JS JavaScript TUTORIAL

Array.prototype.reduceRight()

Summary: in this tutorial, you will learn how to use the JavaScript Array `reduceRight()` method to reduce all values in an array to a single value.

Introduction to JavaScript Array reduceRight() method

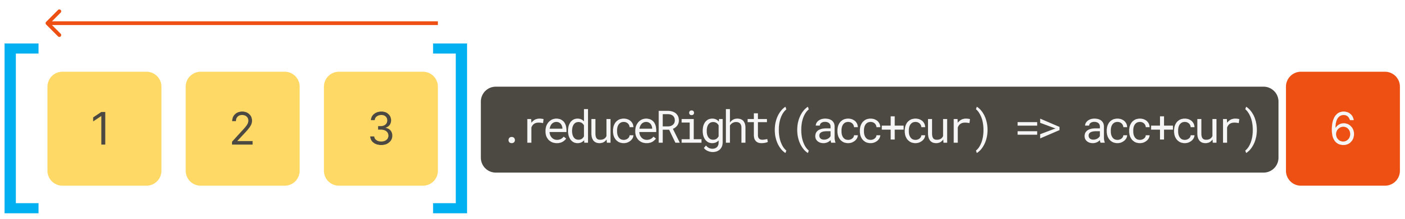
The `reduceRight()` method applies a function to an accumulator and each element in an array from *right to left* to reduce them into a *single value*.

Here's the syntax of the Array `reduceRight()` method:

```
const value = array.reduceRight(callbackFn, initialValue)
```

In this syntax:

- `callbackFn` is a callback function that applies to an accumulator and each value, from the right to left, of the `array` .
- `initialValue` is the value to use as accumulator to the first call of the `callbackFn` . If you skip the `initialValue` , the `reduceRight()` method will use the last element of the `array` as the initial value.



The `callbackFn` has the following syntax:

```
callbackFn(accumulator, currentValue, index, array)
```

The `callbackFn` is a function that execute each element in the array:

- `accumulator` is the accumulated value from previously returned in the last invocation of the `callbackFn` .
- `currentValue` is the element in the array being processed.
- `index` is the index of the current element.
- `array` is the array that calls the `reduceRight()` method.

The `reduceRight()` method returns a value resulting from the reduction.

JavaScript Array reduceRight() method examples

Let's take some examples of using the `reduceRight()` method.

Basic JavaScript reduceRight() method example

The following example uses the `reduceRight()` method to calculate the total of numbers in an array:

```
const numbers = [1, 2, 3, 4];
const total = numbers.reduceRight((sum, n) => sum + n, 0);

console.log({ total });
```

Output:

```
{ total: 10 }
```

JavaScript reduceRight vs. reduce() example

The `reduceRight()` method processes the array elements from right to left, while the `reduce()` method process them from left to right.

For example:

```
const chars = ['E', 'C', 'M', 'A'];
const result = chars.reduceRight((a, c) => a + c, '');

console.log({ result });
```

Output:

```
{ result: 'AMCE' }
```

In this example, the `reduceRight()` method reduces the chars array from right to left: `A`, `M`, `C`, and `E`.

If you use the `reduce()` method, it'll concatenate the string from left to right:

Output:

```
{ result: 'ECMA' }
```

Constructing HTML from HTML elements

The following example uses the `reduceRight()` method to construct `HTML` string from elements:

```
const elements = [
  '<section>',
  '<p>',
  'Hello, JavaScript!',
  '</p>',
  '</section>',
];

const html = elements.reduceRight((a, c) => {
  result = c + a;
  console.log({ result });
  return result;
}, '');
```

```
console.log({ html });
```

Output:

```
{ result: '</section>' }
{ result: '</p></section>' }
{ result: 'Hello, JavaScript!</p></section>' }
{ result: '<p>Hello, JavaScript!</p></section>' }
{ result: '<section><p>Hello, JavaScript!</p></section>' }
{ html: '<section><p>Hello, JavaScript!</p></section>' }
```

In this example, we display the immediate result of each step of the `reduceRight()` method to make it more clear. The final result is a valid HTML snippet constructed from the elements.

Summary

- Use the `reduceRight()` method to reduce an array into a single value by executing a function on each element of an array from right to left.