

Javascript Pointers (They do exist!)



VINCENT PASSANISI | JULY 20, 2021



Javascript is often criticized for being a simple language, but if you take a closer look, there is actually a lot of nuance to it. For example, other languages often have pointers which are a way to store a reference to a specific place in memory. In C, Golang, and C# pointers are made using the ‘&’ operator. At first glance, there seems to be a conspicuous absence of this functionality for javascript. However, it turns out that there are pointers in JavaScript, but they have just been implemented in a different way. In



fact, variables that are an object or an array are always just pointers. Here is an example:

```
1 //js pointer example
2 const obj1 = {
3   city: 'Boston',
4   state: 'Massachusetts'
5 };
6
7 const obj2 = obj1;
8 obj2.city = 'Springfield';
9
10 console.log(obj1); // {city: 'Springfield', st
```

You can see that mutating obj2 also mutates obj1 because, under the hood, the value of obj1 is really just a pointer to the place in memory that is storing the information in the object. So, obj2 is just the same reference at the same place in memory where the object is stored.

What if you don't want this behavior? What if you really want to make a copy of obj1 and store it in obj2? Since ES2015 the most obvious way would be like this:

```
1 const obj2 = { ...obj1 };
```

This tells the interpreter that you want a completely new object in a new place in memory, and that you want the contents of that object to be the same as the contents of obj1. Don't forget though that obj2 is still a pointer, just to a new object. Alternatively, this also works (although I think it is less intuitive):

```
1 const obj2 = Object.assign({}, obj1);
```

When you take a step back, this functionality begs the question, why are objects and arrays pointers? I think this is a difficult question to answer, but it comes down to the fact that Javascript was meant to be a language that evolves and changes over time. The internet as a whole is

constantly evolving as we find new ways to use it, and we are always thinking of new ways that we would like to use it. In the same way, it is necessary to have a language that can change along with it. Pointers deliver powerful functionality to the engineer, but at the same time, you might not want to be locked into using a specific operator or character just for pointers in a language that may want to change in the future.

Moogsoft is the AI-driven observability leader that provides intelligent monitoring solutions for smart DevOps. Moogsoft delivers the most advanced cloud-native, self-service platform for software engineers, developers and operators to instantly see everything, know what's wrong and fix things faster.

See Related Posts by Topic: [ENGINEERING](#) [IT](#)
[INDUSTRY](#) [OUR COMPANY](#) [OUR SOLUTIONS](#)

About the author



Vincent Passanisi

I am a Software Engineer at Moogsoft, living in Santa Cruz California. I love all web-related technologies including JavaScript, TypeScript, CSS, and much more!

[All Posts by Vincent Passanisi](#)

Moogsoft Resources

