“How can I use the MAP function in Google Sheets to transform the values in a range based on a specified function?”

Authored by

stats writer

July 1, 2024

The MAP function in Google Sheets allows users to transform the values in a selected range by applying a specified function. This function can be any mathematical operation or formula, and it will be applied to each cell in the range individually. This feature is useful for quickly and easily modifying data in a spreadsheet, such as converting units or performing calculations. By utilizing the MAP function, users can efficiently manipulate their data without having to manually edit each cell. This can save time and improve the accuracy of data analysis in Google Sheets.

MAP function

This function maps each value in the given arrays to a new value by application of a `LAMBDA` function to each value.

Sample Usage

`MAP(A1:A5, LAMBDA(cell, cell*2))`: `MAP` function with range as input.

`MAP(A1:A5, B1:B5, LAMBDA(cell1, cell2, MAX(cell1, cell2)))`: `MAP` function with multiple ranges as input.

`MAP(UNIQUE(A1:A10), LAMBDA(number, number + 1))`: `MAP` function with array as input.

Syntax

```
MAP(array1, , LAMBDA)
```

`array1`: An array of range to be mapped.`array2, ...`: Additional arrays or ranges to be mapped.`LAMBDA`: A `LAMBDA` function that's mapped to each value in the given arrays to obtain a new mapped value.
Syntax:`LAMBDA(name1, , formula_expression)`**Requirements:**The `LAMBDA` must have exactly 1 `name` argument for each array passed, along with a `formula_expression` which uses those names. When `LAMBDA` is applied, the names resolve to the current values being mapped in the passed arrays.

Notes

The passed `LAMBDA` function should accept exactly as many `name` arguments as the number of input arrays given to `MAP`, otherwise an `#N/A` error is returned. These arguments correspond to the values in the input arrays which are being mapped to a new value.

Values in the input arrays should map to a single value. Array results for mapped values aren't

supported.

A `named function` can be passed for the `LAMBDA` parameter and behaves like a `LAMBDA` in this case. Learn more about named functions.
There should be exactly as many argument placeholders defined for it as the number of input arrays passed to `MAP`.The `named function` shouldn't be followed by parenthesis.

Examples

Simple doubling operation with MAP

Example data:

	A	B	C	D
1	1	2		
2	3	4		
3			2	4
4			6	8

Example: Input this formula in `C3:=MAP(A1:B2, LAMBDA(cell, cell*2))`

Map a comma separated values to hyphenated SKU codes

Example Data:

	A	B
1	Jeans,Black,XL	Jeans-Black-XL
2	Shorts,Brown,S	Shorts-Brown-S
3	Tshirt,Red,L	Tshirt-Red-L
4	Skirt,Pink,M	Skirt-Pink-M

Example: Input this formula in `B1: =MAP(A1:A4, LAMBDA(item, JOIN("-", SPLIT(item, ",")))`

Map multiple input ranges to the max value in each data set

Example Data:

	A	B	C	D	E	F	G	H
1	38.9	17.8		42	20.2		38.6	21.2
2	39.2	19.6		37.8	17.1		34.6	21.2
3	34.1	18.1		41.1	17.6		36.6	17.8

Example: `=MAP(A1:B3, D1:E3, G1:H3, LAMBDA(valA, valB, valC, MAX(valA, valB, valC)))`

Result:

	A	B
1	42	21.2
2	39.2	21.2
3	41.1	18.1

Use a named function as a LAMBDA to count cells which have numbers in them

Make a Copy

Example Data:

	A	B	C
1	13 Going On 30	2 Fast 2 Furious	12 Angry Men
2	Eternal Sunshine of the Spotless Mind	Friday the 13th	No Country for Old Men

Named function: `CONTAINS_NUMBER` is a `named function` which checks if the given string value contains any number.

Formula definition: `=ARRAYFORMULA(OR(ISNUMBER(SPLIT(cell, " "))))`, where cell is an argument placeholder defined for `CONTAINS_NUMBER`.

Example: `=COUNTIF(MAP(A1:C2, CONTAINS_NUMBER), true)`

Result:

3

Common Errors

The dimensions of the input arrays don't match

If the dimensions of the input arrays don't match, this error occurs:

"Array arguments to MAP are of different size."

Example:`=MAP(C1:C4, D1:D2, LAMBDA(x, x+1))`

In this example, array `C1:C4` doesn't match the size of array `D1:D2`.

The passed LAMBDA doesn't have exactly as many name arguments as the number of input arrays

If the passed `LAMBDA` doesn't have exactly as many `name` arguments as the number of input arrays given to `MAP`, this error occurs:

"Wrong number of arguments to LAMBDA. Expected 3 arguments, but got 2 arguments.

Example:`=MAP(C1:C4, D1:D4, LAMBDA(cell, cell+1))`

In this example, `LAMBDA` was given only 1 `name` argument cell, even though we passed 2 arrays to `MAP`.

The last parameter of MAP wasn't a LAMBDA

If the last parameter of `MAP` wasn't a `LAMBDA`, this error occurs:

"Argument must be a LAMBDA."

Example:`=MAP(C1:C3, 3)`

The LAMBDA passed to MAP was incorrect

If the 1 or more `name` arguments aren't valid, this error occurs:

"Argument 1 of function LAMBDA is not a valid name."

Example:`=MAP(C1:C3, LAMBDA(C1, C1+1))`

In this example, `C1` is an invalid `name` since it clashes with a range.

The application of LAMBDA on the input array(s) maps to multiple values or another array

If the application of `LAMBDA` on the input array(s) maps each value to multiple values or another array, this error occurs:

"Single value expected. Nested array results are not supported."

Example:`=MAP(E1, LAMBDA(word, SPLIT(word, " ")))`

In this example, we try to map the text in the cell to an array of words.

Related functions

LAMBDA function: This function lets you create and return a custom function with a set of `names` and a `formula_expression` that uses them.REDUCE function: This function reduces an array to an accumulated result. BYROW function: This function groups an array by rows.BYCOL function: This function groups an array by columns.SCAN function: This function scans an array and produces intermediate values.MAKEARRAY function: This function creates a calculated array of specified dimensions.Create & use named functions: This function lets users create and store custom functions, similar to `LAMBDA`.