# **Functions in JavaScript**

A function in JavaScript is a reusable block of code that performs a specific task. You define it once, and then you can run it whenever you need that task done in your program.

Example :

// Function Definition

function welcomeMsg(name) {

return ("Hello " + name + " welcome to GeeksforGeeks");

}

let nameVal = "User";

// calling the function

console.log(welcomeMsg(nameVal));

The returned value from the welcomeMsg function will be printed as the output.

## **Function Invocation**

The function code you have written will be executed whenever it is called.

* Triggered by an event (e.g., a button click by a user).
* When explicitly called from JavaScript code.
* Automatically executed, such as in self-invoking functions.

## **Why Functions?**

* Functions can be used multiple times, reducing redundancy.
* Break down complex problems into manageable pieces.
* Manage complexity by hiding implementation details.
* Can call themselves to solve problems recursively.

## **Function Expression**

It is similar to a function declaration without the function name. [Function expressions](https://www.geeksforgeeks.org/javascript-function-expression) can be stored in a variable assignment.

**Syntax:**

let geeksforGeeks= function(paramA, paramB) {  
 // Set of statements  
}

Example :

const square = function (number) {

return number \* number;

};

const x = square(4); // x gets the value 16

console.log(x);

## **Arrow Function:**

Arrow function is one of the most used and efficient methods to create a function in JavaScript because of its comparatively easy implementation. It is a simplified as well as a more compact version of a regular or normal function expression or syntax.

**Syntax:**

let function\_name = (argument1, argument2 ,..) => expression

# **JavaScript Function Parameters**

Function parameters in JavaScript act as placeholders for values that the function can accept when it’s called.

**Syntax:**

function Name(paramet1, paramet2, paramet3,...) {  
 // Statements  
}

These are the types of parameters that can be used in JavaScript

* Defaults Parameter
* Function Rest Parameter
* Arguments Object
* Arguments Pass by Value
* Objects passed by Reference

### **1. Defaults Parameter**

Default parameters in JavaScript are utilised to set initial values for named parameters in case no value or undefined is passed when the function is called.

### **2. Function Rest Parameter**

In JavaScript, the rest parameter syntax enables a function to accept an unlimited number of arguments, which are then gathered into an array.

### **3. Arguments Object**

The arguments object is an inherent feature in JavaScript functions. It serves as a local variable in all non-arrow functions. You can analyze the arguments passed to a function using its arguments object.

### **4. Arguments Pass by Value**

In a function call, the parameters are called as arguments. The pass-by value sends the value of the variable to the function. It does not send the address of the variable. If the function changes the value of arguments then it does not affect the original value.

### **5. Objects passed by Reference**

In Pass by Reference for objects, the function receives the address of the variable rather than the value itself as the argument. If we alter the value of the variable inside the function, it affects the variables outside the function as well.

# **JavaScript Anonymous Functions**

An anonymous function is simply a function that does **not have a name**. Unlike named functions, which are declared with a name for easy reference, anonymous functions are usually created for specific tasks and are often assigned to variables or used as arguments for other functions.