JavaScript Scope

❮ Previous Next ❯

Scope determines the accessibility (visibility) of variables.

JavaScript Function Scope

In JavaScript there are two types of scope:

- Local scope
- Global scope

JavaScript has function scope: Each function creates a new scope.

Scope determines the accessibility (visibility) of these variables.

Variables defined inside a function are not accessible (visible) from outside the function.

Local JavaScript Variables

Variables declared within a JavaScript function, become **LOCAL** to the function.

Local variables have **local scope**: They can only be accessed within the function.

Example

```
// code here can not use carName

function myFunction() {
    var carName = "Volvo";

    // code here can use carName

}
```

Try it Yourself »

Since local variables are only recognized inside their functions, variables with the same name can be used in different functions.

Local variables are created when a function starts, and deleted when the function is completed.

Global JavaScript Variables

A variable declared outside a function, becomes **GLOBAL**.

A global variable has **global scope**: All scripts and functions on a web page can access it.

Example

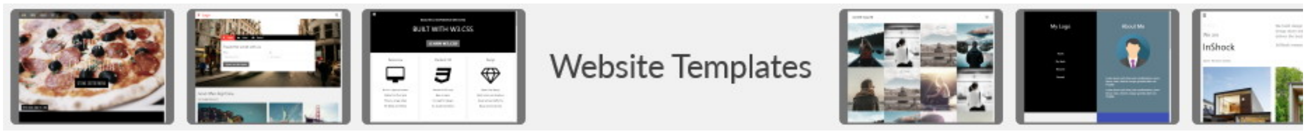
```
var carName = " Volvo";

// code here can use carName

function myFunction() {

    // code here can use carName

}
```

Try it Yourself »

JavaScript Variables

In JavaScript, objects and functions are also variables.

Scope determines the accessibility of variables, objects, and functions from different parts of the code.



Automatically Global

If you assign a value to a variable that has not been declared, it will automatically become a **GLOBAL** variable.

This code example will declare a global variable **carName**, even if the value is assigned inside a function.

Example

```
myFunction();
```

```
// code here can use carName

function myFunction() {
    carName = "Volvo";
}
```

Try it Yourself »

Strict Mode

All modern browsers support running JavaScript in "Strict Mode".

You will learn more about how to use strict mode in a later chapter of this tutorial.

Global variables are not created automatically in "Strict Mode".

Global Variables in HTML

With JavaScript, the global scope is the complete JavaScript environment.

In HTML, the global scope is the window object. All global variables belong to the window object.

Example

```
var carName = "Volvo";

// code here can use window.carName
```

Try it Yourself »

Warning

Do NOT create global variables unless you intend to.

Your global variables (or functions) can overwrite window variables (or functions).
Any function, including the window object, can overwrite your global variables and functions.

The Lifetime of JavaScript Variables

The lifetime of a JavaScript variable starts when it is declared.

Local variables are deleted when the function is completed.

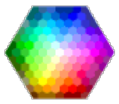
In a web browser, global variables are deleted when you close the browser window (or tab), but remain available to new pages loaded into the same window.

Function Arguments

Function arguments (parameters) work as local variables inside functions.

❮ Previous Next ❯

COLOR PICKER



HOW TO

Tabs
Dropdowns
Accordions
Convert Weights
Animated Buttons
Side Navigation
Top Navigation