

[< Previous](#)[Next >](#)

## JavaScript Function

JavaScript provides functions similar to most of the scripting and programming languages.

In JavaScript, a function allows you to define a block of code, give it a name and then execute it as many times as you want.

A JavaScript function can be defined using **function** keyword.

Syntax:

```
//defining a function
function <function-name>()
{
    // code to be executed
};

//calling a function
<function-name>();
```

The following example shows how to define and call a function in JavaScript.

## Example: Define and Call a Function

```
function ShowMessage() {  
    alert("Hello World!");  
}
```

```
ShowMessage();
```

Try it

In the above example, we have defined a function named ShowMessage that displays a popup message "Hello World!". This function can be execute using () operator e.g. ShowMessage().

## Function Parameters

A function can have one or more parameters, which will be supplied by the calling code and can be used inside a function. JavaScript is a dynamic type scripting language, so a function parameter can have value of any data type.

## Example: Function Parameters

```
function ShowMessage(firstName, lastName) {  
    alert("Hello " + firstName + " " + lastName);  
}
```

```
ShowMessage("Steve", "Jobs");  
ShowMessage("Bill", "Gates");  
ShowMessage(100, 200);
```

Try it

You can pass less or more arguments while calling a function. If you pass less arguments then rest of the parameters will be undefined. If you pass more arguments then additional arguments will be ignored.

### Example: Function Parameters

```
function ShowMessage(firstName, lastName) {  
    alert("Hello " + firstName + " " + lastName);  
}  
  
ShowMessage("Steve", "Jobs", "Mr."); // display Hello Steve Jobs  
ShowMessage("Bill"); // display Hello Bill undefined  
ShowMessage(); // display Hello undefined undefined
```

Try it

### The Arguments Object

All the functions in JavaScript can use arguments object by default. An arguments object includes value of each parameter.

The arguments object is an array like object. You can access its values using index similar to array. However, it does not support array methods.

### Example: Arguments Object

```
function ShowMessage(firstName, lastName) {  
    alert("Hello " + arguments[0] + " " + arguments[1]);  
}
```

```
ShowMessage("Steve", "Jobs");  
  
ShowMessage("Bill", "Gates");  
  
ShowMessage(100, 200);
```

Try it

An arguments object is still valid even if function does not include any parameters.

### Example: Arguments Object

```
function ShowMessage() {  
    alert("Hello " + arguments[0] + " " + arguments[1]);  
}  
  
ShowMessage("Steve", "Jobs"); // display Hello Steve Jobs
```

Try it

An arguments object can be iterated using for loop.

### Example: Iterate all Arguments

```
function ShowMessage() {  
  
    for(var i = 0; i < arguments.length; i++){  
        alert(arguments[i]);  
    }  
}
```

```
    }  
  }  
  
  ShowMessage("Steve", "Jobs");
```

Try it

## Return Value

A function can return zero or one value using return keyword.

### Example: Return value From a Function

```
function Sum(val1, val2) {  
    return val1 + val2;  
};  
  
var result = Sum(10,20); // returns 30  
  
function Multiply(val1, val2) {  
    console.log( val1 * val2);  
};  
  
result = Multiply(10,20); // undefined
```

Try it

In the above example, a function named Sum adds val1 & val2 and return it. So the calling code can get the return value and assign it to a variable. The second function Multiply does not return any value, so result variable will be undefined.

A function can return another function in JavaScript.

### Example: Function Returning a Function

```
function multiple(x) {  
  
    function fn(y)  
    {  
        return x * y;  
    }  
  
    return fn;  
}  
  
var triple = multiple(3);  
triple(2); // returns 6  
triple(3); // returns 9
```

Try it

### Function Expression

JavaScript allows us to assign a function to a variable and then use that variable as a function. It is called function expression.

## Example: Function Expression

```
var add = function sum(val1, val2) {  
    return val1 + val2;  
};  
  
var result1 = add(10,20);  
var result2 = sum(10,20); // not valid
```

Try it

## Anonymous Function

JavaScript allows us to define a function without any name. This unnamed function is called anonymous function. Anonymous function must be assigned to a variable.



Anonymous function is useful in passing callback function, creating closure or Immediately invoked function expression.

## Example: Anonymous Function

```
var showMessage = function (){  
    alert("Hello World!");  
};  
  
showMessage();  
  
var sayHello = function (firstName) {  
    alert("Hello " + firstName);  
};  
  
showMessage();
```

```
sayHello("Bill");
```

Try it

## Nested Functions

In JavaScript, a function can have one or more inner functions. These nested functions are in the scope of outer function. Inner function can access variables and parameters of outer function. However, outer function cannot access variables defined inside inner functions.

### Example: Nested Functions

```
function ShowMessage(firstName)
{
    function SayHello() {
        alert("Hello " + firstName);
    }

    return SayHello();
}
```

```
ShowMessage("Steve");
```

Try it



Points to Remember :

1) JavaScript a function allows you to define a block of code, give it a name and then execute it as many times as you want.



- 1) JavaScript a function allows you to define a block of code, give it a name and then execute it as many times as you want.
- 2) A function can be defined using function keyword and can be executed using () operator.
- 3) A function can include one or more parameters. It is optional to specify function parameter values while executing it.
- 4) JavaScript is a loosely-typed language. A function parameter can hold value of any data type.
- 5) You can specify less or more arguments while calling function.
- 6) All the functions can access arguments object by default instead of parameter names.
- 7) A function can return a literal value or another function.
- 8) A function can be assigned to a variable with different name.
- 9) JavaScript allows you to create anonymous functions that must be assigned to a variable.



Share



Tweet



Share



Whatsapp

[< Previous](#)[Next >](#)

TutorialsTeacher.com is optimized for learning web technologies step by step. Examples might be simplified to improve reading and basic understanding. While using this site, you agree to have read and accepted our terms of use and privacy policy.

---

✉ [feedback@tutorialsteacher.com](mailto:feedback@tutorialsteacher.com)

## TUTORIALS

- › ASP.NET Core
- › ASP.NET MVC
- › IoC
- › Web API
- › C#
- › LINQ
- › Entity Framework
- › AngularJS 1
- › Node.js
- › D3.js
- › JavaScript
- › jQuery
- › Sass
- › Https

## E-MAIL LIST

Subscribe to TutorialsTeacher email list and get latest updates, tips & tricks on C#, .Net, JavaScript, jQuery, AngularJS, Node.js to your inbox.

Email address

GO

We respect your privacy.

---

[HOME](#) [PRIVACY POLICY](#) [TERMS OF USE](#) [ADVERTISE WITH US](#)

