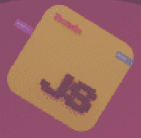
JavaScript Hoisting



Hey Everyone 👋

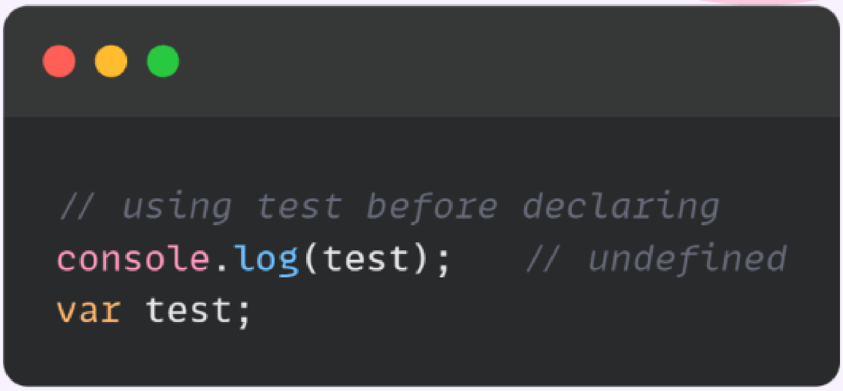
In this post, we will learn about Javascript Hoisting with the help of Examples.

Do Like, save and Share This Post If You Found This Helpful.

Next →

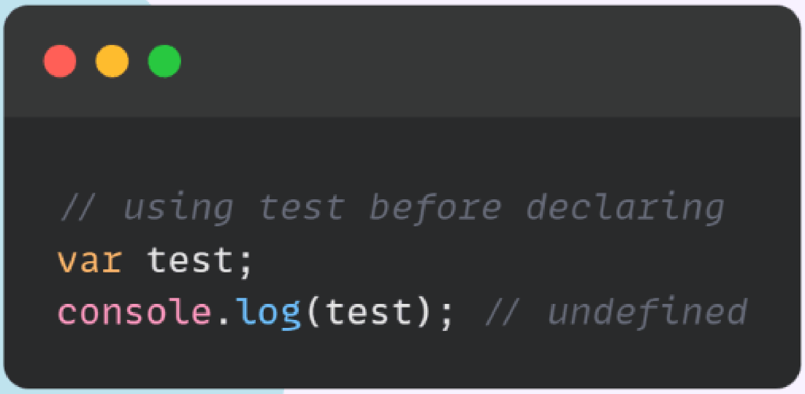
Javascript Hoisting

- **Hoisting** in Javascript is a behavior in which a function or variable can be used **before declaration**.



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

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

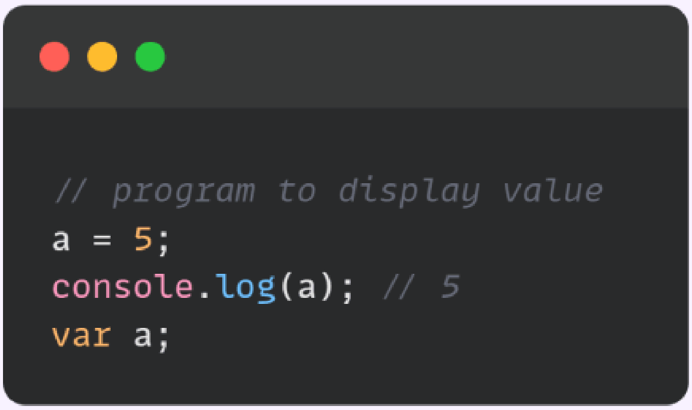


```
// using test before declaring
var test;
console.log(test); // undefined
```

