3 Advanced Javascript

Let us learn

- Features of JavaScript, difference between client side scripting and server side scripting.
- Looping structures.
- DOM Objects and window object in JavaScript.
- Inbuilt objects such String, Math, Array, Date and Number with its properties and Methods.
- Simple JavaScript programs to do validations and user interaction.

3.1 Introduction

There is variety of scripting languages used to develop dynamic websites. JavaScript is an interpreted scripting language. An interpreted language is a type of programming language that executes its instructions directly and freely without compiling machine language instructions in precious program. Program is a set of instructions used to produce various kinds of outputs. JavaScript was initially created to "make webpages alive". The programs in this language are called scripts.

3.1.1 Features of JavaScript :

- JavaScript is light weight scripting language because it does not support all features of object oriented programming languages.
- No need of special software to run JavaScript programs
- JavaScript is object oriented scripting language and it supports event based programming facility. It is case sensitive language.
- JavaScript helps the browser to perform input validation without wasting the user's time by the Web server access.
- It can handle date and time very effectively.
- Most of the JavaScript control statements syntax is same as syntax of control statements in other programming languages.
- An important part of JavaScript is the ability to create new functions within scripts. Declare a function in JavaScript using function keyword.
- Software that can run on any hardware platform (PC, Mac, SunSparc etc.) or software platform (Windows, Linux, Mac OS etc.) is called as platform independent software. JavaScript is platform independent scripting language. Any JavaScript-enabled browser can understand and interpreted JavaScript code. Due to different features, JavaScript is known as universal client side scripting language.

There are two types of scripting : Server side scripting and Client side scripting.

Client-side Scripting : In this type, the script resides on client computer (browser) and that can run on the client. Basically, these types of scripts are placed inside an HTML document.

Server-side Scripting : In this type, the script resides on web server. To execute the script it must be activated by client then it is executed on web server.

3.1.2 Difference between Server side scripting and client side scripting

1. Server-side scripting is used at the backend, where the source code is not visible or hidden at the client side (browser). On the other hand, client-side scripting is used at the frontend which users can see from the browser. So Server-side scripting is more secure than client-side scripting.
2. When a server-side script is processed it communicates to the server. As against, client-side scripting does not need any server interaction.
3. The client-side scripting language involves languages such as HTML5, JavaScript etc. In contrast, programming languages such as PHP, ASP.net, Ruby, ColdFusion, Python, C# etc. are server side scripting languages.
4. Server-side scripting is useful in customizing the web pages and implements the dynamic changes in the websites. Conversely, the client-side scripts are generally used for validation purpose and effectively minimize the load to the server.
5. Special software (web server software) is required to execute server-side script, whereas client side scripts requires web browser as an interface.

Do you know?

There are some popular framework / libraries.

- ***Angular JS :*** *It is a java script based open source frontend web framework devloped mainly for single page application.*
- ***Vue JS :*** *It is javascript based frame work for building interactive user interface (UI). It can be easly integrated with other projects and libraries.*
- ***React :*** *It consists of javascript libraries for building UI for single page application and mobile application.*

3.2 Switch case and Looping Structures

Previous year we have learnt different basic syntax of javascript such as variable declaration, if structure, function etc. Let us learn some extra features:

3.2.1 Switch Case statement

JavaScript has a built–in multiway decision statement known as Switch. The switch statement test the value of given expression against a list of case values and when match is found, a block of statement associated with that case is executed. There should not be duplicity

between the cases. The value for the case must be similar data type as the variable in switch. The default statement is not mandatory.

Syntax :

```
switch(expression)
{
case value1:
        statement block 1;
        break;
case value2:
        statement block 2;
        break;
……..
case value n:
        statement block n;
        break;
default:
      statement block ;
}
```

Program :

```
<!DOCTYPE html>
<head><title>Javascript Program
</title></head>
<body>
<h1> use of switch case </h1>
<script type="text/javascript">
var day=6;
switch(day)
{
case 1: alert("Monday");   break;
case 2: alert("Tuesday");  break;
case 3: alert("Wednesday");    break;
case 4:alert("Thursday");  break;
case 5: alert("Friday");           break;
case 6: alert("Saturday");  break;
case 7: alert("Sunday");           break;
default: alert("Invalid day");
 }
</script></body></html>
```

Output :

Note : 'language' attribute of <Script> is replaced by 'type' attribute in all the programs as it is standardised.

