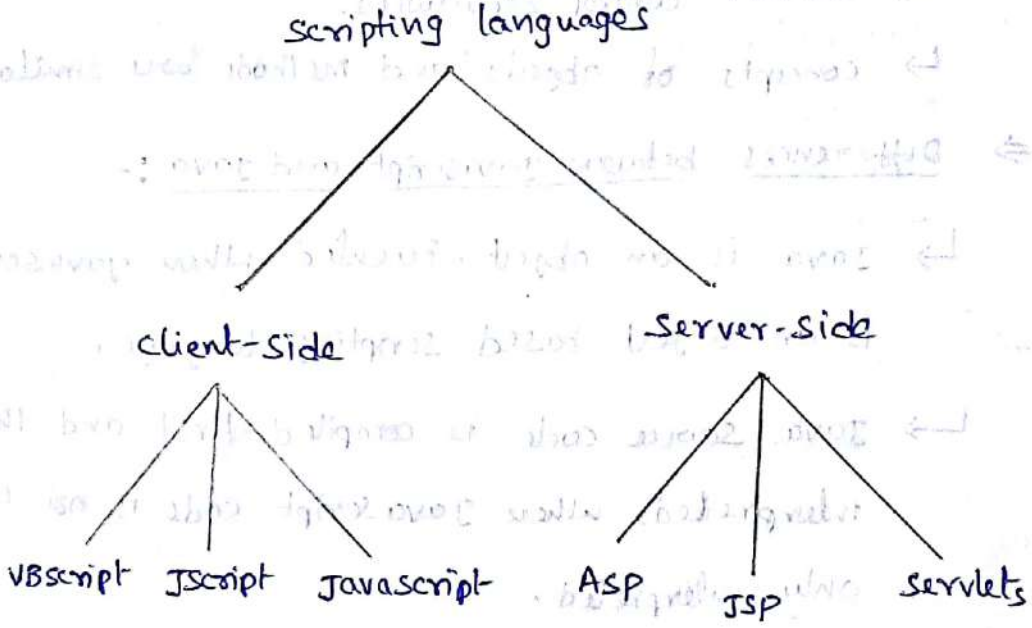
Java Script Languages

* Scripting Languages :-

These are specialized programming languages, which are used enhance functionality and appearance of webpages

These are two types

- client-side scripting language used for simple validations at client-side.
- Server-side scripting language used for database validation.



⇒ <script> tag :-

This is used to includes the script into html document.

format :-

```
<script language = " name of script ">
- - -
</script>
```

⇒ The scripting languages are used to modify document's content dynamically.

⇒ Javascript is a client-side scripting language.

* Introduction to Javascript:-

Javascript is an object-based scripting language, which is designed to enhance functionality of webpages that are developed with HTML.

Javascript is a client-side scripting language.

⇒ Similarities between Javascript and Java:-

↳ Both have same kind of operators.

↳ Similar control structures.

↳ concepts of objects and methods are similar.

⇒ Differences between Javascript and Java:-

↳ Java is an object-oriented where Javascript is an object-based scripting language.

↳ Java source code is compiled first and then interpreted, where Javascript code is not compiled only interpreted.

Note:- In the object-based programming language, we can use pre-defined objects only.

The object-oriented programming language supports to create new objects and to use the objects.

⇒ Benifits of Javascript:-

↳ Widely supported in web browsers

↳ web surfaces don't need a special plug-in to use your script.

↳ It gives the easy access to document object and can most of them.

* Variables :-

A variable is a named location that is used to store any value for that particular program.

Rules for naming the variable :-

→ Names must begin with a letter or digit or underscore.
→ spaces are not allowed in between variable name.
→ Names are case-sensitive.
→ Reserved word won't use as variable name.
⇒ All variables can be declared by using one keyword i.e "var".

Ex:- var a;
var sum = 10;

* Datatypes :-

Javascript supports following datatypes

↳ Numeric
↳ strings
↳ Boolean
↳ null

Ex:- var a = true; → boolean
var num = 10; → numeric
var name = "Madhu"; → string
var age = null; → null

* operators in Javascript :-

↳ Arithmetic operators :-

+ → Addition
- → Substraction
* → Multiplication
/ → Division
% → Modulus

↳ Equality operators :-

== → Is equal to
!= → Is not equal to

↳ Relational operators :-

< → Less than
> → greater than
<= → lessthan or equal to
>= → greater than or equal to