NEXT →

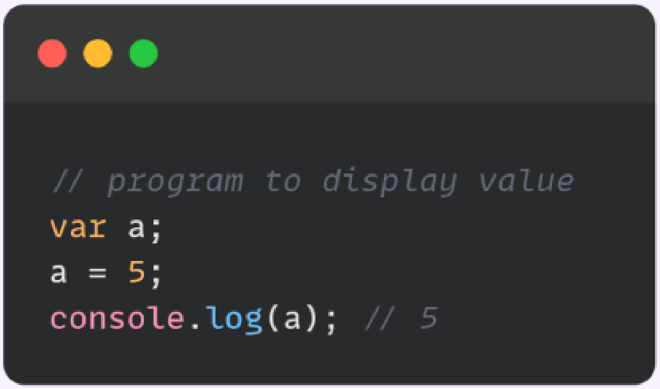
Variable Hoisting

- In terms of variable and constant, keyword var is hoisted, and let and const does not allow hoisting.



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

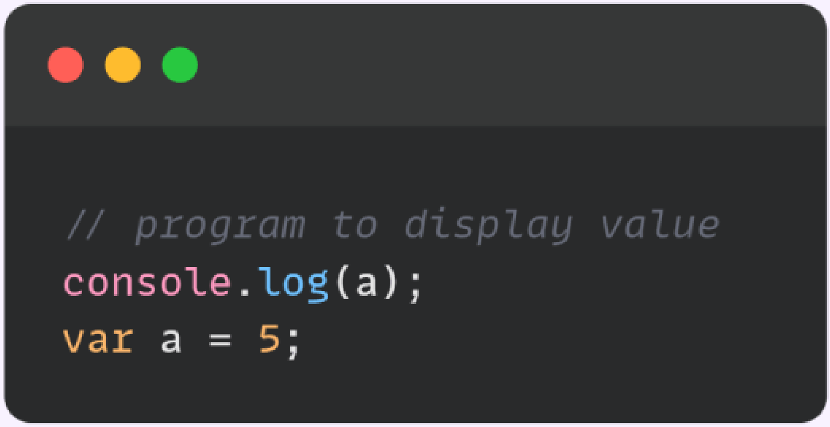
- In above example, variable a is used before declare it, displays the output 5
- the program behave as :



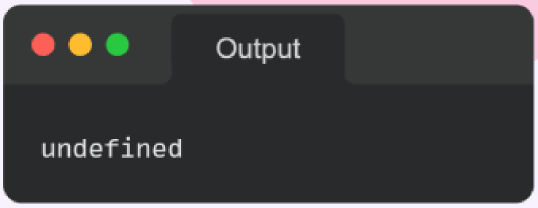
```
// program to display value
var a;
a = 5;
console.log(a); // 5
```

Next →

- However in Javascript, **initializations are not hoisted.**

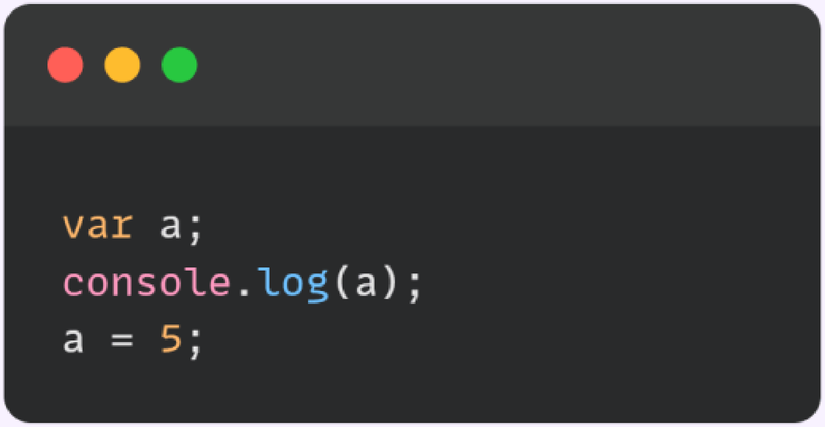


```
// program to display value
console.log(a);
var a = 5;
```

