

Chapter 6

Client Side Scripting Using Java Script

Java Script

Java script is a **client side scripting** language used to validate data. It is embedded in HTML document. The **<SCRIPT>** tag is used to include scripts in an HTML page. JavaScript was developed by **Brendan Eich**. It is supported by almost all web browsers. JavaScript was first known as **LiveScript**. It was developed under the name **Mocha**.

<SCRIPT> Tag

The **<SCRIPT>** tag is used to include (embed) script in an HTML page. The Language attribute of **<SCRIPT>** tag specifies the type of scripting language used.

Example:-

```
<SCRIPT Language="JavaScript">  
</SCRIPT>
```

The document.write () function

The function **document.write** which writes a string into our HTML document.

First JavaScript

```
<html>  
  
<body>  
  
  <script language="javascript" >  
  
    document.write("Welcome to JavaScript") ;  
  
  </script>  
  
</body>  
  
</html>
```

Note:- We use Notepad for creating the above script and save the file with extension *.HTML* or *.HTM*. JavaScript ignores spaces, tabs, and newlines that appear in JavaScript programs. JavaScript is a case-sensitive language.

Creating functions in JavaScript

A function is a self-contained unit of a program that perform a specific task. It is an independent part of a program which is executed when it is called. A function is often referred to as a '**black box**'. Normally functions are placed in the < **HEAD**> Tag of < **HTML**> Tag.

Defining a function

Functions must be declared before they are used. A function is defined using the function keyword. The function statements are enclosed in curly brace. Once a function is defined it can be called many times. A function has two parts **function header** and **function body**.

Example:-

```
function print( )
{
    document.write("Welcome to JavaScript Programming");
}
```

Note:- A function in JavaScript does not return type.

Calling a function

A function can be called using its name such as, print().

Data types in JavaScript

Data type specifies the type of data and the operations that can be performed on the data. Data types in JavaScript are classified into two primitive data type and composite data type.

Primitive data types

The three primitive data types in JavaScript are **Number**, **String** and **Boolean**.

Number:- They include integers and floating point numbers.

Strings:- A string is a combination of characters, numbers or symbols enclosed within double quotes.

Boolean:- A boolean data can be either True or False.

Variables in JavaScript

A variable is a data type whose value changes during program execution. Variables in JavaScript are declared using the **var** keyword.

Example:- var a;

Declares a variable a.

Example:- var msg="Hello";

Note:- *JavaScript determines the type of variable only when assigning a value.*

Example:-

A web page to find the sum of two numbers

<HTML>

<HEAD>

 <TITLE>JAVASCRIPT_DEMO</TITLE>

<SCRIPT Language="JavaScript">

function sum()

{

 var a,b,c;

 a=10;

 b=2;

 c=a+b;

 document.write("Sum is");

 document.write(c);

}

</SCRIPT>

</HEAD>

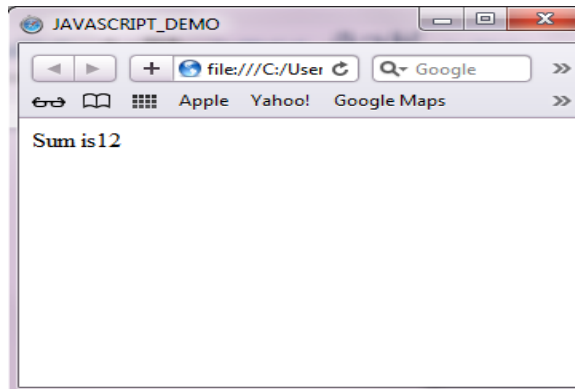
<BODY>

<SCRIPT Language="JavaScript">

HSSLIVE

```
sum( );  
</SCRIPT>  
</BODY>  
</HTML>
```

Output:-



Operators in JavaScript

An operator is a symbol used to perform a specific task. Operators in JavaScript are classified into two, Arithmetic operator, Relational operator, and Logical operator.

Arithmetic Operator

Arithmetic operator is used to perform arithmetic operations.

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus
++	Increment
--	Decrement

Relational Operator

Relational operator is used for comparison operations.

Operator	Example	Same As
=	x = y	x = y
+=	x += y	x = x + y
-=	x -= y	x = x - y
*=	x *= y	x = x * y
/=	x /= y	x = x / y
%=	x %= y	x = x % y

Assignment Operator

The assignment operator(=) operator is used to assign a value to a variable.

