Chapter-6
Client Side Scripting Using Java Script

A scripting language is a lightweight programming language with less complexity.

- JavaScript is a client side scripting language and an interpreted language.
- Client side scripting languages are used for validation of data at the client side itself.
- This reduces network traffic and work load of the server.
- JavaScript is developed by Brendan Eich for the Netscape browser .
- Server side scripting languages are executed at the server and the web page produced is returned to the client browser.
- Server stores huge amount of data in the form of a database. So server side scripting languages may have to to interact with these database, but a client need not.
- JavaScript is a case-sensitive language, which means keywords (lower case only), variable names, function names and identifiers should be typed with a consistent casing of letters. To denote identifiers, it is common to use camel Case names.
- When the first character of each word is capital , like DateOfBirth, JoinTime, it is known as Upper Camel Case. When the first letter of each word except the first word is capitalised, it is called lowerCamelCase, like dateOfBirth, joinTime.
- JavaScript ignores spaces, tabs, and newlines that appear in JavaScript programs.
- JavaScript was first known as LiveScript.

Scripts are small programs embedded in the web pages to interact with the user and make dynamic pages. JavaScript is a client side scripting language used for validate data, to add dynamism and interactivity to web pages. This reduces the workload of the server. It is embedded in HTML document.
The **<SCRIPT>** tag and **</SCRIPT>** tag is used to include scripts in an HTML page.
The **Language** attribute specifies what scripting language we are using, here it is JavaScript. ***<SCRIPT Language =”JavaScript”>*** tells the browser that the code that follows is a JavaScript code.
In JavaScript **document.write( )** function is used to print a text in the body section of an HTML page.

Creating Functions in JavaScript: A function is a set of reusable codes used to perform a particular task. It can be called anywhere and any number of times in the program. There are built-in functions and user defined function. We have to define a function, before it is used. To declare a function, the keyword **function** is used. It is better to include the function definition within the HEAD section. To execute a function, it must be called by using its function name. In C++, the function has a return type, but in JavaScript there is no return type.

The syntax is

```
<SCRIPT Language =”JavaScript”>
function        function_name( )
{
        statements;
}
</SCRIPT>
```

Data types in JavaScript:

Data type specifies the type of data and the operations that can be performed on the data. There are three basic data types in JavaScript.
a). Number: Includes integers and floating point numbers. Eg. 12, 3.14, -7 etc.
b). **String**: Includes characters, numbers, symbols enclosed within double quotes. Eg. “hss@12”, “fifty50” etc.
c): **Boolean**: The only values are True and false. The values are not in double quotes.

Variables in JavaScript:
In JavaScript, data can be temporarily stored in variables, which are the named locations in the memory. A variable has a name, value and memory address.
Variables should be declared with the keyword **var** before using that variable in JavaScript program. The variables should be separated by comma. There is no need to specify the data type.

Syntax is **var variable_name;**
Eg. Var x, y, z; x=5; y="Five"; z=false;

Here x is of type number, y is of type string, z is of type boolean.
[Note: If a variable is declared, but not given a value, then the script engine is unable to understand its type and so it is declared as ***undefined***. That is '***undefined***' is a special datatype to represent variables that are not defined using var.].
To find the type of a variable, we use the function **typeof( )**.
Eg. **Document.write(typeof(y));**
Suppose a variable is declared, but not given a value. Then the script engine can't know its data type and declared as ***undefined.***
Operators in JavaScript: An operator is a symbol used to perform a specific task.

