

arrayReplaceAll

Description

Returns a new array where all instances of a specified value are replaced with a new value.

- The element type must match the array type.
- Supported element types include `string`, `bool`, `number`, `interval`, `timestamp`, `regexp`, and `enum`.

Syntax

Like many functions in DataPrime, `arrayReplaceAll` supports [two notations](#), **function** and **method** notation. These interchangeable forms allow flexibility in how you structure expressions.

Function notation

```
arrayReplaceAll(array: array<T>, value: T, newValue: T): array<T>
```

Method notation

```
(array: array<T>).arrayReplaceAll(value: T, newValue: T): array<T>
```

Arguments

Name	Type	Required	Description
array	array	true	The array to modify
value	T	true	The value to replace
newValue	T	true	The replacement value, must match the array type

Example

Use case: Replace outdated schema values in arrays

Suppose you have a list of values where some still use an outdated schema. Consider the following inputs:

```
{
  "values": ["NewVal1", "NewVal2", "NewVal3"]
},
{
  "values": ["OldVal1", "NewVal2", "NewVal3"]
}
```

By replacing all occurrences of "OldVal1" with "NewVal1", you ensure data consistency before further processing.

Example query

Function notation

```
create updated_values from arrayReplaceAll(values, 'OldVal1', 'NewVal1')
```

Method notation

```
create updated_values from values.arrayReplaceAll('OldVal1', 'NewVal1')
```

Example output

The result will replace all outdated values:

```
{
  "values": ["NewVal1", "NewVal2", "NewVal3"]
},
{
  "values": ["NewVal1", "NewVal2", "NewVal3"]
}
```

 Last updated: October 19, 2025

Was this helpful?

Leave your feedback here.

☐ Yes ☐ No

Send

© 2025 Coralogix. All rights reserved.

Generated on: November 18, 2025

Source: <https://coralogix.com/docs/dataprime/language-reference/functions-reference/array/arrayreplaceall/>