JavaScript

Introduction to JavaScript

- JavaScript is the world's most popular programming language.
- It is a lightweight, cross-platform, and interpreted scripting language.
- JavaScript programming is used for Web development.
- JavaScript can be used for Client-side developments as well as Server-side developments.
- Useful libraries for the client-side are AngularJS, ReactJS, VueJS and so many others.
- The useful framework which is the most famous these days is node.js.
- Brendan Eich, a Netscape Communications Corporation programmer, created JavaScript in September 1995.
- When Eich created JavaScript in 1995, he created it for Netscape Navigator and it was originally going to be named LiveScript but was renamed.

Download IDE for working with JavaScript

In this course i am using Visual Studio Code IDE for working in JavaScript. So you can download form this link - https://code.visualstudio.com/download

Different ways to print in JavaScript

Using document.write()
Using innerHTML
Using alert()
Using console.log()
Using console.warn()
Using console.error()

Example - document.write()

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>Using document.write() function</title>
</head>
<body>
   <script>
   document.write(2+3);
   </script>
</body>
</html>
```

Example - innerHTML

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Using innerHTML</title>
</head>
<body>
   <h1 id="demo"></h1>
<script>
 document.getElementById("demo").innerHTML = 5 + 2;
 </script>
</body>
</html>
```

Example - alert()

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Using alert() function</title>
</head>
<body>
 <script>
```

```
    alert("Welcome to JavaScript");
    </script>
</body>
</html>
```

Example - console.log(), consol.warn(), consol.error()

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Using alert() function</title>
</head>
<body>
    <script>
    // console.log(2+8);
    // console.warn("You can write any Warning message here")
 // console.error("You can write any arror message here")
    </script>
</body>
</html>
```

Comments in JavaScript

If you want to explain any line in your code then you can use comments.

Two ways to use comments

1. Single Line Comments

```
//write what you want to explain.
```

2. Multi-line Comments

```
/* --------------
----------------
--------------*/
```

Variables in JavaScript

Variables are containers (you can say like a box) for storing data.

4 Ways to Declare a JavaScript Variable:

- Using var
- Using let
- Using const
- Using nothing

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Using alert() function</title>
</head>
<body>
 <script>
 //1. Using var keyword for creating variables
 /*
 var a = 2;
 var b = 3;
 var a = 5;
 var c = a + b;
 console.log(c);
 */

 // 2. Using let keyword for creating variables
 // error will show like - Identifier 'a' has already
 /*
 let a = 2;
 let b = 3;
 let a = 5;
 let c = a + b;
 console.log(c);
 */
```

```
// 3. using nothing
/*
a = 2;
b = 3;
a = 5;
c = a + b;
console.log(c);
*/

// Using const keyword
// (A const variable cannot be reassigned and Redeclared)
/*
const pi = 3.14;
// pi = pi + 5; (This is wrong)
let r = 4;
let cr = pi*r*r;
console.log(cr);
*/
</script>
</body>
</html>
```

Data Types in JavaScript

Numbers, Strings, Booleans, Array, Objects & More.

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Using alert() function</title>
</head>
<body>
    <script>
   // Numbers datatype
   let a = 3.5
   let name = "Abhishek"
```

```
  console.log(a);
  console.log(name);

  // Booleans datatype
  let x = true
  let y = false

  console.log(x);
  console.log(y);

// Array datatype

let student_name = [”Raj”, ”Kamal”, ”Rohan”, ”Sonu”];

console.log(student_name[2]);

// Object datatype

let person = {
firstName : ”Raj”,
lastName : ”Singh”,
age : 40,
mobile : ”1001001001”
};

console.log(”Person name is ” + person.firstName + ” & ” + ”Mobile no. is ” +
person.mobile);

</script>
</body>
</html>
```

Operators in JavaScript

Arithmetic Operators (+, -, *, /, %, **, ++, --)

Assignment Operators (=) or +=, -=, *=, /=

Comparison Operators (>, <, >=, <=, ==, !=, ?)

Logical Operators (&&, ||, !)