Example:- var a=10;

Assigns 10 to the variable a.

Arithmetic Assignment Operator

The arithmetic assignment operators includes +=, -=, *=, /=, %= .They are used to simplify the use of assignment operator.

Example:-

a+=10 is equal to a=a+10

a%=10 is equal to a=a%10

Logical Operator

Logical operators are used for comparison.

Given that **x = 6** and **y = 3**, the table below explains the logical operators:

Operator	Description	Example
&&	and	(x < 10 && y > 1) is true
	or	(x == 5 y == 5) is false
!	not	!(x == y) is true

Arithmetic Operator Precedence

The following table lists JavaScript arithmetic operators, ordered from highest to lowest precedence:

Operator	Precedence
()	Expression grouping
++ --	Increment and decrement
* / %	Multiplication, division, and modulo division
+ -	Addition and subtraction

String addition

The + operator can be used to add(Join) two strings.

Example:-

```
var a,b,c
a="Hello";
b="How are you";
c=a+b;
document.write(c);
```

Output:-

Hello How are you

Number function in JavaScript

The **Number()** function is used to convert a string data into a number.

Example:-

```
Number("42"); //returns 42
```

Number("eggs"); //returns NaN String that can't be converted to number returns NaN.

Number(true); //returns 1

Number(false); //returns 0

Note:-

NaN stands for Not a Number.

Control structures in JavaScript

Control structures are used to alter the normal sequence of execution of a program. The important

Control structures in JavaScript are,

1)if statement

The if statement executes a group of statements based on a condition. The syntax is

```
if(test_expression)
{
    Statements;
}
```

2)Switch statement

The Switch statement is a multi-branching statement, which executes statement based on value of the expression. The syntax is

```
switch(expression)
{
    case value1:statement1;break;
    case value2:statement2;break;
    - -----
    Default:statement;
}
```

3)for Loop

The for loop executes a group of statements repeatedly. The syntax is

```
for(initialisation;expression;update_statement)
{
    statements;
}
```

4) While Loop

The while loop executes a group of statements repeatedly based on a condition. The syntax is

```
while(expression)
{
    statements;
}
```

Built-in Functions

Built-in functions are also called methods. The following are the important methods in JavaScript.

1) alert() function

The alert() function is used to display a message on the screen. The syntax is

```
alert("message");
```

2) isNaN() function

The isNaN() function is used to check if a value is a number or not. The function returns **True** if the value is not a number. The syntax is

```
isNaN(test_value);
```

3) toUpperCase () function

This function converts a string into uppercase.

Example:-

```
var a,b;
```

HSSLIVE

```
a="abcd";  
b=a.toUpperCase( );  
document.write(b);
```

Output:ABCD

4)toLowerCase() function

This function converts a string to lowercase.

5)charAt() function

The **charAt()** method returns the character at the specified index in a string. The index of the first character is 0, the second character is 1, and so on.

Example:-

```
var str = "HELLO WORLD";  
var res = str.charAt(0);
```

returns H

length Property

The length property returns the length of a string(**number of characters**).

Example:-

```
var a="Welcome";  
var len=a.length;  
document.write(len);
```

Output:-

7

Note:-*A function has parenthesis but property does not have parenthesis.*

Accessing values in a textbox using JavaScript

Names are provided to textboxes to access them.

The following HTML code is used to specify name to textbox.

<INPUT Type="Text" Name="txtnum";

The above textbox can be accessed as

```
var n= document.txtnum.value;
```

The above statement stores the value in txtnum in variable n.

JavaScript Events

JavaScript's interaction with HTML is controlled using events. JavaScripts reacts to events(click,keydown,keyup etc).Events are actions that are initiated by the user.The important events in JavaScript are,

Mouse Events

Event	Description
<u>onclick</u>	The event occurs when the user clicks on an element
<u>oncontextmenu</u>	The event occurs when the user right-clicks on an element to open a context menu
<u>ondblclick</u>	The event occurs when the user double-clicks on an element
<u>onmousedown</u>	The event occurs when the user presses a mouse button over an element
<u>onmouseenter</u>	The event occurs when the pointer is moved onto an element
<u>onmouseleave</u>	The event occurs when the pointer is moved out of an element
<u>onmousemove</u>	The event occurs when the pointer is moving while it is over an element
<u>onmouseover</u>	The event occurs when the pointer is moved onto an element, or onto one of its children
<u>onmouseout</u>	The event occurs when a user moves the mouse pointer out of an element, or out of one of its children
<u>onmouseup</u>	The event occurs when a user releases a mouse button over an element

