Functions

Objectives

- What are functions? Real-world Examples

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 exec
uted but I don’t want to run all the code
```

Code 2 : Write three functions of above :

1. **printInfo()**

2. **sum()**
3. **multiply()**

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

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

function multiply()
{
    var x = 4;
    var y = 8;
    var multiply = x*y;
    console.log(x*y);
}

printInfo();
sum();
multiply();
```

Passing Value Using Parameters

Code 3 : Write 4 functions :

1. **subtraction**
2. **division**
3. **modulus**
4. **addition**

```
function subtraction(x, y)
{
  var subtraction = x - y;

  console.log(subtraction);
}


function division(e, c)
{
  var division = e/c;

    console.log(division);
}


function modulus(t, q)
{
  var val = t%q;

    console.log(val);
}


function addition(first,second)
{
  var third = 20;
  var sum = first+second+third;
  console.log(sum);
}


var a=2;
var b=30;

addition(a, b);
subtraction(a, b);
division(a, b);
modulus(a, b);
```

Passing Value + Returning Value

Code 4 : Write 4 functions :

1. **subtraction**
2. **division**
3. **modulus**
4. **addition**

```
function subtraction(x, y)
{
  var subtraction = x - y;

  return subtraction;
}


function division(e, c)
{
  var division = e/c;
```

```
    return division;
}


function modulus(t, q)
{
  var val = t%q;

    return val;
}


function addition(first,second)
{
  var third = 20;
  var sum = first+second+third;

  return sum;
}


var a=2;
var b=30;

var output_1 = addition(a, b);
console.log("Addition",output_1);

var output_2 = subtraction(a, b);
console.log("Subtraction",output_2);

var output_3 = division(a, b);
console.log("Division",output_3);

var output_4 = modulus(a, b);
console.log("Modulus",output_4);
```

Code 5 : Checking whether a number is even or not

```
// Checking whether a number is even or not
function checkEven(n)
{
  if(n%2 == 0)
  {
    return true;
  }
  else
  {
    return false;
  }
}


var n = 6;
var z = checkEven(n);

if(z == true)
{
  console.log(n,"is even");
}
```

Code 6 : Write 2 functions

1. sumOfN : Find sum from 1 to n
2. multiplyBy2 : multiple the value by 2

```
function sumOfN(n)
{
  sum = 0;
  for(i=1; i<=n; i++)
  {
    sum = sum + i;
  }
  console.log("Sum is ",sum);

  return sum;
}


function multiplyBy2(x)
{
  var output2 = x*2;
  console.log(output2);
}


var n = 5;
var output = sumOfN(n);
multiplyBy2(output);
```

Code 7 : Write 2 functions

1. **findSum : Find Sum of two numbers**
2. **findSquare : Square of a number**

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

function findsquare(x)
```

```
{
  var val;
  val = x*x;
  console.log("val is",val);
}

var a = 4;
var b = 5;
var z = findSum(a,b);
console.log("z is ",z);
findsquare(z);
```

Use of Return

- The output of one function can be used as input for other.
- The result of sum is acting as input for square function

Code 8

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

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

var a = 5;
var b=15;
var sum = add(a,b);
var sqr = square(sum);
console.log(sqr);
```

There are several ways to define a function in JavaScript:

- **Function Declaration:**

  ```
  function greet() {
    console.log("Hello, World!");
  }
  ```

- **Function Expression:**
  - Anonymous function expression:

    ```
    const greet = function() {
      console.log("Hello, World!");
    };
    ```

  - Named function expression:

    ```
    const greet = function greetFunction() {
      console.log("Hello, World!");
    };
    ```

- **Arrow Functions (ES6+):**

  ```
  const greet = () => {
    console.log("Hello, World!");
  };
  ```

Calling a Function

```
greet(); // Calls the function, logging "Hello, World!" to the c
```

IIFE

An IIFE (Immediately Invoked Function Expression) is a design pattern in JavaScript used to execute functions as soon as they are defined.

The IIFE pattern is particularly useful for creating a private scope for variables or functions, helping to avoid polluting the global namespace and providing a mechanism for organizing code without exposing the internals to the global scope.

```
(function() {
    // Code goes here
})();
```

```
(function(a, b) {
    console.log(a + b); // Outputs: 3
})(1, 2);
```

Local Scope vs Global Scope

There are two kind of variables i.e local variable and Global Variable

- local variable has a scope only to its function
- Global Variables can be accessed by any one.

According to the below code, the orphan child **treated as Global Variable** can be accessed by any of the function , those variables which defined inside functions can only be accessed only from the function

Code 9

```
function kishorilal_Family()
{
  var kishori_son = "chunnu";
  console.log(kishori_son);
  orphan_child = "laalu";
}


function badrilal_Family()
{
  var badrilal_son = "hari om";
}


function rajeshram_Family()
{
  var rajeshram_son = "munnu";
}


var orphan_child = "billu";
kishorilal_Family();
```

```
console.log(orphan_child);
```

\Inbuilt Function

- What is Documentation ?
- Why we need Documentation ?
- How it is helpful ?
- How to use it ?
- Introduce MDN docs

https://replit.com/@varunbhatt1/Inbuiltfunctions#index.js

IW Session Problems

Problem 1

```
//  Create a function to check if a number is Prime or Not


function isPrime(num)
{
    var count = 0;
    for(var i = 1; i<=num; i++)
    {
      if(num%i == 0)
      {
        count++;
      }
    }
    if(count==2)
    {
      return true;
    }
    else
    {
      return false;
    }

  }


  var result = isPrime(5);
  console.log(result);
```

Problem 2

```
// Use the above function to print the Primes from 2 to a given


function isPrime(num)
{
    var count = 0;
    for(var i = 1; i<=num; i++)
    {
      if(num%i == 0)
      {
        count++;
      }
    }
    if(count==2)
    {
      return true;
    }
    else
    {
      return false;
    }

  }



var limit = 100;

for(var i = 2; i<=limit; i++)
{
  var result = isPrime(i);
  if(result == true)
  {
    console.log(i,"is prime");
  }
}
```

Problem 3

```
// Problem 3: Use the same function to print Non-Primes from 2 
```

```
function isPrime(num)
{
    var count = 0;
    for(var i = 1; i<=num; i++)
    {
      if(num%i == 0)
      {
        count++;
      }
    }


    if(count==2)
    {
      return true;
    }
    else
    {
      return false;
    }

  }



var limit = 100;

for(var i = 2; i<=limit; i++)
{
  var result = isPrime(i);
  if(result != true)
  {
    console.log(i,"is non prime");
  }
}
```

Problem 4

```
// Problem 4: Write a function to check if the char is a small c

function isSmallCase(x)
{
  var lower = "abcdefghijklmnopqrstuvwxyz";

  for(var i=0; i<lower.length; i++)
  {
    if(x == lower[i])
    {
      return true;
    }

  }

  return false;
}


var result = isSmallCase("C");
console.log(result);
```

Problem 5

```
// Write a function to replace spaces in a given string with -
```

```javascript
function modifyString(str)
{
  var output = "";

  for(var i = 0; i<str.length; i++)
  {
    if(str[i] == " ")
    {
      output = output+"-";
    }
    else
    {
      output = str;
    }
  }

  return output;
}


var str = "Masai School";
console.log(modifyString(str));
```