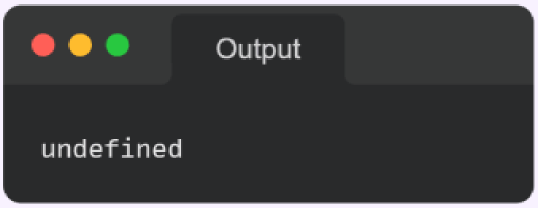


```
undefined
```

- The above program behave as :



```
var a;
console.log(a);
a = 5;
```



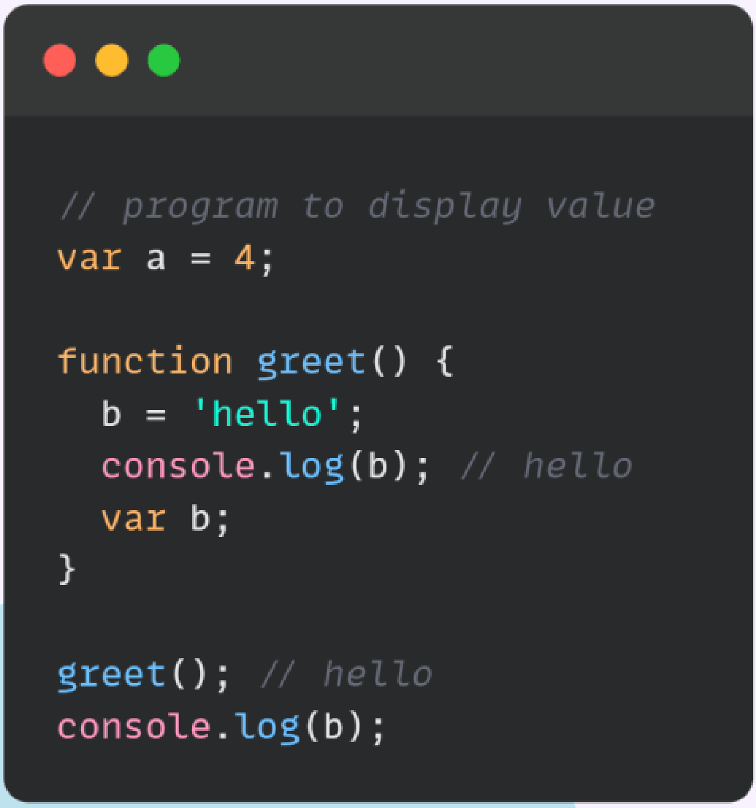
```
undefined
```

- Only the **declaration is moved** to the memory in the compile phase.

NEXT →

5

- Also, when the variable is used inside the function, the variable is hoisted only to the top of the function.

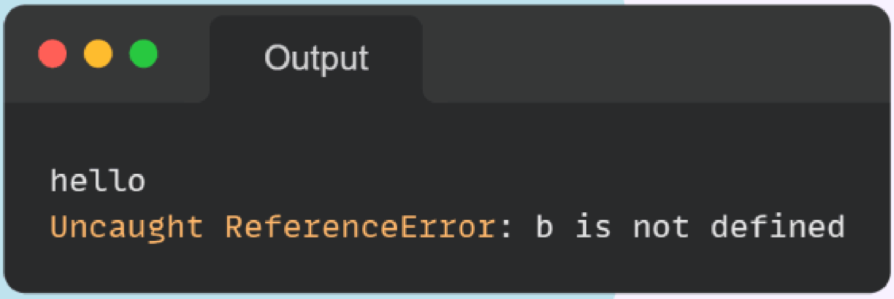


```
// program to display value
var a = 4;

function greet() {
  b = 'hello';
  console.log(b); // hello
  var b;
}

greet(); // hello
console.log(b);
```

