# JAVA SCRIPT

## INTRODUCTION JAVA SCRIPT:

Script means small piece of code. Java script you can easily create interactive web pages. It is designed to add interactivity to HTML pages. Scripting languages are two kinds one is client-side other one is servers-side scripting. In general client-side scripting is used for verifying simple validation at client side, server-side scripting is used for database verifications. VBScript, java script and J script are examples for client-side scripting and ASP,JSP, servlets etc.are examples of server-side scripting.

Web pages are two types

1. Static web page

2. Dynamic webpage

- ❖ Static web page where there is no specific interaction with the client
- ❖ Dynamic web page which is having interactions with client and as well as validations can be added.
  Simple HTML script is called static web page, if you add script to HTML page it is called dynamic page. Netscape navigator developed java script.
  Microsoft‟s version of java script is J script.
- ➢ Java script code as written between <script> ---- </script>tags
- ➢ All java script statements end with a semicolon
- ➢ Java script ignores white space
- ➢ Java script is case sensitive language
- ➢ Script program can save as either. Js or. html

**Similarities between java script and java:**

1. Both java script and java having same kind of operators

2. Java script uses similar control structures of java

3. Nowadays both are used as languages for use on internet.

4. Labeled break and labeled continue both are similar

**Difference between java script and java:**

1. Java is object –oriented programming language where as java script is object-based programming language.
2. Java is full –featured programming language but java script is not.
3. Java source code is first compiled and the client interprets the code. Java script code is not compiled, only interpreted.

4. Inheritance, polymorphism, threads are not available in JavaScript.
The syntax of the script tag is as follows:

<script language=“““scripting language name“““>

---

</script>

The language attribute specifies the scripting language used in the script. Both Microsoft internet explorer and Netscape navigator use java script as the default scripting language. The script tag may be placed in either the head or the body or the body of an HTML document.

Ex: <script language=“““java script“““>
---
---
---
</script>

Variable declaration in java script:

Variables are declared with the var key word

Var variable name=value

Ex: var x=20

The variable is preceded by the var, it is treated as local variable otherwise variable is treated as global variable.

web technologies

Comments in java script:

Single line comment- //

Multi-line comment- /**/

operators in java script:

Arithmetic operators
Relational operators
Logical operators
Assignment operator
Increment decrement operators
Conditional operator (ternary)
Bitwise operators

Control structures:

- If statement
- Switch
- While
- Do-while
- For
- Break
- Continue

Control structures syntax and working as same as java language.

**BASICS OF OBJECTS IN JAVA SCRIPT:**

Basic objects are document object and window object

**Document object:**

To display some information on the screen.

Syntax: document. Write („"any message"");

Document. Writeln („"any message"");

In this document is object name and write () or writeln () are methods. Symbol period is used as connector between object name and method name. The difference between these two methods is carriage form feed character that is new line character automatically added into the document. Write ln(),but it is not included in document. Write ().

Window objects:

Window object there are 3 types of popup boxes.

1. Alert box
2. Confirm box
3. Prompt box

1. Alert box is used error message to user

Syntax: window. Alert („"string message"");

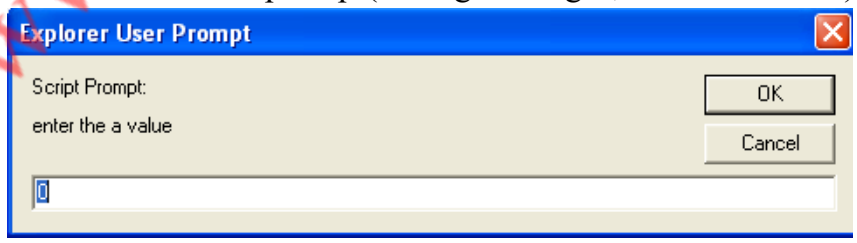

2. Conform syntax

Window.confirm("you want to save?")

(figure)

3. Prompt box for accepting data syntax

Variable=window.prompt("string message","default values");

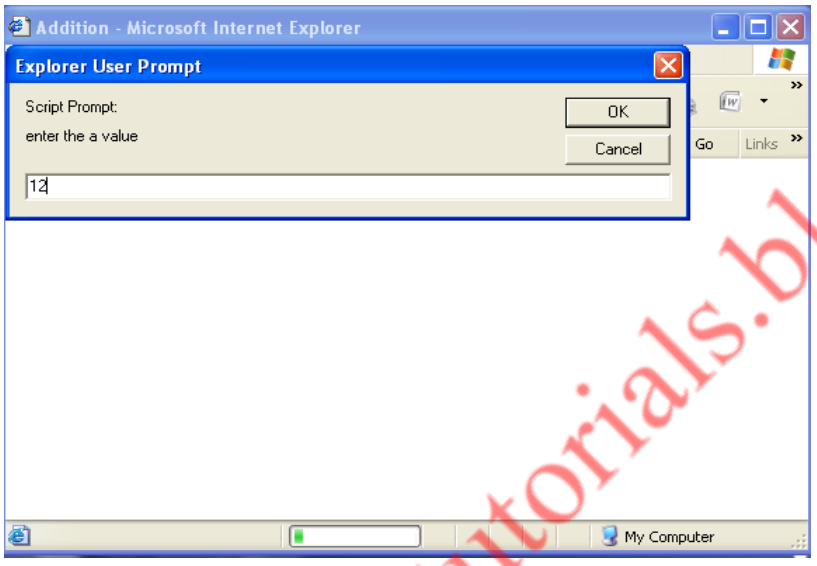


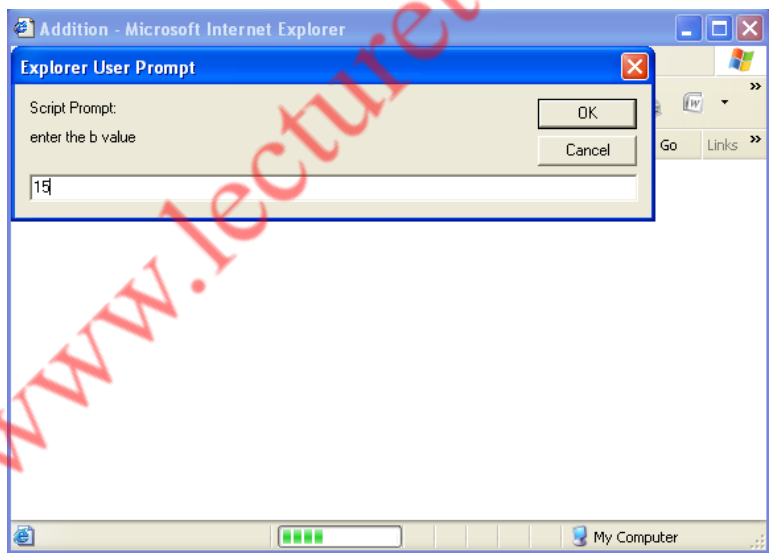

Example 1:

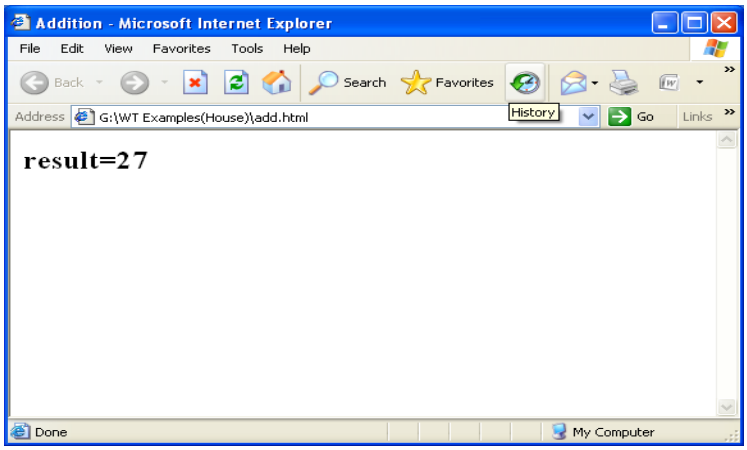
Write a java script addition of two numbers.

<html>

```
<head>
<title>addition</title>
<script language=”java script”>
Var s1,s2,a,b,c;
S1=window.prompt(“enter the a value”,”0”);
S2=window.prompt(“enter the b value”,”0”);
A=parseInt(s1);
b=parseInt(s2);
c=a+b;
document.write(“<h2>result=”+c+</h2>”);
</script>
</head>
</html>
```



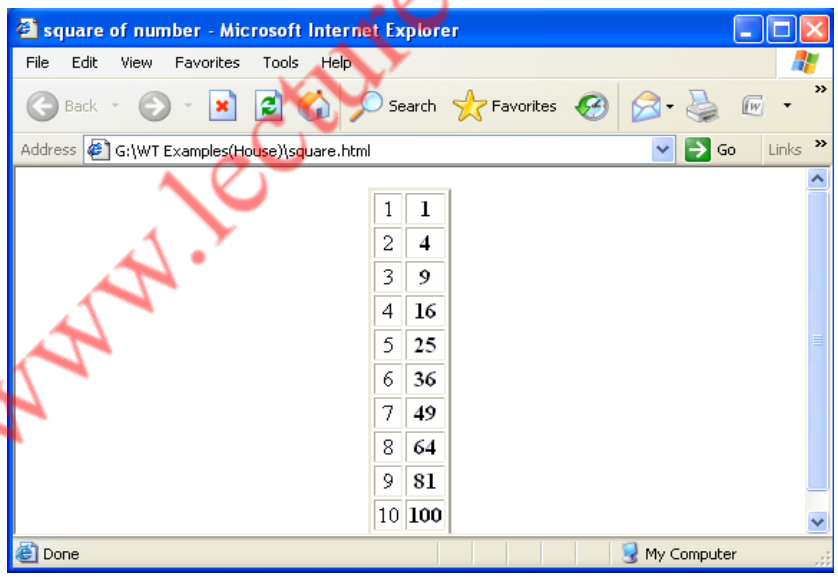






Example2:

Write a program to display 1 to 10 and square of number.

```
<html>
<head>
<title>square of number</title>
<script language=”java script”>
Var i=1,n;
Document .writeln(“<center><table border=1>”);
While(i<=10)
{
Document.writeln(“<tr><td>”+i+”<td>”+(i*i)+”</tr>”);
I++;
}
Document.writeln(“</table></center>”);
</script>
</head>
</html>
```




Example 3:

Write a program to display table for given number

```
<html>
<head>
```

```
<title>nth table </title>
<script language=”java script”>
Var i=i,n;
n=parseInt(window.prompt(“enter n value”,”0”));
Document .writeln(“<center><table border=1>”);
For(i=1;i<=10;i++)
{
Document.writeln(“<tr><td>”+n+”<td>X<td>”+i+”<td>=”+(n*i)+”</tr>”);

}
Document.writeln(“</table></center>”);
</script>
</head>
</html>
```

(figure)

**EVENTS:**

ONCLICK event:

ONCLICK event is used for responding for mouse click action.when the user presses a button, it can call any function through ONCLICK event. For example:

```
<INPUT TYPE=”BUTTON”VALUE=”OK”ONCLICK=”fun()”>
```

When the button OK is clicked it calls the functioin display and according to action given in fun() function it will work.