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Operators</title>
</head>
<body>
    <script>
        //Arithmetic Operators (+, -, *, /, %, **, ++, --)
        /*
        let a = 8;
        let b = 2;
        let c = a + b;
        console.log(c);
        */
//Assignment Operators (=) or +=, -=, *=, /=
 /*
 let a = 3;
 let b = a;
 console.log(b);
 b += 2;
 console.log(b);
 */

 //Comparison Operators (>, <, >=, <=, ==, !=)
 /*
 let a = 5;
 let b = 6;
 let c = a == b;
 console.log(c);
 */

 //Logical Operators (&&, ||, !)
 /*
 let a = 6;
```

```
let b = 4;
 let c = 3;

 let d = a>b && a>c;
 console.log(d);
 */
 </script>
</body>
</html>
```

Conditional Statements

If you want to perform different actions based on different conditions then you can use conditional statements.

if, else, else if and switch statements

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Conditional Statements</title>
</head>
<body>
    <script>
        let age = 41;
        // simple if
        // if(age>=18 && age<=40){
        //     console.log("You can fill this form.");
        // }
 // else statement
 // if(age>=18 && age<=40){
 // console.log("You can fill this form.");
 // } else {
 // console.log("You can't fill this form.");
 // }
```

```
//else if Statement
      /*
      let totalmarks=100;
      let percentage;

      percentage = totalmarks/500*100;
      if(percentage>=90){
         console.log("A+");
         }
      else if(percentage>=80){
         console.log("A");
         }
      else if(percentage>=70){
         console.log("B+");
         }
      else if(percentage>=50){
         console.log("B");
         }
else if(percentage>=35){
 console.log("C");
 }
 else {
 console.log("Fail");
 }
 */

 </script>
</body>
</html>
```

Switch Statement

```
switch(expression) {
  case 1:
    // code block
    break;
  case 2:
    // code block
    break;
  default:
    // code block
}
```

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Switch Statements</title>
</head>
<body>
    <script>
        /*
        let choice = 3;
        switch(choice){
            case 1: console.log("You selected one number.");
            break;
            case 2: console.log("You selected two number.");
            break;
            default : console.log("Wrong Choice");
        }
        */
/*
 let choice = 1;
 let a = 2;
 let b = 3;

 switch(choice){
 case 1 : console.log(a+b);
 break;
 case 2 : console.log(a*b);
 break;
 case 3 : console.log(a-b);
 break;
 case 4 : console.log(a/b);
 break;
 default : console.log("wrong choice");
 }
 */
 </script>
</body>
</html>
```

Loops in JavaScript

If you want to do any work again and again in programming then you can do by using loops.

There are three loops in JavaScript-

For, While and Do while

For Loop Example

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Looping-for, while</title>
</head>
<body>
    <script>
        //for loop example 1
        /*let i;
        for(i=0;i<=10;i++){
            console.log(i);
        }*/

 //for loop example 2
 /*let no=2;
 let b,i;
 for(i=1;i<=10;i++){
 b=no*i;
 console.log(no+"*"+i+"="+b);
 }*/
 </script>
</body>
</html>
```

While and Do while Loop Example

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>While & Do while loop</title>
</head>
<body>
<script>
//while loop
/* let i = 1;
 while(i<=10){
 console.log(i);
 i++;
 } */

 //do while loop
 /* let i=1;
 do{
 console.log(i);
 }while(i>=10);
 */
 </script>
</body>
</html>
```

Functions in JavaScript

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

Built-in functions - log(), write(), alert(), pow(), date()

User defined functions - These are created by us.

Example of both type of function

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Functions</title>
</head>
<body>
    <script>
        //Built-in functions
        /*let a=Date();
        let b = Math.pow(2, 3);
        console.log(a);
        console.log(b);
        */

        //User defined functions
        //Creating function
        function demo(){
            //console.log("I am function");
            //alert("I am function");
            //document.write("I am function");
        }
    </script>
 <form>
 <input type = "button" onclick = "demo()" value = "Demo Function"> </input>
 </form>
</body>
</html>
```

Why we create functions?

