

JavaScript Temporal Dead Zone

Aug 29, 2017

The **Temporal Dead Zone** is a behavior in JavaScript that occurs when declaring a variable with the `let` and `const` keywords, but not with `var`. A shorthand you'll often hear to describe it is that "Let's don't hoist," but this is not technically true. Read on for a brief description of what's really occurring.

In JavaScript, variable declarations (but not assignments) are **hoisted to the top of the scope**. The code below works as expected:

```
function myFunc(){  
  var greeting = "Hello World!";  
  console.log(greeting);  
}
```

```
myFunc(); // "Hello World!"
```

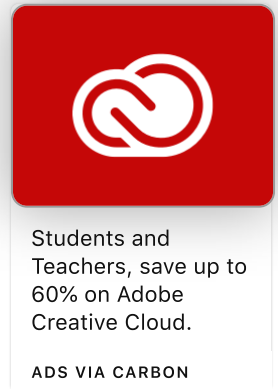
But if you reverse the order and try to run `console.log` on a variable before creating it?

```
function myFunc(){  
  console.log(greeting);  
  var greeting = "Hello World!";  
}
```

Which of these three options will be the output?

- 'Hello World!'
- An error saying `greeting` is not yet defined
- `Undefined`

The answer is `undefined` because the variable declaration is **hoisted** to the top of the scope. So in the eyes of the JavaScript interpreter, the code really looks as follows:



```
function myFunc(){  
  var greeting;  
  console.log(greeting);  
  greeting = 'Hello World!'  
}
```

The JavaScript interpreter works in a two-step process:

- **compile time:** run through all code looking for variable/function declarations
- **runtime:** execute the code including assignments and function invocations

Therefore on the first line, `greeting` is defined but has no assignment. JavaScript automatically provides the value `undefined` to defined variables without any variable. On line 2 the result will be `undefined` because the assignment does not occur until line three.

But if you use either the `let` or `const` keywords to declare a variable, this same code will throw an error:

```
function myFunc() {  
  console.log(greeting);  
  let greeting = 'Hello World!';  
}
```

```
myFunc(); // ReferenceError: greeting is not defined
```

```
function myFunc() {  
  console.log(greeting);  
  const greeting = 'Hello World!';  
}
```

```
myFunc(); // ReferenceError: greeting is not defined
```

This is the **Temporal Dead Zone** where we're trying to access a variable that has not yet been initialized (it has been declared and therefore exists, but has no value, not even `undefined`). It's common to hear the phrase, "let/const don't hoist" as a shorthand to remember this behavior. But technically something else is going on.

When we use the `var` keyword, two things actually happen:

1. at **compile time**, the variable is added to the enclosing lexical scope
2. at **runtime**, when the scope is entered any variables added to the lexical environment are initialized to the `undefined` value so they are available to use in the scope

The `let` and `const` keywords do step 1—so technically they do hoist—but not step 2, the assignment to `undefined`. Therefore it's more accurate to say that `let/const` **do hoist but don't get initialized**.

The end result is the same. Just remember that when using `let/const` you should *always move variable declarations to the top of your scope* to avoid the **temporal dead zone**.

Want to improve your JavaScript? I have a list of [recommended JavaScript books](#).

Receive the latest tutorials/writings by email.

Subscribe

No spam. Promise. Unsubscribe at any time.

[1 Comment](#) [wsvincent.com](#) [Disqus' Privacy Policy](#)[1 Login](#) ▾[♥ Recommend 1](#) [🐦 Tweet](#) [f Share](#)[Sort by Best](#) ▾

LOG IN WITH

OR SIGN UP WITH DISQUS [?](#)**Vishweshvar** • 4 months ago

Hey there..., I loved the way you explained diff between let/const and var in terms of hoisting. But i have a doubt. Suppose i declare(with let/const) and fail to assign a value. When i try to log the variable in console, i will get 'undefined', if the variable is in temporal dead zone until initialized, why does it not throw 'reference error' in this case. Or at what stage does the variable receives 'undefined' value from JS interpreter or engine.

[^](#) | [v](#) • [Reply](#) • [Share](#) ▾[✉ Subscribe](#) [🔗 Add Disqus to your site](#) [Add Disqus](#) [⚠ Do Not Sell My Data](#)

© William Vincent

[About](#) [Books](#) [Reading List](#)