Keyboard Events

Event	Description
<u>onkeydown</u>	The event occurs when the user is pressing a key
<u>onkeypress</u>	The event occurs when the user presses a key
<u>onkeyup</u>	The event occurs when the user releases a key

Q.Create a web page to create that display the square of a number

<HTML>

<HEAD>

<TITLE>JAVA SCRIPT</TITLE>

<SCRIPT Language="JavaScript">

function square()

{

var num,sqr;

num=document.frmSquare.txtNum.value;

sqr=num*num;

document.frmSquare.txtSqr.value=sqr;

}

</SCRIPT>

</HEAD>

<BODY>

<FORM Name="frmSquare">

<CENTER>

Enter a Number :

```
<INPUT Type="text" name="txtNum">
```

```
<BR><BR>
```

Square :

```
<INPUT TYPE="text" name="txtSqr">
```

```
<BR><BR>
```

```
<INPUT Type="button" value="Square" onClick="square()">
```

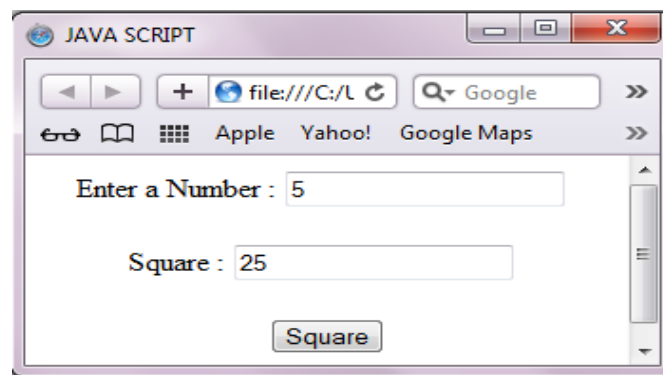
```
</CENTER>
```

```
</FORM>
```

```
</BODY>
```

```
</HTML>
```

Output:-



Different methods to add script to a web page

Script can be placed inside the **<BODY>** or **<HEAD>** tag. Placing the script in **<HEAD>** tag helps to execute scripts faster as the head section is loaded before the body section. The main disadvantage is that scripts that are to be executed while loading the page will not work when they are placed in the **<HEAD>** section.

Scripts can be placed into an external file, saved with **.js** extension. When scripts are placed in an external file, it can be used by multiple HTML pages and also helps to load pages faster. The file is linked to **HTML** file using the **<SCRIPT>** tag. For example,

```
<SCRIPT Type="text/JavaScript" Src="processdata.js">
```

The above code links the external file named process.js to the HTML page.

Note:- When *Src* tag is specified code in the **<SCRIPT>** tag may not work depending on web browser.

1) Create a web page to display first 10 Natural numbers

```
<HTML>
```

```
<HEAD>
```

```
<TITLE> Demo_Program</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<SCRIPT Language="JavaScript">
```

```
var i;
```

```
for(i=1;i<=10;i++)
```

```
{
```

```
    document.write(i+"<BR>");
```

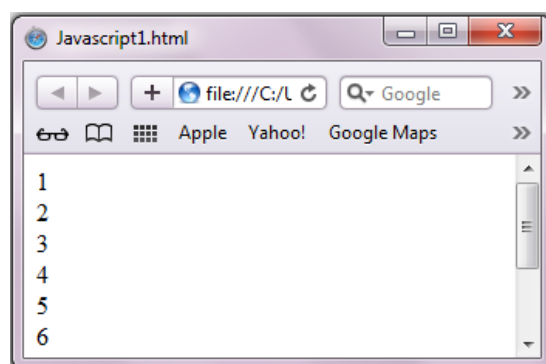
```
}
```

```
</SCRIPT>
```

```
</BODY>
```

```
</HTML>
```

