JavaScript Scope

Scope is the set of variables you have access to.

JavaScript Scope

In JavaScript, objects and functions are also variables.

Local JavaScript Variables

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  
  
}

Global JavaScript Variables

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

Example

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

[Try it Yourself »](http://www.w3schools.com/js/tryit.asp?filename=tryjs_scope_global)

Automatically Global

Example

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

[Try it Yourself »](http://www.w3schools.com/js/tryit.asp?filename=tryjs_local_global)

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.

Global variables are deleted when you close the page.

Function Arguments

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

Global Variables in HTML

Example

// code here can use window.carName  
  
function myFunction() {  
    carName = "Volvo";  
}

[Try it Yourself »](http://www.w3schools.com/js/tryit.asp?filename=tryjs_scope_window)