

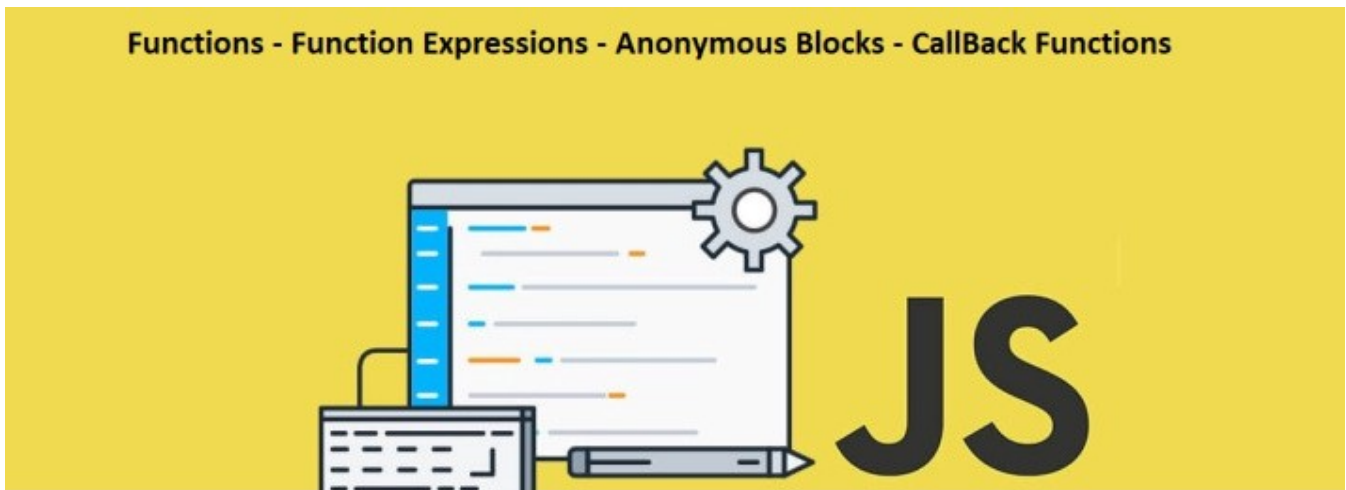
[Open in app](#)[Get started](#)

Priya Pandey · [Follow](#)

Jul 23, 2021 · 5 min read



# Function and Scope in JavaScript.



Function

## What is a function?

The function is a store of steps that we take to perform certain actions. 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 the function keyword.

A **function definition** is a **function declaration**, or **function statement** consists of the function keyword, followed by:

- The name of the function.
- A list of parameters to the function, enclosed in parentheses and separated by commas.
- The JavaScript statements that define the function, enclosed in curly brackets`{}`. and curly brackets are known as **blocks of code**.
- By writing the name of the function followed by open and close brackets is known



[Open in app](#)[Get started](#)

For example, the following code defines a simple function.

```
function ShowMessage() {  
    alert("Hello World!");  
}  
  
ShowMessage();//function Execution or function call.
```

### What is the difference between parameters and arguments in JavaScript?

When talking about the functions, the parameters and arguments of the term are often interchangeably used as if they were the same thing but there is a very subtle difference.

- *Parameters are placeholder. the variables listed as a part of the function definition. the parameter defines when we defining a function.*

#### Syntax:

```
function Name(paramet1, paramet2, paramet3,paramet4) {  
    // Statements  
}
```

#### # Parameter Rules:

There is no need to specify the data type for parameters in JavaScript function definitions.

It does not perform type checking based on the passed-in JavaScript function.

It does not check the number of received arguments.

- The value that we pass to calling a function is known as function arguments.

#### #Arguments are Passed by Value

The parameters in a function call are the function's arguments.



[Open in app](#)[Get started](#)

If a function changes an argument's value, it does not change the parameter's original value.

Changes to arguments are not visible outside the function.