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Function Example</title>
</head>
<body>
    <script>
        function printline(){
            let i;
            for(i=1;i<=40;i++){
                document.write("-");
            }
        }
document.write("Welcome to JavaScript");
document.write("<br>");
printline();
document.write("<br>");
document.write("Welcome to JavaScript");
document.write("<br>");
document.write("Welcome to JavaScript");
document.write("<br>");
printline();
</script>

</body>
</html>
```

Function with parameters

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>function with parameters</title>
</head>
<body>
 <script>
 function sum(a,b){
 console.log(a+b);
 }
 </script>
 <form>
 <input type = "button" onclick = "sum(2,5)" value = "Click">
 </form>
</body>
</html>
```

Basic Arithmetic Calculator Project Example 1

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Basic Calculator Project 1</title>
</head>
<body>
    <script>
        function addNumbers()
            {
                var num1 = parseInt(document.getElementById("value1").value);
                var num2 = parseInt(document.getElementById("value2").value);
                var ans = document.getElementById("answer");
                ans.value = num1 + num2;
            }
    </script>
```

```
<form>
   Enter Value_1 = <input type="text" id="value1" name="value1"/><br><br>
   Enter Value_2 = <input type="text" id="value2" name="value2"/><br><br>
   <input type="button" name="Sumbit" value="Click" onclick="addNumbers()"/>
   <input id="btnReset" type="reset"/></td><br><br>
   Answer = <input type="text" id="answer" name="answer"/>
   </form>
</body>
</html>
```

Example 2

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Basic Calculator Project 2</title>
</head>
<body>
 <script>
 var a,b,c;
 function calAdd(firstnumber,secondnumber){
 a=parseInt(firstnumber);
 b=parseInt(secondnumber);
 c=a+b;
 alert("Result="+c);
 }

 function calSub(firstnumber,secondnumber){
 a=parseInt(firstnumber);
 b=parseInt(secondnumber);
 c=a-b;
 alert("Result="+c);
 }
 function calMul(firstnumber,secondnumber){
 a=parseInt(firstnumber);
 b=parseInt(secondnumber);
 c=a*b;
 alert("Result="+c);
 }
```

```
function calDiv(firstnumber,secondnumber){
 a=parseInt(firstnumber);
 b=parseInt(secondnumber);
 c=a/b;
 alert("Result="+c);
 }
</script>
 <center>
 <h2>Basic Calculator</h2>
 <form>
 <table border="1">
 <tr>
 <td>First Number:</td>
 <td><input id="txtFirstnumber" type="text"/></td>
 </td></tr>
 <tr>
 <td>Second Number:</td>
 <td><input id="txtSecondnumber" type="text"/></td>
 </tr>
 <tr>
 <td><input type="button" value="Add"
onClick="calAdd(txtFirstnumber.value,txtSecondnumber.value);"/> 
 <input type="button" value="SUB"
onClick="calSub(txtFirstnumber.value,txtSecondnumber.value);"/></td>
 <td><input type="button" value="MUL"
onClick="calMul(txtFirstnumber.value,txtSecondnumber.value);"/> 
 <input type="button" value="DIV""
onClick="calDiv(txtFirstnumber.value,txtSecondnumber.value);"/>
<input id="btnReset" type="reset"/></td>
 </tr>
 </table>
 </form>
</center>
</body>
</html>
```

JavaScript Events

Here is a list of some common HTML events:

Event	Description
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user presses a keyboard key
onload	The browser has finished loading the page

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>JavaScript Events</title>
</head>
<body onload="test4()">
 <script>
 function test1(){
 alert("Button1 tested!");
 }