↳ Logical operators :-

&& → logical AND
|| → logical OR

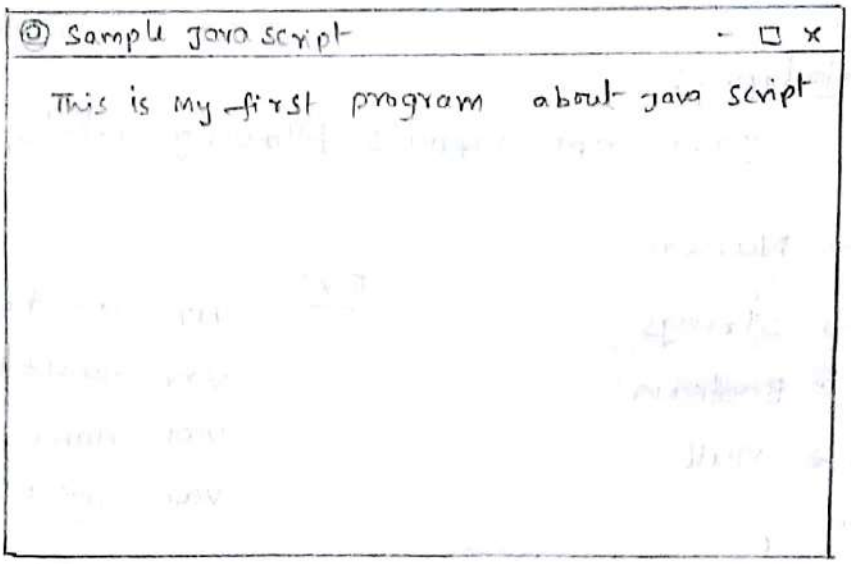
* programming with Java scripts :-

Example:- The following example program displays the normal text.

Isample.html

```
<html>
<head>
<title> Sample Java script</title>
<script language = "java script" >
<!-- This indicates the comment -->
document.writeln ("This is my first program");
document.writeln (" about javascript");
</script>
</head>
</html>
```

o/p :-



In the above example,

⇒ <script> tag is used to including the script into html document.

⇒ "<!-- -->", This is used to display or write the comment Text.

⇒ "document.write()" is used to display some text on screen

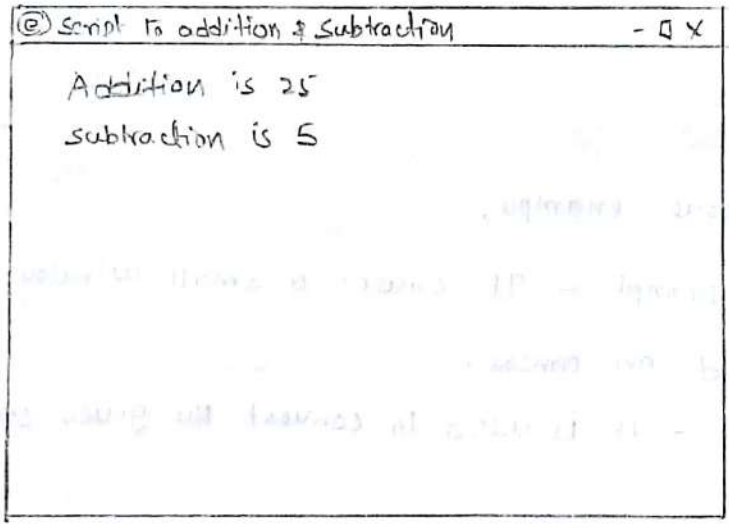
Note:- A good practice to deploy the script in the HEAD region of the HTML code.

Example:- The following script demonstrats the addition and substraction of two numbers.

AS.html

```
<html>
<head>
<title> script to addition and substraction </title>
<script language = "javascript">
  var n1, n2, add, sub;
  n1 = 10;
  n2 = 15;
  add = n1 + n2;
  sub = n2 - n1;
  document.writeln("Addition is " +add);
  document.writeln("subtraction is" + sub);
</script>
</head>
<body>
</body>
</html>
```

O/P:-



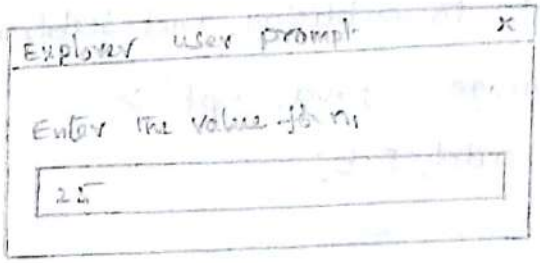
In the example, we can assign the values directly to variables. instead of this we can also provide these values dynamically during run time.

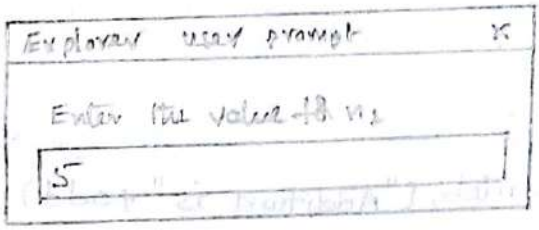
To do that, just replace the code from line number 6 to 7 by following code.

