A Drip of JavaScript

Emulating Block Scope in JavaScript

Originally published in the A Drip of JavaScript newsletter.

While there are many issues that trip up developers coming from other languages, variable scoping may be number one. The fundamental problem is that many expect variables to be scoped to a particular block (like a `for` loop), but in JavaScript variables declared with `var` are scoped to the nearest parent function.

First let's take a look at how this can go wrong.

```
var avatar = "Ang";
var element = "Air";

var elements = [
    "Air",
    "Earth",
    "Fire",
    "Water"
];

for (var i = 0; i < elements.length; i++) {
    var element = elements[i];
    console.log(avatar + " has mastered " + element);
}

// Outputs: "Ang's primary element is Water"
console.log(avatar + "'s primary element is " + element);
```

A developer used to a language with block scoping might not see anything wrong with the code above, and would expect `"Ang's primary element is Air"` instead of the actual result.

This issue is easily avoided once you become aware of it. Avoiding variable declarations within blocks tends to prevent any confusion.

But suppose that we really wanted to use block scoping in JavaScript. How would we go about it? We could do something like this.

```
var avatar = "Ang";
var element = "Air";

var elements = [
    "Air",
    "Earth",
    "Fire",
    "Water"
];

for (var i = 0; i < elements.length; i++) {
    (function() {
        var element = elements[i];
        console.log(avatar + " has mastered " + element);
    })();
}

// Outputs: "Ang's primary element is Air"
console.log(avatar + "'s primary element is " + element);
```

This solution uses an IIFE to emulate block scoping. Since functions are JavaScript's scoping mechanism, we define and immediately invoke a new function on each pass through the loop, therefore approximating the behavior of a block scope.

While this isn't idiomatic JavaScript, it does help give you an idea of just how flexible JavaScript can be.

It is also worth noting that drafts of the ECMAScript 6 specification (the next version of JavaScript) include a `let` keyword which is used to define block-scoped variables. If you are interested in playing around with this proposed keyword, it is available in FireFox.

Thanks for reading!

Josh Clanton