3.2.2 Looping Statement

While creating programming logic, we need to execute some statements repeatedly. Iteration refers to the execution of statement or a group of statements of code for a fixed number of times or till the condition is satisfied. The condition should be boolean condition. Some commonly used JavaScript looping statements are:

1. for…….loop

This loop executes statements as long as condition becomes true, control comes out from the loop when condition becomes false. Benefit of for-loop is that it combines initialization, condition and loop iteration (increment or decrement) in single statement.

Syntax :

```
for(initialization;condition;iteration)
{
statement block;
}
```

Initialization is assigning initial value to the variable, which executes only once, and then the condition is checked. Loop will execute statements in statement block till the condition is true. When condition becomes false control is transferred out of the loop and it will execute remaining program. Iteration means increment or decrement value of a running variable.

Example :

```
for(i=1;i<=5;i++)
{
  document.writeln(i);
}
```

Output:
1
2
3
4
5

```
for(i=5;i>=1;i--)
{
  document.writeln(i);
}
```

Output:
5
4
3
2
1

2. While…..loop

This loop executes statements as long as the condition is true. As soon as condition becomes false control comes out of the loop.

Syntax:

```
initialization;
while(condition)
{
  statement block;
}
```

The statement within the loop may be a single line or a block of statements. If the statement within loop is a single line then the curly parenthesis is optional. Here loop will be executed repeatedly as long as the condition is true. Note that if condition always true then loop would be executed infinitely so after some execution condition becomes false.

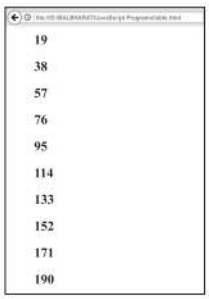
Example :

```
var i=1
while(i<=5)
{
  document.writeln(i);
i=i+1;
}
```

Program for loop

```
<!DOCTYPE html>
<head><title>Table-I</title>
<script type="text/javascript">
function display()
{
   var i,a;
   a=form1.t1.value
   for(i=1;i<=10;i++)
     {
       document.write(a*i + "<br/>");
     }
}
   </script></head>
<body>
<form name="form1">
Enter number to display table:-
<input type="text" name="t1">
 <input type="button" value=" Display
Table" onClick="display()">
</body>
</html>
```

Output :



Do it yourself

Find syntax of do.....while() loop and difference between while() and do... while() loop.

Break and continue statements

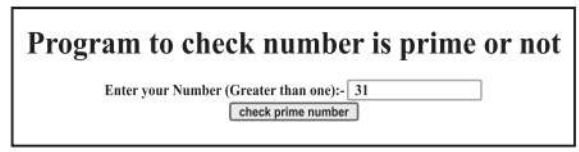
Break statement is used to jump out of loop. It is used to make an early exit from a loop. When keyword break is encountered inside the loop, control automatically passes to the next statement after the loop.

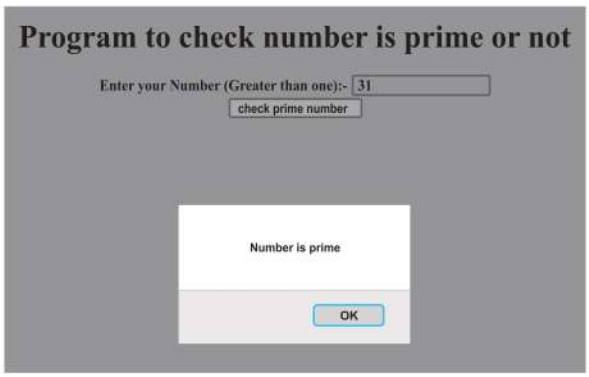
Sometimes in looping it may be necessary to skip statement block and take the control at the beginning for next iteration. This is done by using **'continue'** statement in JavaScript.