Arithmetic operators:

Operator	Description	Example
+	Adds two numbers or join two strings.	5+3 returns 8. "a"+"b" returns ab. x="20" , y=5, x+y returns 205.
-	subtraction	10 – 7 returns 3
*	Multiplication	5*3 returns 15
/	division	5/2 returns 2.5
%	Modulus (returns the remainder)	5%2 returns 1
++	Increment (when prefixed, the value is incremented in the current statement, and when suffixed, the value is incremented after the current statement.	Y=10, x=++y , x=11 y=10, x= y++, x=10
- -	Decrement	Y=10, x=--y, x=9 y=10, x==y-- , x=10

The assignment operators includes =, +=, - =, * =, / = , % =. They are used to simplify the expression.
Ex. a+=10 is equal to a=a+10 , a% =10 is equal to a=a%10 .
Relational (Comparison operator): Includes operators ==, !=, <, <=, >, >=. The result of a relational operator is either TRUE or FALSE.
Ex. 5==10 returns false. 15>=10 returns true.
Logical operators: Includes && (AND), || (OR), ! (NOT).
String addition: The + operator can be used to join two strings.
Ex a="Hello"; b= "Abu". a+b returns Hello Abu.
Note: number() is a function in JavaScript, and it converts a string type data containing numbers to number type.
Ex. X="10"; y=5; number(x) + y returns the value 15.
Number("fruit"); //returns NaN String that can't be converted to number returns NaN. (Not a Number - NaN)
Number(true);//returns 1 Number(false) ; return 0
Control structures in JavaScript:
Control structures are used to change the sequence of execution of a program.
a). if statement and if-else statement.

The syntax is

```
if (test condition)
{
        statements;
}
else
{
                statements;
}
```

Example.

```
<html>
<head> <title> java Script </title> </head>
        <font color=green size=3>
<body>
<SCRIPT Language="JavaScript">
var mark=20;
 if(mark>=30)
{
          document.write("The student is passed");

}
 else
 {
          document.write("The student is failed ");
 }
 </SCRIPT>
</body>
</html>
```

b). SWITCH statement: A switch statement is used to select a particular group of statements to be executed among several other group of statements.

Syntax is

```
switch(expression)
   {
      case value_1:  statements;
                     break;
     .....................................
      case value_n:  statements;
                     break;
      default:    statements;
   }
```

```
<html>
<head><title>java Script</title></head>
 <font color=green size=3>
<body>
<script language="JavaScript">
      var       i,   s;
for(i=1; i<=10; i++)
{
         s=i*i;
         document.write(s);
         document.write("<br>");
}
</script></body></html>
```

c) for loop: It is used to execute a group of instructions repeatedly.

Syntax is

```
for(initialisation;test_expression;update_expression)
{
        statements;
}
```

Ex.

d). **WHILE loop.**

While loop executes a group of statements repeatedly based on a condition. The loop variable must be in initialised outside the loop and updated inside the body. The body of the loop will be executed until the expression becomes false.

The syntax is

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

BUILT-IN functions: They are also called ***methods***.

a). alert( ) function :

It is used to display a message on the screen (at the time of data validation).

Ex. alert ("welcome to JS ");

b). isNaN( ) function : It is used to check whether a value is a Number or Not (***NaN – Not a Number***) . The function returns ***TRUE if the value is not a number.***

Ex. isNaN("A12"); [true], isNaN(7.5); [false], isNaN("5"); [false], alert(isNaN("A")); [true]

c). toUpperCase( ) function: This function converts a string into upper case.

Ex.

```
<html> <head> <title> Java Script</title> </head>
<font color=green size=3>
<h2> Convert into Upper Case </h2>
<body>
<script language="JavaScript">
        var x, y;
        x="abcdEF";
        y=x.toUpperCase( );
        alert(y);
</scrpt> </body></html>        //The output is ABCDEF
```

d). toLowerCase( ) function :

This function converts a string into lower case.

```
Ex.   var x, y ;
      x="JAVA";
      y=x.toLowerCase();
        alert(y);        //The output is 'java' .
```

e). charAt( ) function: Returns the character in the specified index.

charAt(0) returns the first character, charAt(1) returns second character in the string etc.

```
Ex. var x, y;  x="welcome";
    y=charAt(3);    Output is 'c'.
```

f). length property: [***The function has parenthesis , but property does not have parenthesis.***]

This property returns the number of characters (length) of the string.

```
Ex.    var x,n;
       x="computer";
       n=x.length;
       alert(n);      //Output is 8.
```

Accessing values in a text box using JavaScript :

JavaScript's interaction with HTML is handled through EVENTS that occur when the user or the browser manipulates a page.

