

## JavaScript

JavaScript is high level, interpreted programming language that allows you to add interactivity and dynamic behavior to the web pages.

It is primarily used for client-side web development, enabling you to manipulate and control the content of a webpage directly within the user's browser.

JavaScript can be directly embedded within HTML document using `<script>` tag

```
<html>
  <head>
    <title> JavaScript Example </title>
  </head>
  <body>
    <h1 id="greet"> Hello, World! </h1>
    <script>
      var greetingE = document.getElementById('greet');
      greetingE.innerHTML = 'Hello, JavaScript!';
    </script>
  </body>
</html>
```

## Features

1. Light weight and interpreted language
2. Supports OOPs
3. Used for client-side scripting in web browsers
4. Cross platform compatibility
5. Provides built-in functions & APIs

## JavaScript Identifiers

Identifiers are used to name variables, functions, objects or other elements of within the code. It should be a unique name. There are certain rules and restrictions that must be followed when we name in identifier

- 1) name must be unique
- 2) first letter of identifier should be of upper, lower case, underscore, dollar sign
- 3) should not be a reserved keyword
- 4) They are case sensitive
- 5) no space should be there b/w letters

## JavaScript Operators

- 1) Arithmetic Operators      exponentiation  
 $+ , - , * , / , ^ , +t , -t , ( + )$

## 2) Assignment Operators

=, +=, -=, \*=, /=, /=, \*\*=

## 3) Comparison Operators

==, !=, >, <, >=, <=, ?

## 4) Logical Operators

&&, ||, !

## 5) Type Operators

typeof: returns the type of a variable

instanceof: returns true if an object  
is of an instance of an object type

## 6) Bitwise Operators

&, |, ~, ^, <<, >>

## JavaScript Loops

### 1) For loop

Iterates the elements for a fixed no  
of times

for (initialization; condition; iteration)  
{  
 code  
}

### 2) While loop

Iterates the elements for the infinite no  
of times

while (condition)

{  
 code  
}

### 3) do while loop

Iterates the elements for the infinite no  
of times like while loop but code is executed  
at least once whether condition is T or F

do {

code  
} while (condition);

## JavaScript Array

JavaScript Array represents the array  
a collection of similar types of elements.  
3 ways to construct array are

1. var arr = ["Aman", "Sonu", "Monu"];

2. var arr = new Array();

arr[0] = "Aman";

arr[1] = "Sonu";

arr[2] = "Monu";

3. var arr = new Array("Aman", "Sonu", "Monu");

## array.length

PAGE No. \_\_\_\_\_  
DATE \_\_\_\_\_

## JavaScript Array Methods

concat()      forEach()  
push()      array.forEach(function  
pop()      {array});  
slice()      document.write(array);  
indexof()  
sort()

<html>

<head>

<title> Array Handling Methods </title>

</head>

<body>

<script>

var fruits = ["apple", "banana", "orange"];

// accessing array element

document.write(fruits[0] + "<br>");

// modifying array element

fruits[1] = "grape";

document.write(fruits.join(", ") + "<br>");

// Array length

document.write(fruits.length + "<br>");

// Adding elements to array

fruits.push("mango");

join() method is used to concatenate elements of an array to a single string

document.write(fruits.join(", ") + "<br>");

// Removing last element of array

var lastE = fruits.pop();  
document.write(lastE + "<br>");

// Combining arrays

var c = fruits.concat(["pear", "kiwi"]);  
document.write(c.join(", ") + "<br>");

// slicing an array

var s = fruits.slice(1, 3);  
document.write(s.join(", ") + "<br>");

// Iterating over array

fruits.forEach(function(fruit){  
document.write(fruit + "<br>");  
});

// sorting of array

var sa = fruits.sort();  
document.write(sa.join(", ") + "<br>");

// Index of an element

var i = fruits.indexOf("orange");  
document.write(i + "<br>");

</script>

</body>

</html>

## JavaScript Function

Function is a block of code that is designed to perform a specific or calculate value. It allows to reuse the code and execute whenever needed.

Functions can be parameterized or non-parameterized

## User define function

This function is developed by the developer to perform specific task based on their requirements.

```
function greet(name)
```

```
    {  
        document.write("Hello, " + name + "!");  
    }  
  
greet("John");
```

## Built-in function

These are native functions or predefined functions which are provided by the JavaScript

e.g. alert()

parseInt()

Math.random()

```
<html>
```

```
</head>
```

```
<title> Addition Text Box Program </title>
```

```
<style>
```

```
.container {
```

```
    margin-top: 50px;
```

```
    text-align: center;
```

```
}
```

```
input {type="text"} {
```

```
width: 200px;
```

```
padding: 5px;
```

```
margin-bottom: 10px;
```

```
}
```

```
button {
```

```
padding: 10px 20px;
```

```
font-size: 16px;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="container">
```

```
<input type="text" id="num1" placeholder= "Enter first number"><br>
```

```
<input type="text" id="num2" placeholder= "Enter second number"><br>
```

```
<button onclick="addNumbers()">Add</button><br>
```

```
<input type="text" id="result" placeholder= "Result" readonly>
```

</div>

<script>

function addNumbers()

```
    var num1 = parseFloat(document.  
        getElementById("num1").value);  
    var num2 = parseFloat(document.  
        getElementById("num2").value);
```

var sum = num1 + num2;

```
    document.getElementById("result").  
        value = sum;
```

}

</script>

</body>

</html>

### Pop UP Boxes

Javascript provides various popup or dialog boxes that allow developers to interact with the user.