Program :

```
<!DOCTYPE html>
<html><head><title>Prime number</title>
<script type="text/javascript">
function display()
{
   var a,ans;
   a=parseInt(form1.t1.value);
   ans=1;
   for(i=2;i<a;i++)
     {
      if(a%i==0)
        {
         ans=0;
          break;
        }
     }
     if(ans==1)
       alert("Number is prime");
     else
       alert("Number is not prime");
}
</script></head>
<body>
<h1 align="center"> Program to check number is prime or not </h1>
<form name="form1" style="text-align:center">
Enter your Number (Greater than one):-<input type="text" name="t1"> <br>
  <input type="button" value="check Prime number"  onClick="display()">
</body></html>
```

Output :





3.3 Objects in JavaScript

JavaScript is an object based scripting language. Almost everything is an object in JavaScript. A JavaScript object is an entity having state (properties) and behavior (methods). An object can group data together with functions needed to manipulate it. Look around you, you will find many examples of real world objects. Such as table, board, television, bicycle, shop, bus, car, monitor etc. All these tangible things are known as objects. Take an example of car object. It has properties like name, model, weight, color etc. and methods like start, stop, brake etc. All cars have same properties but contain different values from car to car. All cars have same methods but perform differently.

Object	Properties	Methods
car	car.name=Ferrari	car.start()
	car.model=F430	car.drive()
	car. weight=1517kg	car.brake()
	car.color=red	car.stop()

Properties and methods of object's are accessed with '.' operator. JavaScript supports 2 types of objects built-in objects and user defined objects.

1. Built in objects such as Math, String, Array, Date etc.
2. JavaScript gives facility to create user defined objects as per user requirements. The 'new' keyword is used to create new object in JavaScript.

e.g.

```
d= new Date();
// 'd' is new instance created for Date object.
```

DOM (Document Object Model) :

When HTML document is loaded into a web browser, it becomes a document object. It defines logical structure of document. The way in which HTML document content is accessed and modified is called as Document Object Model. It is programming interface for HTML and XML (Extensible Markup Language) documents.

The standardization of DOM was founded by W3C (World Wide Web Consortium) which works for standardization of web technologies. According to W3C :

"The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."

Following diagram shows hierarchy of DOM object:

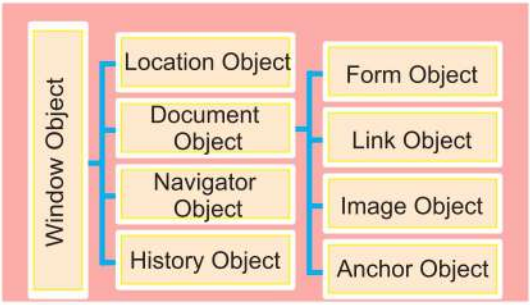


Fig. 3.1 Document Object Model

Following are some of the predefined methods and properties for DOM object.

Property	Description
head	Returns the <head> element of the document
title	Sets or returns title of the document.
URL	Returns full URL of the HTML document.
body, img	Returns <body>, <img> elements respectively.
Method	**Description**
write()	Writes HTML expressions or JavaScript code to a document.
writeln()	Same as write(), but adds a newline character after each statement.
getElementById()	There are many ways of accessing form elements, of which the easiest is by getElementById() method. In which id property is used to find an element.

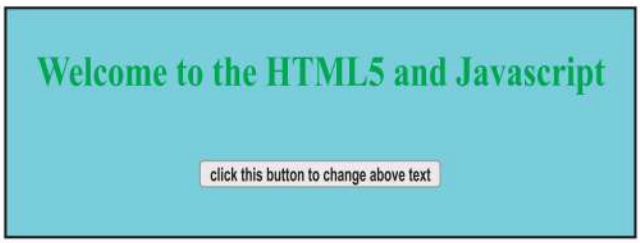
The innerHTML Property

The innerHTML property is useful for getting html element and changing its content. The innerHTML property can be used to get or change any HTML element, including <html> and <body>.

```
<!DOCTYPE html>
<html>
<head>
<script type="text/javascript">
function changeText3()
{
var style="<h2 style= 'color:green'>";
var text="Welcome to the HTML5 and Javascript";
 var closestyle="</h2>";
document.getElementById('para').
innerHTML =style+text+closestyle;
    }
    </script></head>
    <body style="background-color:cyan">
<h1 align="center">
<p id="para">Welcome to the site</p>
<input type="button"onclick=" "changeText3()" value="click this button to change above text">
</h1>
</body>
</html>
```

Output :

Before button click



After button click

Window Object :

