# Var and let

## Var

var is a keyword used to create a variable. it is a placeholder for another value that has been used during the entire execution of that function. The var is a function scoped, means any variable created using var is exist during the entire execution of that function.

## let

let is also a keyword used to create variables. it works same as any other variable but it is a block scope, means the existence of that variable is only till the block gets execute, once the block get executed the variable pops out of the variable list.

### First code

example of var and let

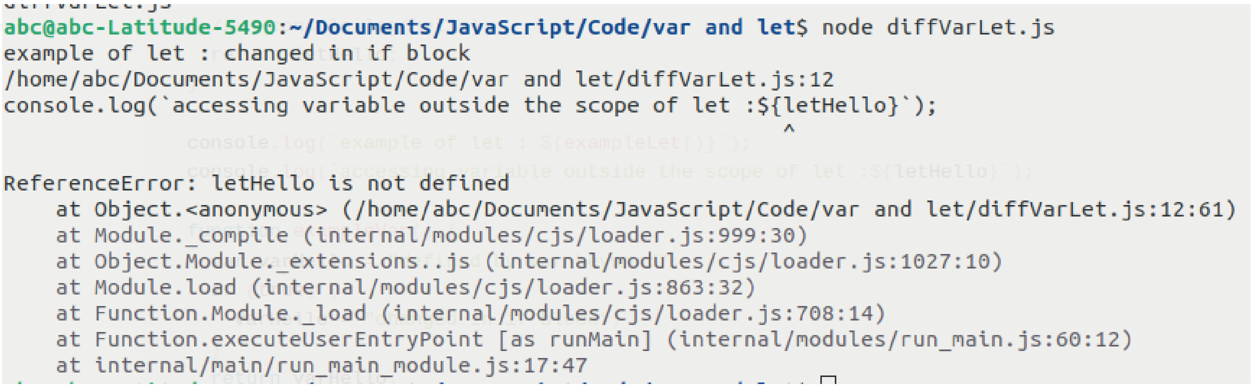
```
function exampleLet() {
  let letHello = "assigned at the time of creation";
  if (true) {
    // hello = 45;
    letHello = "changed in if block";
  }
  //   console.log(hello);
  return letHello;
}

console.log(`example of let : ${exampleLet()}`);
console.log(`accessing variable outside the scope of let :${letHello}`);

function exampleVar() {
  var varHello = "defined by var keyword";
  if (true) {
    varHello = "changed in if block";
  }
  return varHello;
}

console.log(`example of var : ${exampleVar()}`);
console.log(`accessing variable outside the scope of var : ${varHello}`);
```

output of the above snippets



```
abc@abc-Latitude-5490:~/Documents/JavaScript/Code/var and let$ node diffVarLet.js
example of let : changed in if block
/home/abc/Documents/JavaScript/Code/var and let/diffVarLet.js:12
console.log(`accessing variable outside the scope of let :${letHello}`);
                                                            ^

ReferenceError: letHello is not defined
    at Object.<anonymous> (/home/abc/Documents/JavaScript/Code/var and let/diffVarLet.js:12:61)
    at Module._compile (internal/modules/cjs/loader.js:999:30)
    at Object.Module._extensions..js (internal/modules/cjs/loader.js:1027:10)
    at Module.load (internal/modules/cjs/loader.js:863:32)
    at Function.Module._load (internal/modules/cjs/loader.js:708:14)
    at Function.executeUserEntryPoint [as runMain] (internal/modules/run_main.js:60:12)
    at internal/main/run_main_module.js:17:47
```