 function test2(){
 alert("Button2 tested!");
 }

 function test3(){
 alert("You pressed a key inside the input field");
 }

 /* function test4(){
 alert("Welcome to JavaScript");
 } */
 </script>
```

```
<h1 id="demo"></h1>
<button onclick="document.getElementById('demo').innerHTML=Date()">View Current Time</button>
<br><br>
<button onclick="test1()">Test Button1</button>
<br><br>
<button onmouseover="test2()">Test Button2</button>
<br><br>
<input type="text" onkeydown="test3()"></input>
</body>
</html>
```

Strings in JavaScript

A string is a sequence of one or more characters that may consist of letters, numbers, or symbols.

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Strings and It's Methods</title>
</head>
<body>
 <script>
 let mystring = "Welcome to JavaScript";
 console.log(mystring);

 // string methods
 // console.log(mystring.length);
 // console.log(mystring.indexOf("to"));
 // console.log(mystring.search("J"));
 // console.log(mystring.slice(10,21));
// The replace() method replaces only the first match
// console.log(mystring.replace("JavaScript","Programming"));
// toUpperCase()
// toLowerCase()
```

```
 // joins two or more strings
/*let text1 = "Hello";
let text2 = "World!";
let text3 = text1.concat(" ",text2);

 console.log(text3);*/

 // console.log(mystring.charAt(1));

 //Template Literals
 /*let price = 10;
 let tax = 0.30;

 let total = `Total bill is: ${price + tax}`;
 console.log(total);*/
 </script>
</body>
</html>
```

Arrays in JavaScript

If you want to store more than one value at a time in a single variable then you can use array variable.

Syntax:

```
let array_name = [item1, item2, ...n];
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Array and Its Methods</title>
</head>
<body>
 <h1 id="demo"></h1>
 <h1 id="demo1"></h1>
 <h1 id="demo2"></h1>
```

```
<script>
// Creating Array Method 1
// let programming = ["Java","C","C++","Python"];
// console.log(programming);

// Creating Array Method 2 (by using new keyword)
// const students = new Array("Sunil", "Raj", "Mark");
// console.log(students);
// You can also access full array by using ID.
// let softwares = ["VLC", "Notepad", "VSCode"];
 // document.getElementById("demo").innerHTML = softwares;

 // Accessing Array Elements
 // let student = students[1];
 // console.log(student);

 // Changing an Array Element
 // students[1] = "Komal";
 // console.log(students);

 // Arrays are special kinds of objects.
 // Because of this, you can store different types of values in the same Array.
 // const employee = {firstName:"Suresh", lastName:"Singh", age:45,
salary:55000};
 // document.getElementById("demo").innerHTML = employee.firstName;

// Length Property
 // let programming = ["Java","C","C++","Python"];
 // let length = programming.length;
 // console.log(length);
// Access array elements by using loop
 // let students = ["Sunil", "Raj", "Mark"];
 // let len = students.length;

 // for (let i = 0; i < len; i++) {
 // console.log(students[i]);
 // }
```

```
// ********** Array Methods **********

// 1. Converting Arrays to Strings
// let students = ["Sunil", "Raj", "Mark"];
// let myString = students.toString();

// 2. Converting Arrays to Strings (Join())
// let students = ["Sunil", "Raj", "Mark"];
// let myString = students.join("_");
// console.log(myString);

// 3. pop() Method - removes the last element from an array
// let softwares = ["VLC", "Notepad", "VSCode"];
// document.getElementById("demo1").innerHTML = softwares;
// softwares.pop();
// document.getElementById("demo2").innerHTML = softwares;

// 4. push() Method - adds a new element to an array (but at the end)
 // let softwares = ["VLC", "Notepad", "VSCode"];
 // document.getElementById("demo1").innerHTML = softwares;
 // softwares.push("Photoshop");
 // document.getElementById("demo2").innerHTML = softwares;

 // 5. shift() Method - removes the first array element and
 // "shifts" all other elements to a lower index.

 // let softwares = ["VLC", "Notepad", "VSCode"];
 // document.getElementById("demo1").innerHTML = softwares;
 // softwares.shift();
 // document.getElementById("demo2").innerHTML = softwares;

// 6. unshift() Method - adds a new element to an array at the beginning

 // let softwares = ["VLC", "Notepad", "VSCode"];
 // document.getElementById("demo1").innerHTML = softwares;

// softwares.unshift("Photoshop","Tally");
// document.getElementById("demo2").innerHTML = softwares;
```

```
// 7. concat() Method - creates a new array by concatenating existing arrays

// let students = ["Sunil", "Raj", "Mark"];
// let marks = ["300", "400", "350"];
// let data = marks.concat(students);
// console.log(data);
</script>
</body>
</html>
```

JavaScript Date Objects

Date objects are created with the new Date() constructor.

Example

```
<script>
        let a = new Date();
        console.log(a);
    </script>
```

Date Methods

you can use get and set the year, month, day, hours, minutes, seconds, and milliseconds of date object, using either local time or UTC (universal, or GMT) time.

