



JavaScript  
Article

## Back to Basics: JavaScript Hoisting

By Colin Ihrig    February 02, 2013

Variable declarations are one of the most basic aspects of any programming language. However, JavaScript has a little quirk, known as *hoisting*, which can turn an innocent looking declaration into a subtle bug. This article explains what hoisting is, and how you can avoid being burned by it.

JavaScript is an extremely flexible language, and will happily allow you to declare a variable almost anywhere. For example, the following immediately-invoked function expression (IIFE) declares three variables and then displays them using an alert dialog box. As a side note, you should never use alert boxes, but we're trying to prove a point here.

```
(function() {  
  var foo = 1;  
  var bar = 2;  
  var baz = 3;  
  
  alert(foo + " " + bar + " " + baz);  
})();
```

This looks like sane JavaScript code. As expected, it displays the string `"1 2 3"`. Now, assume that the alert is moved, as shown below.

```
(function() {  
  var foo = 1;  
  alert(foo + " " + bar + " " + baz);  
  var bar = 2;  
  var baz = 3;  
})();
```

If someone actually wrote this code, it was probably by mistake. Clearly, the alert takes place before `bar` and `baz` are declared. However, this is perfectly valid JavaScript, which does not generate an exception. Instead, the alert displays `"1 undefined undefined"`.

Based on our previous experiment, it seems that you can reference variables that don't exist yet. Now, let's take the same IIFE, but remove the `baz` declaration altogether, as shown below. Suddenly, we have a `ReferenceError` because `baz` is not defined.

```
(function() {  
  var foo = 1;  
  alert(foo + " " + bar + " " + baz);  
  var bar = 2;  
})();
```

This is truly interesting behavior. To understand what's going on here, you have to understand hoisting. Hoisting is the JavaScript interpreter's action of moving all variable and function declarations to the top of the current scope. However, only the actual declarations are hoisted. Any assignments are left where they are. Therefore, our second example IIFE actually translates to the following code.

```
(function() {  
  var foo;  
  var bar;  
  var baz;  
  
  foo = 1;  
  alert(foo + " " + bar + " " + baz);  
  bar = 2;  
  baz = 3;  
})();
```

Now it makes sense why the second example didn't generate an exception. After hoisting, **bar** and **baz** are actually declared before the alert statement, albeit with undefined values. In the third example, **baz** was removed completely. Therefore, there was nothing to hoist, and the alert statement resulted in an exception.

## Function Hoisting

As previously mentioned, function declarations are also hoisted. However, functions that are assigned to variables are not hoisted. For example, the following code will work as expected due to function declaration hoisting.

```
foo();  
  
function foo() {  
  alert("Hello!");  
}
```

However, the following example will fail spectacularly. The variable declaration for **foo** is hoisted before the function call. However, since the assignment to **foo** is not hoisted, an exception is thrown for trying to call a non-function variable.

```
foo();  
  
var foo = function() {  
  alert("Hello!");  
};
```

## Conclusion

Hoisting is an easy to understand, but often overlooked nuance of the JavaScript language. Without a proper understanding of hoisting, your programs are susceptible to subtle bugs. To help avoid these bugs, many developers (and linting tools) advocate for a single variable declaration statement at the very beginning of every scope. Since this is how the JavaScript interpreter essentially sees your code, there is validity to this rule – even if I am personally guilty of breaking it.

Was this helpful?



More:

[functions](#), [Hoisting](#)



Meet the author

**Colin Ihrig** [Twitter](#) [Google+](#) [LinkedIn](#) [GitHub](#)

Software Engineer. Freelance Technology Writer.

Comments for this thread are now closed.



8  
Comments

SitePoint

[Login](#)

[Recommend](#) **2** [Share](#)

Sort by Best



**irfan** • 4 years ago

I wish I had known about this when I first started learning JavaScript. it would have saved me so much time and frustration. Thank you for the article.

[^](#) [v](#) • [Share](#)



**Peter** • 4 years ago

Excellent explanation about hoisting - Thanks.

[^](#) [v](#) • [Share](#)



**imJohnBon** • 4 years ago

Hey this was a great article, totally understand hoisting now. Before it just seemed like one of those weird programming terminologies that would take me a while to comprehend, so I never bothered. Hope you write more like this!

[^](#) [v](#) • [Share](#)



**Colin Ihrig** [→](#) imJohnBon • 4 years ago

Thanks for reading Jay!

[^](#) [v](#) • [Share](#)



**Madryt** • 4 years ago

Great article and straight to the point. I don't know if this is really the best place to ask but do you guys have any idea where to hire some professional writers? Thanks in advance :)

[^](#) [v](#) • [Share](#)



**Colin Ihrig** [→](#) Madryt • 4 years ago

Thank you Madryt. I'm sorry, but I'm not sure where you can hire writers. Maybe a site like elance?

Although we compensate our writers, we don't normally actively "hire."

^ | v • Share >



**Aurelio De Rosa** → Madryt • 4 years ago



Colin this is a very good and well explained article. I think all the programmers starting with JavaScript should read this piece. Good work!

@Madryt: I think this is a little off-topic, however, I think the JSPro writers are good ones and maybe someone can be interested in.

^ | v • Share >



**Colin Ihrig** → Aurelio De Rosa • 4 years ago



Thank you Aurelio!

1 ^ | v • Share >

[Subscribe](#)

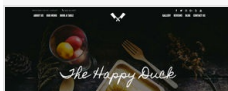
[Add Disqus to your site](#)

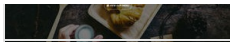
[Privacy](#)

**DISQUS**

LATEST THEMES >

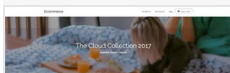
PREMIUM THEME





## SitePoint WordPress Restaurant Theme

PREMIUM THEME



## SitePoint WordPress Ecommerce Theme

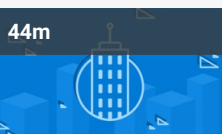
PREMIUM THEME



## SitePoint WordPress Portfolio Theme

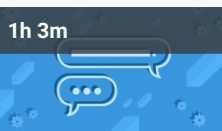
## LATEST COURSES >

PREMIUM COURSE



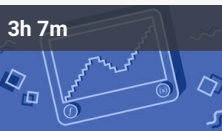
### Introducing TypeScript

PREMIUM COURSE



### MS Bots

PREMIUM COURSE



### Functional JavaScript Programming

## LATEST BOOKS >

PREMIUM BOOK



### ECMAScript 2015: A SitePoint Anthology

PREMIUM BOOK



### Jump Start Git

PREMIUM BOOK



### Full Stack JavaScript Development with MEAN

Get the latest in JavaScript, once a week, for free.

Enter your email

Subscribe



Facebook



LinkedIn



Twitter

## About

[Our Story](#)  
[Advertise](#)  
[Press Room](#)  
[Reference](#)  
[Terms of Use](#)  
[Privacy Policy](#)  
[FAQ](#)  
[Contact Us](#)  
[Contribute](#)

## Visit

[SitePoint Home](#)  
[Themes](#)  
[Podcast](#)  
[Forums](#)  
[Newsletters](#)  
[Premium](#)  
[References](#)  
[Versioning](#)

## Connect



© 2000 – 2017 SitePoint Pty. Ltd.