Example6:

Write a program to display a button OK and display your name when it is clicked.

```
<html>
<head>
<title> onclick example</title>
<script language=”java script”>
Function fun()
{
Window .alert(“hai,this is Raju”);
}
</script>
</head>
<body>
<form>
<INPUT TYPE=”BUTTON” VALUE=”OK“ ONCLICK=”fun()”>
</form>
</body>
```

</html>

**onsubmit event:**

Example 7:

```
<html>
<head>
<title> onsubmit example</title>
script language=”java script”>
Function fun()
{
Window .alert(“hai,this is raju”);
}
</script>
</head>
<body>
<form name =”f1” onsubmit=”fun()”>
<INPUT TYPE=”BUTTON” VALUE=”submit” >
</form>
</body>
</html>
```

**onload event:**

```
<html>
<head>
<title> onload example</title>
<script language=”java script”>
Function fun()
{
Window .alert(“hai,this is ramesh”);
}
</script>
</head>
<body onload=”fun()”>
</body>
</html>
```

**Accepting values from text box:**

In javascript we can refer the entered values from the textbox which is defined in HTML forms. By taking the name parameter we can perform this task. Simply we can give as follows

Variable=document. Formname.textboxname.value;

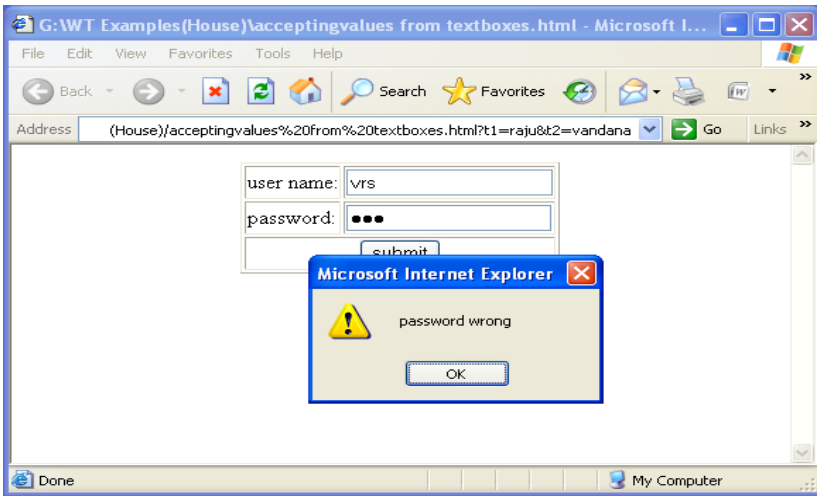
For example, form name is “form1” and text box name is “user” then you can assign its values in java script variable as follows:

Str=document .form1.user.value;

Example 9:

Write a program to login form

```
<html>
<head>
<script language="javascript">
function fun()
{
var s1="vrs";
var s2="yrn";
var s3,s4;
s3=document.f1.t1.value;
s4=document.f1.t2.value;
if((s1==s3)&&(s2==s4))
window.alert("user name and password correct");
else if(!(s1==s3)&&(s2==s4))
window.alert("user name wrong");
else
if((s1==s3)&&!(s2==s4))
window.alert("password wrong");
else
if(!(s1==s3)&&!(s2==s4))
window.alert("both user name and password wrong");
}
</script>
</head>
<body>
<center>
<form name="f1" onsubmit="fun()">
<table border="1">
<tr><td>user name:<td><input type="text" name="t1" size=20></tr>
<tr><td>password:<td><input type="password" name="t2" size=20></tr>
<tr><td colspan=2 align="center"><input type="submit" value="submit"></tr>
</table>
</form>
</center>
</body>
</html>
```

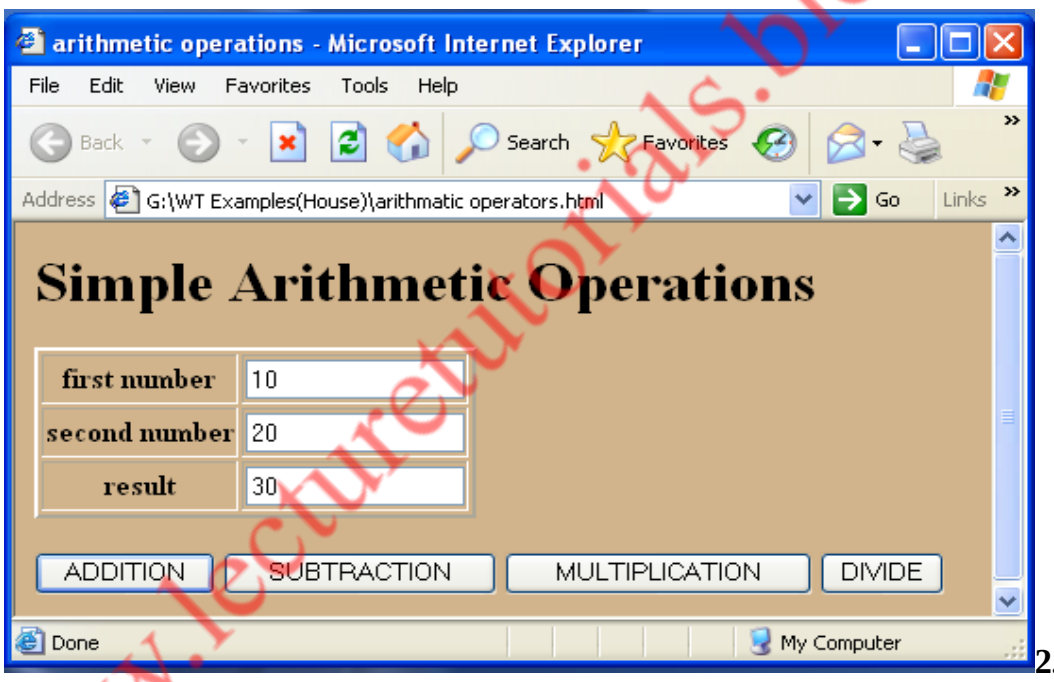



Example 10:

Write a program to display a form with three text fields,two for accepting numbers and thirdone for displaying result.form should contain four buttons with labels ADD,SUBTRACT,MULTIPLY and DIVIDE .perform the respective arthimetic operations and display the result in the third text box.

```
<html>
<head>
<title>arithmetic operations</title>
<script language ="javascript">
function cal(op)
{
var a,b,c=0;
a=parseInt(document.form1.first.value);
b=parseInt(document.form1.second.value);
switch(op)
{
case '+':c=a+b;break;
case '-':c=a-b;break;
case '*':c=a*b;break;
case '/':c=a/b;break;
}
document.form1.result.value=c;
}
</script>
</head>
<body bgcolor=tan>
<form name="form1">
<h1>Simple Arithmetic Operations</h1>
<TABLE border='2'>
<TR>
<TH> first number
<TD><input type="text" name="first" size=15>
```

```
</TR>
<TR>
<TH>second number
<TD><input type="text" name="second" size=15>
</TR>
<TR>
<TH>result
<TD><input type="text" name="result" size=15>
</TR>
</TABLE>
<p>
<INPUT TYPE =BUTTON VALUE="ADDITION" ONCLICK=cal("+")>
<INPUT TYPE =BUTTON VALUE="SUBTRACTION" ONCLICK=cal("-")>
<INPUT TYPE =BUTTON VALUE="MULTIPLICATION"
ONCLICK=cal("*")>
<INPUT TYPE =BUTTON VALUE="DIVIDE" ONCLICK=cal("/")>
</body>
</html>
```




**2.6. RECURSION IN JAVASCRIPT:**

A recursive function is a function that calls itself either directly or on directly through another function.

The following recursion example in java script shows the evaluation for the factorial of n. a recursive definition of the factorial function is arrived by observation that is N!=N*(N-1)!

Example 11:

Write a program to display the factorial of a given number.

```
<html>
<head>
<title>recursion for factorial</title>
<script language="javascript">
  var a =window.prompt("enter number for factorial:","0");
var num=parseInt(a);
document.writeln("factorial of "+num+ "is" +factorial(num));
function factorial(num)
{
if(num<=1)
return 1;
else
return num*factorial(num-1);
}
</script>
</head>
</html>
```

ITERATION: if a function or structure executes by following a repetition structure mechanism it is called iterative process. It terminates when a loop continuation condition fails.

Common features of recursion and iteration:

- Both are based on a control structure.
- Both involve repetition.
- Both involve a termination test.
- Both can occur infinitely.

**2.6.1. Recursion VS iteration:**

| S.NO | RECURSION | ITERATION |
|---|---|---|
| 1 | Recursion uses selection structure(such as if, if/else or switch) | Iteration uses a repetition structure(such as for, while or do-while) |
| 2 | Recursion achieves repetition through repeated function calls | Iteration explicitily uses the repetition structure. |
| 3 | Recursion terminates when a base case is recognized | Iteration terminates when the loop continuation condition fails. |
| 4. | Recursion keeps producing simpler versions of original problem until | Iteration keeps modifying the counter until the counter assumes |

| | | |
|---|---|---|
| | the base case is reached | a value that makes the loop continuation condition fail. |
| 5. | Infinite recursion occurs if the recursion step does not reduce the problem each time. | An infinite loop occurs if the loop continuation test never becomes false. |
| 6. | Recursion can be expensive in both processor time and memory space. | Iteration is less expensive in both processor time and memory space. |
| 7 | Recursion consumes considerable memory | Iteration consumes considerable memory |

Example 12:

Write java script that three integers from the user and outputs their sum,average,largest.use alert dialog box to display results.