They are commonly used to display message, gather information or confirm actions.

Three types of Pop UP boxes are

#### 1) Alert Box

It displays a message to the user which is commonly important information or notification.

```
alert("Hello");
```

#### 2) Confirm Box

Used to get a binary (Yes or No) response from the user.

```
var result = confirm("Are you sure?");
```

```
if(result){
```

}

```
else{
```

}

#### 3) Prompt Box

It allows user to enter text input.

```
var name = prompt("Enter name");
```

```
if(name != null & name != ""){
```

}

```
else{
```

}

<html>

<head>

<title> Dialog Box Example </title>

<script>

```
function showAlert() {
```

```

    alert ("This is alert Box");
}

function showPrompt() {
    var name = prompt("enter name")
    if (name != null & name != "") {
        alert ("Hello " + name);
    } else {
        alert ("you did not enter a name");
    }
}

function showConfirm() {
    var res = confirm ("are you sure");
    if (res) {
        alert ("item deleted");
    } else {
        alert ("item not deleted");
    }
}

<script>
</head>
<body>
    <button onclick = "showAlert()"> Alert</button>
    <button onclick = "showPrompt()"> Prompt</button>
    <button onclick = "showConfirm()"> Confirm</button>
</body>

```

### JavaScript Errors

Errors are problem that occur during the execution of code and are recoverable.

**Reference Error:** occurs when you try to access a variable or function that doesn't exist.

**TypeError:** occurs when you try to perform an operation on a value of unexpected type.

**SyntaxError:** occurs when JavaScript Engine encounters invalid syntax.

**RangeError:** occurs when you use a value that is outside the range.

### JavaScript Exception

Exceptions are the special conditions which break the normal flow of executing code ~~but~~ as it can be recovered unlike errors.

**try :** a code block to run (try to run)

**catch :** a code block to handle any error

**finally :** a code block to run regardless of the result

throw: defines a custom error

```
<html>
<head>
<title> Exception Handling </title>
<script>
    function divide(a, b) {
        try {
            if (b == 0)
                throw new Error("cannot divide by zero");
            return a/b;
        } catch (error) {
            document.write(`An exception occurred: ${error.message}`);
        }
    }

```

function handleClick()

```
var num1 = parseInt(document.getElementById('num1').value);
var num2 = parseInt(document.getElementById('num2').value);
var result = divide(num1, num2);
document.getElementById('result').value = result;
```

<body>

```
<label for="num1"> Number 1: </label>
<input type="number" id="num1">
<br>
<label for="num2"> Number 2: </label>
<input type="number" id="num2">
<br>
<button onclick="handleClick()"> Divide </button>
<p id="result"> </p>
</body>
</html>
```

### Form Validation

It refers to the process of checking the data entered in the form by the user to check that it meets the requirement or not.

```
<html>
<head>
<head>
<title> Form Validation </title>
<script>
</script>
```

```

function validateform() {
    var name = document.myform.name.value;
    var x = document.myform.email.value;
    var firstp = document.myform.pass.value;
    var secondp = document.myform.pass2.value;
    var num = document.myform.num.value;
    var value = document.myform.empty.value;
}

```

If firstp == null || num != ""

if (name == null || name == "")

    alert("Name can't be blank");  
    return false;    3

    else if (name.length < 10){  
        alert("Name must be of atleast 10  
              character long");

    3. return false;

if ( isNaN(num))

    alert("not a number");  
    return false;

3. else return true;

if (value == "")

    alert("Please enter a value");  
    return false;

3. else return true;

i) (firstp != secondp)

    alert("password must be same");  
    return false;

3

var dotposition = n.indexOf('@');

var dotposition = n.lastIndexOf('.',');

if (dotposition < 1 || dotposition > dotposition + 2)

    || dotposition + 2 > n.length)

    alert("Enter valid email");

    return false;

    return true;

</script>

</head>

<body>

<form name="myform" onsubmit="return validateform();"

Name: <input type="name" name="name"><br>

Password: <input type="password" name="pass"><br>

ReType-Password: <input type="password"  
                  name="pass2"><br>

Email: <input type="email" name="email"><br>

Digit: <input type="number" name="num"><br>

Text: <input type="text" name="empty"><br>

<input type="submit" value="register">

</form> </body> </html>