution.

Type	Name	Syntax	Description
Statistical	F.TEST	<code>F.TEST(range1, range2)</code>	See F.TEST .
Statistical	FDIST	<code>FDIST(x, degrees_freedom1, degrees_freedom2)</code>	See F.DIST.RT .
Statistical	FINV	<code>FINV(probability, degrees_freedom1, degrees_freedom2)</code>	See F.INV.RT
Statistical	FISHER	<code>FISHER(value)</code>	Returns the Fisher transformation of a specified value.
Statistical	FISHERINV	<code>FISHERINV(value)</code>	Returns the inverse Fisher transformation of a specified value.
Statistical	FORECAST	<code>FORECAST(x, data_y, data_x)</code>	Calculates the expected y-value for a specified x based on a linear regression of a dataset.
Statistical	FORECAST.LINEAR	<code>FORECAST.LINEAR(x, data_y, data_x)</code>	See FORECAST
Statistical	FTEST	<code>FTEST(range1, range2)</code>	Returns the probability associated with an F-test for equality of variances. Determines whether two samples are likely to have come from populations with the same variance.
Statistical	GAMMA	<code>GAMMA(number)</code>	Returns the Gamma function evaluated at the specified value. .
Statistical	GAMMA.DIST	<code>GAMMA.DIST(x, alpha, beta, cumulative)</code>	Calculates the gamma distribution, a two-parameter continuous probability distribution.
Statistical	GAMMA.INV	<code>GAMMA.INV(probability, alpha, beta)</code>	The GAMMA.INV function returns the value of the inverse gamma cumulative

Type	Name	Syntax	Description
			distribution function for the specified probability and alpha and beta parameters. .
Statistical	GAMMADIST	GAMMADIST(x, alpha, beta, cumulative)	See GAMMA.DIST
Statistical	GAMMAINV	GAMMAINV(probability, alpha, beta)	See GAMMA.INV .
Statistical	GAUSS	GAUSS(z)	The GAUSS function returns the probability that a random variable, drawn from a normal distribution, will be between the mean and z standard deviations above (or below) the mean. .
Statistical	GEOMEAN	GEOMEAN(value1, value2)	Calculates the geometric mean of a dataset.
Statistical	HARMEAN	HARMEAN(value1, value2)	Calculates the harmonic mean of a dataset.
Statistical	HYPGEOM.DIST	HYPGEOM.DIST(num_successes, num_draws, successes_in_pop, pop_size)	See HYPGEOMDIST
Statistical	HYPGEOMDIST	HYPGEOMDIST(num_successes, num_draws, successes_in_pop, pop_size)	Calculates the probability of drawing a certain number of successes in a certain number of tries given a population of a certain size containing a certain number of successes, without replacement of draws.
Statistical	INTERCEPT	INTERCEPT(data_y, data_x)	Calculates the y-value at which the

Type	Name	Syntax	Description
			line resulting from linear regression of a dataset will intersect the y-axis (x=0).
Statistical	KURT	<code>KURT(value1, value2)</code>	Calculates the kurtosis of a dataset, which describes the shape, and in particular the "peakedness" of that dataset.
Statistical	LARGE	<code>LARGE(data, n)</code>	Returns the nth largest element from a data set, where n is user-defined.
Statistical	LOGINV	<code>LOGINV(x, mean, standard_deviation)</code>	Returns the value of the inverse log-normal cumulative distribution with given mean and standard deviation at a specified value.
Statistical	LOGNORM.DIST	<code>LOGNORM.DIST(x, mean, standard_deviation)</code>	See LOGNORMDIST
Statistical	LOGNORM.INV	<code>LOGNORM.INV(x, mean, standard_deviation)</code>	See LOGINV
Statistical	LOGNORMDIST	<code>LOGNORMDIST(x, mean, standard_deviation)</code>	Returns the value of the log-normal cumulative distribution with given mean and standard deviation at a specified value.
Statistical	MARGINOFERROR	<code>MARGINOFERROR(range, confidence)</code>	Calculates the amount of random sampling error given a range of values and a confidence level.
Statistical	MAX	<code>MAX(value1, [value2, ...])</code>	Returns the maximum value in a numeric dataset.