Output:-



2) Create a web page to Print Even numbers up to 100

```
<HTML>

<HEAD>

<TITLE> Even_Numbers</TITLE>

</HEAD>

<BODY>

<SCRIPT Language="JavaScript">

var i;

for(i=2;i<=100;i+=2)

{

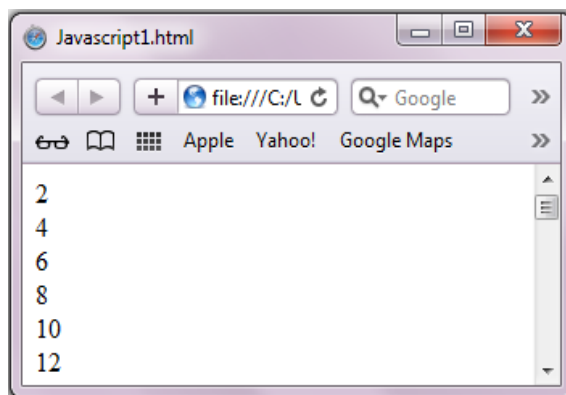
    document.write(i+"<BR>");

}

</SCRIPT>

</BODY>

</HTML>
```

Output:-**3) Design a web page to find the sum of first 10 Natural numbers**

```
<HTML>

<HEAD>

<TITLE> Sum_Of_Natural_Numbers</TITLE>

</HEAD>

<BODY>

<SCRIPT Language="JavaScript">

var i,s=0;

for(i=1;i<=10;i++)

{

s=s+i;

}

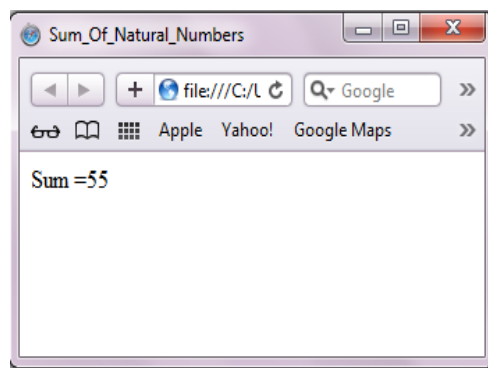
document.write("Sum =" +s);

</SCRIPT>

</BODY>

</HTML>
```

Output:-



4)Design a web page to find the sum of n Natural numbers

```
<HTML>

<HEAD>
```

```
<TITLE>Sum</TITLE>

<SCRIPT Language="JavaScript">

function sum( )

{

    var sum=0,i,limit;

    limit=Number(document.frmsum.txtlimit.value);

    for(i=1;i<=limit;i++)

    {

        sum=sum+i;

    }

    document.frmsum.txtsum.value=sum;

}

</SCRIPT>

</HEAD>

<BODY>

<FORM Name="frmsum">

    <CENTER>

        Enter the limit:

        <INPUT Type="text" Name="txtlimit">

        <BR><BR>

        Sum:

        <INPUT Type="text" name="txtsum">

        <BR><BR>

        <INPUT Type="button" value="Sum" onClick="sum( )">

    </CENTER>
```

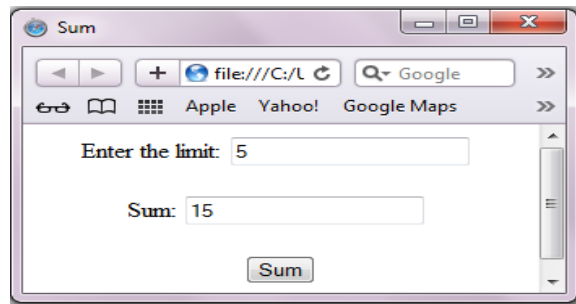
HSSLIVE

</FORM>

</BODY>

</HTML>

Output:-



5)Design a web page to perform arithmetic operations such as addition,substraction etc.

<HTML>

<HEAD>

<TITLE>Calculator</TITLE>

<SCRIPT Language="JavaScript">

function plus()

{

var num1,num2,result;

num1=document.frmcalc.txtNum1.value;

num2=document.frmcalc.txtNum2.value;

result=Number(num1)+Number(num2);

document.frmcalc.txtResult.value=result;

}

function minus()

{

var num1,num2,result;

```
num1=document.frmcalc.txtNum1.value;
num2=document.frmcalc.txtNum2.value;
result=Number(num1)-Number(num2);
document.frmcalc.txtResult.value=result;
}

function div( )
{
    var num1,num2,result;
    num1=document.frmcalc.txtNum1.value;
    num2=document.frmcalc.txtNum2.value;
    result=Number(num1)/Number(num2);
    document.frmcalc.txtResult.value=result;
}

function mul( )
{
    var num1,num2,result;
    num1=document.frmcalc.txtNum1.value;
    num2=document.frmcalc.txtNum2.value;
    result=Number(num1)*Number(num2);
    document.frmcalc.txtResult.value=result;
}

</SCRIPT>

</HEAD>

<FORM name=frmcalc>

<CENTER>
```