Eg. When a user clicks a button, that click is an event.

Events refers to actions that are detected by the programming language when we perform a particular task. [An

event commonly occurs when a user clicks the mouse button, web page is loaded, or form field is changed. Events are handled by a special function, known as event handler, which handles a particular event when the event is triggered. Some of the events are listed here

EVENT	Description
onClick	Occurs when the user clicks on an object
onMouseEnter	Occurs when the mouse pointer is moved onto an object
onMouseLeave	Occurs when the mouse pointer is moved out of an object
onKeyDown	Occurs when the user is pressing a key on the keyboard
onKeyUp	Occurs when the user releases a key on the keyboard
OndblClick	Occurs on double clicking the mouse button
onMouseWheel	Occurs on rotating the mouse wheel
onSubmit	Occurs on submitting a form

Forms

Forms are used to entering data and there are various controls like textbox, checkbox, radio button, etc in a web page. To process data entered in the form, scripting languages are essential. Giving Name's to FORM, Textbox, are essential to access them. If we do not give any name to a web page element, the JavaScript cannot access that element.

Form control elements: Form contains object such as text box, radio buttons, check boxes etc. Anything enclosed within the tag pair <form> and </form> called the form object.

The syntax is

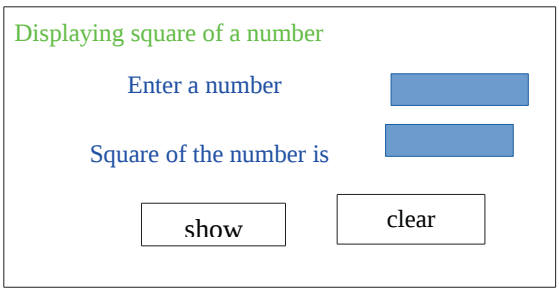
document.form_name.control_name.property;

Form Validation:

It is the process of verifying whether a form has been filled correctly before it is processed. When a user enters data into a form field, it is possible that the user can make a mistake or enter incorrect data in the fields. We can check these mistakes or incorrect data input by validating those fields by using server side validation [uses Java Server Pages (JSP), Active Server Pages (ASP) to validate a form .], or Client Side Validation [Uses Java Scripts or VB Scripts to validate a form].

Create a web page that displays the square of a number.

```
<html><head><title>JavaScript</title>
<font color=green size=3>
<h2> Displaying square of a number</h2></font>
<font color=blue>
<SCRIPT Language="JavaScript">
function displaysquare( )
{
      var n,   s;
      n=document.frmsqr.txtnum.value;
      s=n*n;
      document.frmsqr.txtsqr.value=s;
}
</script>  </head>
<body>
<form Name="frmsqr">
<center>
```



*Enter a number   <input type="text" name="txtnum">

*
*Square of the number is   <input type= "text" name="txtsqr">

*
<input type="button" value="show" onClick="displaysquare( )">
<input type="reset" value="clear">
* </form></body></html>*

Note: Practically we give name of the function and write body element first, because we want to give *name* to *form, text, button* etc. Then we use these names in the function. Here name of the form is ' ***frmsqr*** ' , name of the first text is ' ***txtnum*** ' , name of the second text is ' ***txtsqr*** ' , name of the function is ' ***displaysquare( )*** ' . When the user click the button named '**show**' [value = "show"] , function named ' ***displaysquare( )*** ' is called. ***Document*** refers the body section of the web page. The expression ' ***n=document.frmsqr.txtnum.value;*** ' means , the value entered in the textbox named ' **txtnum**', which is in the form named ' **frmsqr** ' , of the web page to be displayed is assigned to the variable ' **n** '. Then find the square of the entered number [n*n] and assigned to the variable ' s ' . This value is displayed in the second text box. ** ** [non breaking space] is written to adjust the space between the text and text box.
[ie. Enter number]

Ways to add scripts to a web page:

a). Inside <BODY> section. SCRIPTS can be placed inside the <body> section. Here the scripts element placed inside the BODY element runs when a Web page starts loading in a web browser.