Type	Name	Syntax	Description
Statistical	MAXA	<code>MAXA(value1, value2)</code>	Returns the maximum numeric value in a dataset.
Statistical	MAXIFS	<code>MAXIFS(range, criteria_range1, criterion1, [criteria_range2, criterion2], ...)</code>	Returns the maximum value in a range of cells, filtered by a set of criteria. ↗
Statistical	MEDIAN	<code>MEDIAN(value1, [value2, ...])</code>	Returns the median value in a numeric dataset.
Statistical	MIN	<code>MIN(value1, [value2, ...])</code>	Returns the minimum value in a numeric dataset.
Statistical	MINA	<code>MINA(value1, value2)</code>	Returns the minimum numeric value in a dataset.
Statistical	MINIFS	<code>MINIFS(range, criteria_range1, criterion1, [criteria_range2, criterion2], ...)</code>	Returns the minimum value in a range of cells, filtered by a set of criteria. ↗
Statistical	MODE	<code>MODE(value1, [value2, ...])</code>	Returns the most commonly occurring value in a dataset.
Statistical	MODE.MULT	<code>MODE.MULT(value1, value2)</code>	Returns the most commonly occurring values in a dataset. ↗
Statistical	MODE.SNGL	<code>MODE.SNGL(value1, [value2, ...])</code>	See MODE
Statistical	NEGBINOM.DIST	<code>NEGBINOM.DIST(num_failures, num_successes, prob_success)</code>	See NEGBINOMDIST
Statistical	NEGBINOMDIST	<code>NEGBINOMDIST(num_failures, num_successes, prob_success)</code>	Calculates the probability of drawing a certain number of failures before a certain number of successes given a probability of success in independent trials.

Type	Name	Syntax	Description
Statistical	NORM.DIST	<code>NORM.DIST(x, mean, standard_deviation, cumulative)</code>	See NORMDIST
Statistical	NORM.INV	<code>NORM.INV(x, mean, standard_deviation)</code>	See NORMINV
Statistical	NORM.S.DIST	<code>NORM.S.DIST(x)</code>	See NORMSDIST
Statistical	NORM.S.INV	<code>NORM.S.INV(x)</code>	See NORMSINV
Statistical	NORMDIST	<code>NORMDIST(x, mean, standard_deviation, cumulative)</code>	Returns the value of the normal distribution function (or normal cumulative distribution function) for a specified value, mean, and standard deviation.
Statistical	NORMINV	<code>NORMINV(x, mean, standard_deviation)</code>	Returns the value of the inverse normal distribution function for a specified value, mean, and standard deviation.
Statistical	NORMSDIST	<code>NORMSDIST(x)</code>	Returns the value of the standard normal cumulative distribution function for a specified value.
Statistical	NORMSINV	<code>NORMSINV(x)</code>	Returns the value of the inverse standard normal distribution function for a specified value.
Statistical	PEARSON	<code>PEARSON(data_y, data_x)</code>	Calculates r, the Pearson product-moment correlation coefficient of a dataset.
Statistical	PERCENTILE	<code>PERCENTILE(data, percentile)</code>	Returns the value at a given percentile of a dataset.
Statistical	PERCENTILE.EXC	<code>PERCENTILE.EXC(data, percentile)</code>	Returns the value at a given percentile of

Type	Name	Syntax	Description
			a dataset, exclusive of 0 and 1. .
Statistical	PERCENTILE.INC	PERCENTILE.INC(data, percentile)	See PERCENTILE
Statistical	PERCENTRANK	PERCENTRANK(data, value, [significant_digits])	Returns the percentage rank (percentile) of a specified value in a dataset.
Statistical	PERCENTRANK.EXC	PERCENTRANK.EXC(data, value, [significant_digits])	Returns the percentage rank (percentile) from 0 to 1 exclusive of a specified value in a dataset.
Statistical	PERCENTRANK.INC	PERCENTRANK.INC(data, value, [significant_digits])	Returns the percentage rank (percentile) from 0 to 1 inclusive of a specified value in a dataset.
Statistical	PERMUTATIONA	PERMUTATIONA(number, number_chosen)	Returns the number of permutations for selecting a group of objects (with replacement) from a total number of objects. .
Statistical	PERMUT	PERMUT(n, k)	Returns the number of ways to choose some number of objects from a pool of a given size of objects, considering order.
Statistical	PHI	PHI(x)	The PHI function returns the value of the normal distribution with mean 0 and standard deviation 1. .
Statistical	POISSON	POISSON(x, mean, cumulative)	See POISSON.DIST