Number 1

```
<INPUT Type="text" name="txtNum1">
```

```
<BR><BR>
```

Number 2

```
<INPUT Type="text" name="txtNum2">
```

```
<BR><BR>
```

Answer

```
<INPUT Type="text" name="txtResult">
```

```
<BR><BR>
```

```
<INPUT Type="button" value="+" onClick="plus()">
```

```
<INPUT Type="button" value="-" onClick="minus()">
```

```
<INPUT Type="button" value="*" onClick="mul()">
```

```
<INPUT Type="button" value="/" onClick="div()">
```

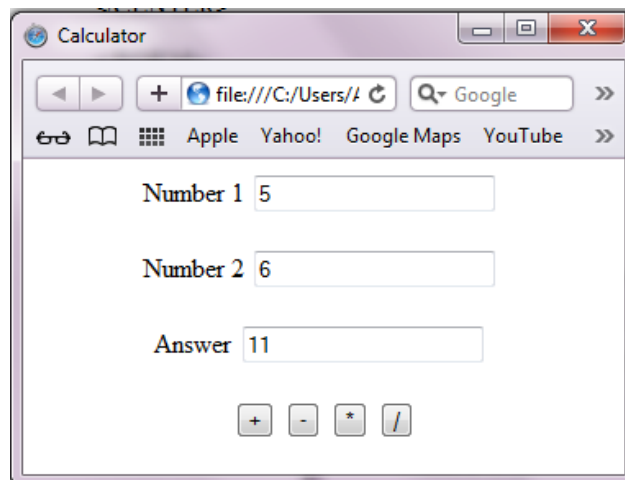
```
</CENTER>
```

```
</FORM>
```

```
</BODY>
```

```
</HTML>
```

Output:-



6)Design a web page to check if a number is Even or Odd

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Even_Odd</TITLE>
```

```
<SCRIPT Language="JavaScript">
```

```
function check( )
```

```
{
```

```
var num1,rem;
```

```
num1=document.frmcheck.txtNum1.value;
```

```
rem=Number(num1)%2;
```

```
if(rem==0)
```

```
{
```

```
document.frmcheck.txtresult.value="Even";
```

```
}
```

```
else
```

```
{
```

```
document.frmcheck.txtresult.value="Odd";
```

```
}
```

```
}
```

```
</SCRIPT>
```

```
</HEAD>
```

```
<FORM name=frmcheck>
```

```
<CENTER>
```

Enter a number :

```
<INPUT Type="text" name="txtNum1">
```

```
<BR><BR>
```

The number is :

```
<INPUT Type="text" name="txtresult">
```

```
<BR><BR>
```

```
<INPUT Type="Button" Value="show" onClick="check ( )">
```

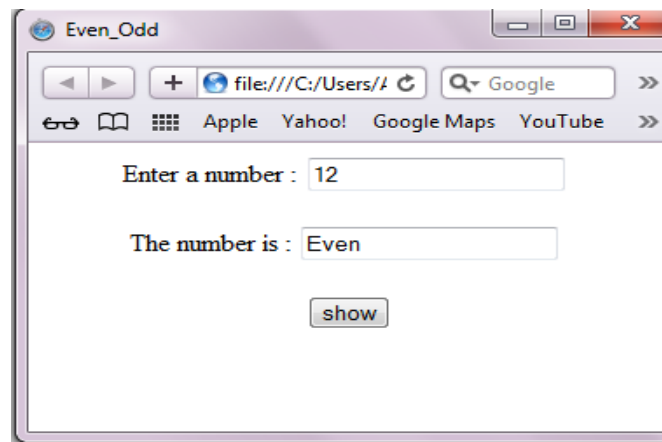
```
</CENTER>
```

```
</FORM>
```

```
</BODY>
```

```
</HTML>
```

Output:-



Conclusion:-

- **JavaScript** was developed by **Brendan Eich**.
- **<SCRIPT>** Tag is used to embed script in an HTML page.
- In JavaScript, a variable without a value, has the value **undefined**.
- Java script is case sensitive language.
- JavaScript is interpreted by **JavaScript engine**.
- Document represents the body section of the web page.
- Every Javascript statement ends in a semicolon.
- JavaScript uses the keyword **function** to define a function.
- JavaScript functions has no return type.
- JavaScript variables are declared using the keyword **var**.
- The **alert()** function is used to display a message on the screen.