Ex. <body>
<script language = "JavaScript"> script codes ..... </script>
</body>

b). inside the <HEAD> section. Scripts can be placed inside the <head> section. The script placed inside the HEAD element runs when the user perform some action, such as click the submit button or a link. This is widely accepted method.

c). JavaScript in an External file. We can write SCRIPTS in a file and save it as a separate file with extension ' .js ' . We can use the same JavaScript code in several web pages. This enhances the speed of page loading. We have to link this file using the 'SRC' attribute of the SCRIPT element. Ex. *<HEAD> <script type="text/JavaScript" src="check.js"></script></head>*
Here the code links the external file named **check.js** .

Create a web page to find the Sum of two numbers :

```
<html><head><title>Java Script</title>
<font color=green size=3>
<h2> Sum of two numbers </h2>
</font>
<font color=blue>
<script language="JavaScript">
function displaysum( )
{
        var n1, n2, s;
        n1=Number(document.frmsum.fnum.value);
        n2=Number(document.frmsum.snum.value);
        s=n1+n2;
        document.frmsum.txtsum.value=s;
 }
</script></head>
<body>
<form Name="frmsum">
<center>
    Enter First number     <input type="text" name="fnum"><br> <br>
   Enter Second number    <input type="text" name="snum"><br> <br>
   Sum of the numbers is     <input Type= "text" name="txtsum"><br> <br>
```

```
      <input Type="button"   value="Total"    onClick="displaysum( )">
      <input Type="reset"     value="clear">
 </center>
</form>
</body></html>
```

Create a web page that displays sum of natural numbers up to a given limit.

```
<html><head><title>Java Script</title><font color=green size=3>
<h2> Sum of first  N  natural numbers   </h2>  </font>          <font color=blue>
<script language="JavaScript">
function sumnat( )
{
          var    i,    n,    s=0;
          n=Number(document.frmsum.txtn.value);
          for(i=1;    i<=n;     i++)
          s=s+i;
          document.frmsum.txtsum.value=s;
 }
 </script></head>
 <body><form name="frmsum">
 <center>
 Enter the limit         <input type="text" name="txtn"> <br> <br>
 Sum of the  numbers  is             <input type= "text"  name="txtsum">    <br> <br> <br>
 <input Type="button"     Value="Total"     onClick="sumnat( )">
 <input Type="reset" value="clear"></center></form>  </body> </html>
```

Product of first N natural numbers.

```
 <html><head><title>Java Script </title><font color=green size=3>
  <h2> Product of first  N  natural numbers </h2></font>
  <font color=blue>
  <script language="JavaScript">
   function sumnat(  )
  {
              var i, n, s=1;
              n=Number(document.frmsum.txtn.value);
             for(i=1; i<=n; i++)
                        s=s*i;
             document.frmsum.txtsum.value=s;
}
 </script></head><body><form Name="frmsum">
 <center>
          Enter the limit         <input type="text"  name="txtn">    <br> <br>
          Sum of the  numbers  is     <input Type= "text" Name="txtsum"> <br> <br>
          <input Type="button" Value="Total" onClick="sumnat( )">
          <input Type="reset" value="clear">
 </center></form></body></html>
```

Even or Odd

```
   <html><head><title>Java Script</title>
    <font color=green size=3>
   <h2> Even or Odd    </h2>
   </font>
```

```
<font color=blue>
<script language="JavaScript">
function evenorodd( )
{                   var n,res;
                    n=Number(document.frmnum.num.value);
                   if(n%2==0)
                     res="The number is Even";
                     else
                     res="The number is Odd";
                     document.frmnum.txtres.value=res;
    }
    </script></head>
    <body>
              <form Name="frmnum">
           <center>
           Enter a number   <input type="text"  name="num">        <br> <br>
           Even Or Odd ?      <input type= "text" name="txtres">            <br> <br> <br>
           <input Type="button" Value="Result" onClick="evenorodd()">
           <input Type="reset" value="clear">
          </center>
           </form></body></html>
```

