

# Remove an Item From an Array with JavaScript

📍 OSCON, Austin, TX • May 8-11 • Save 20% PC20DWALSH

By David Walsh on February 6, 2013

💬 21



One operation that seems to be more difficult than it should be in every programming language is removing a value from an array. It's such an easy concept mentally that it skews our programmatic view of the task. In JavaScript the splice method is of huge help in removing an item from an array.

## {Track:js}



## JavaScript Splice

One `splice` coupled with an `indexOf` removes the item from an array:

---

```
// Start with an initial array
var array = ["a", "b", "c"];

// Find and remove item from an array
var i = array.indexOf("b");
if(i !== -1) {
    array.splice(i, 1);
}
```

---

Of course if you'd like to remove multiple occurrences of the same string/number, you'll need add a bit more logic:

---

```
for(var i = array.length-1; i--){
    if (array[i] === "b") array.splice(i, 1);
}
```

---

You may be thinking that the `filter` method would work...

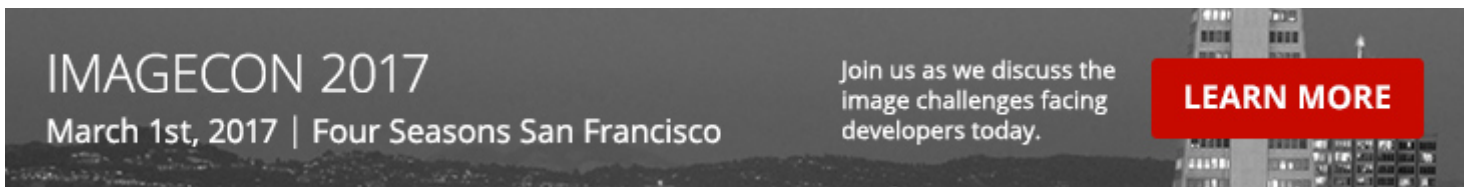
---

```
array.filter(function(i) {  
    return i !== "b"  
});
```

---

...but that will return a new array, thus not modifying the original.

Removing a given value from an array isn't too difficult of a task when you have a reliable snippet nearby!



## Recent Features



### Designing for Simplicity

Before we get started, it's worth me spending a brief moment introducing myself to you. My name is Mark (or [@integralist](#) if Twitter happens to be your communication tool of choice) and I currently work for BBC News in London England as a principal engineer/tech...

### Responsive and Infinitely Scalable JS Animations

Back in late 2012 it was not easy to find open source projects using `requestAnimationFrame()` - this is the hook that allows Javascript code to synchronize with a web browser's native paint loop. Animations using this method can run at 60 fps and deliver fantastic...

## Incredible Demos



### Fading Links Using jQuery: `dwFadingLinks`

UPDATE: The jQuery website was down today which caused some issues with my example. I've made everything local and now the example works. Earlier this week, I posted a MooTools script that faded links to and from a color during the mouseover and mouseout events.

## WebKit Marquee CSS: Bringin' Sexy Back

We all joke about the days of [Web yesteryear](#). You remember them: stupid animated GIFs (flames and "coming soon" images, most notably), lame counters, guestbooks, applets, etc. Another "feature" we thought we had gotten rid of was the marquee. The marquee was a rudimentary, javascript-like...

## Discussion

### Brandon Benvie



It's important to note that splicing an array is an incredibly slow operation and is very difficult or impossible for engines to optimize. If you have performance sensitive uses of collections of items that need to be mutated at arbitrary points then you'll likely want to look into alternative data structure implementations. Two new ones being introduced as base language features in ES6 are Map and Set, which force uniqueness but also allow for linear time item deletions. While these features only exist in Spidermonkey and V8 behind a flag, there are shims available and they will eventually be in all engines.

### David Walsh



Awesome tip Brandon, thank you!

### Brandon Benvie



Oops, quick correction. That's *\*constant\** time, not *\*linear\** time.

### Steve Fink



nit: you start out looking past the end of the array. I think you want `array.length - 1`.

I haven't benchmarked, but if you're expecting to remove a lot of elements, it might be faster to do something like:

```
for (var i = 0, j = 0; i < array.length; i++) {  
  if (array[i] !== "b")
```

```
    array[j++] = array[i];  
}  
array.length = j;
```

## MaxArt



I don't know how much is "a lot" of elements, but your code (although nice) doesn't seem to be very fast... At least, not faster than `.splice`:

<http://jsperf.com/removing-items-from-array>

In this case I'm just trying to remove 10% of the elements, maybe with a higher ratio the scenario would be different.

## David



@MaxArt: Your code has a little mistake: Instead of checking for `b !== 5`, you should check for `b[i] !== 5`. Otherwise, you just get "Array" for `b`, which is always false. Without this little mistake, Steve's code is way faster.

## Daniel Imms



I'm quite surprised by those results, Steve's version should be faster than shifting all elements back one index each splice. I guess splice must have a clever implementation in the other browsers and Firefox 21 just has a beast JS interpreter :P

## Anthony Articque



`indexOf` doesn't work on array in IE :(

## MaxArt



It does in IE9+.

