

Example: In this example, we will declare a block using curly braces “{ }”, and inside that block, we will declare a variable having a certain value in it. We will call that variable outside the blocked scope to see what output it actually displays upon calling.

Javascript

```
{  
    let x = 13;  
}  
// Throws a reference error  
// since x is declared inside a block which  
// cannot be used outside the block  
console.log(x);
```

Output:

```
Uncaught ReferenceError: x is not defined
```

Scope Chain

- JavaScript engine uses scopes to find out the exact location or accessibility of variables and that particular process is known as Scope Chain.
- Scope Chain means that one variable has a scope (it may be global or local/function or block scope) is used by another variable or function having another scope (may be global or local/function or block scope).
- This complete chain formation goes on and stops when the user wishes to stop it according to the requirement.

Example: In this example, we will first declare a global scope variable which we will use in the later part of the code, and then we will declare the main function inside which we will do some stuff. We will declare another local/function scoped variable inside that main function and just after that we will declare two nested functions (having local/function scope) within the main function itself.



- First, the nested function will print the value of the local/function scoped variable and the second nested function will display the value of the globally scoped variable.
- We will call the two nested functions inside the main function and then call the main function outside its declaration namespace.

Javascript

```
// Global letiable
let global_letiable = 20;

function main_function() {
  // Local letiable
  let local_letiable = 30;

  let nested_function = function () {

    // Display the value inside the local letiable
    console.log(local_letiable);
  }

  let another_nested_function = function () {

    // Displays the value inside the global letiable
    console.log(global_letiable);
  }

  nested_function();
  another_nested_function();
}

main_function();
```

Output

```
30
20
```