Problem in detail:

Use four different functions for accepting data, finding sum, average caluculation and to find largest among them.every time display the data along with output. Use four buttons to call the respective functions.

```
<html>
<head>
<title> numbers</title>
<script language=”javascript”>
Var a,b,c,data;
Var aNUM,bNUM,cNUM;
Function acc()
{
A=window.prompt(“enter first number”);
B=window.prompt(“enter second number”);
C=window.prompt(“enter third number”);
aNUM=parseInt(a);
bNUM=parseInt(b);
cNUM=parseInt(c);
data=a+” ,”+b+” ,”+c;
}
Function sum()
{
Result=aNUM+bNUM+cNUM;
Window.alert(“sum of “+data+”numbers is “+result);
}
Function avg()
{
Avg=(aNUM+bNUM+cNUM)/3;
Window.alert(“average of “+data+”number is “+avg);
```

```
}
Function largest()
{
Larg=Math.max(math.max(a.NUM,bNUM),cNUM);
Window.alert("largest of "+data+"numbers "+larg);
}
</script>
</head>
<body bg color="tan" text="black">
<form>
<table border="2">
<caption><h3>sum,average and largest</h3></caption>
<colgroup>
<col span="2"ALIGN="right">
</colgroup>
<tr><th>to accept numbers
<td><input type="button" value="accept" onclick="acc()">
</tr>
<tr><th> to sum of those numbers
<td><input type="button" value="sum" onclick="sum()"></tr>
<tr><th>to get average of those numbers
<td><input type="button" value="average" onclick="average()"></tr>
<tr><th>to get largest among those numbers
<td><input type="button" value="largest" onclick="largest()"></tr>
</table>
</form>
</body>
</html>
```

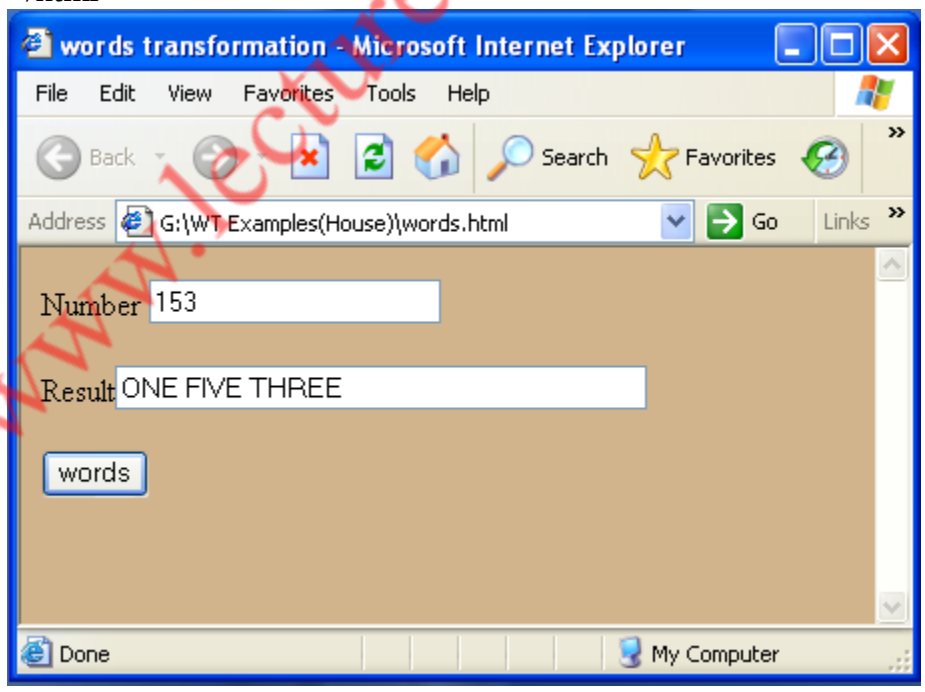
(figure)

Example 13:

Write a program for word equivalent of given number.

```
<html>
<head>
<title>words transformation</title>
<script language="javascript">
function dispwords()
{
var n=0,k=0,i=0,temp=0,r=" ";
var ar=new Array();
n=parseInt(document.f1.num.value);
temp=n;
while(temp>0)
{
k=temp%10;
ar[i++]=words(k);
temp=Math.floor(temp/10);
}
ar.reverse();
```

```
r=ar.join(" ");
document.f1.res.value=r;
}
function words(k)
{
switch(k)
{
case 0:return"ZERO";
case 1:return"ONE";
case 2:return"TWO";
case 3:return"THREE";
case 4:return"FOUR";
case 5:return"FIVE";
case 6:return"SIX";
case 7:return"SEVEN";
case 8:return"EIGHT";
case 9:return"NINE";
}
}
</script>
</head>
<body bgcolor=TAN>
<form name=f1>
Number <input type=text name=num>
<p>
Result<input type=text size=40 name=res>
<p>
<input type=button value="words" onclick="dispwords()">
</form>
</body>
</html>
```




**Functions in Java script:**

Function is a piece of code that performs specific talk.

Functions are two types 1. Library Functions2.User defined Functions

Library Functions in Java script:

1. **Eval:** This function takes a string representing java script code to execute. The interpreter evaluates the code and executes dynamically.
2. **isFinite:** This function takes a numeric value and returns true if the argument results a finite numeric. Otherwise it returns false.
3. **isNaN:** This function takes a numeric argume nt and returns true if the argument is not a number otherwise returns false.
4. **parseFloat:** It accepts a string as argument and converts into its equivalent float value. If the conversion fails then a value NaN is returned.
5. **parseInt:** It accepts a string as argument and converts into its equivalent numeric. If the conversion fails then a value NaN is returned.

Example:

Write a program to accept a number from user and display whether it is Armstrong or not. Before testing for Armstrong, check user has entered number or not.

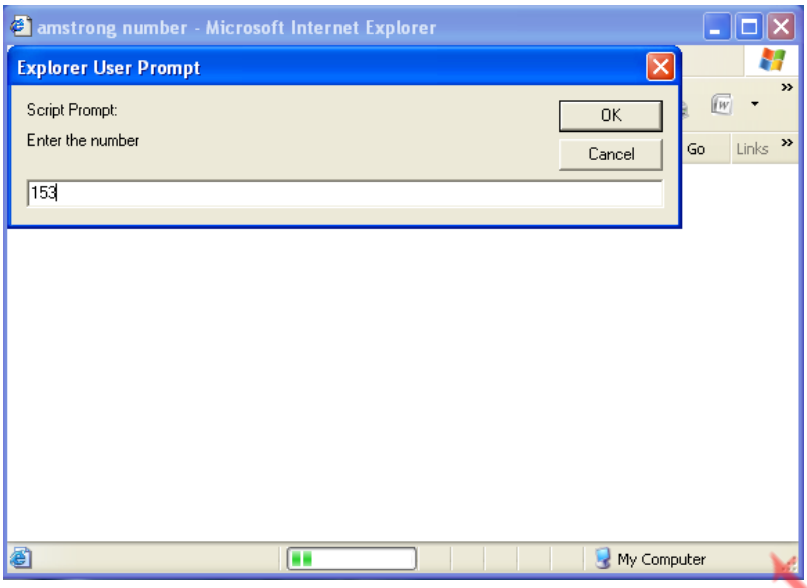
For Example n=153

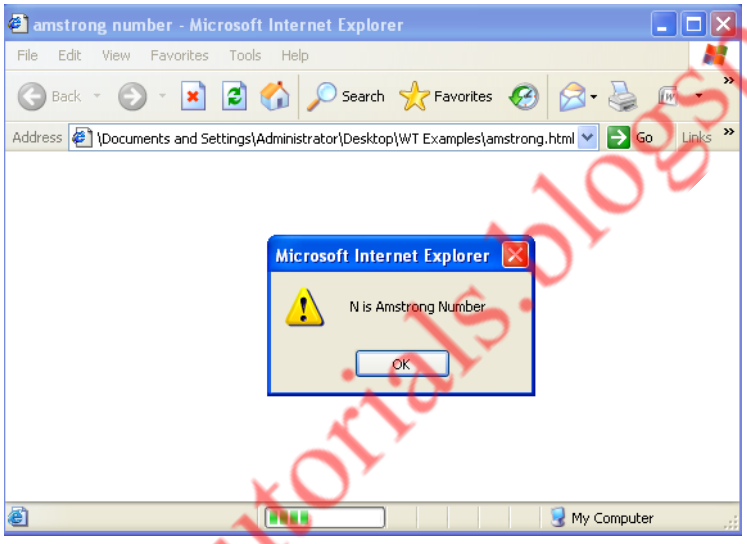
$1^3+5^3+3^3=1+125+27=153=x$

x=n Armstrong

x!=n not Armstrong

```
<html>
<head>
<title>amstrong number</title>
<script language="javascript">
var n,r,x,sum=0;
do
{
 n=parseInt(window.prompt("Enter the number","0"));
 }
while(isNaN(n));
x=n;
while(n!=0)
{
 r=n%10;
 sum=sum+r*r*r;
 n=parseInt(n/10);
}
if(x==sum)
        window.alert("N is Amstrong Number")
        else
        window.alert("N is Not amstrong Number");
</script>
</head>
</html>
```




**User-Defined Functions:**

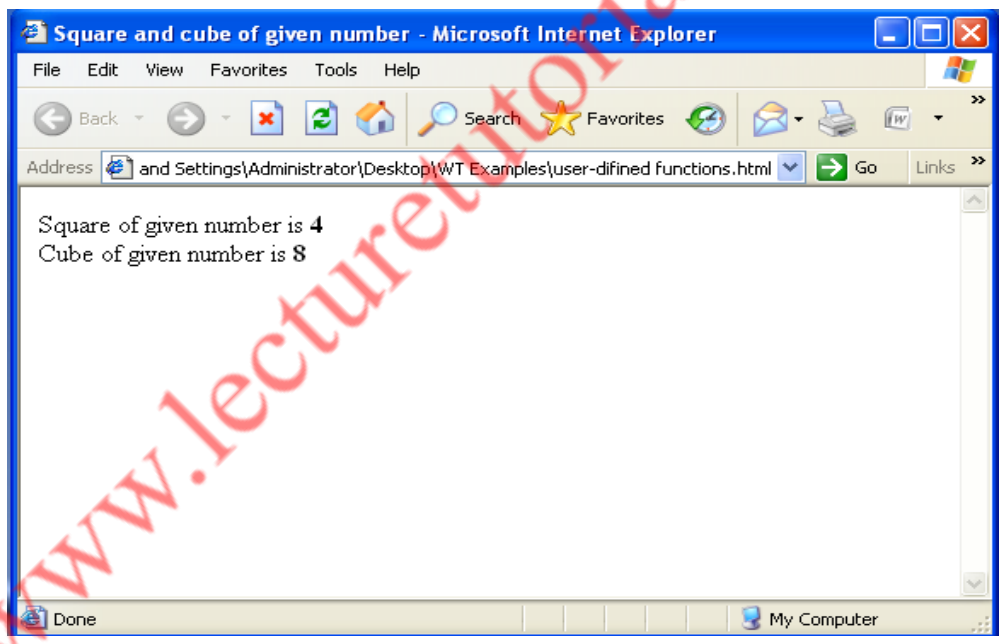
Divide and conquer can be achieved in Javascript by using functions. Functions that you create start with the command "function" and are followed by the name for the function. A function can be defined as follows:

```
Function function-name(parameter-list)
{
        Declerations;
                Statements;
        ----------
        ----------
        return[expression];
}
```

