

ARRAYSORT Function

Contents:

- *Basic Usage*
- *Syntax and Arguments*
 - *array_ref*
 - *order_enum*
- *Examples*
 - *Example - Student progress across tests*

Sorts array values in the specified column, array literal, or function that returns an array in ascending or descending order.

- This function calculates based on the outer layer of an array. If your array is nested, the sorting of inner elements is not factored.

Wrangle vs. SQL: This function is part of Wrangle , a proprietary data transformation language. Wrangle is not SQL. For more information, see *Wrangle Language*.

Basic Usage

Array literal reference example:

```
arraysort([A,B,C,D],descending)
```

Output: Returns the following array: [D,C,B,A].

Column reference example:

```
arraysort(myArrays,ascending)
```

Output: Returns the arrays listed in the `myArrays` column sorted in ascending order.

Syntax and Arguments

```
arraysort(array_ref,order_enum)
```

Argument	Required?	Data Type	Description
array_ref	Y	string	Name of Array column, Array literal, or function returning an Array to apply to the function
order_enum	Y	string (enumerated value)	Order is defined as either: <ul style="list-style-type: none">• ascending (default)• descending

For more information on syntax standards, see *Language Documentation Syntax Notes*.

array_ref

Name of the array column, array literal, or function returning an array whose array values you wish to sort.

- Multiple columns and wildcards are not supported.

Usage Notes:

Required?	Data Type	Example Value
Yes	String (column reference or function) or array literal	myArray1

order_enum

String literal indicating the order by which the referenced arrays should be sorted:

- *ascending* - (default) lowest values for the valid data type are listed first.
- *descending* - Null/empty values are sorted first, followed by mismatched values. Then, the array values that are valid for the specified data type are listed in descending order.

Usage Notes:

Required?	Data Type	Example Value
No	String (enumerated value)	<i>descending</i>

Examples

Tip: For additional examples, see *Common Tasks*.

Example - Student progress across tests

This example illustrates how to return n-based number of elements in an array.

Functions:

Item	Description
ARRAYLEN Function	Computes the number of elements in the arrays in the specified column, array literal, or function that returns an array.
ARRAYELEMENTAT Function	Computes the 0-based index value for an array element in the specified column, array literal, or function that returns an array.
ARRAYSORT Function	Sorts array values in the specified column, array literal, or function that returns an array in ascending or descending order.

Source:

Here are some student test scores. Individual scores are stored in the `SCORES` column. You want to:

1. Flag the students who have not taken four tests.

2. Compute the range in scores for each student.

LastName	FirstName	Scores
Allen	Amanda	[79, 83,87,81]
Bell	Bobby	[85, 92, 94, 98]
Charles	Cameron	[88,81,85]
Dudley	Danny	[82,88,81,77]
Ellis	Evan	[91,93,87,93]

Transformation:

First, you want to flag the students who did not take all four tests:

Transformation Name	New formula
Parameter: Formula type	Single row formula
Parameter: Formula	IF(ARRAYLEN(Scores) < 4,"incomplete","")
Parameter: New column name	'Error'

This test flags Cameron Charles only.

The following transform sorts the array values in highest to lowest score:

Transformation Name	Edit column with formula
Parameter: Columns	Scores
Parameter: Formula	ARRAYSORT(Scores, 'descending')

The following transforms extracts the first (highest) and last (lowest) value in each student's test scores, provided that they took four tests:

Transformation Name	New formula
Parameter: Formula type	Single row formula
Parameter: Formula	ARRAYELEMENTAT(Scores,0)
Parameter: New column name	'highestScore'

Transformation Name	New formula
Parameter: Formula type	Single row formula
Parameter: Formula	ARRAYELEMENTAT(Scores,3)
Parameter: New column name	'lowestScore'

Tip: You could also generate the `Error` column when the `Scores4` column contains a null value. If no value exists in the array for the `ARRAYELEMENTAT` function, a null value is returned, which would indicate in this case an insufficient number of elements (test scores).

You can now track change in test scores:

Transformation Name	New formula
Parameter: Formula type	Single row formula
Parameter: Formula	<code>SUBTRACT(highestScore, lowestScore)</code>
Parameter: New column name	'Score_range'

Results:

LastName	FirstName	Scores	Error	lowestScore	highestScore	Score_range
Allen	Amanda	[87,83,81,79]		79	87	8
Bell	Bobby	[98,94,92,85]		85	98	13
Charles	Cameron	[88,85,81]	incomplete		88	
Dudley	Danny	[88,82,81,77]		77	88	11
Ellis	Evan	[93,93,91,87]		87	93	6