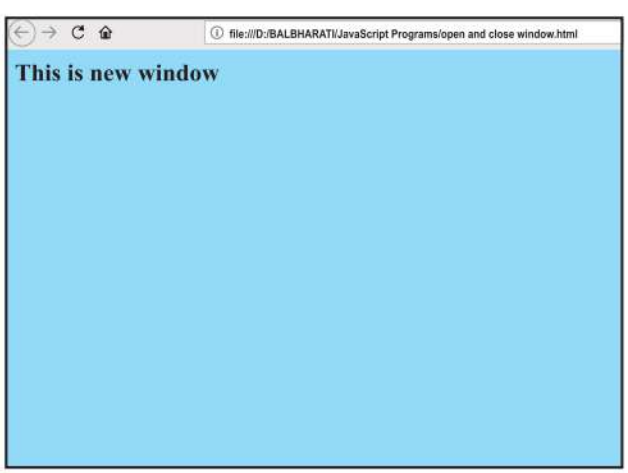
At the very top of the object hierarchy is the window object. Window object is parent object of all other objects. It represents an open window in a browser. An object of window is created automatically by the browser. Window object represents an open window in a browser. An object of window is created automatically by the browser. Following table shows some of the methods and properties for window object.

Property	**Description**
name	Sets or returns the name of a window.
location	Returns the Location object for the window.
document	Returns the Document object for the window.
status	Sets or returns the text in the status bar of a window.
closed	Returns a Boolean value indicating whether a window has been closed or not.
Method	**Description**
alert()	Displays the alert box containing message with ok button.
confirm()	Displays the confirm dialog box containing message with ok and cancel button.
prompt()	Displays a dialog box to get input from the user.
open()	Opens the new window.
close()	Closes the current window.
blur()	Removes focus from the current window.
focus()	Sets focus to the current window.
print()	Prints the content of current window.
setTimeout()	Calls a function or evaluates an expression after a specified number of milliseconds.

Program :

```
<!DOCTYPE html>
<html>
<head>
<title >Window Opener and Closer
</title>
<script type="text/javascript">
function makeNewWindow()
{
var newwin=window.open();
newwin.document.write("<h1>This is
new window</h1>");
newwin.document.body.style.
backgroundcolor="skyblue";
}
</script></head>
<body><form>
<input type="button" value="Create
New Window"
onClick="makeNewWindow()">
</form></body>
</html>
```

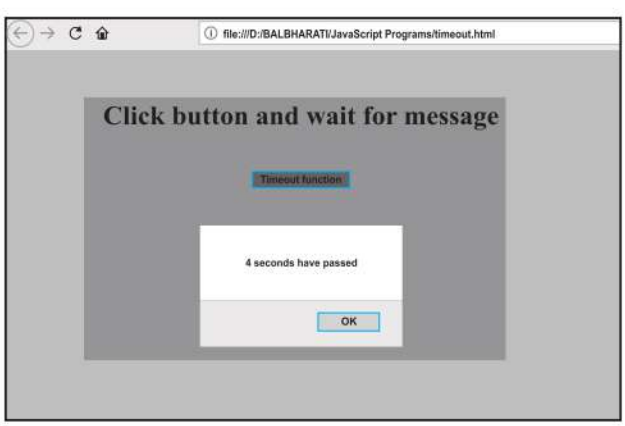
Output :



Program :

```
<!DOCTYPE html>
<html>
<head>
<script type="text/javascript">
function sampleFunction()
{
  window.setTimeout(next(), 4000);
}
function next()
{
    alert("4 seconds have passed");
  }
</script></head>
<body style="background-color:cyan">
<h1 align="center">
Click button and wait for message
</h1>
<input type="button" value="Timeout
function"
onClick="sampleFunction()">
</body>
</html>
```

Output :



Do it yourself

1. *Write JavaScript program to display status"Hi this is status property"in status bar of window.*
2. *Write JavaScript program to set three different background color to webpage on onClick, onMousemove and onMouseover events of button object.*

3.4 JavaScript Event

Events are actions done by the user or an application that occurs on the webpage. In previous year we studied different keyboard events (onKeypress, onKeydown, onkeyup) and mouse events (onClick, onMousemove, onMouseout, onMouseover). Similarly there are some more events used with form objects.

Input and other object Events:

Event handler	Description
onblur	It occurs when user leaves field or losses focus of an element.
onfocus	It occurs when an element gets focus.
onchange	It occurs when user changes content of an element or selects dropdown value. E.g. for textbox, password, select box, textarea etc.
onselect	It occurs when user selects some text of an element.
onsubmit	It occurs when user clicks submit button.
onreset	It occurs when user clicks reset button.
onload	It occurs when page/image has been loaded.
onunload	It occurs when document/page has been unloaded or closes.