Example:

Write a program to display square and cube of a given number by using user-defined functions.

```
<html>
<head>
        <title>Square and cube of given number</title>
        <script language="JavaScript">
        var a;
        a=parseInt(window.prompt("Enter number","0"));
        document.writeln("Square of given number is
<b>"+square(a)+"</b><br>");
        document.writeln("Cube of given number is <b>"+cube(a)+"</b>");
        function square(k)
        {
           return k*k;
        }
        function cube(k)
        {
           return k*k*k;
        }
        </script>
</head>
</html>
```




## Arrays:

Array is a collection of similar type of elements which can be referred by a common name. Any element in an array is referred by an array name followed by [position of the element]. The particular position of element in an array is called array index or subscript.

Arrays are two types1. Single dimensional arrays

2. Multi dimensional arrays

**Single dimensional array:**

Array Declaration: "new" operator is used to declare and allocate memory for array. Operator new is also known as dynamic memory allocation operator.

Syntax: var variablename=new array(size);

Ex: var a=new array(10);

*Initialization of array elements:*

If you want to initialize the array elements with zeros then you can use for loop for that purpose, observe the following example:

```
var num=new array(10);
for(i=0;i<num.length;i++)
{
   num[i]=0;
}
```

The same thing can be achieved through for/in control structure, that enables to process each element in an array.

For example,

```
var num =new array(10);
for(var i in num)
{
     num[i]=0;
}
```

The above statements show the way of usage for/in control structure.

**Two Dimensional Array:**

For example we want required 3 rows 2 cols.

First create 3 rows

```
var a =new array(3);
```

| a[0] |
|---|
| a[1] |
| a[2] |

Then create 2 cols each row

```
for(i=0;i<3;i++)
{
   a[i]=new array(2);
}
```

| a[0][0] | a[0][1] |
|---|---|
| a[1][0] | a[1][1] |
| a[2][0] | a[2][1] |

**Objects in JavaScript:-**

An object is a collection of variables and functions(methods).

Syntax: object.method()

There are several objects in javascript programming languages

| Name of the Object | Purpose of the Object |
|---|---|
| Document | To get complete control on document along with displaying contents |
| Window | To get dialog control and provision for accepting data from user |
| Math | Used for mathematical functions and standard constants in mathematics |
| String | String manipulation can be done and also include to generate HTML |
| Date | For date manipulations, mainly to get current date and time |
| Number | To get number constants |
| Boolean | Wrapper class for Boolean type |

**Methods in MATH Object:-**

| Method | Meaning | Example |
|---|---|---|
| Math.abs(x) | Returns the absolute value | Math.abs(-20) is 20 |
| Math.ceil(x) | Returns the ceil value | Math.ceil(5.8) is 6<br>Math.ceil(2.2) is 3 |
| Math.floor(x) | Returns the floor value | Math.floor(5.8) is 5<br>Math.floor(2.2) is 2 |
| Math.round(x) | Returns the round value, nearest integer value | Math.round(5.8) is 6<br>Math.round(2.2) is 2 |
| Math.trunc(x) | Removes the decimal places it returns only integer value | Math.trunc(5.8) is 5<br>Math.trunc(2.2) is 2 |
| Math.max(x,y) | Returns the maximum value | Math.max(2,3) is 3<br>Math.max(5,2) is 5 |
| Math.min(x,y) | Returns the minimum value | Math.min(2,3) is 2<br>Math.min(5,2) is 2 |
| Math.sqrt(x) | Returns the square root of x | Math.sqrt(4) is 2 |
| Math.pow(a,b) | This method will compute the $a^b$ | Math.pow(2,4) is 16 |
| Math.sin(x) | Returns the sine value of x | Math.sin(0.0) is 0.0 |
| Math.cos(x) | Returns cosine value of x | Math.cos(0.0) is 1.0 |
| Math.tan(x) | Returns tangent value of x | Math.tan(0.0) is 0 |
| Math.exp(x) | Returns exponential value i.e $e^x$ | Math.exp(0) is 1 |
| Math.random(x) | Generates a random number in beween 0 and 1 | Math.random() |
| Math.log(x) | Display logarithmic value | Math.log(2.7) is 1 |

**String Object:-**

A String is a collection of characters; these may be including any kind of special characters, digits, normal characters and other characters.Strings may be written with help of single quotation or double quotation marks.

Methods of STRING Object:-

| Method | Meaning | Example |
|---|---|---|
| toLowerCase() | It converts the given string into lowercase | var s=”VRSYRN”<br>s.toLowerCase()<br>output: vrsyrn |
| toUpperCase() | It converts the given string into upper case | var s=” vrsyrn”<br>s.toLowerCase()<br>output: VRSYRN |
| substring(n) | It returns starting of n th position | Var s=” vrsyrn”<br>s.substring(4)<br>ouput: rn |
| substring(m,n) | It returns starting from mth character up to nth character, not include nth character | var s=” vrsyrn”<br>s.substring(0,4)<br>ouput: vrsy |
| Substr(m,n) | It return starting from mth character upto n characters | var s=” vrsyrn”<br>s.substr(1,3)<br>ouput: rsy |
| charAt(index) | It gives the ith character of string | Var s=”vrsyrn”<br>s.charAt(2)<br>output: s |
| charCodeAt(index) | It returns ASCII value of the character present at the given index | Var s=”teamwork”<br>s.charCodeAt(2)<br>output:97 |
| indexOf(character) | It gives the position of first occurrence of x to a given string | Var s=”teamwork”<br>s.indexOf(„m“)<br>output:3 |
| indexOf(character,n) | It gives the position of x that occurs after nth position in the string | Var s=”teamwork”<br>s.indexOf(„m“,2)<br>output:3 |
| contact(string) | It returns concatenation of s1and s2. Java script strings can be concatenated using „+“ operator | Var s1=”vrs”<br>Var s2=”yrn”<br>s1.concat(s2)<br>output:vrsyrn |
| split(string) | It specifies the split string from the given string | Var s=”vrs,yrn”<br>s.split(„„“)<br>output: vrs ,yrn |

**Date object:**

This object is used for obtaining the date and time. This date and time value is based on computer“s local time(system“s time).first we have to create a new date object.for this purpose we are using the new operator with out any operator(to current date): var d1=new Date()

With date information:var d2=new Date(yyyy,mm,dd)
With date and time information: var d3=new d3=new
Date(yyyy,mm,dd,hh,mm,ss,millis)
Methods in date object:

Java script date object provides several methods, they can be classified is string form, get methods and set methods. All these methods are provided in the following table.

| METHOD | MEANING |
|---|---|
| getDate() | Returns 1 to 31 ,day of the month |
| Get Day() | Returns 0 to 6, Sunday to Saturday respectively. |
| getMonth() | Returns 0 to 11, jan to dec respectively. |
| getFullYear() | Returns four-digit year number |
| getHours() | Returns 0 to 23 |
| get Minitues() | Returns 0 to 59 |
| Getseconds() | Returns 0 to 59 |
| setDate(v) | To set the date,day,month and full year |
| setDay(v) | |
| Setmonth(v) | |
| Setfullyear(Y,m,d) | |
| Sethours(v) | To set the hours,minutes,seconds,time |
| Setminutes(v) | |
| Setseconds(v) | |
| Settime(v) | |

**Boolean object:**

Wrapper object for Boolean data type is Boolean object. To manipulate Boolean data types in java script program these objects are provided. Constructor to create Boolean object is

Var n=new Boolean(Boolean value)

For example,

Var n=new Boolean(true);

It accepts parameters, true, false, 0, 1 number.NaN or empty string. If it empty string or numbers.NaN it treats as false values.

Methods in Boolean object:
Tostring()-to convert to string for true or false values.
Valueof()-to get value of Boolean object.
2.7.5. NUMBER OBJECT:
Object wrapper in JavaScript is number object, for any number data type, this act as wrapper class. Same methods are applied for number object also. Constructor is as follows:
Var n=new number (any value)
For example
Var n=number(432,123);
Properties of number object:
Number.MAX_VALUE.number.MIN_VALUE,
number.NaN,number.NEGITIVE_INFINITY and number.POSITIVE_INFINITY.
Number.NaN is for not a number.
Object:
Window.open(): syntax:window.open(file name,name,properties)
Window.close(): syntax:window.close()
**Properties:**
Directories -yes or no
Height -number of pixels
Width -number of pixels.
Location -yes or no
Menubar -yes or no
Resizable -yes or no
Scrollobars -yes or no
Status -yes or no
Toolbar -yes or no
Always lowered -yes or no
Alwaysraised -yes or no
Dependent -yes or no
Hotkeys -yes or no

Example:
open("ram.html","window","width=300,height=300,status=no,toolbar=no,menubar=no");
**2.8. DYNAMIC HTML:**
2.8.1 What is the difference between HTML and DHTML

| HTML | DHTML |
|---|---|
| HTML is used to create static web pages. | DHTML is used to create dynamic web pages. |
| Html is consists of simple html tags. | DHTML is made up to html |

| | |
|---|---|
| | tags+cascading style sheets+javascript. |
| Creation of html web pages is simplest but less interactive. | Creation of DHTML is complex but more interactive. |

**EVENT HANDLING:**

vents are triggers that call one of your function. An event could be action Such as clicking on a button or placing your mouse over an image. for example we will use the onclick event for starting our form validation scripts, and the on mouse overevent for creating graphics images that change when you place your cursor over them.