Digital Clock Project

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Digital Clock</title>
</head>
```

```
<body>
   <style>
      #clock {
       background-color: lightgrey;
       width: 300px;
       border: 5px solid rgb(0, 83, 0);
       padding: 15px;
       margin: 15px;
       font-size: 50px;
       font-family: Arial, Helvetica, sans-serif;
    }
   </style>

<center><div id = "clock" onload="currentTime()"></div></center>

 <script>
 function currentTime() {
 let date = new Date();
 let hour = date.getHours();
 let min = date.getMinutes();
 let sec = date.getSeconds();
 let session = "AM";

 if(hour > 12){
 session = "PM";
 }

 hour = (hour < 10) ? "0" + hour : hour;
 min = (min < 10) ? "0" + min : min;
 sec = (sec < 10) ? "0" + sec : sec ;

 let time = hour + ":" + min + ":" + sec + " " + session;

 document.getElementById("clock").innerText = time;
 let t = setTimeout(function(){ currentTime() }, 1000);

}

 currentTime();
```

```
</script>
</body>
</html>
```

The HTML DOM (Document Object Model)

When you open your web page in a browser, the browser creates a document object model of the page.

With the object model, JavaScript gets all the power it needs to create dynamic HTML.

- You can access HTML elements.
- You can show your HTML dynamically.
- By using JavaScript you can change all the HTML elements & HTML attributes in the page. You can also change CSS styles in the page.
- By using JavaScript you can remove existing HTML elements and attributes.
- By using JavaScript you can add new HTML elements and attributes.
- By using JavaScript you can react to all existing HTML events in the page.

Now I clear the all above mentioned points through practical.

Open this HTML file (DOM) in browser

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>HTML DOM</title>
</head>
<body>
    <h1 id="demo"></h1>

    <script>
        document.getElementById("demo").innerHTML = "Welcome to JavaScript";
    </script>

</body>
</html>
```

Example : Change HTML dynamically

Type this command in console

document.getElementById('demo').style.border = '3px solid blue'

document.getElementById('demo').style.background = 'blue'

Example : Main HTML FILE

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>HTML DOM</title>
</head>
<style>
    /*
.c1 {
    background: blueviolet;
    width: 300px;
    border: 5px solid green;
    padding: 5px;
    margin: 5px;
  }
.bgchange {
 background: red;
}*/
</style>
<body>
 <h1 id="demo"></h1>
 <!--
 <div class="c1">
 <h1>This is heading.</h1>
 </div>
 <div class="c1">
 <p>This is paragraph</p>
 </div>
 -->
```

```
<script src="js/DOM_Methods.js"></script>
<script>
   document.getElementById("demo").innerHTML = "Welcome to JavaScript";
</script>

</body>
</html>
```

JavaScript File Which Name Is (DOM_Methods.js)

```
// Example : JavaScript can change all the HTML elements
// & HTML attributes in the page.
// let eleId = document.getElementById("demo");
// eleId.innerHTML = "Welcome to DOM";
// eleId.style.background='yellow';

// Example : You can also change CSS styles in the page.
// let eleclass = document.getElementsByClassName("c1");
// eleClass[0].style.background = "yellow";
// eleClass[0].style.border = "10px solid black";

// Example : JavaScript can add new HTML elements and attributes.
// let eleClass = document.getElementsByClassName("c1");
// eleClass[0].classList.add("bgchange");

// Example : JavaScript can remove existing HTML elements and attributes.
// eleClass[0].classList.remove("bgchange");

// let tg = document.getElementsByTagName("div");
// console.log(tg);

