



# A Drip of JavaScript

## Reordering Arrays with Array#sort

Originally published in the [A Drip of JavaScript newsletter](#).

In [last week's issue](#) I briefly mentioned the standard `sort` method available on all JavaScript arrays. It does just what you'd expect.

```
// Produces [1, 2, 3]
[3, 2, 1].sort();

// Produces ["a", "b", "c"]
["c", "b", "a"].sort();

// Produces [1, 2, "a", "b"]
["b", 2, "a", 1].sort();
```

As you can see, it sorts by dictionary order (numbers first, followed by letters). But what if you want to sort your array differently? For example, listing "better" cars first? How would you go about that? Fortunately, `sort` accepts a custom comparison function.

```
var cars = [
  "Honda Civic",
  "Ford Taurus",
  "Chevy Malibu"
]
```

```

cars.sort(function(a, b) {
    // Default to 0, which indicates
    // no sorting is necessary
    var returnVal = 0;

    // If `a` is a Chevy, subtract 1
    // to move `a` "up" in the sort order
    // because Chevys are awesome.
    if (a.match(/Chevy/)) {
        returnVal = returnVal - 1;
    }

    // If `b` is a Chevy, add 1
    // to move `b` "up" in the sort order
    if (b.match(/Chevy/)) {
        returnVal = returnVal + 1;
    }

    return returnVal;
});

// Outputs:
// ["Chevy Malibu", "Honda Civic", "Ford Taurus"]
console.log(cars);

```

The comparison function compares two values and returns a number.

- If it returns a negative number, `a` will be sorted to a lower index in the array.
- If it returns a positive number, `a` will be sorted to a higher index.
- And if it returns `0` no sorting is necessary.

When you pass `sort` a comparison function, that function will be called repeatedly and given different values from the array until the array has been completely sorted.

It is important to note that returning `0` does not guarantee that `a` and `b` will remain in the same order. It merely means that sorting is not necessary. JavaScript

engines may choose to keep them in the same order, but that is not part of the language spec.

Thanks for reading! If you enjoyed this issue, why not subscribe to the newsletter?

---

© 2015. All rights reserved.