

# 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.

## Basic Usage

### Array literal reference example:

```
derive type:single value:ARRAYSORT([A,B,C,D],descending)
```

**Output:** Generates the following array in the new column: [D,C,B,A].

### Column reference example:

```
derive type:single value:ARRAYSORT(myArrays,ascending) as:'myArrays_ascending'
```

**Output:** Generates the new `myArrays_ascending` column containing the arrays listed in the `myArrays` column sorted in ascending order.

## Syntax and Arguments

```
derive type:single value: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"><li>• ascending (default)</li><li>• descending</li></ul>

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.
- For more information on the rules of sorting, see *Sort Order*.

### Usage Notes:

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

### Examples

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

### Example - Student progress across tests

This example covers the following functions:

- `ARRAYLEN` - Returns 1-based number of elements in an array. See *ARRAYLEN Function*.
- `ARRAYELEMENTAT` - Returns array element based on 0-based index parameter. See *ARRAYELEMENTAT Function*.
- `ARRAYSORT` - Returns array sorted in ascending or descending order. See *ARRAYSORT Function*.

### 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]

### Transform:

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

```
derive type:single value:IF(ARRAYLEN(Scores) < 4,"incomplete","") as:'Error'
```

This test flags Cameron Charles only.

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

```
set column: Scores value: 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:

```
derive type:single value:ARRAYELEMENTAT(Scores,0) as:'highestScore'
```

```
derive type:single value:ARRAYELEMENTAT(Scores,3) as:'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:

```
derive type:single value:SUBTRACT(highestScore,lowestScore) as:'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