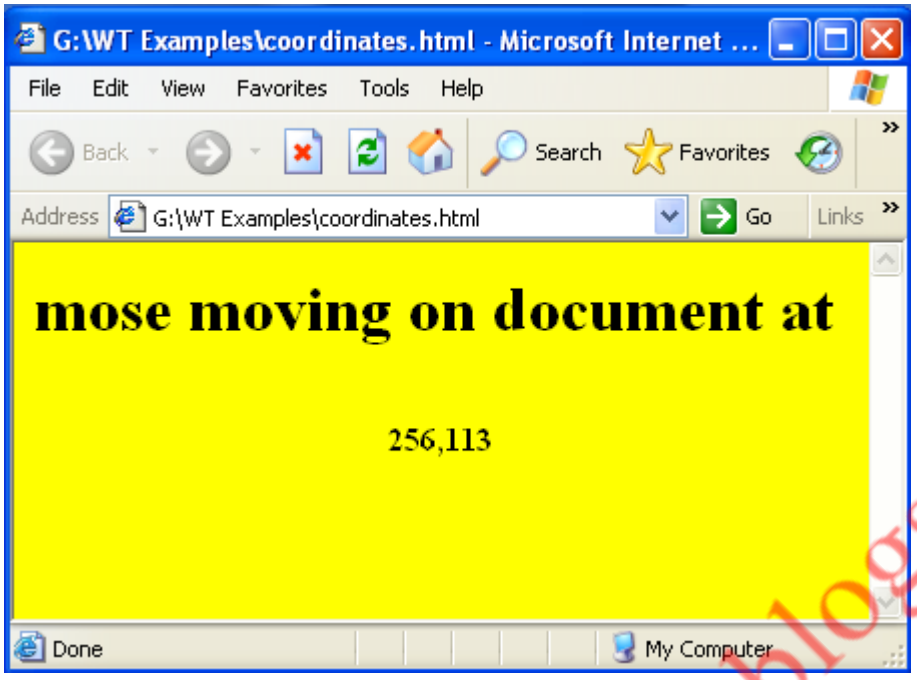
**Advantages of events:**

1. You can create user –interactive forms.
2. Validate the forms immediately when elements lost their focus
3. Display messages when some components receive focus
4. You can handle errors
5. Process the forms with help of submit button
6. You can write events when objects are selected
7. Also you can handle resize of windows or frames
8. Scripts can respond to user inter actions
9. Change the page according i.e. add dynamism to the page
10. It makes web applications more responsive and user-friendly]

**Mouse Events:**

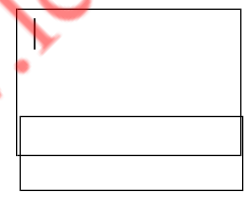
```
<HTML>
<head>
<script language="javascript">
function disp()
{
 c.innerText=event.offsetX+","+event.offsetY;
}
</script>
</head>
<body bgcolor="yellow" onmousemove="disp()">
<h1>mose moving on document at</h1>
<center>
<br>
```

```
<b id=c></b>
</center>
</body>
</html>
```

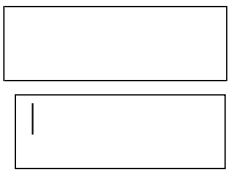



**FOCUSING EVENTS:**

Focus is nothing but the position of the cursor where it is placed .In forms, focus is moved between various elements on the form. For example, you have simple form as follows:



In the first field cursor is blinking, if you press tab key then the cursor is moved to second field and they appears as follows:



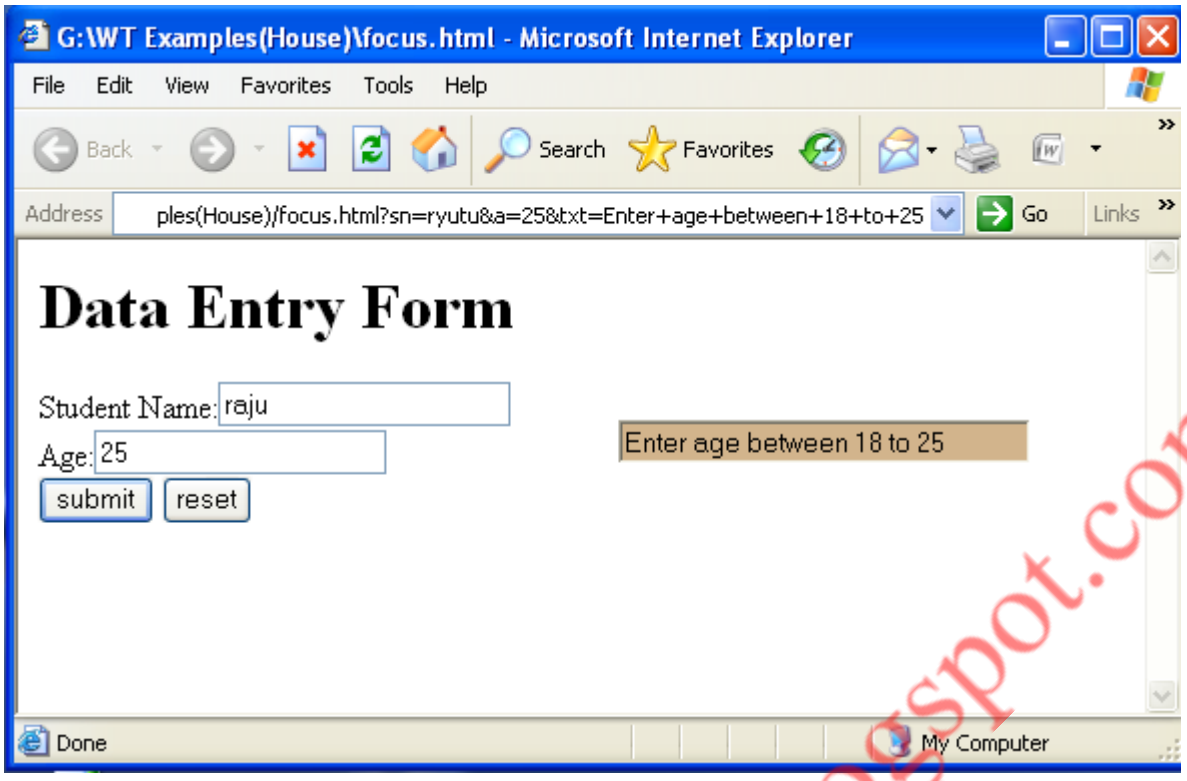
That means first field is loosing its focus .second field is gaining its focus. The events that fired during this are ONFOCUS and ONBLUR events. ONFOCUS is fired when the element gained focus and ONBLUR event is fired when

the element looses its focus. These events are very important if you want to handle the form elements. In general ONFOCUS event is used if user wants to display some messages when some particular graphical element receives focus. ONBLUR is used to verify given condition is satisfied after the user entered some data and pressed tab button. Following program is used to explain the concept of ONFOCUS and ONBLUR events.

**Example 16:**

Write a program to display a form that accepts student name, age, father name. When age field receives its focus display message that age should be below 18 to 25. After losing its focus from age field verify user entered in between given values or not display respective message.

```
<html>
<head>
<script language="javascript">
function disp()
{
 document.f1.txt.value="Enter age between 18 to 25";
}
</script>
</head>
<body>
<form name="f1">
<h1>Data Entry Form</h1>
Student Name:<input type="text" name="sn">
<br>
Age:<input type="text" name="a" onfocus="disp()">
<br>
<input type="submit" value="submit">
<input type="reset" value="reset">
<input type=text name="txt" size=30 onkeyup="clear()"
style="position:absolute;left:300;top:90;background-color:tan;color:black;">
</form>
</body>
 </html>
```




**Key events:**

Sometimes you want to check keys that user pressed and according to that you want add dynamism to your web page. Java script provides many key events; with the help of them you can easily get these effects. Important
Key events that are supported by java script are

1. ONKEYDOWN
2. ONKEYPRESS
3. ONKEYUP

Key up event is used to restrict the user entry to a text field. Now the one more example is here. Numerical text fields, these fields accept only numerical values in to restrict the user to enter only numerical values into a text then he should depend on the key events. When a key is pressed, he should verify that the key is numeric or not. If is numeric display the value into the text field otherwise restrict that key.
For this we used the logic as follows

```
If (s.length=1)
{
        If (event.keycode<48|event. Keycode>58)
        Document. Form 1. Txt. Value=s. substring (0, s. length-1)
}
Else
        If (event. Keycode<49|event. Keycode>58)
         Document. Form . txt. Value=``””;
```

First time when user presses other then number, then we clear the text field, after entering some numeric values and if he presses any alphabet we display up to

previous length that is only numeric value. This can be achieved through parseInt() instead of substring().

**Form processing and on change event:**

ONSUBMIT is invoked when the user submits the form. ONRESET is invoked when the user resets the form. If you want to cancel default action of the event then you can use window. Event. Return value=false.

ONCHANGE event is invoked when the user changed a value in the SELECT element, or text changed in a text field, it can be called as follows.

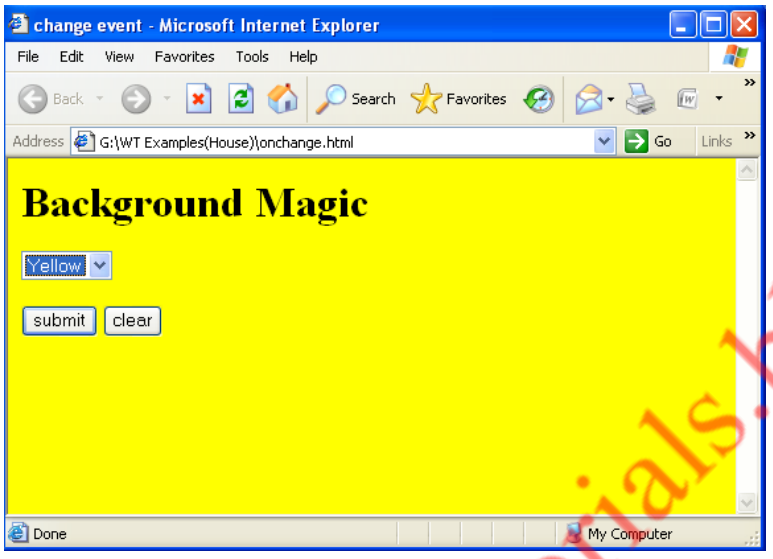
<SELECTID=``sel“““ onchange=change colar()“““>

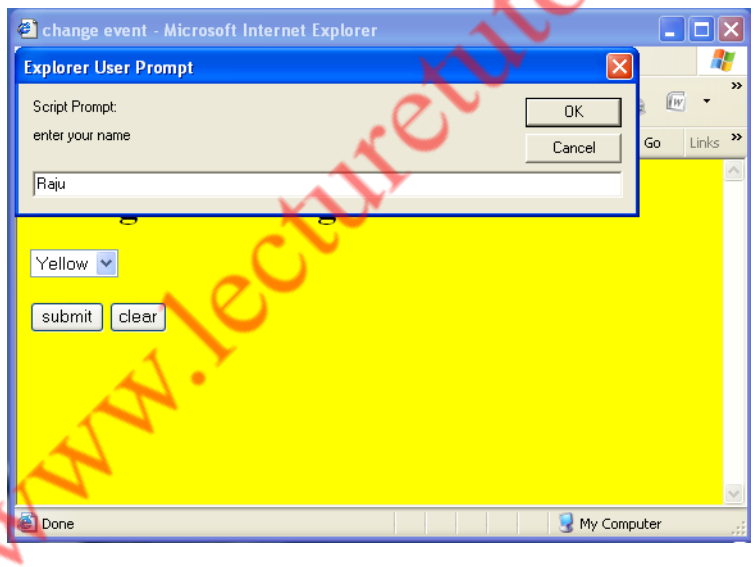
If the options in the combo box are changed then this change color() method is called is called.