3.5 JavaScript Built-in Objects

JavaScript has several built-in or core language objects. These built-in objects are available regardless of window content and operates independently of whatever page browser has loaded. These objects provide different properties and methods that are useful while creating live web pages.

String Object :

String is used to store zero or more characters of text within single or double quotes. String object is used to store and manipulate text.

Property	Description
length	Returns the number of characters in a string
Method	**Description**
charAt()	Returns the character at the specified position (in Number).
indexOf()	Returns the index of the first occurence of specified character in given string, or -1 if it never occurs, so with that index you can determine if the string contains the specified character.
lastIndexOf()	Returns the index of the last occurence of specified character in given string.
substr()	Returns the characters you specified: (14,7) returns 7 characters, from the 14th character.
substring()	Returns the characters you specified: (7,14) returns all characters between the 7th and the 14th.
trim()	The trim() method removes whitespace from both sides of a string
toLowerCase()	Converts a string to lower case
toUpperCase()	Converts a string to upper case

Example :

```
var str="Information Technology";
document.write ("length of string is :-" + str.length);
document.write ("Substring is :-" + str.substr (12,10));
```

Output :

```
Length of string is :-22
Substring is :- Technology
```

Math Object :

The built-in Math object includes mathematical constants and functions. You do not need to create the Math object before using it. Following table contains list of math object methods: e.g. var x=56.899; alert(Math.ceil(x));

Method	Description
abs(x)	Returns the absolute value of a number.
cbrt(x)	Returns the cube root of a number.
ceil(x)	Returns the next integer greater than or equal to a given number (rounding up).

Method	Description
floor(x)	Returns the next integer less than or equal to a given number (rounding down).
max(x, y, ...)	Returns the highest-valued number in a list of numbers.
min(x, y, ...)	Returns the lowest-valued number in a list of numbers.
pow(x, y)	Returns the base to the exponent power, that is, xy.
random(x)	Returns a random number between 0 and 1 (including 0, but not 1).
sqrt(x)	Returns the square root of a number.

Do it yourself

1. Write event driven JavaScript program to take number as user input and find its square root and cube root.

Date Object :

The date object is used to create date and time values. It is created using new keyword. There are different ways to create new date object.

var currentdate=new Date();

var currentdate=new Date(milliseconds);

var currentdate=new Date(dateString);

var currentdate=new Date(year, month, day, hours, minute, seconds, milliseconds);

Method	Description
getDate()	Returns the day of the month (from 1-31)
getDay()	Returns the day of the week (from 0-6)
getFullYear()	Returns the year (four digits).
getHours()	Returns the hour (from 0-23).
getMinutes()	Returns the minutes (from 0-59).
getMonth()	Returns the month (from 0-11).
getSeconds()	Returns the seconds (from 0-59).
getTime()	Returns the number of milliseconds since midnight Jan 1, 1970.
now()	Returns the number of milliseconds since midnight Jan 1, 1970.
setDate()	Sets the day of the month of a date object.
setFullYear()	Sets the full year of a date object.
setHours()	Sets the hours of a date object.
setMinutes()	Set the minutes of a date object.
setMonth()	Sets the month of a date object.
setSeconds()	Sets the seconds of a date object.
setTime()	Sets a date to a specified number of milliseconds after/before Jan 1, 1970.

Number Object :

It helps us to work with numbers. Primitive values (like 34 or 3.14) cannot have properties and methods, but with JavaScript it is available with primitive values.

Property	Description
MIN_VALUE	Returns the largest minimum value.
MAX_VALUE	Returns the largest maximum value.
NaN	It represents 'Not a Number' value.
Method	**Description**
isInteger()	It determines whether the given value is a Integer
parseFloat()	It converts the given string into a floating point number.
parseInt()	It converts the given string into a integer number.
isFixed()	It returns the string that represents a number with exact digits after a decimal point.