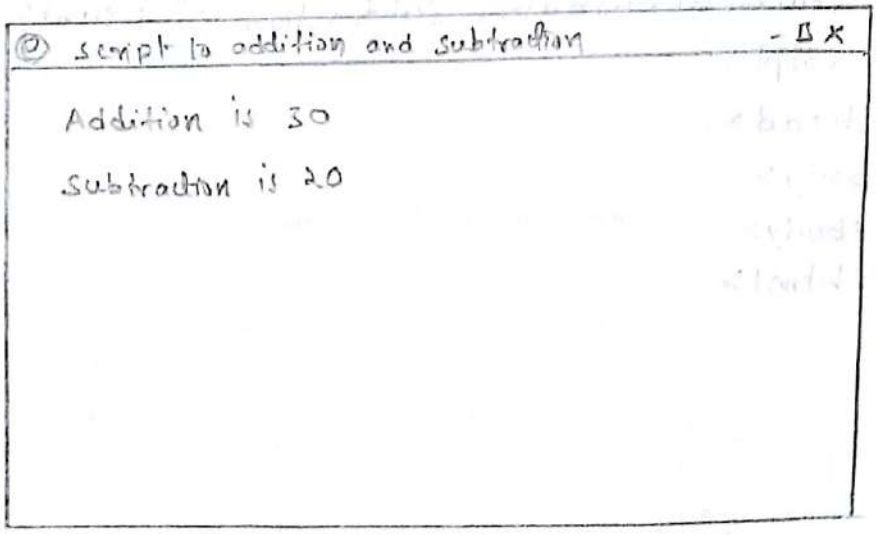
```
n1 = parseInt (window.prompt ("Enter n1 value"));
n2 = parseInt (window.prompt ("Enter n2 value"));
```

now, The output will be generated in following manner

o/p:-







In the above example,

⇒ window.prompt - It causes a small window to be displayed on console.

⇒ parseInt - It is used to convert the given string into Integer.

Similarly

⇒ parseFloat - It is used to convert a given string into floating value.

* conditional statements :-

↳ if statement :-

Syntax :-
```
if (condition)
{
 Statements;
}
```

↳ if-else statement :-

Syntax :-
```
if (condition)
{
 statements;
}
else
{
 statements;
}
```

↳ else-if ladder :-

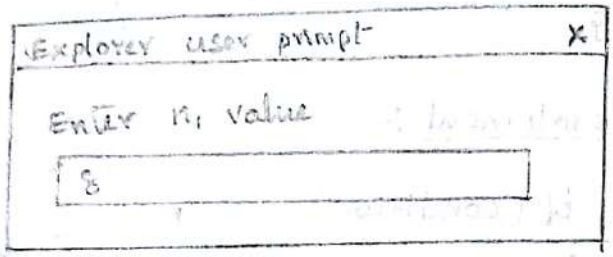
Syntax :-
```
if (condition)
{
 statements;
}
else if (condition)
{
 statements;
}
else
{
 statements;
}
```

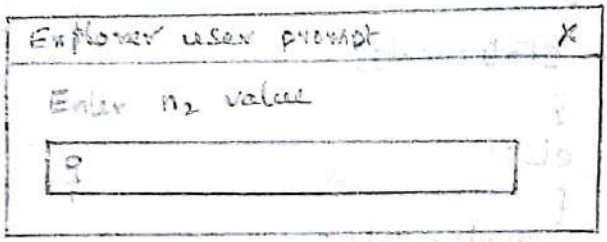
Example :- The following example finds maximum of three num - bers.

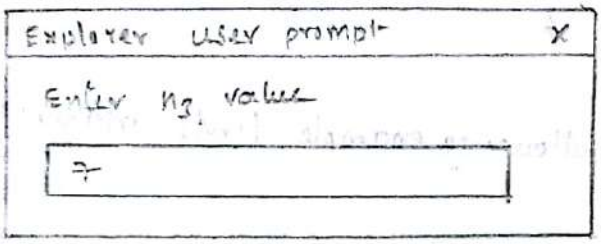
```
<html>
<head>
<title> maximum of 3 numbers </title>
<script language = "javascript">
 var n1, n2, n3;
 n1 = parseInt (window.prompt ("Enter n1 value"));
 n2 = parseInt (window.prompt ("Enter n2 value"));
 n3 = parseInt (window.prompt ("Enter n3 value"));
 if (n1 > n2 && n1 > n3)
 {
  document.writeln ("maximum is " + n1);
 }
 else if (n2 > n1 && n2 > n3)
 {
  document.writeln ("maximum is " + n2);
 }
 else
 {
  document.writeln ("maximum is " + n3);
 }
```