Type	Name	Syntax	Description
Statistical	POISSON.DIST	POISSON.DIST(x, mean, [cumulative])	Returns the value of the Poisson distribution function (or Poisson cumulative distribution function) for a specified value and mean. .
Statistical	PROB	PROB(data, probabilities, low_limit, [high_limit])	Given a set of values and corresponding probabilities, calculates the probability that a value chosen at random falls between two limits.
Statistical	QUARTILE	QUARTILE(data, quartile_number)	Returns a value nearest to a specified quartile of a dataset.
Statistical	QUARTILE.EXC	QUARTILE.EXC(data, quartile_number)	Returns value nearest to a given quartile of a dataset, exclusive of 0 and 4. .
Statistical	QUARTILE.INC	QUARTILE.INC(data, quartile_number)	See QUARTILE
Statistical	RANK	RANK(value, data, [is_ascending])	Returns the rank of a specified value in a dataset.
Statistical	RANK.AVG	RANK.AVG(value, data, [is_ascending])	Returns the rank of a specified value in a dataset. If there is more than one entry of the same value in the dataset, the average rank of the entries will be returned.
Statistical	RANK.EQ	RANK.EQ(value, data, [is_ascending])	Returns the rank of a specified value in a dataset. If there is more than one entry of the same value in

Type	Name	Syntax	Description
			the dataset, the top rank of the entries will be returned.
Statistical	RSQ	<code>RSQ(data_y, data_x)</code>	Calculates the square of r, the Pearson product-moment correlation coefficient of a dataset.
Statistical	SKEW	<code>SKEW(value1, value2)</code>	Calculates the skewness of a dataset, which describes the symmetry of that dataset about the mean.
Statistical	SKEW.P	<code>SKEW.P(value1, value2)</code>	Calculates the skewness of a dataset that represents the entire population. .
Statistical	SLOPE	<code>SLOPE(data_y, data_x)</code>	Calculates the slope of the line resulting from linear regression of a dataset.
Statistical	SMALL	<code>SMALL(data, n)</code>	Returns the nth smallest element from a data set, where n is user-defined.
Statistical	STANDARDIZE	<code>STANDARDIZE(value, mean, standard_deviation)</code>	Calculates the normalized equivalent of a random variable given mean and standard deviation of the distribution.
Statistical	STDEV	<code>STDEV(value1, [value2, ...])</code>	Calculates the standard deviation based on a sample.

Type	Name	Syntax	Description
Statistical	STDEV.P	STDEV.P(value1, [value2, ...])	See STDEVP
Statistical	STDEV.S	STDEV.S(value1, [value2, ...])	See STDEV
Statistical	STDEVA	STDEVA(value1, value2)	Calculates the standard deviation based on a sample, setting text to the value 0.
Statistical	STDEVP	STDEVP(value1, value2)	Calculates the standard deviation based on an entire population.
Statistical	STDEVPA	STDEVPA(value1, value2)	Calculates the standard deviation based on an entire population, setting text to the value 0.
Statistical	STEYX	STEYX(data_y, data_x)	Calculates the standard error of the predicted y-value for each x in the regression of a dataset.
Statistical	T.DIST	T.DIST(x, degrees_freedom, cumulative)	Returns the right tailed Student distribution for a value x.
Statistical	T.DIST.2T	T.DIST.2T(x, degrees_freedom)	Returns the two tailed Student distribution for a value x.
Statistical	T.DIST.RT	T.DIST.RT(x, degrees_freedom)	Returns the right tailed Student distribution for a value x.
Statistical	T.INV	T.INV(probability, degrees_freedom)	Calculates the negative inverse of the one-tailed TDIST function.

Type	Name	Syntax	Description
Statistical	T.INV.2T	T.INV.2T(probability, degrees_freedom)	Calculates the inverse of the two-tailed TDIST function.
Statistical	T.TEST	T.TEST(range1, range2, tails, type)	Returns the probability associated with Student's t-test. Determines whether two samples are likely to have come from the same two underlying populations that have the same mean. .
Statistical	TDIST	TDIST(x, degrees_freedom, tails)	Calculates the probability for Student's t-distribution with a given input (x).
Statistical	TINV	TINV(probability, degrees_freedom)	See T.INV.2T
Statistical	TRIMMEAN	TRIMMEAN(data, exclude_proportion)	Calculates the mean of a dataset excluding some proportion of data from the high and low ends of the dataset.
Statistical	TTEST	TTEST(range1, range2, tails, type)	See T.TEST .
Statistical	VAR	VAR(value1, [value2, ...])	Calculates the variance based on a sample.
Statistical	VAR.P	VAR.P(value1, [value2, ...])	See VARP
Statistical	VAR.S	VAR.S(value1, [value2, ...])	See VAR
Statistical	VARA	VARA(value1, value2)	Calculates an estimate of variance based on a sample, setting text to the value 0 .
Statistical	VARP	VARP(value1, value2)	Calculates the variance based on an