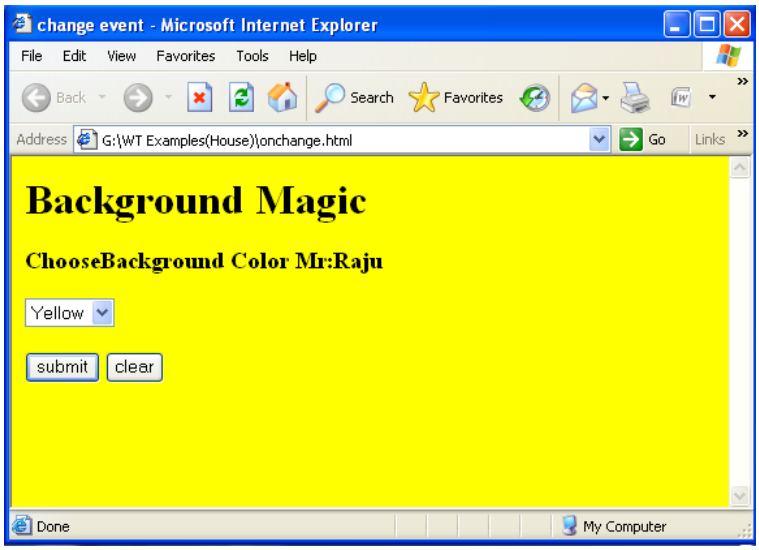
Example 17:write a program to change the background color when the user selected one of the colors from the combo box. When the user press submit button accept user name and display it.if he clicks on rest button display an alert message that he pressed resetting option.

```
<html>
<head>
<title>change of colors</title>
<script language=”java script”>
Function change color()
{
Document.body.style.backgroundcolor=document.form1.se1.value;
}
Function change text()
{
Var nam=window.prompt(“enter your name”,”name”);
s.inner text=”choose background color Mr/Ms :”+nam;
window.event.return value=false;
}
function say bye()
{
Window.alert(“now you are resetting “);
}
</script>
</head>
<boy bgcolor=tan>
<h1>BACKGROUND MAGIC</h1>
<form name=form1 onsubmit=change Text()  onreset=”saybye()”>
<h3 id=s>choose background color</h3>
```

```
<select id=”se1” onchange=”changecolor()”>
<option value=”red”>Red
<option value=”blue”>Blue
<option value=”yellow”>Yellow
<option value=”green”>Green
</select>
<input type=submit value=”submit”>
<input type=reset value=”reset”>
</form>
</body>
</html>
```






**2.9.5 Timer methods:**

HTML elements can be modified according to code. JavaScript is having setInterval()method in window object that is used for running a function according to given time interval. For example window.setInterval("disp()",1000),the disp() method calls for every 1000 milliseconds. There is another method setTiout(),the specified function called after waiting number of milliseconds specified along with setTimeout() and clearInterval().,these are used for stopping the timeout timer or interval timer. These functions are needed because window.setInterval()program will run continuously ,to stop it we need clearInterval()method.

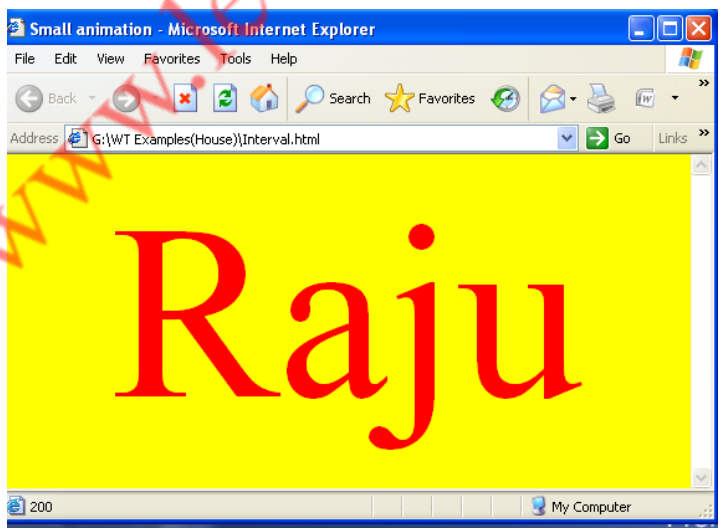
These two methods are used as follows.
To start timer
T1.window.setInterval("disp()",100);
To stop Timer
Window.clearInterval(t1);;

Example 18:
write a program to display your name by increasing font sizes according to time intervals.

```
<html>
<head>
<title>Small animation</title>
<script language="javascript">
var t1;
var c=0;
function start()
{
        t1=window.setInterval("incr()",100);
}
```

```
function incr()
{
        c=c+10;
        t.style.fontSize=c;
        window.status=c;
        if(c==200)
                window.clearInterval(t1);
        //t.style.color=c;

}
</script>
</head>
<body bgcolor="yellow" onload="start()">
<center>
<p id="t" style="color:red">Raju</p>
</body>
</html>
```


**2.10. COLLECTIONS:**

Arrays of related object on a page can be called as collections.dynamic HTML object model is contains various collections.
Some of the collections are all, children, frames etc.
Collection all"contains all the HTML elements of the current web page.as all is an array of elements ,it contains the property length to get the number of tags on the web page;other property of this collection is tag name that is used to get the specified tag name. to display all the tag names of the current page,just run a for loop and add individual tag to a string.

There is a property inner text for html element;it refers to the text contained in that element.property inner HTML is similar to inner Text, but it can include HTML formatting. The outerHTML property is similar to innerHTML property;it includes the enclosing HTML tags as well as the content inside them.

**SUMMMARY OF COLLECTIONS:**

| Collection name | usage |
|---|---|
| all | All elements on the web page |
| children | Children elements for individual element |
| anchors | Collection<A name>tags |
| links | Collection<AHREF>tags |
| applets | Collection pf <APPLET>tags |
| embeds | Collection of <EMBED>tags |
| images | Collection of <IMG>tags |
| forms | Collection of FORMS |
| frames | Representing each frame |
| scripts | Collection contains all the <script>tags |
| stylesheets | Represent each style element |

**FILTERS AND TRANSITIONS:**
Horizontal and vertical filters:

Fundamental filters supported by dynamic HTML are flipv and fliph popularly known as flip filters. These filters create mirror effects for the images or text. These are vertical in nature if filters is flipv and horizontal in the case of fliph. Simple effects are as follows:

In dynamic HTML creation of filter is very simple.only thing you require is style sheets. In styles, you have a special style that is filter style,which is used for all the

kinds of filters.the value passed to filter is fliph for horizontal flip and flipv for vertical flip. If you require both filters just apply both of them.

Example:

Style=”filter:fliph”for horizontal flip

Style=”filter:flipV” for vertical flip

Style =”filter:fliph flipv” for both flips.

Example 19:

Write a program to display vertical flip for ramesh

```
<html>
<body bgcolor=tan>
<table>
<tr><th  size=40px;filter:flipv”>Ramesh</tr>
</table>
</body>
</html>
```

Example 20:

Write a DHTML and java script program to accept username and display the given name along with four buttons that contains horizontal flip,vertical flip, both flips and normal buttons with actions.

```
<html>
<head>
<script language="javascript">
var k;
function change(k)
{
var k;
        switch(k)
        {
                case 1:a.style.filter='fliph';break;
                case 2:a.style.filter='flipv';break;
                case 3:a.style.filter='fliph flipv';break;
                case 4:a.style.filter='';
        }
}
</script>
</head>
<body onload="a.innerText=window.prompt('Your Name')">
<form name="f1">
```

```
<p id="a" align="center" style="background-color:tan;font-size:80px;">Your
Name</p>
<input type=button onClick="change(1)" value="horizontal">
<input type=button onclick="change(2)" value="vertical">
<input type=button onclick="change(3)" value="Horizontal&vertical">
<input type=button onclick="change(4)" value="Normal">
</form>
</body>
</html>
```

## filtering colors:chroma filter:

One of the option available in graphics software is you can turn off the color that you don‟t want. For image enhancements these kinds of options are useful. One such kind of filter is chroma filter. Without using any graphics software you can achieve these effects on your web page with the help chroma filter. Dynamically transparent effects can be obtained.

