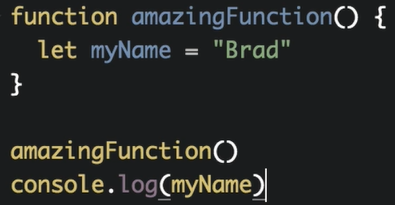
Scope:

let declares a variable that is block-scoped (curly brackets {} of the amazingFunction).

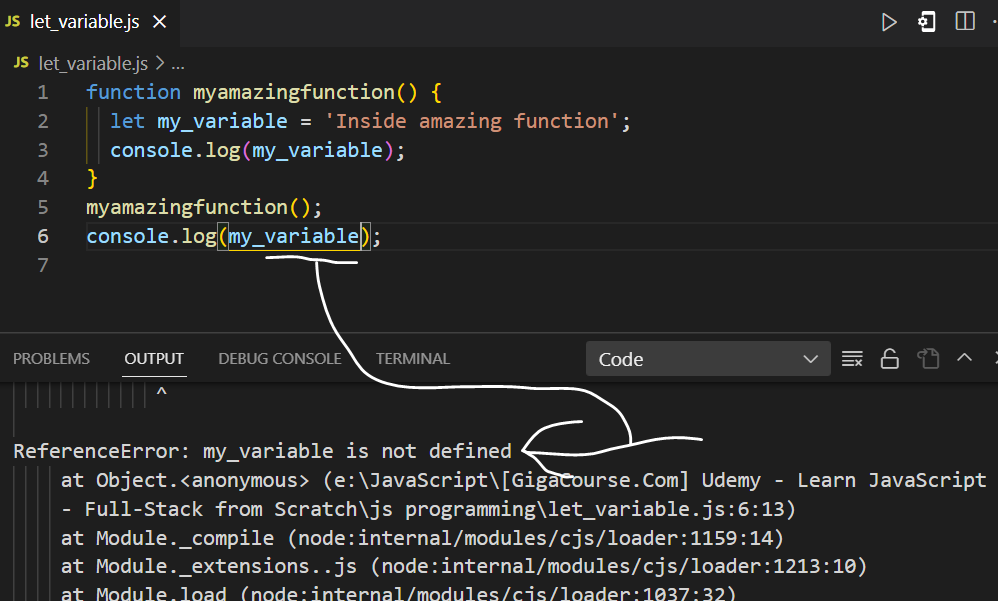
Curly brackets of while loop, for loop ,function and if statement is a block.

Hence, console.log(myName) can only be accessed inside amazing function, not outside.





Attempting to access the my\_variable outside the myamazingfunction will give error.



**Globally declared let variable:**

Globally declared let variable can be accessed an where inside (inside function, inside if, inside for loop, inside while loop ) the program.

The function call occurs first then last console.log is executed

Text

Description automatically generated

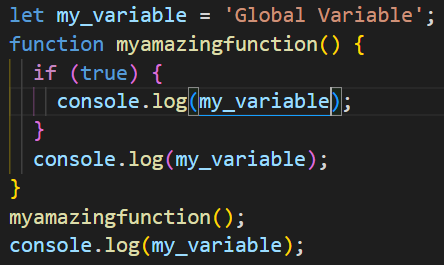
Text

Description automatically generated

**Scope is like a one-way street:**

Code can reach outward to access the variable, code cannot reach inward to access the variable.

Scope begins inward and moves outward.



When if code in if statement is executed, JavaScript sees if there is my\_variable within current scope (with in the curly brackets {} of if or block).

Block is the curly brackets of if, function, for loop and while loop.

If my\_variable is not found with in the current scope then, JavaScript will look one level up (with in the curly brackets of myamazingfunction), even still my\_variable is not found within one level up then JavaScript will move one level up (outside the curly brackets of myamazingfunction) and upon finding the my\_variable JavScript will print it on console.

Graphical user interface, text

Description automatically generated

**Each time we declare variable using a let key word brand new variable is created:**

// Following my\_variable is  new global variable is accessible anywhere

let my\_variable = 'Global Variable';

function myamazingfunction () {

  // Following my\_variable is a new variable and is only accessible anywhere inside myamazingfunction

  let my\_variable = 'I am a brand-new variable inside myamazingfunction';

  console.log(my\_variable);

  if (true) {

    // Following my\_variable is a new variable and is accessible anywhere inside the myamazingfunction

    let my\_variable = 'I am a brand-new variable inside if';

    console.log(my\_variable);

  }

}

**Output:**

Text

Description automatically generated

**Mutating global variable:**

// Followong my\_variable is  new global variable is accessible anywhere

let my\_variable = 'Global Variable';

function myamazingfunction() {

  // Updating and mutating global variable inside myamazingfunction

  // Accessing my\_variable global variable

  my\_variable = 'I am a mutated varible inside myamazingfunction';

  console.log(my\_variable);

  if (true) {

    // Updating and mutating global variable inside if

    my\_variable = 'I am a mutated  variable inside if';

    console.log(my\_variable);

  }

}

myamazingfunction();

console.log(my\_variable);

Inside if statement ,JavaScript will look whether variable my\_variable exists inside the current block(curly brackets of if) of if statements. If the variable my\_variable does not exist in the current local scope (curly brackets of if ) , then JavaScript will look one level up(inside myamazing function) for the variable my\_variable. If still variable my\_variable does not exist inside myamazing function then JavaScript will continue to look one level up until variable my\_variable is found and variable my\_variable is updated where it accessed and set to some particular value.