Array Object :

An array is an object that can store a collection of items. JavaScript arrays are used to store multiple values in single variable. An array is a special variable which can hold more than one value at a time. Arrays become really useful when you need to store large amounts of data of the same type. You can create an array in JavaScript as given below.

```
var fruits=["Mango","Apple","Orange","Grapes"];
        OR
var fruits=new Array("Mango","Apple","Orange","Grapes");
```

You can access and set the items in an array by referring to its indexnumber and the index of the first element of an array is zero. arrayname[0] is the first element, arrayname[1] is second element and so on.

e.g.
```
var fruitname=fruits[0];
document.getElementById("demo").inner HTML=fruits[1];
```

Property	Description
index	The property represents the zero-based index of the match in the string
length	Reflect number of elements in array.
Method	**Description**
concat()	Joins two or more arrays, and returns a copy of the joined arrays
copyWithin()	Copies array elements within the array, to and from specied positions.

Method	Description
find()	Returns the value of the first element in an array that satisfies a test in testing.
forEach()	Calls a function for each array element.
indexOf()	Search the array for an element and returns its position.
isArray()	Checks whether an object is an array.
pop()	Removes the last element of an array, and returns that element.
push()	Adds new elements to the end of an array, and returns the new length.
reverse()	Reverses the order of the elements in an array.
sort()	Sorts the elements of an array.

Program Using array object :

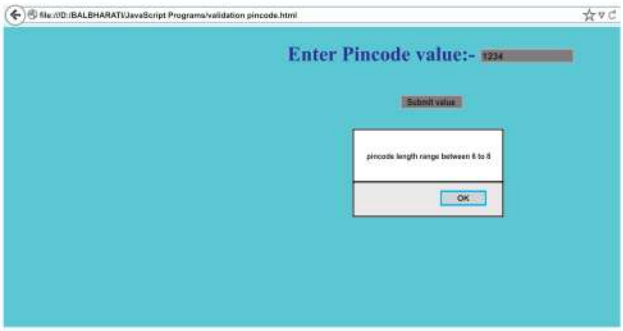
```
<!DOCTYPE html>
<html><head>
<title>use of array methods</title>
</head>
<body style="color:blue;background-color:pink;font-size:30px">
<script type="text/javascript">
var city = ['Pune', 'Kolhapur', 'Mumbai', 'Nashik', 'Latur', 'Nagpur'];
document.write("Original array elements are <br>");
document.write(city);
document.write("<br><br>Copy elements at end to the beginning <br>");
document.write(city.copyWithin(0, 3));
city = ['Pune', 'Kolhapur', 'Mumbai', 'Nashik', 'Latur', 'Nagpur'];
document.write("<br><br>Copy elements at the beginning to the end<br>");
document.write(city.copyWithin(3, 0));
city = ['Pune', 'Kolhapur', 'Mumbai', 'Nashik', 'Latur', 'Nagpur'];
document.write("<br><br>Copy first 3 elements to middle<br>");
document.write(city.copyWithin(2, 0, 3));
city = ['Pune', 'Kolhapur', 'Mumbai', 'Nashik', 'Latur', 'Nagpur'];
document.write("<br><br>Adding an element to an array<br>");
document.write(city.push('Kokan'));
document.write(city);
document.write("<br><br>Reversing an array element <br>");
document.write(city.reverse());
</script> </body></html>
```

Output :

```
Original array elements are
Pune,Kolhapur,Mumbai,Nashik,Latur,Nagpur

Copy elements at end to the beginning
Nashik,Latur,Nagpur,Nashik,Latur,Nagpur

Copy elements at the beginning to the end
Pune,Kolhapur,Mumbai,Pune,Kolhapur,Mumbai

Copy first 3 elements to middle
Pune,Kolhapur,Pune,Kolhapur,Mumbai,Nagpur

Addind an element to an array
7Pune,Kolhapur,Mumbai,Nashik,Latur,Nagpur,Kokan

Reversing an array element
Kokan,Nagpur,Latur,Nashik,Mumbai,Kolhapur,Pune
```

Validation program in JavaScript :

```
<!DOCTYPE html>
<html><head><title>Pincode Validation</title></head>
<body style="color:blue;background-color:cyan"><form name="form1">
<h1 align="center">
Enter Pincode value:-<input type="text" name="t1"><br><br>
<input type="button" value="Submit value" onClick="validate()"></h1>
<script type="text/javascript">
function validate()
{
 var pincode; pincode=form1.t1.value;
if(pincode.length==0)
   {
alert("please check, enter value");
     form1.t1.focus();
}
else if(Number.isInteger(pincode))
    {
alert("please, enter integer number only");
   form1.t1.focus();
}
else if(pincode.length<6||pincode.length>8)
 {
  alert("pincode length range between 6 to 8");
  form1.t1.focus();  }
else
alert("Pincode is accepted");
}
</script> </body></html>
```



Note : isInteger() is supported by version Mozilla Firefox 16 and higher.

