

Functions

Suppose , we have wrote the code for calculating the sum of two numbers, calculating the difference of two numbers and calculating the multiplication of two numbers in single file.

When I execute the code file, all the three code will execute but What if I want only to run Addition code or subtraction code only.

I need some tool through which I able to control the different block of code.

For Example :

In Amazon , there are different functionalities implemented like Showing products, Adding/Deleting to cart, Orders, perform payments, etc.

- Every functionalities was written separately.

In Instagram, there are different functionalities like posting the image, commenting , chatting, etc.

- For each functionality, their individual code is written.
- Those code execution depends upon the button you are hitting

Thus , we will going to understand HOW TO CONTROL OUR CODE ?

Therefore to achieve that we have something known as functions.

Functions

A **JavaScript function** is a block of code designed to perform a particular task.

A **JavaScript function** is executed when "something" invokes it (calls it).

- To solve above problem , we will create three functions of names addition, subtraction and multiplication .
- After creating the function, we will put the respective code inside them.
- Now, we can control the code by calling it. It depends on us How we are calling it. Whichever function will get called, it will run.

Code 1 : Write three block of code :

1. **print length of the name**
2. **Sum of two numbers**
3. **Multiplication of two numbers**

```
var name = "Shubham";  
console.log(name, name.length);
```

```
var a = 3;  
var b = 5;  
var sum = a + b;  
console.log("Sum is ",sum);
```

```
var x = 4;  
var y = 8;  
var multiply = x*y;  
console.log(x*y);
```

If I execute the above code, All the three code will get executed but I don't want to run all the code

Code 2 : Using Functions

```
// Printing Name and length of the name  
var name = "Varun";
```

```
function sheru(){  
    var name = "Shiro";  
    console.log(name);  
    console.log("length ",name.length);  
}
```

```
// Sum of two numbers
function sum_of_two_numbers(){
    var a = 2;
    var b = 3;
    console.log("Sum ",a+b);
}

function print_numbers(){
    // printing from 1 to 10
    for(var i = 1; i<10; i++){
        console.log(i);
    }
}
```

Code 3 : Functions using return values

```
function add(a, b){
    var sum = a + b;
    return sum;
}

function square(x){
    var y = x*x;
    return y;
}

function cube(x){
    var z = x*x*x;
    return z;
}

var output1 = add(2,3);
console.log("output1 is ", output1);

var output2 = cube(output1);
console.log("output2 is ", output2);

var answer = square(output2);
console.log("answer is ", answer);
```

Code 4 : Local Scope vs Global Scope

```
var outside_child = "chintu"; // Global Variables

function sukhbeer_singh(){
  var sukhbeer_child = "rahul"; // local Variables
  console.log("My child name is ",sukhbeer_child);
}

function kalam_singh(){
  var kalam_child = "rajat"; // local Variables
  console.log("My child name is ",kalam_child);
}

function rajendra_singh(){
  var rajendra_child = "rocky"; // local Variables
  console.log("My child name is ",rajendra_child);
}

sukhbeer_singh();
console.log(outside_child);
```