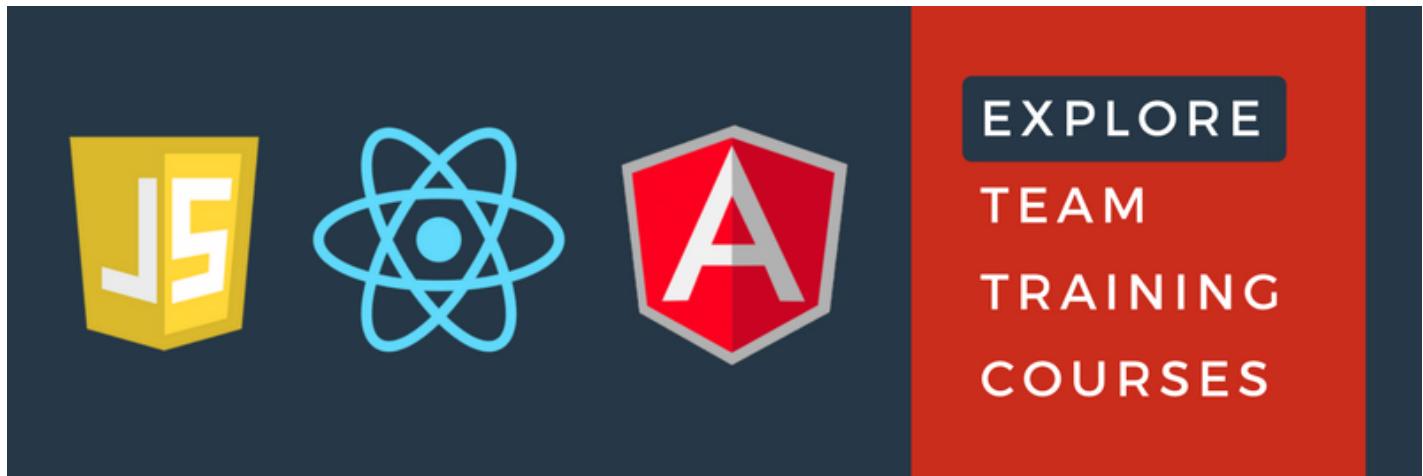




# How to Clear an Array in JavaScript



**Arrays are awesome!** They allow you to store multiple values in a convenient, indexed set. In JavaScript, arrays can be [declared literally](#) or they can be initialized using the [Array constructor function](#). **But wait...** What if you want to *empty* an array instead of creating one? Hmm... perhaps not as straightforward. Have no fear, there are some relatively easy ways to go about emptying an array in JavaScript. In the following examples, I'll examine three different ways in which you can clear an array. I'll then present some quirkiness related to one of the methods, as well as how to address it.

Here's our example array:

```
1 var arr = [1, 2, 3, 4, 5];
2
3 console.log(arr); // [1, 2, 3, 4, 5]
```

<http://codepen.io/anon/pen/adjwde?editors=0012>

\*Note here, that our array could be populated with any type of data, not just numbers.

## Now, to clear it...

### Method 1



method couldn't be simpler; just set the value of the array in question to an array.

```
1 var arr = [1, 2, 3, 4, 5];
2
3 arr = [];
4
5 console.log(arr); //[]
```

<http://codepen.io/anon/pen/ZQjyWw?editors=0012>

### Method 2

This method for emptying an array uses a for loop (could be any kind of loop) and the `Array.prototype.pop()` method.

```
1 var arr = [1, 2, 3, 4, 5];
2
3 for (var i = arr.length; i > 0; i--) {
4
5   arr.pop();
6
7 }
8
9 console.log(arr); //[]
```

<http://codepen.io/anon/pen/VeBWjj?editors=0012>

### Method 3

This third method is perhaps less intuitive, however, it's as simple as the first and actually produces a more robust result. Simply set the `length` of the array to 0.

```
1 var arr = [1, 2, 3, 4, 5];
2
3 arr.length = 0;
4
5 console.log(arr) //[]
```

<http://codepen.io/anon/pen/GoBENm?editors=0012>

Piece of cake! Three methods for clearing/emptying an array in JavaScript. **Hold on...** What was that quirkiness I was talking about earlier? Let's have a look...



**Method 1** is easy; just set the array to an empty array. However, you should be aware of the fact that setting an array equal to an empty one doesn't affect all of the references to that array. Here's what I mean....

```
1 var arr = [1, 2, 3, 4, 5];
2
3 var arr2 = arr;
```

Empty out the original array and...

```
1 arr = [];
2
3 console.log(arr); // []
4
5 console.log(arr2); // [1, 2, 3, 4, 5]
```

<http://codepen.io/anon/pen/adjwpb?editors=0012>

Whoops! Emptying the original array by setting it equal to an empty array cleared *it* out, but it didn't clear a reference to it. This could cause some issues! **Methods 2** and **3** take care of the problem:

```
1 var arr = [1, 2, 3, 4, 5];
2
3 var arr2 = arr;
4
5 for (var i = arr.length; i > 0; i--) {
6
7   arr.pop();
8
9 }
10
11 console.log(arr); // []
12
13 console.log(arr2); // []
```

<http://codepen.io/anon/pen/WrKORy?editors=0012>

and

```
1 var arr = [1, 2, 3, 4, 5];
2
3 var arr2 = arr;
4 arr.length = 0;
5
6
7 console.log(arr); // []
8
9 console.log(arr2); // []
```

<http://codepen.io/anon/pen/ZQjyLP?editors=0012>

As you can see, Methods 2 and 3 address the matter; emptying out the array and references to it. As to the question of which is the preferred method, there is a debate. Some claim performance is worse with Method 3 ([discussion here](#)). And some claim that readability is lost with Method 3. Still others don't see these issues as drawbacks and prefer the simple elegance of writing length = 0.

Regardless of the pros and cons associated with each of the aforementioned methods, you now know three distinct ways in which you can empty an array in JavaScript. Additionally, you are now aware of a quirk associated with one of the methods.

Use what you've learned. Go forth and clear those arrays!

---

Share this:



**Chase Allen**



Chase is a self-taught HTML, CSS, and JavaScript enthusiast. He enjoys creating web pages and web applications. His interests include art, music, movies, food, tech, and just general musings.

February 2, 2016 by **Chase Allen** in

javascript



## Comments

Comment

Name

Email

Add Comment