Type	Name	Syntax	Description
			entire population.
Statistical	VARPA	<code>VARPA(value1, value2,...)</code>	Calculates the variance based on an entire population, setting text to the value 0.
Statistical	WEIBULL	<code>WEIBULL(x, shape, scale, cumulative)</code>	Returns the value of the Weibull distribution function (or Weibull cumulative distribution function) for a specified shape and scale.
Statistical	WEIBULL.DIST	<code>WEIBULL.DIST(x, shape, scale, cumulative)</code>	See WEIBULL
Statistical	Z.TEST	<code>Z.TEST(data, value, [standard_deviation])</code>	Returns the one-tailed P-value of a Z-test with standard distribution. .
Statistical	ZTEST	<code>ZTEST(data, value, [standard_deviation])</code>	See Z.TEST .

11. Text

Type	Name	Syntax	Description
Text	ARABIC	<code>ARABIC(roman_numeral)</code>	Computes the value of a Roman numeral.
Text	ASC	<code>ASC(text)</code>	Converts full-width ASCII and katakana characters to their half-width counterparts. All standard-width characters will remain unchanged. .
Text	CHAR	<code>CHAR(table_number)</code>	Convert a number into a character according to the current Unicode table.
Text	CLEAN	<code>CLEAN(text)</code>	Returns the text with the non-printable ASCII characters removed.

Type	Name	Syntax	Description
Text	CODE	<code>CODE(string)</code>	Returns the numeric Unicode map value of the first character in the string provided.
Text	CONCATENATE	<code>CONCATENATE(string1, [string2, ...])</code>	Appends strings to one another.
Text	DOLLAR	<code>DOLLAR(number, [number_of_places])</code>	Formats a number into the locale-specific currency format.
Text	EXACT	<code>EXACT(string1, string2)</code>	Tests whether two strings are identical.
Text	FIND	<code>FIND(search_for, text_to_search, [starting_at])</code>	Returns the position at which a string is first found within text.
Text	FINDB	<code>FINDB(search_for, text_to_search, [starting_at])</code>	Returns the position at which a string is first found within text counting each double-character as 2.
Text	FIXED	<code>FIXED(number, [number_of_places], [suppress_separator])</code>	Formats a number with a fixed number of decimal places.
Text	JOIN	<code>JOIN(delimiter, value_or_array1, [value_or_array2, ...])</code>	Concatenates the elements of one or more one-dimensional arrays using a specified delimiter.
Text	LEFT	<code>LEFT(string, [number_of_characters])</code>	Returns a substring from the beginning of a specified string.
Text	LEFTB	<code>LEFTB(string, num_of_bytes)</code>	Returns the left portion of a string up to a certain number of bytes. .
Text	LEN	<code>LEN(text)</code>	Returns the length of a string.
Text	LENB	<code>LENB(string)</code>	Returns the length of a string in bytes." .
Text	LOWER	<code>LOWER(text)</code>	Converts a specified string to lowercase.

Type	Name	Syntax	Description
Text	MID	MID(string, starting_at, extract_length)	Returns a segment of a string.
Text	MIDB	MIDB(string)	Returns a section of a string starting at a given character and up to a specified number of bytes. .
Text	PROPER	PROPER(text_to_capitalize)	Capitalizes each word in a specified string.
Text	REGEXEXTRACT	REGEXEXTRACT(text, regular_expression)	Extracts matching substrings according to a regular expression.
Text	REGEXMATCH	REGEXMATCH(text, regular_expression)	Whether a piece of text matches a regular expression.
Text	REGEXREPLACE	REGEXREPLACE(text, regular_expression, replacement)	Replaces part of a text string with a different text string using regular expressions.
Text	REPLACE	REPLACE(text, position, length, new_text)	Replaces part of a text string with a different text string.
Text	REPLACEB	REPLACEB(text, position, num_bytes, new_text)	Replaces part of a text string, based on a number of bytes, with a different text string. .
Text	REPT	REPT(text_to_repeat, number_of_repetitions)	Returns specified text repeated a number of times.
Text	RIGHT	RIGHT(string, [number_of_characters])	Returns a substring from the end of a specified string.
Text	RIGHTB	RIGHTB(string, num_of_bytes)	Returns the right portion of a string up to a certain number of bytes. .
Text	ROMAN	ROMAN(number, [rule_relaxation])	Formats a number in Roman numerals.
Text	SEARCH	SEARCH(search_for, text_to_search, [starting_at])	Returns the position at which a string is first found within text.