Do it yourself

1. *Find more Math object methods useful for trigonometric functions.*
2. *Write JavaScript Program to create simple calculator using JavaScript Math object.*

Summary

- JavaScript is light weight scripting language. It is platform independent language.
- There are two types of scripts; client side script and server side scripts. Client side scripts reside on client machine and server side script resides on web server.
- JavaScript provide 'switch…case' as multi way decision statement.
- For....loop, while…loop and do…while are commonly used looping structures in JavaScript.
- DOM (Document Object Model) is a programming interface for HTML and XML documents. It defines logical structure of document.
- Window object is parent object of all other objects hence its methods can be used without specifying it.
- JavaScript is event based language support objects events such as onBlur, onFocus, onChange, onSelect, onSubmit, onLoad, onUnload, onResize etc.
- JavaScript supports built-In objects such as Date, String, Math, Number and array etc. These objects contain number of properties and methods that are useful while creating interacting web pages.

Exercise

Q.1 Fill in the blanks.

1. ----------script resides on server computer.
2. --------- statement is used to jump out of loop.
3. ---------- defines logical structure of document.
4. ---------- property of window object returns Boolean value indicating whether window is closed or not.
5. ---------- event occurs when an element looses its focus.

Q.2. State whether given statement is true or false.

1. JavaScript is case sensitive language.
2. Math.ceil() function is used to return the nearest integer less than or equal to given number.
3. MAX_VALUE property of number object returns smallest possible value.
4. getDay() method of Date object returns month in number.
5. onKeydown event occurs when user moves mouse pointer.

Q.3. Multiple choice questions. Select one correct answer.

1. JavaScript is ---------- language.

   a) Compiled

   b) Interpreted

   c) Both a and b

   d) None of the above

2. Select correct method name of String object----------.

   a) charAt() b) characterAt()
   c) valueAt() d) lengthAt()

3. ---------- method displays message box with Ok and Cancel button.

   a) Confirm() b) Alert()
   c) both a and b d) None of these

4. We can declare all types of variables using keyword ----------.

   a) var b) dim
   c) variable d) declare

5. Trace ouput of following JavaScript code.

   var str="Information Technology";

   document.write

   (str. lastIndexOf("o");

   a) 18 b) 19
   c) 20 d) 21

Q.4. Multiple choice questions. Select two correct answer.

1. Valid two methods of Date object are ---------- and ----------.

   a) setTime()

   b) getValidTime()

   c) getTime()

   d) setValidTime()

2. Properties of document object are ---------- and ----------.

a) URL b) title

c) name d)status

3. ---------- and ---------- are event / event handler used with text object in JavaScript.

a) onBlur b) onMove

c) onFocus d) onAction

Q.5 Multiple choice questions. Select three correct answers.

1. Select three correct methods of window object----------

a) write() b) alert()

c) writeln() d) close()

e) open() f) charAt()

2. JavaScript features are ---------- ---------- , ----------and ---------- ----------.

a) supports event based facilities

b) is platform dependent language

c) case insensitive scripting language

d) provide inbuilt objects

e) can handle date and time effectively

f) requires special software to run

3. Inbuilt objects in JavaScript are ----------, ---------- and ----------.

a) Time b) Date

c) Inheritance d) Array

e) Number f) function

Q.6 Explain the following.

1. What are similarities and differences between client side scripting and server side scripting.
2. Briefly explain features of JavaScript.
3. Explain switch…….....case conditional statement in JavaScript with example.

Q.7 Write event driven JavaScript program for the following.

1. Display Addition, multiplication, division and remainder of two numbers, which were accepted from user.
2. Display number sequence from 100 to 150 in following format.

   (100 101 102.....150)
3. Find and display factorial of given number.
4. Accept any string from user and count and display number of vowels occurs in it.

Q.7 Match the following.

A	B
ceil()	Writes HTML expression or javascript code to a document.
floor()	Sets focus to current window.
write()	Removes white spaces from both sides of string.
focus()	Returns next integer greater than or qual to to given number.
trim()	Returns the next integer less than or equal to given number.

❑❑❑