// makeElement = document.createElement('h1');
// makeElement.innerText = "This is created heading";
// tg[0].appendChild(makeElement);
```

JavaScript Regular Expressions (RegExp)

In JavaScript, a Regular Expression (RegEx) is an object that describes a sequence of characters used for defining a search pattern.
RegExp can be used for text search and text replace operations.

Syntax : /pattern/modifiers;

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Regular Expression</title>
</head>
<body>
    <h1 id = "demo"></h1>
    <script>
        // Pattern Matching
        // let pattern = /[a-z]/i;
        // let pass = "12345";
        // console.log(pattern.exec(pass));

        // let pattern = /[a-z][0-9]/g;
        // let pass = "a12345";
        // //console.log(pattern.exec(pass));
        // console.log(pattern.test(pass));

        // String search() With a Regular Expression
        // let text = "Welcome to Microsolution!";
        // let t = text.search(/microsolution/i);
        // document.getElementById("demo").innerHTML = t;
       // String replace() With a Regular Expression
       // let text = "Welcome to Microsolution!";
       // let t = text.replace(/Microsolution!/i,"JavaScript!");
       // document.getElementById("demo").innerHTML = t;
```

```
        // javascript regex password pattern
        // let pattern = /^(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*[!@#\$%\^&\*])(?=.
{8,})/g;
        // let pass = "Test@12345";
        // let t = pattern.test(pass);
        // if(t==true){
        //     console.log("Strong Password");
        // } else {
        //     console.log("Weak Password");
        // }
</script>
</body>
</html>
```

Validation in JavaScript

Validation is a method to authenticate the user. JavaScript provides the facility to validate the form on the client-side so data processing will be faster than server-side validation.

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>JavaScript Validation</title>
</head>
<script>
    function validate_form(){
    // can't be empty validation
    let name = document.myform.name.value;

    if (name==null || name==""){
      alert("Name can't be blank");
      return false;
    }
```

```
    // email validation
    let x=document.myform.email.value;
    let atposition=x.indexOf("@");
    let dotposition=x.lastIndexOf(".");

    if (atposition<1 || dotposition<atposition+2 || dotposition+2>=x.length){
    alert("Please enter a valid e-mail address");
    return false;
    }

    // Retype Password Validation
    let firstpassword = document.myform.password1.value;
    let secondpassword = document.myform.password2.value;

    if(firstpassword == secondpassword){

    // javascript regex password pattern
    // strong password validation
    let pattern = /^(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*[!@#\$%\^&\*])(?=.{8,})/g;
    let pass = document.myform.password1.value;
    let t = pattern.test(pass);
    if(t==true){
        return true;
    } else {
      alert("Weak Password");
      return false;
    }
    }
    else{
    alert("password must be same!");
    return false;
    }
}
</script>
<body>
    <form name="myform" action="https://www.youtube.com/playlist?
list=PLnSDvcENZlwDOiSN10wHqcZUF4jLoxUp-" method="post" onsubmit="return
validate_form()" >
```

```
Name: <input type="text" name="name"><br><br>
   Email: <input type="text" name="email"><br><br>
   Password: <input type="password" name="password1" /><br><br>
   Confirm Password: <input type="password" name="password2"/><br><br>
   <input type="submit" value="SUBMIT">
   </form>
</body>
</html>
```

Exception Handling in JavaScript

What is Exception?
Exception is an abnormal condition in a program.

Exception Handling
In programming, exception handling is a process for handling the abnormal statements in the code.

There are following statements that handle if any exception occurs:

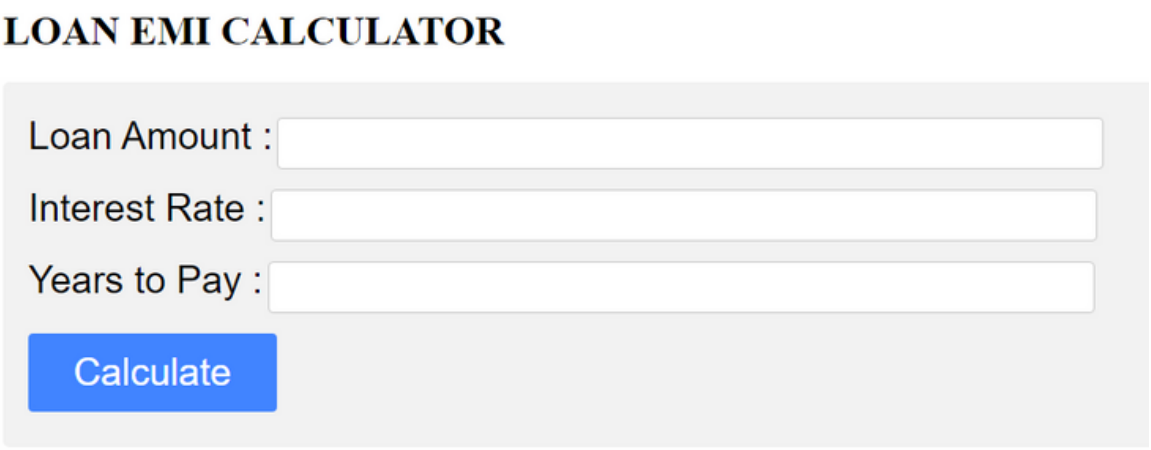
try...catch statements
throw statements
try...catch...finally statements.

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Exception Handling</title>
</head>
```

```
<body>
  <script>
    // try and catch example
    // try {
    //     alertt("Welcome!");
    //  }
    // catch(err) {
    //     console.log("Please check your syntax!");
    // }

    // throw example
    // let saving = 5000;
    // let withdrawal = 6000;
    // try{
    //     if(saving>withdrawal){
    //         saving -= withdrawal;
    //         console.log("your current balance is - "+saving);
    //     }
    //     else {
    //         throw "!not enough funds";
    //     }
    // }
    // catch(err){
    //     console.log(err);
    // }

    // finally example
    // try {
    //     alartt("Welcome!");
    //  }
    // catch(err) {
    //     let a = err.message
    //     console.log(a);
    // }
    // finally{
    //     console.log("test");
    // }
  </script>
</body>
</html>
```