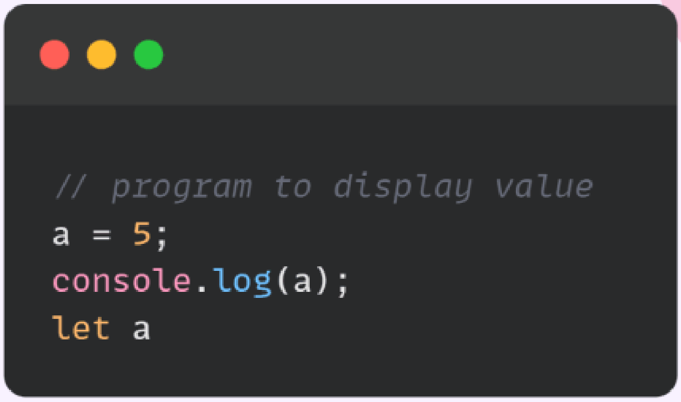
- b is hoisted to the top of the function greet and become a local variable.
- hence b is only accessible inside the function.
- b is not become global variable



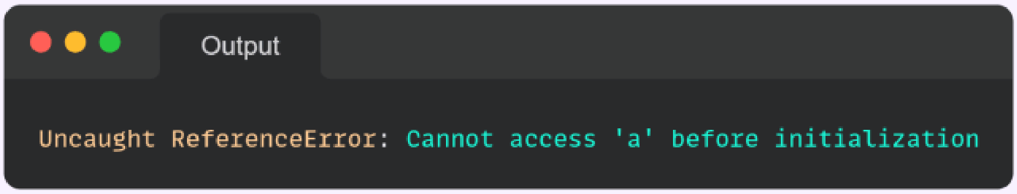
```
hello
Uncaught ReferenceError: b is not defined
```

NEXT →

- if a variable is used with the let or const keyword, that variable is not hoisted.



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



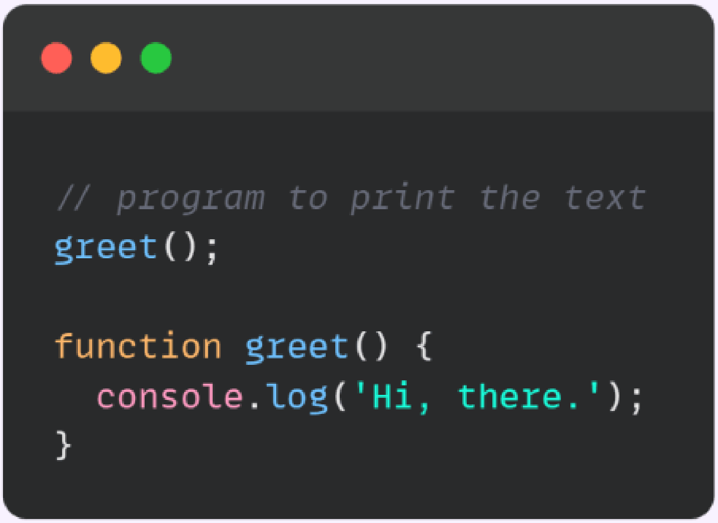
```
Uncaught ReferenceError: Cannot access 'a' before initialization
```

- **NOTE** : While using let, the variable must be declared first.

Next →

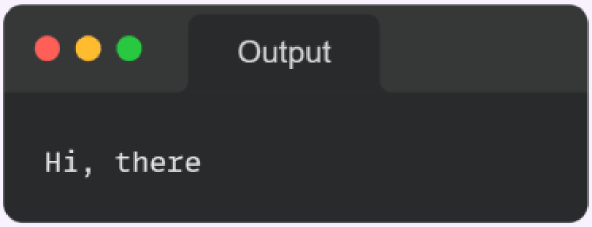
Function Hoisting

- a function can be called before declare it.



```
// program to print the text
greet();

function greet() {
  console.log('Hi, there.');
}
```