Upper case to Lower case and vice versa

```
<html><head><title>Java Script </title>
<font color=green size=3>
<h2> Capital into small and vice versa   </h2>
</font>
<font color=blue>  <script   language="JavaScript">
 function upper( )
{        var a,b;
          a=document.frmcase.txt1.value;
          b=a.toUpperCase();
         document.frmcase.txt2.value=b;
}
 function lower( )
 {                  var a,b;
                    a=document.frmcase.txt1.value;
                    b=a.toLowerCase( );
                    document.frmcase.txt2.value=b;
  }
</script></head>
<body>
<form Name="frmcase">
<center>
Enter a word   <input type="text"  name="txt1"  size="30">  <br> <br>
After Changing the Case       <input type= "text" name="txt2" size="30"> <br> <br>
<p>
<input type="button" name=s1  value="To Upper Case"  onClick="upper( )">
<input Type="button" name=s2 Value="To Lower Case" onClick="lower()">
<input Type="reset" value="clear">
</p>
</center>
```

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

Ex. SWITCH .. CASE

```
<html> <head> <title>Java Script</title> </head>
<font color=green size=3>
<body>
<script language="JavaScript">
 var  day;
 day=4;
switch(day)
 {
   case 1:        document.write("Sunday ");
                    break;
   case 2:        document.write("Monday ");
                    break;
   case 3:        document.write("Tuesday ");
                    break;
   case 4:        document.write("Wednesday ");
                    break;
   case 5:        document.write("Thursday ");
                    break;
   case 6:       document.write("Friday ");
                    break;
   case 7:       document.write("Saturday ");
                    break;
   default:       document.write("Invalid day ");
 }
</script></body></html>
```

Ex. WHILE LOOP.

```
<html><head><title>Java Script</title> </head>
<font color=green size=3>
<h3> WHILE  loop  </h3>
</font>
<font color=blue>
 <body>
        <script language="JavaScript">
         var  n;
        n=1;
         while(n<20)
        {
                document.write(n);
                document.write("<br>");
                n+=2;
        }
      </script></body></html>
```

Questions:

1. “TRUE and False are used to represent Boolean values”. State if the given statement is correct or not. (1) (March 2016)
2. Explain the use of for loop with an example. (3) (March 2016)
3. Develop a web page that implements a JavaScript function that takes two numbers as

input and displays their product. (2) (March 2016)
4. Give the function in JavaScript that converts a string type data containing numbers to number type. (1) (SAY 2016)
5. Design a web page with form tag which accepts a number in a textbox and another textbox which should display either odd or even. Write a function in JavaScript to check whether the number is odd or even. (2) (SAY 2016)
6. Develop a web page that accepts a number after validation and prints the factorial of it. (2) (SAY 2016)
7. JavaScript provides a large number of built-in functions.
(a) Name any two of them with an example. (2) (SAY 2016)
(b) The property which returns the size of the string is _____. (1) (SAY 2016)
8. A virtual machine for executing JavaScript code is _____. (1) (March 2017)
9. Discuss about six built-in functions used in JavaScript. (3) (March 2017)
10. Design a procedure in JavaScript that takes two strings as input and displays the concatenated string as output. (2) (March 2017)
11. State whether the following statements are true or false:
(a) JavaScript is the only client side scripting language.
(b) JavaScript is a case sensitive language.
(c) The keyword used to declare a variable in JavaScript is VAR. (3) (SAY 2017
12. Predict the output of the following code:

```
<html>
<body>
<script language="JavaScript">
        var i, s=0;
        for (i=1;i<=10; i+=2)
            s=s+i;
            document.write("sum="+s);
</script>
</body>
</html>
```

(2) (SAY 2017)
13. What is an external JavaScript file? Write the advantage of using an external JavaScript file. (3) (March 2017)
14. Write a JavaScript which inputs the name, rollno and date_of_birth of a student. Date of birth contains month, day and year. Month should be selected from a drop-down list. (3) (SAY 2017)
15. Write JavaScript statements to create a number and string variables. (March 2020)
16. Briefly explain about any two built-in functions available in JavaScript. (March 2020)
17. What are the different control structures used in JavaScript ? Explain any one with an example.(March 2020)