Project (Loan EMI Calculator)



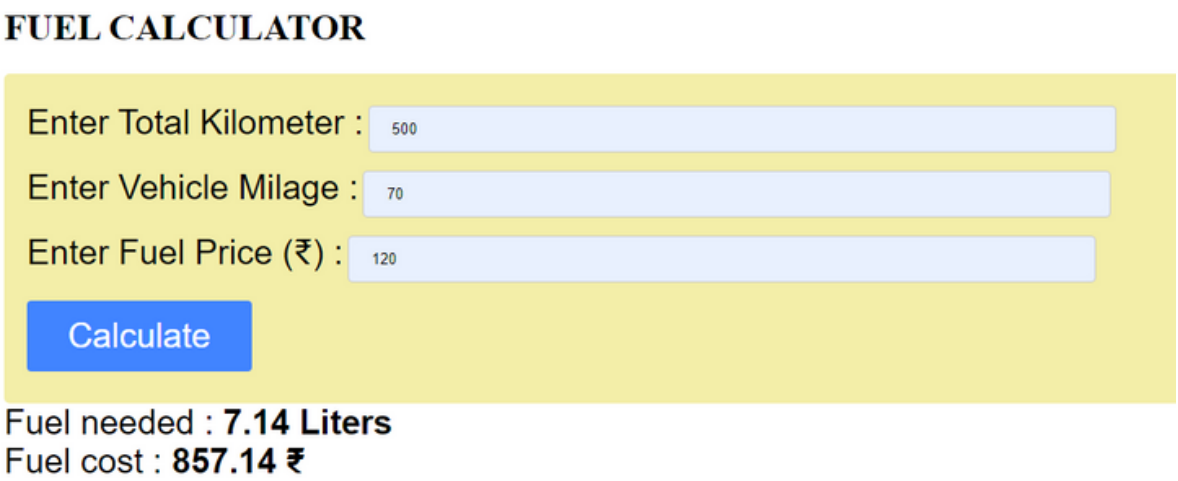
Code :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title> Loan EMI Calculator</title>
</head>
<style>
    input[type=text], select {
      width: 50%;
      padding: 12px 20px;
      margin: 8px 0;
      display: inline-block;
      border: 1px solid #ccc;
      border-radius: 4px;
      box-sizing: border-box;
    }

    label {
    font-size: 30px;
    font-family: Arial, Helvetica, sans-serif;
    }
```