#Example:-

```
//Basic function adding numbers.  
function addNumbers (a,b){  
  return a+b;  
}  
addNumbers (2,22); //arguments  
  
//
```

### What is the Difference between console.log and return?

console.log is a function that logs the arguments you pass to the web console  
**console.log** prints the output in the console.

Return is a statement that specifies the value returned from a function.

example:-

```
``` # console.log  
  
function hello(){  
  console.log('Hello');  
}  
hello();  
//This will print Hello undefined in the console.  
  
``` #return  
function addNumbers (a,b){  
  return a+b;  
}
```





function its always return a value but the function doesn't contain a return statement its will always return undefined .

## Different ways to defining a function

### \* Function Declaration

A function is starting with a function keyword is known as a function declaration. a list of **parameters** in a pair of parenthesis and a pair of curly braces that delimits the body code.

```
//Function Declaration

function add(numA , numB){
return numA + numB;
}
add(10 ,25);
```

### \* Function Expression

When we create a function and assign it to a variable, known as a function expression. A function expression has to be stored in a variable and can be accessed using *variableName*.

*some rules to follow in a function expression*

- **Function Expression** allows us to create an anonymous function that doesn't have any function name which is the main difference between Function Expression and Function Declaration.
- A function expression has to be defined first before calling it or using it as a parameter. you can't call function expression before the function definition.
- function expression in JavaScript is not hoisted, unlike function declaration.

**Syntax for Function Expression (anonymous) :**



[Open in app](#)[Get started](#)

```
addNumbers(10, 25);
```

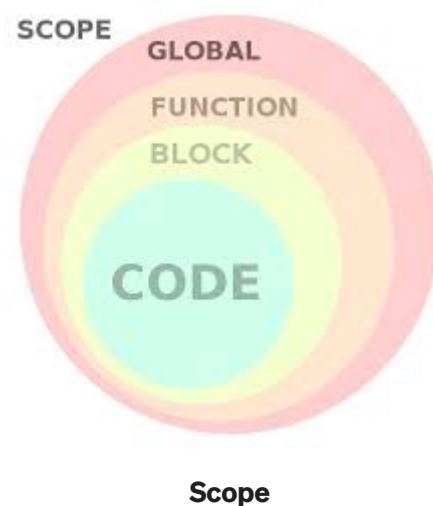
## Arrow function

arrow function is a short way of writing anonymous function expression. the way we make it short by removing a function keyword from it. and adding a arrow function to it(=>)

```
const addNumbers =(numA , numB) => numA + numB;
```

## What is Scope?

The scope is one of the most important topics in JavaScript



## The basics of scope

*The scope is how we can restrict access to any variable in a certain area of the code.*

The scope is the accessibility of variables, functions, and objects in some particular part of your code during runtime. Runtime is a time during which a computer program is executing. In other words, scope determines the visibility of variables and other resources in areas of your code.

Three types of scope:



[Open in app](#)[Get started](#)

```
{  
let username = "priya"  
}
```

**Function Scope:** function scope has also restricted access to a variable.

```
function sayHello(){  
let oddNumber = 24;  
}  
syahello();
```

**Global Scope:** any Variable that is not inside a block or function that is considered to be a Global Scope variable. This scope is created automatically. Everything that is not defined inside a local scope is automatically in a global scope. If you run your code in a web browser, the global scope will be a `window` object.

```
let user = "priya";
```

A variable declared inside the global scope is named *global* variable. Global variables are accessible from any scope.

In the previous code snippet, `user` is a global variable. This variable can be accessed from any place of the webpage's JavaScript.

## Benefits of scope

*it makes your code more secure. Imagine you have a system with different types of users. Some of these users are admins and others are users. Let's say you give them all full access to all parts of the system. What if something bad happens?*

*For example, what if someone deletes some important files, changes some records, or breaks the system? How will you find out who did it? This might be close to impossible, depending on the number of users of the system. How can you prevent this from*





Open in app

Get started

*accessibility and visibility make your code safer.*

Thank you for reading this. *If you enjoyed this article, share it with your friends and colleagues!* 😊 ✨