In style argument if you pass filter:chroma, then chroma filter starts its processing. By using dynamic features, if you can any type of graphical user components to apply this feature. By using identification of an image you can call filters collection and change its properties are follows.

chImg. Filters(„‟chroma‟‟‟). Color=color code;

example 21:

chroma an image by using paint brush with some black and white colors. Write dynamic html program to apply black and white filters to that image. (used white and black color for understanding purpose)

```
<HTML>
<HEAD>
<TITLE>chroma black and write filter</TITLE>
<SCRIPTLANGUAGE=‟‟‟JAVASCRIPT‟‟‟>
Function sopcolor(c)
{
If (c){
Chimg. Filters(„‟chroma‟‟‟). Color=parseint(c,16);
Chimg.filters(``chroma‟‟‟). Endable=true;
}
Else
Chimg. Filters(``chroma‟‟‟). Enabled=false;
}
</SCRIPT>
</HEAD>
<BODY BGCOLOR=TAN>
<H1>CHROMA FILTER:</H1>
<img id=‟‟‟chimg‟‟‟sic=‟‟‟cir. Jpg‟‟‟style=‟‟‟filter:chroma‟‟‟>
<FORM>
```

```
<INPUT TYPE=BUTTON VALUE=""ALL
COLORS""onclick=stopcolor(0)>
<INPUT TYPE=BUTTON VALUE=""BLACK""
Onclick=stopcolor(000000")>
<INPUT TYPE=BUTTON VALUE=""WHITE"" onclick=stopcplor(FFFFFF")>
</FORM>
</BIDY>
</HTML>
```

## Masking:

Masking effect is obtained by using mask filters. These are created as image masks, with the help of these filters you can get an effect of background image colors applied on the specified text of foreground. Image mask means adding text to image with image colors. Foreground text is in transparent color so that background image can be displayed on the text.

These effects can be achieved in DHTML by using statements.

```
<h1 style=``position: absolute: top:95 ; left:280;filter:mask(color=#000000)"">
```

Followed by an image behind this header so that you can get the mask feel.

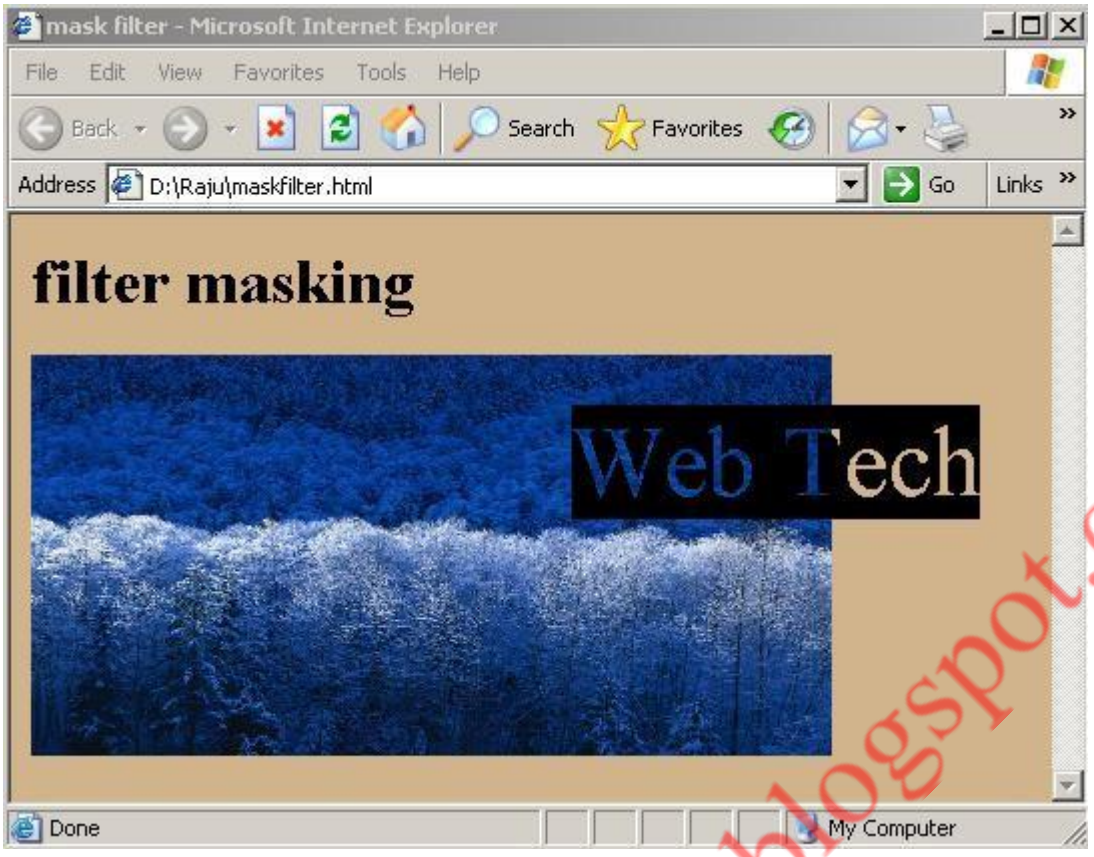
```
<img src=``anyimg. Jpg""width=""400""height=``200"">
```

Better to get both are placed at one location.

Example 22:

Write a DHTML program to display a text on given image. That text should start form the image and moved outwards of the image, so that background color of webpage should apply to text.

```
<html>
<head>
<title>mask filter</title>
</head>
<body bgcolor=tan>
<h1>filter masking</h1>
<div style="position:absolute;top:95;left:280;filter:mask(color= #000000)">
<p style="font-size:50px">Web Tech</div>
<img src="D:\Raju\WT\winter.jpg" width=400 height=200>
</body>
</html>
```




**Other filters:**

Graphics software or photo editors contain an important feature for image enhancement. Those features are applying some negative image or gray scale effects to the picture. Automatically by clicking one of the menu options. DHTML also provides this feature by using filters. There are three important filters of that kind.

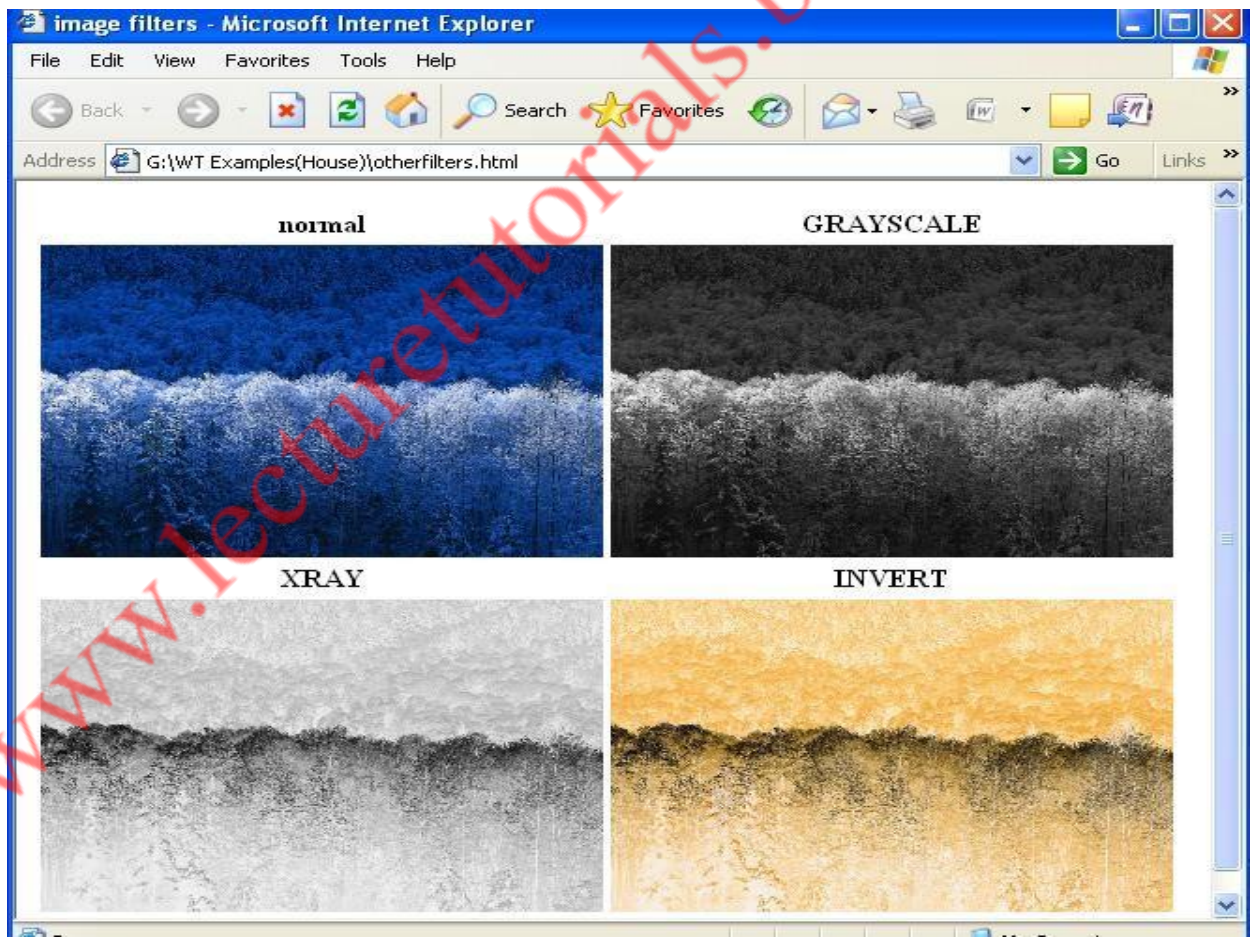
1. **Invert filter:** this applies a negative image effect to the given image, which means that drak areas became light and area became dark.
   Example: <img src=”could06.jpg” style=”filter: invert”>
2. **Gray filter:** this generates a grayscale image effect; in that all colors are stripped from the image and all that remains in brightness data.
   Example:<img src=”could06.jpg” style=”filter:gray”>
3. **X-ray filter:** this generates inversion of grayscale image effect that is called as x-ray effect.
   Example:<img src=”could06.jpg” style=”filter:xray”>

Example 23:

Write a program that displays grayscale,invert and x-ray effects for the given image.

<html>

```
<head>
<title>image filters</title>
</head>
<body>
<table>
<tr><th>normal<th>GRAYSCALE</tr>
<tr>
<td><img src="C:\Documents and Settings\All Users\Documents\My Pictures\Sample Pictures\Winter.jpg" width=300 height=200></td>
<td><img src="C:\Documents and Settings\All Users\Documents\My Pictures\Sample Pictures\Winter.jpg" width=300 height=200 style="filter:gray"></td></tr>
<tr><th>XRAY <th>INVERT</tr>
<tr>
<td><img src="C:\Documents and Settings\All Users\Documents\My Pictures\Sample Pictures\Winter.jpg" width=300 height=200 style="filter:xray"></td>
<td><img src="C:\Documents and Settings\All Users\Documents\My Pictures\Sample Pictures\Winter.jpg" width=300 height=200 style="filter:invert"></td>
</tr>
</table>
</body>
</html>
```




**Shadow effects:**

Now a day‟s normal word packages are also providing facilities to add shadows to your text. For example a message “welcome” with shadow effect observes here.