Similarly:

function myamazingfunction() {

  let my\_variable = 'Global Variable';

  // Updating and mutating global variable inside myamazingfunction

  // Accessing my\_variable global variable

  my\_variable = 'I am a mutated varible inside myamazingfunction';

  console.log(my\_variable);

  if (true) {

    // Updating and mutating global variable inside if

    my\_variable = 'I am a mutated  variable inside if';

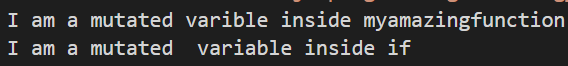
    console.log(my\_variable);

  }

}

myamazingfunction();

**Output:**



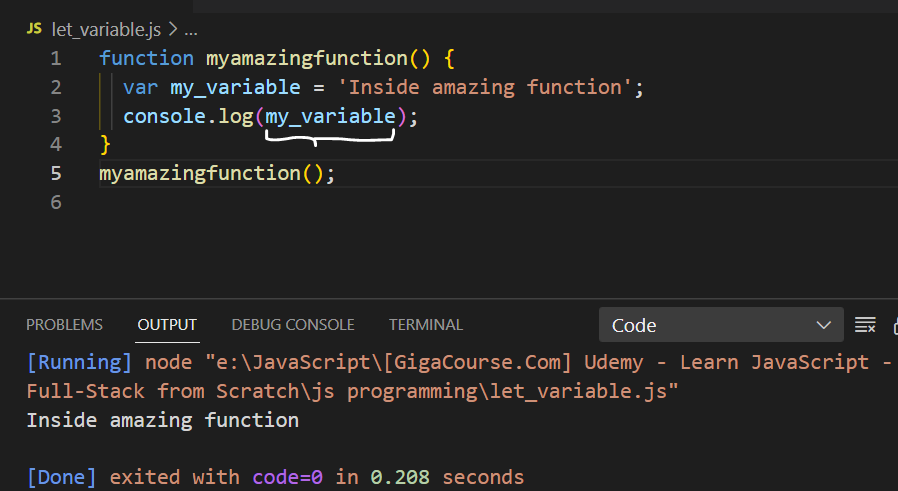
JavaScript will start looking for the variable my\_variable from the if statement and keep looking one level up until variable my\_variable is found inside myamazing function, and variable my\_variable is updated where (inside if statement and myamazing function) it is set to some value.

**Using Var:**

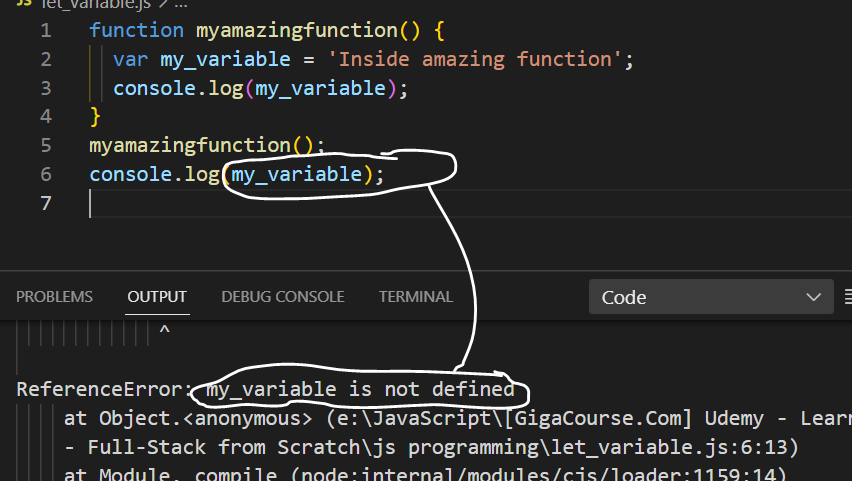
When var variable declared globally, it can be accessed throughout the entire program. When declared locally to a function, it can only be accessed within that function.

Var uses function scope , hence variable declared with var keyword is accessible anywhere inside the function.

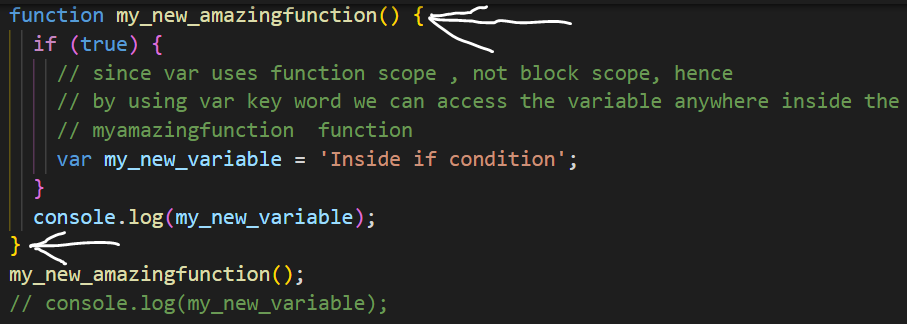
**Locally declared var variable:**



**Attempting to access the locally created var variable outside a function will give error**



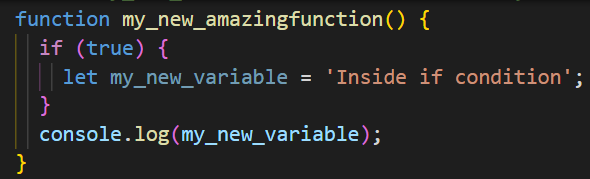
**Since var uses function scope not the block scope so if variable created using var key word inside if (if must be inside function) can be accessed:**



**Output**



**Whereas if we had used let key word then, my\_new\_variable would not be accessible inside my\_new\_amazingfunction, it would only be accessible inside if statement.**



Text

Description automatically generated