

## <!-- ! Functions -->

### Definition:

A function in JavaScript is a block of code designed to perform a specific task.

It is executed when "called"

Functions help to organize code into reusable blocks, making the code more modular and easier to maintain.

## 1. Named Functions in JavaScript

### Definition:

Named functions are typically declared using the `function` keyword followed by a function name, a set of parameters, and a block of code to execute.

#### Syntax of Named Functions

- A named function is declared using the following syntax:

```
function functionName(parameters) {  
    // Function body  
}
```

## 2. Anonymous Function in JavaScript

### Definition:

An **anonymous function** is a function that does not have a name.

### Syntax:

```
function (parameters) {  
    // Function body  
}
```

### 3. Function Expression in JavaScript

#### Definition:

A **function expression** is a way to define a function in JavaScript where the function is assigned to a variable. Function expressions can be either **named** or **anonymous**.

#### Syntax:

```
let variableName = function(parameters) {  
    // Function body  
};
```

### 4. Arrow Functions in JavaScript

#### Definition:

Arrow functions, introduced in ES6 (ECMAScript 2015), provide a shorter and more concise syntax for writing functions.

#### Syntax:

```
// Single parameter and single expression  
let variableName = (parameter) => expression  
  
// Multiple parameters and multiple statements  
let variableName = (parameters) => {  
    // Function body  
}
```

```
// !    how to declare Function
```

```
function hello()  
{  
    console.log('I am hello function....')  
}
```

```
hello()
```

```
// !    take two numbers and add them inside function
```

```
let num1 = 10;  
let num2 = 20;
```

```
function add()  
{  
    console.log(num1 + num2)  
}
```

```
add()
```

```
// !    function with parameters
```

```
function add1(a,b)  
{  
    console.log(`the addition of ${a} and ${b} is ${a+b}`)  
}
```

```
add1(20,50)
```

```
// ! prime number using function

function prime(num)
{

let count = 0;
for(let i =1 ;i<=num ; i++)
{
    if(num % i == 0)
    {
        count++;
    }
}
if(count==2)
{
    console.log(`${num} is prime`)
}
else{
    console.log(`${num} is not prime`)
}

}

prime(99)

prime(97)

prime(100)


// ! function with return value

function sub(a,b)
{
    let c = a - b;
    // console.log(c)
    return c
}

let result = sub(20,10)
console.log(result)
```

```

function hi(name,msg)
{
  //   console.log(` ${name} , ${msg}`)

  let combinedMsg = name.concat(" ",msg)
  return combinedMsg
}

let res = hi("MSD","thala for a reason")
console.log(res)


// !   named function example


// !   Anonymous Fuction

let anno = function ()
{
  console.log('hello I am anonymous function')
}

anno()

console.log('-----')

let sub2 = function(a,b)
{
  //   console.log(a-b)
  return a-b;
}

console.log(sub2(20,10))

console.log('-----')


// !   Arrow Funtion


// !   how to declare arrow Function


// let hello = ()=>{
// }

```

```

// let hi = ()=>{

// }

let multiply = (a,b) =>
{
    console.log(' i am multiply arrow function')
    // console.log(a * b)
    return a*b;
}

console.log(multiply(3,6))
let mul = multiply(4,5)
console.log(mul)

console.log('-----')

// ! Arrow Function In one line

let div = (a,b)=> a/b;
let ans5 =div(4,2)
console.log(ans5)
console.log("-----")

// ! nested function and lexical scoping

let a = 20;
let outer = ()=>{

    let b = 30;

    let inner =()=>{
        var c = 50;

        console.log(a + b + c)
    }

    // console.log(c)    //? it will throw error

    inner()
}

outer()

```