

JS Scope

JavaScript variables have 3 types of scope:

- 1) **Global Scope**
- 2) **Function Scope/Local Scope**
- 3) **Block Scope**

Before ES6 (2015), JavaScript variables only had Global Scope and Function/Local Scope.

Global Scope

- **Internal JS:** JavaScript codes can be written in HTML within the script tag. It is only used rarely when small tasks are needed to be performed.

Global Scope: Any function or variable which is simply kept outside of any loop. Any such variable is called **Global Variable**. And the way it works is known as **Global Scope**.

The function of Global Scope is that any variable kept in Global Scope & the variable which works globally can be accessed from anywhere.

Example: Global JavaScript Variables

```
let carName = "Volvo";  
// code here can use carName  
  
function myFunction() {  
  // code here can also use carName  
}
```

They all have **Global Scope**: Example:

```
var x = 2;           // Global scope  
let x = 2;           // Global scope  
const x = 2;         // Global scope
```

Function Scope/Local Scope

Any scope which is declared within a function is known as **Local Scope**. The characteristics of Local Scope is: the scope/variable cannot be accessed from anywhere except from the place/function it is declared.

Difference between Local Scope and Function Scope

Local scope just refers to the scope available to a given variable, but function scope would refer to variables inside a function.

<https://stackoverflow.com/questions/72536050/difference-between-local-scope-and-function-scope-in-javascript#:~:text=Local%20scope%20just%20refers%20to,only%20accessible%20inside%20the%20function.>

Example: Local Scope

```
// code here can NOT use carName

function myFunction() {
  let carName = "Volvo";
  // code here CAN use carName
}

// code here can NOT use carName
```

They all have **Function Scope**: Example:

```
function myFunction() {
  var carName = "Volvo";    // Function Scope
}

function myFunction() {
  let carName = "Volvo";    // Function Scope
}

function myFunction() {
  const carName = "Volvo";  // Function Scope
}
```

Block Scope

The characteristics of Block Scope is: the variables in any block (each part) cannot be accessed from outside of that block. These variables are **let** and **const**.

Only variable `var` which does not provide Block Scope can be accessed from outside the Block Scope.

Difference between Object and Array

`{}` ==> **Object**

`[]` ==> **Array**

Both Object and Array are non-primitive data.

Array have **Length** and **Index** values. In Array, data is stored directly.

Length: The number of data within an Array.

Index Number: Index Numbers always starts from 0. It is the no. of Length in an Array.

Object and Array are used to store huge amount of data.

Object does not show Length and Index number. In Object, data is stored by means of **Key** and **Value**.

Example:

Key: Value

Roll: 01687

Map Method

Map only works on Array. It works like a loop.

Example of Map Method:

Get the full name for each person:

```
const persons = [  
  {firstname : "Malcom", lastname: "Reynolds"},  
  {firstname : "Kaylee", lastname: "Frye"},  
  {firstname : "Jayne", lastname: "Cobb"}  
];  
  
persons.map(getFullName);  
  
function getFullName(item) {  
  return [item.firstname,item.lastname].join(" ");  
}
```

https://www.w3schools.com/jsref/jsref_map.asp

Sources:

- 1) Class Notes
- 2) https://www.w3schools.com/js/js_scope.asp
- 3) <https://stackoverflow.com/questions/72536050/difference-between-local-scope-and-function-scope-in-javascript#:~:text=Local%20scope%20just%20refers%20to,only%20accessible%20inside%20the%20function.>
- 4) https://www.w3schools.com/jsref/jsref_map.asp