For IE8- there are common shims, but I guess you already know that.

## Al Jey



Even though `filter` does in fact return a new array, it is a much more efficient way to get rid of multiple items at once than a several calls to `splice`, and as such should be preferred.

## John Pittman



Hey everyone, I'm new to contributing but stoked on javascript and the open source community so here I go.

If you're not concerned about keeping the array organized I came up with this which is blazing fast in comparison:

```
if (i > 0) callbacks[i] = callbacks[0];  
callbacks.shift();
```

Otherwise you will need to run a loop to shift all elements after the one you want to remove -1 index which depending on the size of the array could get ridiculous but in the majority of case would still crush splice.

Let me know what you think or send me an email. I'm just getting started on Git to get more involved.

## Brandon Belvin



I was curious about your technique so I built a jsPerf test around it. Unfortunately, it appears `array.splice` still performs better than the method you described.

<http://jsperf.com/array-splice-vs-reassign-and-shift>

It was certainly a good idea and comes close to `splice` (except in FF), but since it destroys the ordering of the array, adds an extra line of code, and still doesn't match `splice`, I'd recommend sticking with `splice`.

I haven't looked into the browser implementation code for `splice`. Maybe that's something you would want to do to find if you can help optimize it further; otherwise, it might help you understand how arrays work behind-the-scenes.

## Michael



Hello David,

I am new to java script and starting to learn about arrays. I understand pop and push. I would like to know how to use the splice code snippet you have posted. How would I use the splice code in my java script file?

```
// tasks.js #2  
// This script manages a to-do list.  
  
// Need a global variable:  
var tasks = [];
```

```

// Function called when the form is submitted.
// Function adds a task to the global array.
function addTask() {
    'use strict';

    // Get the task:
    var task = document.getElementById('task');

    // Reference to where the output goes:
    var output = document.getElementById('output');

    // For the output:
    var message = '';

    if (task.value) {

        // Add the item to the array:
        tasks.push(task.value);

        // Update the page:
        message = 'To-Do';
        for (var i = 0, count = tasks.length; i < count; i++) {
            message += ' ' + tasks[i] + ' ';
        }
        message += ' ';
        output.innerHTML = message;

    } // End of task.value IF.

    // Return false to prevent submission:
    return false;

} // End of addTask() function.

// Initial setup:
function init() {
    'use strict';
    document.getElementById('theForm').onsubmit = addTask;
} // End of init() function.
window.onload = init;

```

Hey guys, I'm okay with js but still learning. I've been looking everywhere on how to time a function to END after a certain period of time, say two days. Any help? Would really appreciate any advice! Thanks in advance.

## Chirag64



Your `for` loop for removing multiple occurrences of the same string/number is wrong. It doesn't have any condition like `i > -1`; . Although it still seems to work, it doesn't really check the last element of the array. So if you had an array like

```
var array = ["a", "c", "b"];
```

, it wouldn't remove the 'b' element.

## wasim



Hey you have explained it very well but will it work if there is same value in array. i mean if duplicate data then what it will do. ?

in my case i want to delete on the bases of key so what can i do ?  
please can you suggest me some solution.

## Timmah



When you do this:

```
for(var i = array.length-1; i--;){  
    if (array[i] === "b") array.splice(i, 1);  
}
```

```
and array = ["a", "b", "b", "c", "d"];
```

since splice re-indexes the array, won't you miss the second "b"?

## carla



in duplicated data

If you remove the `-1` it will work

```
for(var i = arr.length; i--;){  
    if (arr[i] === "b") arr.splice(i, 1);  
}
```

or reset the index

```
for(var i = 0; i < arr.length; i++){
  if(arr[i] === item){
    arr.splice(i, 1);
    i--;
  }
}
```

## Lex



Yes, removing multiple occurrences doesn't work for array like ['a', 'a', 'a'] .  
I think this should work:

```
for(var i = 0; i < array.length; i++){
  while(array[i] == item) array.splice(i,1);
}
```

## Aneudys Amparo



Here my example, It took me like 18hrs trying to do.

I have to array, One is the original but before to give the data to the User i have to extract/delete the data that had been completed by the same User.

```
var offers = server.offers;
var completions = server.completions;

offers.forEach(function(offer, index0) {

  completions.forEach(function(completion, indexC){

    if(offer.id == parseInt(completion.offerId)) {
      //Delete that offer

      offers.splice(index0, 1);

    }

  });

});
```

## Ezra Obiwale





To remove multiple occurrences, I believe this would be much faster than looping through the whole array:

```
var index = array.indexOf("b");
while (index > -1) {
  array.splice(index,1);
  index = array.indexOf("b");
}
```

This can easily be put into a function as such:

```
function removeValueFromArray(value, array) {
  var index = array.indexOf(value);
  while (index > -1) {
    array.splice(index,1);
    index = array.indexOf(value);
  }
  return array;
}
```

Wrap your code in `<pre class="{Language}"></pre>` tags, link to a [GitHub gist](#), [JSFiddle fiddle](#), or [CodePen pen](#) to embed!

☐ Continue this conversation via email

[Post Comment!](#)

[Use Code Editor](#)





