JavaScript is a object-based scripting language and it is light weighted.

It is first implemented by Netscape (with help from Sun Microsystems)

JavaScript was created by Brendan Eich at Netscape in 1995 for the purpose of allowing code

in web-pages (performing logical operation on client side).

Uses of JavaScript

=====================

Client-side validation

Dynamic drop-down menus

Displaying data and time

Validate user input in an HTML form before sending the data to a server.

Build forms that respond to user input without accessing a server.

Change the appearance of HTML documents and dynamically write HTML into separate Windows.

Open and close new windows or frames.

Manipulate HTML "layers" including hiding, moving, and allowing the user to drag them around a browser window.

Build small but complete client side programs .

Displaying popup windows and dialog boxes (like alert dialog box, confirm dialog box and prompt dialog box)

Displaying clocks etc.

1.**Javascript – change HTML content**

<!DOCTYPE html>

<html>

<body>

<h2>What Can JavaScript Do?</h2>

<p id="demo">JavaScript can change HTML content.</p>

<button type="button" onclick='document.getElementById("demo").innerHTML = "Hello JavaScript!"'>Click Me!</button>

</body>

</html>

2.**Change HTML attributes, change HTML styles, Hide HTML elements, show HTML elements**

3**. JavaScript Forms Validation**

The JavaScript provides the facility to validate the form on the client side so processing will be fast than server-side validation. So, most of the web developers prefer client-side form validation using JavaScript.

<html>

<head>

<script>

function form\_validation(){

var name=document.myform.name.value;

//var x = document.forms["myform"]["name"].value;

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

alert("Name can't be blank");

return false;

}

}

</script>

</head>

<body>

<form name="myform" method="post" action="register.php" onsubmit="return form\_validation()" >

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

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

</form>

</body>

</html>

4.Function Expression

<!DOCTYPE html>

<html>

<body>

<p>After a function has been stored in a variable,

the variable can be used as a function:</p>

<p id="demo"></p>

<script>

var x = function (a, b) {return a \* b};

document.getElementById("demo").innerHTML = x(4, 3);

</script>

</body>

</html>

5.Function Hoisting

myFunction(5);  
  
function myFunction(y) {  
    return y \* y;  
}

6. Function as objects - with methods and properties

<!DOCTYPE html>

<html>

<body>

<p>The arguments.length property returns the number of arguments received by the function:</p>

<p id="demo"></p>

<script>

function myFunction(a, b) {

return arguments.length;

}

document.getElementById("demo").innerHTML = myFunction(4, 3);

</script>

</body>

</html>

6.Function Parameters

<!DOCTYPE html>

<html>

<body>

<p>Setting a default value to a function parameter.</p>

<p id="demo"></p>

<script>

function myFunction(x, y) {

if (y === undefined) {

y = 0;

}

return x \* y;

}

document.getElementById("demo").innerHTML = myFunction(4);

</script>

</body>

</html>

7.Function with argument objects

<!DOCTYPE html>

<html>

<body>

<p>Finding the largest number.</p>

<p id="demo"></p>

<script>

function findMax() {

var i;

var max = -Infinity;

for(i = 0; i < arguments.length; i++) {

if (arguments[i] > max) {

max = arguments[i];

}

}

return max;

}

document.getElementById("demo").innerHTML = findMax(4, 5, 6);

</script>

</body>

</html>

8. this – the object that owns the current code

the value of this, when used in a function, is the object that "owns" the function.

The Global Object

When a function is called without an owner object, the value of this becomes the global object.

In a web browser the global object is the browser window.

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript Functions</h2>

<p>In HTML the value of <b>this</b>, in a global function, is the window object.</p>

<p id="demo"></p>

<script>

var x = myFunction();

function myFunction() {

return this;

}

document.getElementById("demo").innerHTML = x;

</script>

</body>

</html>

9. Invoking function as a method

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript Functions</h2>

<p>myObject.fullName() will return John Doe:</p>

<p id="demo"></p>

<script>

var myObject = {

firstName:"John",

lastName: "Doe",

fullName: function() {

return this.firstName + " " + this.lastName;

}

}

document.getElementById("demo").innerHTML = myObject.fullName();

</script>

</body>

</html>

10. Function with properties and methods .

All JavaScript functions are object methods.

If a function is not a method of a JavaScript object, it is a function of the global object

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript Functions</h2>

<p>This example creates an object with 3 properties (firstName, lastName, fullName).</p>

<p>The fullName property is a method:</p>

<p id="demo"></p>

<script>

var myObject = {

firstName:"John",

lastName: "Doe",

fullName: function() {

return this.firstName + " " + this.lastName;

}

}

x = myObject.fullName();

document.getElementById("demo").innerHTML = x;

</script>

</body>

</html>

11. Function with predefined call method

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript Functions</h2>

<p>This <strong>call()</strong> example uses the fullName method of person on myObject:</p>

<p id="demo"></p>

<script>

var person = {

firstName:"John",

lastName: "Doe",

fullName: function() {

return this.firstName + " " + this.lastName;

}

}

var myObject = {

firstName:"Mary",

lastName: "Doe",

}

x = person.fullName.call(myObject);

document.getElementById("demo").innerHTML = x;

</script>

</body>

</html>