```
input[type=submit] {
     width: 15%;
     background-color: #4184ff;
     color: white;
     padding: 14px 20px;
     margin: 8px 0;
     border: none;
     border-radius: 4px;
     cursor: pointer;
     font-size: 30px;
     font-family: Arial, Helvetica, sans-serif;
    }

    input[type=submit]:hover {
      background-color: #45a072;
    }

    div {
      border-radius: 5px;
      background-color: #f2f2f2;
      padding: 20px;
    }
    </style>
<body>
     <h1>LOAN EMI CALCULATOR</h1>

     <div>
       <form onsubmit="return false">
         <label for="LoanAmount">Loan Amount :</label>
         <input type="text" id="amount">
         <br>
         <label for="IntersetRate">Interest Rate :</label>
         <input type="text" id="rate">
         <br>
         <label for="Years">Years to Pay :</label>
         <input type="text" id="years">
         <br>
         <input type="submit" value="Calculate"
onClick="calculate(amount.value,rate.value,years.value);"/>
       </form>
```

```
</div>
    <label for="output" id="demo"> </label>
  <script>
      function calculate(amount,rate,years){
       var p=parseInt(amount);
       var r=parseFloat(rate);
       var n=parseInt(years);

       var monthly_interest = r/12/100;

       var months = n * 12;

       // P x R x (1+R)^N / [(1+R)^N-1] This is Loan EMI Formula

       var x = Math.pow(1 + monthly_interest, months);
       var monthly = (p * x * monthly_interest) / (x - 1);

       var emi = monthly.toFixed(2);
       var totalPayment = (emi * months).toFixed(2);
       var totalInterest = (totalPayment - p).toFixed(2);

       document.getElementById("demo").innerHTML="Loan EMI : " + "<b>" + emi
+ "</b>" + "<br>" + "Total Interest Payable : " + "<b>" + totalInterest + "</b>" + "
<br>" + "Total Payment : " + "<b>" + totalPayment + "</b>";
       }
  </script>
</body>
</html>
```

Project (Fuel Calculator)



Code :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Fuel Calculator</title>
</head>
<style>
    input[type=text], select {
    width: 50%;
    padding: 12px 20px;
    margin: 8px 0;
    display: inline-block;
    border: 1px solid #ccc;
    border-radius: 4px;
    box-sizing: border-box;
  }

  label {
  font-size: 30px;
  font-family: Arial, Helvetica, sans-serif;
  }
```

```
 input[type=submit] {
   width: 15%;
   background-color: #4184ff;
   color: white;
   padding: 14px 20px;
   margin: 8px 0;
   border: none;
   border-radius: 4px;
   cursor: pointer;
   font-size: 30px;
   font-family: Arial, Helvetica, sans-serif;
  }

  input[type=submit]:hover {
   background-color: #45a072;
  }

  div {
   border-radius: 5px;
   background-color: #f3eea8;
   padding: 20px;
  }
  </style>
<body>
   <h1>FUEL CALCULATOR</h1>

    <div>
     <form onsubmit="return false">
       <label for="km">Enter Total Kilometer :</label>
       <input type="text" id="totalkm">
       <br>
       <label for="milage">Enter Vehicle Milage :</label>
       <input type="text" id="milage">
       <br>
       <label for="fuelprice">Enter Fuel Price (₹) :</label>
       <input type="text" id="fuel_price">
       <br>
       <input type="submit" value="Calculate"
onClick="calculate(totalkm.value,milage.value,fuel_price.value);"/>
       <br>
```

```
</form>
    </div>
    <label for="output" id="demo"> </label>
   <script>
      function calculate(a,b,c){
      let totalkm=parseInt(a);
      let milage=parseInt(b);
      let fuel_price=parseFloat(c);

      let x = ( ( totalkm / milage ) * fuel_price);
      let y = totalkm/milage;
      let Total_Cost = x.toFixed(2);
      let total_fuel = y.toFixed(2);
      document.getElementById("demo").innerHTML="Fuel needed : " + "<b>" +
total_fuel + " Liters" + "</b>" + "<br>" + "Fuel cost : " + "<b>" + Total_Cost + " ₹" +
"</b>";
      }
   </script>
</body>
</html>
```

The End

All Programming Languages Video Link -

https://youtube.com/playlist?list=PLnSDvcENZlwAcFgFLD5bzt5Zh428yzQXT