Type	Name	Syntax	Description
Text	SEARCHB	SEARCHB(search_for, text_to_search, [starting_at])	Returns the position at which a string is first found within text counting each double-character as 2.
Text	SPLIT	SPLIT(text, delimiter, [split_by_each], [remove_empty_text])	Divides text around a specified character or string, and puts each fragment into a separate cell in the row.
Text	SUBSTITUTE	SUBSTITUTE(text_to_search, search_for, replace_with, [occurrence_number])	Replaces existing text with new text in a string.
Text	T	T(value)	Returns string arguments as text.
Text	TEXT	TEXT(number, format)	Converts a number into text according to a specified format.
Text	TEXTJOIN	TEXTJOIN(delimiter, ignore_empty, text1, [text2], ...)	Combines the text from multiple strings and/or arrays, with a specifiable delimiter separating the different texts. ↗
Text	TRIM	TRIM(text)	Removes leading and trailing spaces in a specified string.
Text	UNICHAR	UNICHAR(number)	Returns the Unicode character for a number. ↗
Text	UNICODE	UNICODE(text)	Returns the decimal Unicode value of the first character of the text. ↗
Text	UPPER	UPPER(text)	Converts a specified string to uppercase.
Text	VALUE	VALUE(text)	Converts a string in any of the date, time or number formats that Google Sheets understands into a number.

12. Database

Type	Name	Syntax	Description
Database	DAVERAGE	DAVERAGE(database, field, criteria)	Returns the average of a set of values selected from a database table-like array or range using a SQL-like query.
Database	DCOUNT	DCOUNT(database, field, criteria)	Counts numeric values selected from a database table-like array or range using a SQL-like query.
Database	DCOUNTA	DCOUNTA(database, field, criteria)	Counts values, including text, selected from a database table-like array or range using a SQL-like query.
Database	DGET	DGET(database, field, criteria)	Returns a single value from a database table-like array or range using a SQL-like query.
Database	DMAX	DMAX(database, field, criteria)	Returns the maximum value selected from a database table-like array or range using a SQL-like query.
Database	DMIN	DMIN(database, field, criteria)	Returns the minimum value selected from a database table-like array or range using a SQL-like query.
Database	DPRODUCT	DPRODUCT(database, field, criteria)	Returns the product of values selected from a database table-like array or range using a SQL-like query.
Database	DSTDEV	DSTDEV(database, field, criteria)	Returns the standard deviation of a population sample selected from a database table-like array or range using a SQL-like query.
Database	DSTDEVP	DSTDEVP(database, field, criteria)	Returns the standard deviation of an entire population selected from a database table-like array or range using a SQL-like query.
Database	DSUM	DSUM(database, field, criteria)	Returns the sum of values selected from a database table-like array or range using a SQL-like query.
Database	DVAR	DVAR(database, field, criteria)	Returns the variance of a population sample selected from a database table-like array or range using a SQL-like query.

Type	Name	Syntax	Description
Database	DVARP	DVARP(database, field, criteria)	Returns the variance of an entire population selected from a database table-like array or range using a SQL-like query.

13.Parser

Type	Name	Syntax	Description
Parser	CONVERT	CONVERT(value, start_unit, end_unit)	Converts a numeric value to a different unit of measure.
Parser	TO_DATE	TO_DATE(value)	Converts a provided number to a date.
Parser	TO_DOLLARS	TO_DOLLARS(value)	Converts a provided number to a dollar value.
Parser	TO_PERCENT	TO_PERCENT(value)	Converts a provided number to a percentage.
Parser	TO_PURE_NUMBER	TO_PURE_NUMBER(value)	Converts a provided date/time, percentage, currency or other formatted numeric value to a pure number without formatting.
Parser	TO_TEXT	TO_TEXT(value)	Converts a provided numeric value to a text value.