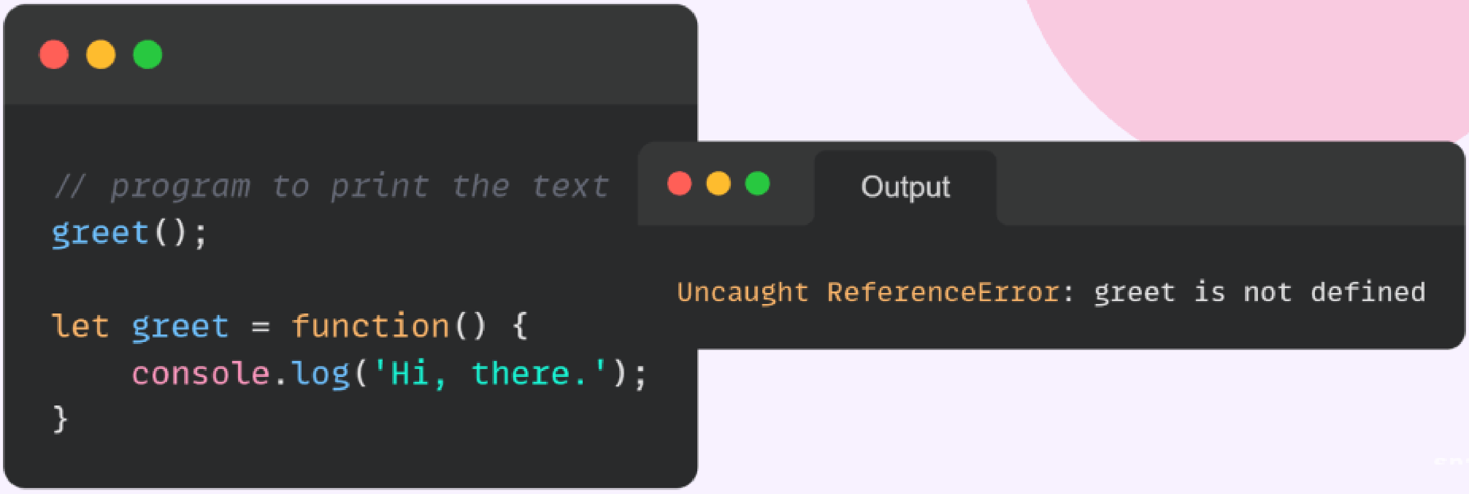


```
Hi, there
```

- In the above program, the function greet is called before declaring it and the program shows the output.
- This is due to hoisting.

NEXT →

- However, when a function is used as an **expression**, an error occurs because only declarations are hoisted.

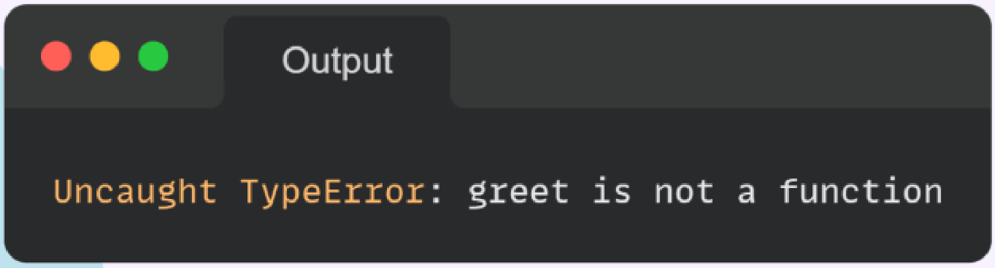


```
// program to print the text
greet();

let greet = function() {
    console.log('Hi, there.');
}
```

```
Uncaught ReferenceError: greet is not defined
```

- that’s why **arrow function is not hoisted.**
- If var was used in the above program, the error would be:



```
Uncaught TypeError: greet is not a function
```

NEXT →

THANKS FOR YOUR ATTENTION

Code.Clash

Imtiyaz Nandasaniya

LIKE AND SAVE IT FOR LATER