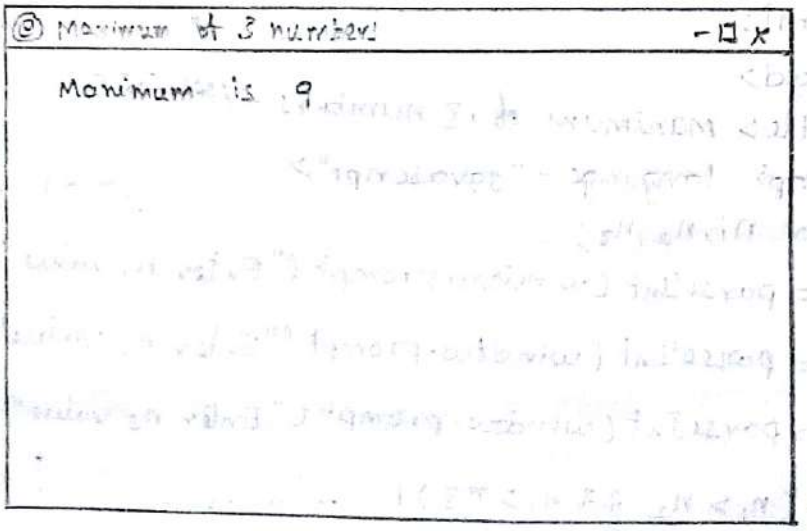
```
</script>
</head>
<body>
</body>
</html>
```

O/p:-







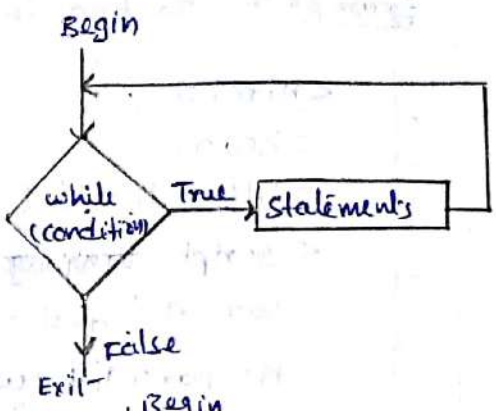


* Looping statements :-

↳ while statement :-

Syntax :-

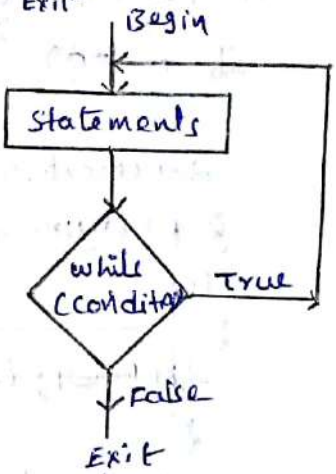
```
while (condition)
{
 statements;
}
```



↳ do-while statement :-

syntax :-

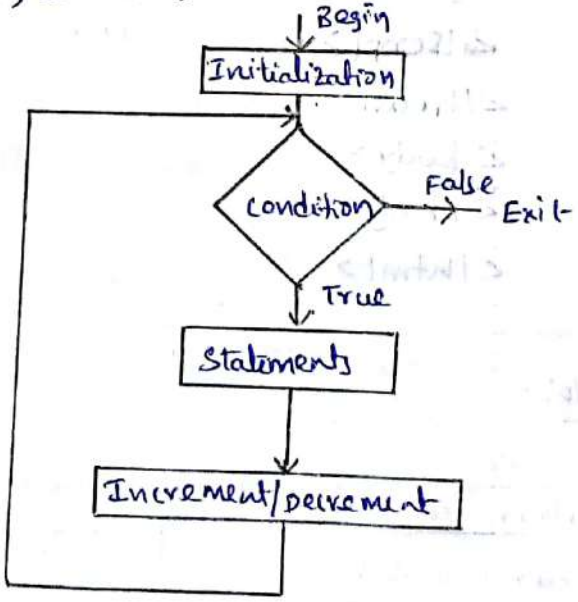
```
do
 {
  statements;
 } while (condition);
```



↳ for statement :-

syntax :-

```
for (initialization; condition; increment/decrement)
 {
  statements;
 }
```