14. Array

Type	Name	Syntax	Description
Array	ARRAY_CONSTRAIN	ARRAY_CONSTRAIN(input_range, num_rows, num_cols)	Constrains an array result to a specified size.
Array	BYCOL	BYCOL(array_or_range, LAMBDA)	Groups an array by columns by application of a LAMBDA function to each column.
Array	BYROW	BYROW(array_or_range, LAMBDA)	Groups an array by rows by application of a LAMBDA function to each row.

Type	Name	Syntax	Description
Array	CHOOSECOLS	CHOOSECOLS(array, col_num1, [col_num2])	Creates a new array from the selected columns in the existing range.
Array	CHOOSEROWS	CHOOSEROWS(array, row_num1, [row_num2])	Creates a new array from the selected rows in the existing range.
Array	FLATTEN	FLATTEN(range1, [range2, ...])	Flattens all the values from one or more ranges into a single column.
Array	FREQUENCY	FREQUENCY(data, classes)	Calculates the frequency distribution of a one-column array into specified classes.
Array	GROWTH	GROWTH(known_data_y, [known_data_x], [new_data_x], [b])	Given partial data about an exponential growth trend, fits an ideal exponential growth trend and/or predicts further values.
Array	HSTACK	HSTACK(range1; [range2, ...])	Appends ranges horizontally and in sequence to return a larger array.
Array	LINEST	LINEST(known_data_y, [known_data_x], [calculate_b], [verbose])	Given partial data about a linear trend, calculates various parameters about the ideal linear trend using the least-squares method.
Array	LOGEST	LOGEST(known_data_y, [known_data_x], [b], [verbose])	Given partial data about an exponential growth curve, calculates various parameters about the best fit ideal exponential growth curve.
Array	MAKEARRAY	MAKEARRAY(rows, columns, LAMBDA)	Returns an array of specified dimensions

Type	Name	Syntax	Description
			Results are calculated by application of a LAMBDA function.
Array	MAP	MAP(array1, [array2, ...], LAMBDA)	Maps each value in the given arrays to a new value by application of a LAMBDA function to each value.
Array	MDETERM	MDETERM(square_matrix)	Returns the matrix determinant of a square matrix specified as an array or range.
Array	MINVERSE	MINVERSE(square_matrix)	Returns the multiplicative inverse of a square matrix specified as an array or range.
Array	MMULT	MMULT(matrix1, matrix2)	Calculates the matrix product of two matrices specified as arrays or ranges.
Array	REDUCE	REDUCE(initial_value, array_or_range, LAMBDA)	Reduces an array to an accumulated result by application of a LAMBDA function to each value.
Array	SCAN	SCAN(initial_value, array_or_range, LAMBDA)	Scans an array and produces intermediate values by application of a LAMBDA function to each value. Returns an array of the intermediate values obtained at each step.
Array	SUMPRODUCT	SUMPRODUCT(array1, [array2, ...])	Calculates the sum of the products of corresponding entries in two equal-sized arrays or ranges.
Array	SUMX2MY2	SUMX2MY2(array_x, array_y)	Calculates the sum of the differences of the

Type	Name	Syntax	Description
			squares of values in two arrays.
Array	SUMX2PY2	SUMX2PY2(array_x, array_y)	Calculates the sum of the sums of the squares of values in two arrays.
Array	SUMXMY2	SUMXMY2(array_x, array_y)	Calculates the sum of the squares of differences of values in two arrays.
Array	TOCOL	TOCOL(array_or_range, [ignore], [scan_by_column])	Transforms an array or range of cells into a single column.
Array	TOROW	TOROW(array_or_range, [ignore], [scan_by_column])	Transforms an array or range of cells into a single row.
Array	TRANSPOSE	TRANSPOSE(array_or_range)	Transposes the rows and columns of an array or range of cells.
Array	TREND	TREND(known_data_y, [known_data_x], [new_data_x], [b])	Given partial data about a linear trend, fits an ideal linear trend using the least squares method and/or predicts further values.
Array	VSTACK	VSTACK(range1; [range2, ...])	Appends ranges vertically and in sequence to return a larger array.
Array	WRAPCOLS	WRAPCOLS(range, wrap_count, [pad_with])	Wraps the provided row or column of cells by columns after a specified number of elements to form a new array.
Array	WRAPROWS	WRAPROWS(range, wrap_count, [pad_with])	Wraps the provided row or column of cells by rows after a specified number of