



A Drip of JavaScript

Sealing JavaScript Objects with `Object.seal`

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

In the last drip we talked about making an object completely immutable. But suppose you need something a little less than full immutability? Then you're in luck.

`Object.freeze` has a little brother named `Object.seal`.

Let's walk through how it works.

```
var rectangle = {  
  height: 5,  
  width: 10  
};  
  
Object.seal(rectangle);  
  
rectangle.depth = 15;  
  
rectangle.width = 7;  
  
// Outputs: {  
//   height: 5,  
//   width: 7  
// }  
console.log(rectangle);
```

As you can see, once the object is sealed, new properties can't be added, but existing properties can still be modified.

In addition to preventing the addition of new properties, a sealed object can't have properties removed via `delete`. For example:

```
delete rectangle.width;

// Outputs: {
//   height: 5,
//   width: 7
// }
console.log(rectangle);
```

`Object.seal` has one final effect. It makes all object properties non-configurable, preventing you from configuring them into a different state with `Object.defineProperty` and similar methods.

```
Object.defineProperty(rectangle, "height", {
  writable: false
});

rectangle.height = 22;

// Outputs: 22
console.log(rectangle.height);
```

In this example, despite attempting to configure the writability to `false`, the `height` property remains writable. (For a refresher on `Object.defineProperty`, [see drip #30](#).)

Attempting to make any of these forbidden modifications to a sealed object will either fail silently or (in strict mode) throw an error.

Fortunately, we also have a method to detect whether an object is sealed.

```
// Outputs: true  
console.log(Object.isSealed(rectangle));
```

Like `Object.freeze`, `Object.seal` is part of the ECMAScript 5 specification, which means it isn't available in older browsers like IE8 and below. If you need to support those browsers, then you'll need to either avoid `Object.seal` or use feature detection to use it only in the browsers that support it.

© 2015. All rights reserved.