* Branching statement :-

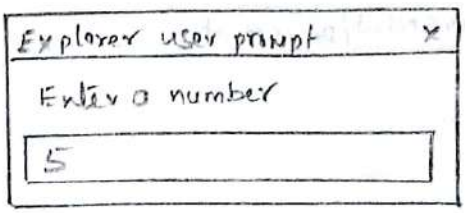
↳ switch case :- syntax :-

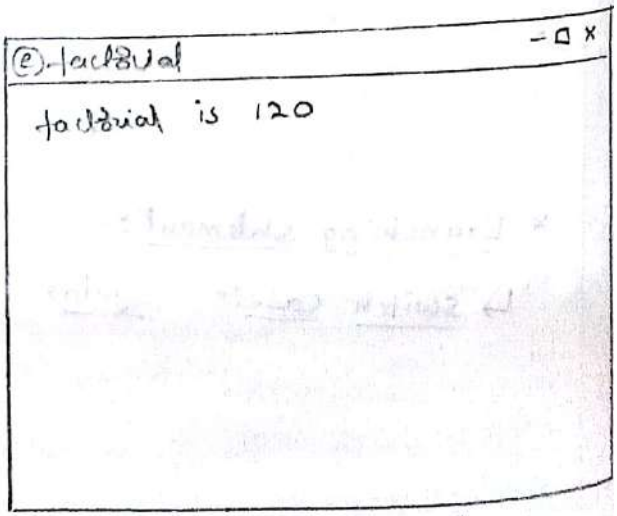
```
switch (expression)
{
 case 1: statements; break;
 case 2: statements; break;
   :
 default: statements;
}
```

Example :- To find factorial of given number.

```
<html>
<head>
<title> factorial </title>
<script language="javascript">
 var n,i,fact=1;
 n= parseInt(window.prompt("Enter a number"));
 if(n==0)
 {
  document.writeln("factorial is 1");
 }
 else
 {
  for(i=1; i<=n; i++)
  {
   fact = fact * i;
  }
 document.writeln("factorial is " +fact);
 }
</script>
</head>
<body>
</body>
</html>
```

O/p:-





* Functions :-

A function is a self-contained block of statement that perform a particular task.

Basically, functions are two types, They are

→ predefined functions

→ user defined functions

↳ pre defined functions :-

These are also called as global functions, because they can be called and used in any part of a program.

→ isFinite() :-

usually the above function takes in a numeric value as an argument and returns true only if the given argument results a finite numeric else it returns false.

Example :- isfinite(5/0); → It returns false

→ isNaN() :-

It returns true states only if the argument is not a number, else it returns a false states.

Example: isNaN(a); → It returns true

→ parseInt() :-

It accepts string as argument and converts into its equivalent numeric.

Example :- parseInt(102); → It converts to 102

→ parsefloat() :-

It accepts string as argument and converts into its equivalent floating value.

Example :- parse float(12); → 12.00

→ Eval() :-

It takes a string as an argument, it is used to evaluting the string.

Example:- Eval (2*3) → It returns 6.

↳ user defined functions:-

These functions are defined by user by using predefined keyword "function".

Each of these functions can have following

→ function name
→ List of parameters
→ List of statements
→ Return type

Syntax:-

```
function function_name (parameters)
{
  statements;
  . . .
}
```

Example:-

```
<html>
<head>
<title> Example for functions </title>
<script language='javascript'>
  var a;
  a = parseInt (window.prompt("Enter a num"));
  document.writeln("Square of given no is"
                   + square(a));
  document.writeln("cube of given no is "
                   + cube(a));
  function square (k)
  {
    var s;
    s = k * k;
  } return s;
```

```
function cube(k)
{
  var c;
  c = k*k*k;
  return c;
}
</script>
</head>
</html>
```

* Objects in Javascript :-

In todays world almost all programming languages uses object-oriented concepts.

In the real wold object is nothing but an entity, which can different from other entities.

In, Javascript objects refers to construct holding data and functions.

Once the Java script gets executed, a separate memory space is reserved for each object, where its data and functions are stored.

The Javascript supports following objects

↳ Document object :-

The wold document refers to the page which will displays the browser windows.

It has following methods

⇒ Write / Writeln ():-

It is used to display the text on the document.

Example:-

```
document.write("Hai");
```

O/P:- Hai

⇒ forms():-

It is used to process the elements in form.

Example:- document.forms (form-name);

⇒ links():-

It is used to hold the number of links in webpage.

Example:- document.links();

⇒ close():-

It is used to stop current process on the document.

Example:- document.close();

↳ Window object :-

It has a title bar, message, a default icon with one or more command buttons.

It supports following methods

⇒ open():- It is used to open a new window. It has two arguments, those are → URL
→ Name of window.

Ex:- window.open("URL", "name");

⇒ Scroll():- It is used to scroll the window easily. It has two arguments, those are → x-coordinate.
y-coordinate.

Ex:- window.scroll(100, 105);

⇒ prompt():- It is used to get the input from the user by displaying small window.

Ex:- window.prompt("Enter a value");

⇒ close():- It is used to close the current window.

