Move item in object to last position

Asked 5 years, 2 months ago Active 8 months ago Viewed 20k times

15

7

I have an array of objects. I want to move a selected object to the last position in the array. How do I do this in javascript or jquery?

Here is some code I have:

```
var sortedProductRow = this.product_row;

for (var s in sortedProductRow) {
    if (sortedProductRow[s]["parent_product_type"] != "")
        // Move this object to last position in the array
}
```

I'm looping through this with a for loop, and I want the output to be ordered so that all objects that does not have a "parent_product_type" value comes first, then those with a value.

javascript arrays sorting

edited Jul 23 '14 at 11:43

asked Jul 23 '14 at 11:33

Johan Dahl
539 1 9 25

1 An **object** is not an **array** and position is meaninglessness, it's only keys and values. read more - stackoverflow.com/q/5525795/104380 – vsync Jul 23 '14 at 11:36

who says `sortedProductRow` is an object? you can iterate an array with for in as well (I don't say it's a good idea, but it's possible) – Ferdi265 Jul 23 '14 at 11:40

1 Hmm ok, then I'm not sure what I need to do. I can loop through this array with a for loop, and the objects come out in the order that they appear within the "array" (or object). So what do I need to do if I want to change the order that they come out of the for loop? – Johan Dahl Jul 23 '14 at 11:41

3 Answers

40

✓

to move an element (of which you know the index) to the end of an array, do this:

```
array.push(array.splice(index, 1)[0]);
```

If you don't have the index, and only the element, then do this:

```
array.push(array.splice(array.indexOf(element), 1)[0]);
```

Example:

```
var arr = [1, 2, 6, 3, 4, 5];
arr.push(arr.splice(arr.indexOf(6), 1)[0]);
console.log(arr); // [1, 2, 3, 4, 5, 6]
```

Run code snippet   Expand snippet

> NOTE:
>
> this only works with Arrays (created with the `[ ... ]` syntax or `Array()`) not with Objects (created with the `{ ... }` syntax or `Object()`)

edited Mar 22 '18 at 10:16
Nikola Lukic
2,125 3 25 42

answered Jul 23 '14 at 11:42
Ferdi265
1,563 13 14

needs some more explanation. Although this solves the problem, no one is learning anything. – Devashish Jul 8 at 9:52

5

Moving **the first element** of an array **to the end** of the same array

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

Run code snippet Expand snippet

or this way

```
var a = [5,1,2,3,4];
var b = a.shift();
a[a.length] = b;
console.log(a); // [1,2,3,4,5]
```

Run code snippet Expand snippet

Moving **any element** of an array **to any position** in the same array

```
// move element '5' (index = 2) to the end (index = 4)
var a = [1, 2, 5, 4, 3];
a.splice(4,0,a.splice(2,1)[0]);
console.log(a); // [1, 2, 4, 3, 5]
```

Run code snippet Expand snippet

or it could be converted to a prototype as well, like this where `x` represents the current position of element while `y` represents the new position in array

```
var a = [1, 2, 5, 4, 3];
Array.prototype.move = function(x, y){
   this.splice(y, 0, this.splice(x, 1)[0]);
   return this;
  };

  a.move(2,4);
  console.log(a); // ["1", "2", "4", "3", "5"]
```

edited Mar 22 '18 at 10:22
Nikola Lukic
2,125 3 25 42

answered Jul 23 '14 at 11:46
hex494D49
7,016 3 26 40

Elegant solution, but this can only move the first element of the array to the end. – Ferdi265 Jul 23 '14 at 11:49

@Ferdi265 I don't know why I was sure that the OP wants to move the first element :) – hex494D49 Jul 23 '14 at 11:50

Thanks but it's not the first element I want to move. I'm looping through the sortedProductRow array/object(?) and when I find a property name "parent_product_type" that has a value, I want to move it to the last position in the array. This way, my array will have only objects without a value for the property "parent_product_type" coming first, then all objects with a value for this property. – Johan Dahl Jul 23 '14 at 11:55

1 @JohanDahl Sorry :) I updated the answer to something more useful :) so give it a try. – hex494D49 Jul 23 '14 at 12:08

Thanks for your answer! I ended up using Ferdi's second method which worked well. – Johan Dahl Jul 23 '14 at 12:30

1

Using an anonymous function you can pass in the array and the value to filter by.

```
let concatToEnd = function (arr, val) {
        return arr.filter(function(x) {
            return x !== val; // filter items not equal to value
        }).concat(arr.filter(function(x) { // concatonate to filtered array
            return x === val; // filter items equal to value
        })
    );
}

// invoke
concatToEnd(array, 'parent_product_type');
```

You could probably shorten this further:

```
let concatToEnd = (arr,val) => arr.filter(x => x !== val).concat(arr.filter(x => x ===
val))
```

This function filters the items which *do not equal* the value passed in, then concatenates the result (to the end of the filtered array) of *another* `filter` function which filters out the items which *do equal* the value you've passed in.

This hasn't been tested for your use-case, but I've used something similar to move all numbers of an array to the end of the index.

answered Jan 15 at 19:41

UncaughtTypeError
5,646 2 11 27