But this slight difference we can‟t observe, but if this shadow is larger then we can easily identify it.to add some depth to your text then you can apply shadow filter to your text. A three –dimensional appearance can be observed to your text if you apply shadow effects shadow filters in DHTML takes two important parameters. One is direction that specifies in which direction you want to display your shadow and other parameters is color.

Tells the color of the shadow. For example, for letter I that shadow appears left side as follows.

As exactly may be you cannot get with simple shadoweffects in DHTML, but you can decided which color you want and which directionthat shadow can be displayed you can fix it.

<p style=”color:blue; font-size:75;position:absolute;

Filter:shadow(direction=50,color=black)”>ramesh

Then the shadow of text ramesh appears 50 degrees, means above right side and display in black color.actual text color is blue.

The shadow directions are 0-top, 45-above right,90-right,135-below right,180-below,225-below left,270-left,315-above left.

Example 24:

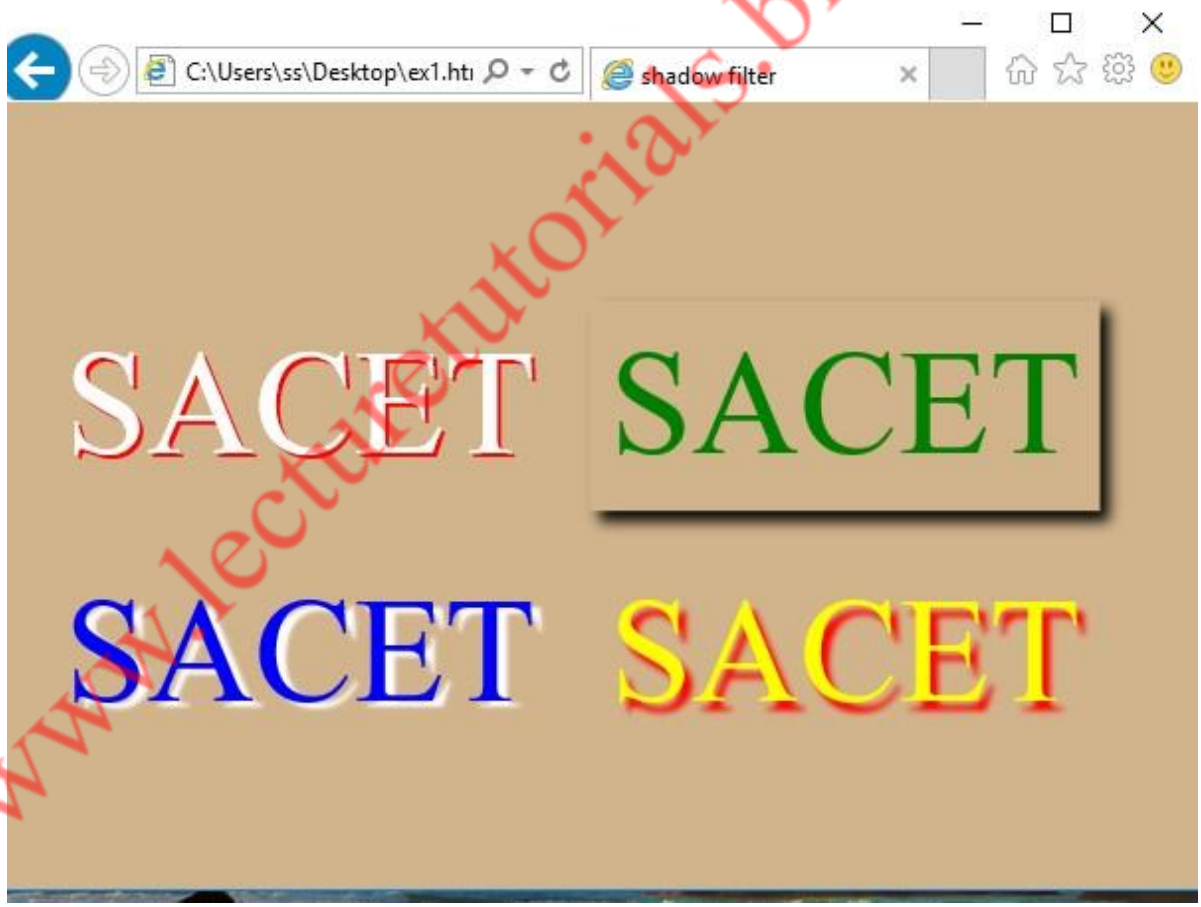
Write a DHTML program that displays shadow for a text.

```
<HTML>
<head>
<title>shadow filter</title>
</head>
<body bgcolor=tan>
<p style="color:white;font-
size:75;position:absolute;top:25;left:25;padding:10;filter:shadow(direction=315,color
=red)">SACET
<p style="color:green;font-
size:75;position:absolute;top:25;left:300;padding:10;filter:shadow(direction=150,colo
r=black)">SACET
<p style="color:blue;font-
size:75;position:absolute;top:150;left:25;padding:10;filter:shadow(direction=150,colo
r=black)">SACET
<p style="color:yellow;font-
size:75;position:absolute;top:150;left:300;padding:10;filter:shadow(direction=150,col
or=blue)">SACET
</body>
</html>
```

**Note: But the above code will be supported for the lower versions of the browsers; higher version of the browsers will support the following code**

```
<HTML>
<head>
<title>shadow filter</title>
</head>
<body bgcolor=tan>
<p style="color:white;font-size:75;position:absolute;top:25;left:25;padding:10;text-shadow:2px 2px 0 red">SACET</p>
<p style="color:green;font-size:75;position:absolute;top:25;left:300;padding:10;box-shadow: 5px 5px 10 black">SACET</p>
<p style="color:blue;font-size:75;position:absolute;top:150;left:25;padding:10;text-shadow:5px 3px 2 white">SACET</p>
<p style="color:yellow;font-size:75;position:absolute;top:150;left:300;padding:10;text-shadow:4px 5px 3 red">SACET</p>

</body>
</html>
```



## Gradient effects

One of the beautiful effect of image graphics is to display an image starting with a color and ending with the complete picture. This can be called as gradient effects. The alpha filter is used to create the above said gradient effect. Alpha filter is as follows

Filter: alpha (style=2, opacity=100, finishopacity=0)”>

It takes three important parameters indicates how the gradient transforms, there are four options are available for this.

0-uniform
1-linear
2-circular
3-rectangular

Other two parameters are opacity indicates the percentage at what percent at what opacity the specified gradient starts and other one finish opacity indicates where it finishes

Pic.filters(“alpha”)opacity=0;

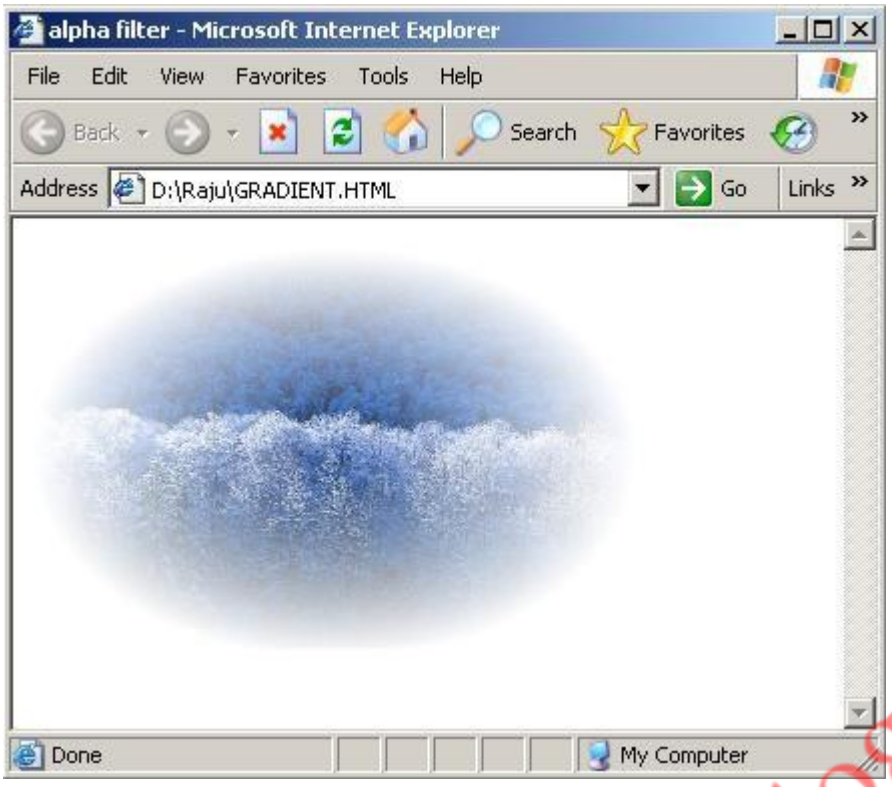
Pic.filters(“alpha”).finish opacity=100;

Indicates that first initial percent of opacity starts with 0 and finished it after achieving complete opacity.

Example 25:

Write a simple program that explains the concept of alpha gradient.

```
<html>
<head>
<title>alpha filter</title>
</head>
<body>
<div id="pic"
style="position:absolute;filter:alpha(style=2,opacity=100,finishopacity=0)">
<img width=300 height=200 src="D:\Raju\WT\winter.jpg">
</div>
</body>
</html>
```




**Note: Present versions of the browsers supports following syntax**

## All Filters

A demonstration of all filter functions:

```
.blur {
   -webkit-filter: blur(4px);
   filter: blur(4px);
}

.brightness {
   -webkit-filter: brightness(0.30);
   filter: brightness(0.30);
}

.contrast {
   -webkit-filter: contrast(180%);
   filter: contrast(180%);
}

.grayscale {
   -webkit-filter: grayscale(100%);
   filter: grayscale(100%);
}

.huerotate {
   -webkit-filter: hue-rotate(180deg);
   filter: hue-rotate(180deg);
}
```

```
.invert {
    -webkit-filter: invert(100%);
    filter: invert(100%);
}

.opacity {
    -webkit-filter: opacity(50%);
    filter: opacity(50%);
}

.saturate {
    -webkit-filter: saturate(7);
    filter: saturate(7);
}

.sepia {
    -webkit-filter: sepia(100%);
    filter: sepia(100%);
}

.shadow {
    -webkit-filter: drop-shadow(8px 8px 10px green);
    filter: drop-shadow(8px 8px 10px green);
}
```