How can I use the SCAN function in Google Sheets?

Authored by

stats writer

July 1, 2024

The SCAN function in Google Sheets is a useful tool that allows users to quickly search for specific data within a spreadsheet. By entering a keyword or phrase, the SCAN function will scan through a designated range of cells and return the first occurrence of the keyword, along with its corresponding cell reference. This function can save users time and effort when trying to locate specific information within a large data set. To use the SCAN function, simply enter the keyword and the range of cells to be searched, and the function will return the desired result.

SCAN function

This function 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.

Sample Usage

```
SCAN(5, A1:A3, LAMBDA(accumulator, current_value, accumulator+current_value))
```

```
SCAN(2, A1:A3, LAMBDA(accumulator, current_value, accumulator*current_value))
```

Syntax

```
SCAN(initial_value, array_or_range, LAMBDA)
```

`initial_value`: The initial `accumulator` value.`array_or_range`: An array or range to be scanned.`LAMBDA`: A `LAMBDA` that's applied to each value in `array_or_range` for scanning it.
Syntax:`LAMBDA(name1, name2, formula_expression)`**Requirements:**The `LAMBDA` must have exactly 2 `name` arguments along with a `formula_expression` which uses those `names`. The `name1` resolves to the current value in the `accumulator` and `name2` resolves to the `current_value` in `array_or_range`, when applying the `LAMBDA`. The `accumulator` is updated in each step to the intermediate value obtained in the previous step.

Notes

The passed `LAMBDA` should accept exactly 2 `name` arguments, otherwise an `#N/A` error is returned. These arguments correspond to `accumulator` and `current_value`, in order. These are explained as:

`name1`: Resolves to the value in the `accumulator`.`name2`: Resolves to the `current_value` in the input array.The `accumulator` is initialized by `initial_value` and updated in each step to the intermediate value obtained in the previous step.

The `current_value` in the input array are found row by row, while you apply the `LAMBDA.`

A `named function` can be passed for the `LAMBDA` parameter and behaves like a `LAMBDA` in this case. Learn more about named functions.

The `named function` must follow the `LAMBDA` syntax for `SCAN` with exactly 2 argument placeholders defined for it.The `named function` shouldn't be followed by parenthesis.

Examples

Return the running total of an array

Example data:

	A
1	4
2	2
3	1

Example:`=SCAN(5, A1:A3, LAMBDA(accumulator, current_value, accumulator+current_value))`

Result:

9
11
12

Return the cumulative percentage of total value

Example data:

	A
1	4
2	2
3	1

Example: `=SCAN(0, A1:A3, LAMBDA(accumulator, current_value, accumulator +`

`current_value/sum(A1:A3)))`

Result:

0.57
0.85
1

Return the running total of an array and restart calculation when the number is 0 with a named function as LAMBDA.

Make a Copy

Example data:

	A
1	4
2	2
3	1
4	0
5	3
6	6

Example:`=SCAN(0, A1:A6, RUNNING_TOTAL_0)`

Named function:`RUNNING_TOTAL_0` is a `named function` which outputs the running total of the array and restarts the calculation when the `current_value` is 0.

Formula definition:`=if(current_value=0, current_value, accumulator+current_value)`, where `accumulator` and `current_value` are argument placeholders defined for `RUNNING_TOTAL_0`.

Result:

4
6
7

0
3
9

Common Errors

The passed LAMBDA doesn't have exactly 2 name arguments

If the `LAMBDA` function doesn't have 2 `name` arguments, this error occurs:

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

Example:`=SCAN(5, C1:C4, LAMBDA(current_value, current_value+1))`

In this example, LAMBDA was given only 1 `name` argument when it needed 2.

The last parameter of SCAN wasn't a LAMBDA

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

"Argument must be a LAMBDA."

Example:`=SCAN(5, C1:C4, 3)`

In this example the last function is `3`, instead of a `LAMBDA` function.

The LAMBDA passed to SCAN was incorrect

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

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

Example:`=SCAN(5, C1:C4, LAMBDA(C1, v, C1+v))`

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

One or more intermediate values produced by the application of LAMBDA are not single values

If the application of `LAMBDA` on the input array produces any non-single intermediate value, this error occurs:

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

Example:`=SCAN(5, C1:C4, LAMBDA(accumulator, value, {accumulator, value}))`

Every application of `LAMBDA` must produce an intermediate value which is a single value only and can't be another array.

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.MAP function: This function maps each value in the given arrays to a new value.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.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`.