

Learn to code — free 3,000-hour curriculum

NOVEMBER 10, 2019 / #JAVASCRIPT

JavaScript Map, Reduce, and Filter - JS Array Functions Explained with Code Examples



Beau Carnes



Learn to code — free 3,000-hour curriculum

perform a transformation or computation. Each will return a new array based on the result of the function. In this article, you will learn why and how to use each one.

Here is a fun summary by Steven Luscher:



Map

The `map()` method is used for creating a new array from an existing one, applying a function to each one of the elements of the first array.

Learn to code — free 3,000-hour curriculum

```
var new_array = arr.map(function callback(element, index, array) {  
  // Return value for new_array  
}, thisArg))
```

In the callback, only the array `element` is required. Usually some action is performed on the value and then a new value is returned.

Example

In the following example, each number in an array is doubled.

```
const numbers = [1, 2, 3, 4];  
const doubled = numbers.map(item => item * 2);  
console.log(doubled); // [2, 4, 6, 8]
```

Filter

The `filter()` method takes each element in an array and it applies a conditional statement against it. If this conditional returns true, the element gets pushed to the output array. If the condition returns false, the element does not get pushed to the output array.

Syntax

```
var new_array = arr.filter(function callback(element, index, array) {  
  // Return true or false  
}, thisArg))
```

Learn to code — free 3,000-hour curriculum

callback, only the `element` is required.

Examples

In the following example, odd numbers are "filtered" out, leaving only even numbers.

```
const numbers = [1, 2, 3, 4];
const evens = numbers.filter(item => item % 2 === 0);
console.log(evens); // [2, 4]
```

In the next example, `filter()` is used to get all the students whose grades are greater than or equal to 90.

```
const students = [
  { name: 'Quincy', grade: 96 },
  { name: 'Jason', grade: 84 },
  { name: 'Alexis', grade: 100 },
  { name: 'Sam', grade: 65 },
  { name: 'Katie', grade: 90 }
];

const studentGrades = students.filter(student => student.grade >= 90);
return studentGrades; // [ { name: 'Quincy', grade: 96 }, { name: 'Alexis', grade: 100 } ]
```

Reduce

The `reduce()` method reduces an array of values down to just one

Learn to code — free 3,000-hour curriculum

Syntax

```
arr.reduce(callback[, initialValue])
```

The `callback` argument is a function that will be called once for every item in the array. This function takes four arguments, but often only the first two are used.

- *accumulator* - the returned value of the previous iteration
- *currentValue* - the current item in the array
- *index* - the index of the current item
- *array* - the original array on which `reduce` was called
- The `initialValue` argument is optional. If provided, it will be used as the initial accumulator value in the first call to the callback function.

Examples

The following example adds every number together in an array of numbers.

```
const numbers = [1, 2, 3, 4];
const sum = numbers.reduce(function (result, item) {
  return result + item;
}, 0);
console.log(sum); // 10
```

Learn to code — free 3,000-hour curriculum

into a single object that shows how many times each string appears in the array. Notice this call to `reduce` passes an empty object `{}` as the `initialValue` parameter. This will be used as the initial value of the accumulator (the first argument) passed to the callback function.

```
var pets = ['dog', 'chicken', 'cat', 'dog', 'chicken', 'chicken', 'rabbit'];

var petCounts = pets.reduce(function(obj, pet){
  if (!obj[pet]) {
    obj[pet] = 1;
  } else {
    obj[pet]++;
  }
  return obj;
}, {});

console.log(petCounts);

/*
Output:
{
  dog: 2,
  chicken: 3,
  cat: 1,
  rabbit: 1
}
*/
```

Video Explanation

Check out the video below from the freeCodeCamp.org YouTube channel. It covers the array methods discussed, plus a few more.

Learn to code — free 3,000-hour curriculum

Array Iteration: 8 Methods - map, filter, reduce, so...



Beau Carnes

I'm a teacher and developer with freeCodeCamp.org. I run the freeCodeCamp.org YouTube channel.

If you read this far, tweet to the author to show them you care.

[Tweet a thanks](#)

Learn to code for free. freeCodeCamp's open source curriculum has helped more than 40,000 people get jobs as developers.

[Get started](#)

freeCodeCamp is a donor-supported tax-exempt 501(c)(3) nonprofit organization (United States

Federal Tax Identification Number: 82-0779546)

Learn to code — free 3,000-hour curriculum

thousands of freeCodeCamp study groups around the world.

Donations to freeCodeCamp go toward our education initiatives, and help pay for servers, services, and staff.

You can [make a tax-deductible donation here](#).

Trending Guides

Git Clone	UX
Agile Methods	Design Thinking
Python Main	Prime Numbers List
Callback	Product Design
Debounce	Digital Design
URL Encode	Coding Games
Blink HTML	SVM
Python Tuple	JavaScript forEach
JavaScript Push	Google BERT
Java List	Create Table SQL
Responsive Web Design	What Is TLS?
What Is an SVG File?	What Is a LAN?
PDF Password Remover	What Is npm?
What Is a PDF?	RSync Examples
What Is Python?	Random Forest

Our Nonprofit

[About](#) [Alumni Network](#) [Open Source](#) [Shop](#) [Support](#) [Sponsors](#) [Academic Honesty](#)

[Code of Conduct](#) [Privacy Policy](#) [Terms of Service](#) [Copyright Policy](#)

[Forum](#)

[Donate](#)

[Learn to code — free 3,000-hour curriculum](#)