Ex:- window.close();

↳ Math object :-

The Math object have the different types of methods. Those are used to perform several mathematical calculations.

It has following syntax,

Math . Method (numeric values);

The following are frequently used mathematical methods.

⇒ min() :-

It displays the minimum of two numeric values.

Ex:- document. write (Math.min (10,5)); // o/p:- 5

⇒ max() :-

It displays the maximum of two numbers

Ex:- document. write(Math. max (10,5)); // o/p: 10

⇒ abs(x) :-

It returns the absolute value of x

Ex: document. write(Math. abs (-43)); // o/p:- 43

⇒ ceil(x) :-

It returns the nearest- integer not less than x.

Ex:- document. write (Math. ceil (5.8)); // o/p: 6

document. write (Math. Ceil (5.1)); // o/p: 6

⇒ round(x) :-

It returns the nearest integer.

Ex: document. write (Math. round (5.8)); // o/p: 6

document. write (Math. round (5.1)); // o/p: 5

⇒ floor() :-

It returns the nearest integer not greater than x.

Ex: document. write (Math. floor (5.8)); // o/p: 5

document. write (Math. floor (5.1)); // o/p: 5

⇒ pow() :- It returns power of one value.

Ex: document. write (Math. pow (2,3)); // o/p: 8

⇒ sqrt():- It returns the square root of given number.

Ex:- document.write(Math.sqrt(9)); // o/p:- 3

⇒ sin():- It returns trigonometric sine value.

Ex:- document.write(Math.sin(90)); // o/p:- 1

⇒ cos():- It returns trigonometric cos value.

Ex:- document.write(Math.cos(0)); // o/p: 1

⇒ tan():- It returns trigonometric tan value.

Ex:- document.write(Math.tan(45)); // o/p:- 1

⇒ log():- It returns logarithmic equavilent value

Ex:- document.write(Math.log(2.71)); // o/p: 1.0

↳ String object :-

In general terms string refer to series of characters enclosed under double quots.

The following are frequently used string methods

⇒ toLowerCase():-

It is used to convert the given string into lower - case letters.

Ex:- var name = "MADHU";
document.writeln(name.toLowerCase());
// o/p:- madhu

⇒ toUpperCase():-

It is used to convert the given string into upper - case letters.

Ex:- var name = "madhu";
document.writeln(name.toUpperCase());
// o/p:- MADHU

⇒ concat():-

It simply combines or concatenates two strings.

Ex:- var name = "Mr"
document.write(name.concat("Madhu")); // o/p: Mr Madhu

⇒ charAt():-

It returns a character based on given index value.

Ex:- var name = "Madhu";

name.charAt(3); // o/p:- h

⇒ substr:-

It is used to extract substring from given string.

It use two arguments, "index" and "length".

Ex:- var name = "Madhu";

name.substr(2,2); // o/p:- dh

⇒ substring():-

It is used to extract substring from given string.

It use two arguments, "index" and "end".

Ex: var name = "Madhu";

name.substring(2,4); // o/p: dhu

⇒ indexOf():-

It returns an index, based on given character.

Ex:- var name = "madhu";

name.indexOf('d'); // o/p:- 2

⇒ lastIndexOf()

It takes a character as an argument and returns the numeric value, which is appearence at last time in string.

Ex:- var name = "Mr Madhu";

name.lastIndexOf('m'); // o/p:- 3.

⇒ length :-

It returns a length of given string.

Ex:- var name = "Madhu";

name.length; // o/p:- 5

↳ Date object :-

This object simply captures the date of the local system at that instant and returns the value.

usage :- var currentdate = new Date();

It has following methods

⇒ toString() :- It returns to string respective of date.

⇒ getDate() :- It returns 1 to 31, day of month

⇒ getDay() :- It returns 0 to 6, sunday to saturday.

⇒ getMonth() :- It returns 0 to 11, jan to Dec.

⇒ getFullYear() :- It returns 4 digit year no.

⇒ getHours() :- It returns 0 to 23.

⇒ getMinutes() :- It returns 0 to 59.

⇒ getSeconds() :- It returns 0 to 59.

⇒ setDate (1--31)
⇒ setDay (0--6)
⇒ setMonth (0--11)
⇒ setFullYear (Y,m,d)

} Sets date, day, month and year.

⇒ setHours (0--23)
⇒ setMinutes (0--59)
⇒ setSeconds (0--59)
⇒ setTime (HH:MM:SS)

} Sets date, hours, Minutes, seconds and time.

↳ Array object :-

Array is a collection of items or elements. In Javascript, Arrays are created using a special keyword 'new'

Syntax :- var Array_name = new Array();

Ex: var numbers = new Array(10);

The Array object supports following Methods

⇒ push() :- It is used to insert data into an array. once the data is pushed, array size gets increased.

