

JavaScript Hoisting

Hoisting in JavaScript is a behavior in which a **function** or a variable can be used before declaration. For example,

```
// using test before declaring
console.log(test); // undefined
var test;
```

Run Code >>

The above program works and the output will be `undefined`. The above program behaves as

```
// using test before declaring
var test;
console.log(test); // undefined
```

Run Code >>

Since the variable `test` is only declared and has no value, `undefined` value is assigned to it.

If you want to learn more about variables, visit [JavaScript Variables](#).

Note: In hoisting, though it seems that the declaration has moved up in the program, the actual thing that happens is that the function and variable declarations are added to memory during the compile phase.

Variable Hoisting

In terms of variables and constants, keyword `var` is hoisted and `let` and `const` does not allow hoisting.

For example,

```
// program to display value
a = 5;
console.log(a);
var a; // 5
```

Run Code >>

In the above example, variable `a` is used before declaring it. And the program works and displays the output `5`. The program behaves as:

```
// program to display value
var a;
a = 5;
console.log(a); // 5
```

Run Code >>