## Second code

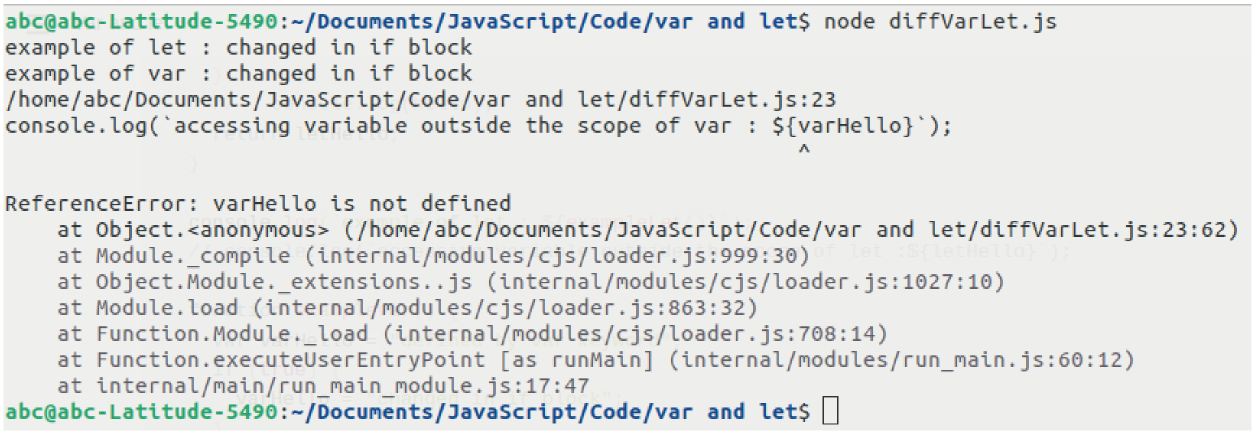
```
function exampleLet() {
  let letHello = "assigned at the time of creation";
  if (true) {
    // hello = 45;
    letHello = "changed in if block";
  }
  //   console.log(hello);
  return letHello;
}

console.log(`example of let : ${exampleLet()}`);
// console.log(`accessing variable outside the scope of let :${letHello}`);

function exampleVar() {
  var varHello = "defined by var keyword";
  if (true) {
    varHello = "changed in if block";
  }
  return varHello;
}

console.log(`example of var : ${exampleVar()}`);
console.log(`accessing variable outside the scope of var : ${varHello}`);
```

output of above snippet



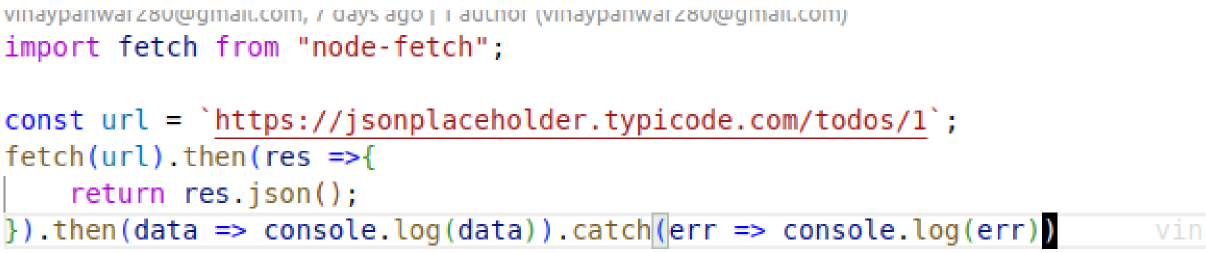
```
abc@abc-Latitude-5490:~/Documents/JavaScript/Code/var and let$ node diffVarLet.js
example of let : changed in if block
example of var : changed in if block
/home/abc/Documents/JavaScript/Code/var and let/diffVarLet.js:23
console.log(`accessing variable outside the scope of var : ${varHello}`);
                                                             ^

ReferenceError: varHello is not defined
    at Object.<anonymous> (/home/abc/Documents/JavaScript/Code/var and let/diffVarLet.js:23:62)
    at Module._compile (internal/modules/cjs/loader.js:999:30)
    at Object.Module._extensions..js (internal/modules/cjs/loader.js:1027:10)
    at Module.load (internal/modules/cjs/loader.js:863:32)
    at Function.Module._load (internal/modules/cjs/loader.js:708:14)
    at Function.executeUserEntryPoint [as runMain] (internal/modules/run_main.js:60:12)
    at internal/main/run_main_module.js:17:47
abc@abc-Latitude-5490:~/Documents/JavaScript/Code/var and let$ 
```

## Const

Const is a keyword which is used to define the variable that are need be constant throughout a program. means the value of those variable shouldn’t get change during the code.

once the const variable is assigned the value of that variable can’t be reassigned after that. we can’t create a variable with same name to existing const variable.



```
vinaypanwar280@gmail.com, 7 days ago | 1 author (vinaypanwar280@gmail.com)
import fetch from "node-fetch";

const url = `https://jsonplaceholder.typicode.com/todos/1`;
fetch(url).then(res =>{
    return res.json();
}).then(data => console.log(data)).catch(err => console.log(err))
```

### More on var, let and const

The differences between var, let, and const variable declaration in JavaScript include: Variables declared with var and const are scoped to the immediate function body.

Variables declared with the var keyword are hoisted. Hoisting means that the variable can be accessed in their enclosing scope even before they are declared.
Variables declared with the let keyword are block-scoped, which means the variables will have scope to the immediate enclosing block.