Ex:- numbers.push(9);

⇒ pop() :- It is used to remove the elements from an array. Ex:- numbers.pop(8);

⇒ sort() :- It is used to arrange the elements in ascending order Ex:- numbers.sort();

⇒ reverse() :- It reverses the elements in an array.

Ex:- numbers.reverse();

Example - var Students = new Array ("Ravi", "Mohan", "kiran");
Students.push("seetha"); O/P: "Ravi", "Mohan", "kiran", "seetha".
Students.pop("Mohan"); O/P: Ravi, kiran, seetha.

---

* Dynamic HTML with Javascript :-

→ DHTML is a combination of HTML, Javascript, CSS and DOM (Document object Model). It creates some interactive and animated web sites.

→ This is done after loading the page and during the viewing process.

→ DHTML code is difficult to be developed and debugged because it is a collection of various Technologies.

The dynamic HTML prouides validation process.

⇒ Data validation :-

validation is process of ensuring that some data might be correct data for a particular application.

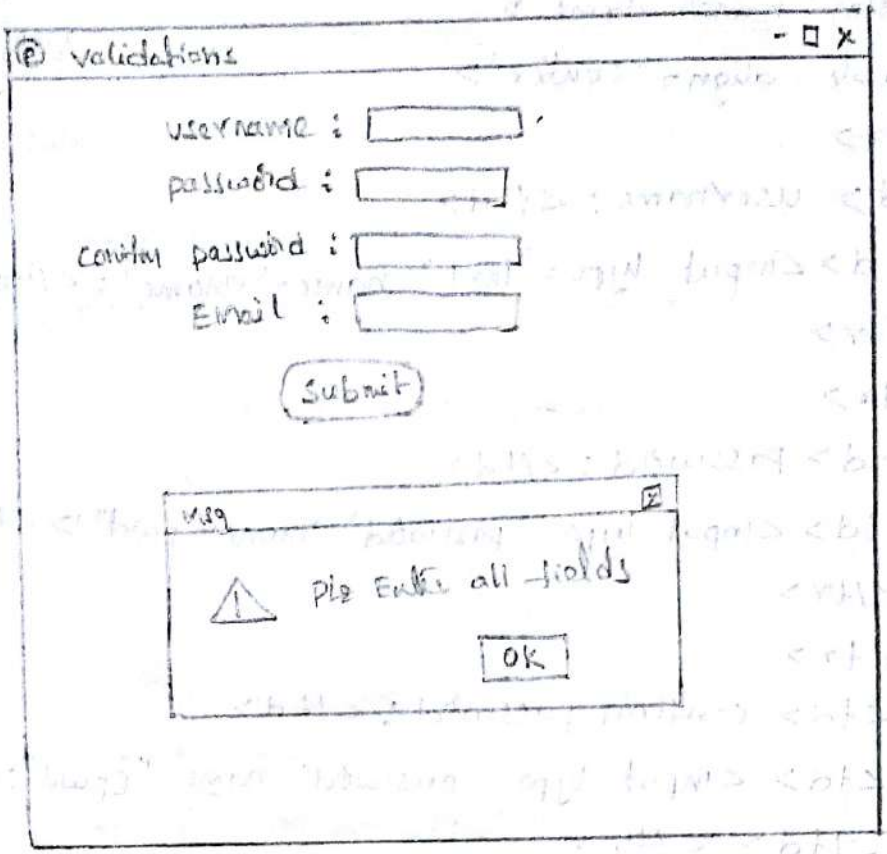
Data validation is a process through which a user is allowed to enter the data required by the organization

Example:-

```
<html>
<head>
<title> validations </title>
<script language = "javascript" >
function validate()
{
 var uname, pwd, cpwd, email;
 uname= document.forms("frm1").uname.value;
 pwd = document.forms("frm1").pwd.value;
 cpwd = document.forms("frm1").cpwd.value;
 email = document.forms("frm1").email.value;
if(uname==" "|| pwd==" "|| cpwd==" "|| email==" ")
 {
   alert("plz enter all details");
 }
else if(uname.length<8)
{
 alert("username must be atleast 8 characters");
}
else if(pwd.length<6)
{
 alert("passwdd must be atleast 6 character");
}
else if(pwd!=cpwd)
{
 alert("passwords didn't match");
}
else
 {
  alert("success");
 }
}
</script>
</head>
```

```
<body>
<form name="frm1">
<table align="center">
<tr>
<td> username : </td>
<td><input type="text" name="uname"></input>
</td>
</tr>
<tr>
<td> password : </td>
<td><input type="password" name="pwd"></input>
</td>
</tr>
<tr>
<td> confirm password : </td>
<td><input type="password" name="cpwd"></input>
</td></tr>
<tr>
<td> E-mail : </td>
<td><input type="text" name="email"></input>
</td></tr>
<tr>
<td colspan=2 align="center">
<input type="button" onclick="validate()"
name="btn" value="submit" /></td>
</tr>
</table>
</form>
</body>
</html>
```

output:-



* Event Handling in Java script :-

Once the event is generated, there is often requirement of code to process these events. such code is known as event handler.

The following are commonly used Event handlers

→ onLoad() :- It invokes as soon as a given web document was loaded.

→ onclick() :- It invokes as soon as whenever any of the page elements are clicked.

→ onchange() :- It invokes whenever data in any of the HTML control (textbox, textarea, etc) gets changed.

→ onDblclick() :- It invokes whenever any of the page elements are clicked twice.

→ onMousemove() :- It invokes as soon as a user passes the mouse pointer over any of the page elements.

→ onSubmit():- It invokes as soon as the user press the submit button on the web page.

→ onUnload():- It invokes as soon as a given web document is closed.

→ onKeyUp():- It invokes as soon as the user releases the key.

→ onKeyDown():- It invokes as soon as the user press the key.

→ onBlur():- It invokes as soon as any text or data turns blur.

Example :-

```
<html>
<head>
<title> Event handlings </title>
</head>
<body onLoad = 'alert ("Example of Javascript events')'>
<h1 align = "center"> Javascript Events </h1>
<br/> <hr/>
<form name = "frm1" onsubmit = 'alert ("submit")'>
onBlur Event :
<input type = "text" value = "click here"
                onBlur = 'alert ("Not clear")'> </input>

onclick Event :
<input type = "text" value = "click here"
                onclick = 'alert ("Clicked")'></input>

Onchange Event :
<input type = "text" value = "click here"
                onchange = 'alert ("changed')'></input>
<br/>
```

onDblclick Event:

```
<input type="button" value="click here"
        onDblclick='alert("Double clicked")'></input
```

onmouseMove Event:

```
<input type="button" value='place here"
        onmouseMove='alert("mouse placed")'>
                                        </input>
```

onkey up Event:

```
<input type="text" value="press any key"
        onkeyup='alert("key up")'></input>
```

onkeyDown Event:

```
<input type="text" value="press any key"
        onkeyDown='alert("KeyDown")'></input>
```

onsubmit Event:

```
<input type='submit" value="submit'></input>
</form>
</body>
</html>
```

---

* opening and closing a Window:-

To open a new window, we usually resort to certain predefined Javascript functions.

The following syntax is

window.open('URL', 'window_name'--);

→ URL: Here we supply the address of the page.

→ Window-name: It specifies the window name.

And it supports different types of attributes, like

width = pixel

height = pixel

scrollbars = yes or NO

To closing a window syntax is

Syntax: window.close();

Example:-

```
<html>
<head>
<title> Window operations </title>
</head>
<body>
<form>
<input type="button" value="New window"
onclick="window.open('login.html', 'login'
width=250, height=200)"> </input>

<input type="button" value="close"
onclick="window.close()"> </input>.

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

Example programs:-

*) Write a javascript to find given number is amstrong or not.

```
<html>
<head>
<title> Amstrong </title>
<script language = "javascript" >
 var  n, temp, sum=0, r;
 n = parseInt (window.prompt("Enter n value"));
 temp= n;
 while (n>0)
 {
  r = parseInt (n%10);
 sum = sum + r*r*r;
  n= parseInt (n/10);
 }
 if ( temp==sum)
 {
 document.write( "Amstrong");
 }
 else
 {
  document.write ("NOT");
 }
</script>
</head>
</html>
```

8) Write a Javascript to find given number is pallendram or not.

```
<html>
<head>
<title> pallendram </title>
<script language = "javascript">
 var n, temp, sum = 0, r;
n = parseInt ( window. prompt (" Enter n value "));
temp = n;
while ( n > 0)
{
 r = parseInt (n % 10);
 sum = sum * 10 + r;
  n = parseInt (n / 10);
}
if ( temp = = sum)
 {
  document. writeln (" pallendram");
 }
 else
  {
   document. writeln (" NOT ");
  }
</script>
</head>
< body>
 </body>
 </html>
```

*) Write a javascript to find the prime numbers upto given numbers.

```
<html>
<head>
<title> prime numbers </title>
<script language = "javascript" >
 var n, count, i, j;
 n= parseInt (window.prompt ("Enter a value"));
 document.write(" prime numbers are");
 for(i=1; i<=n; i++)
  {
    count=0;
  for(j=1; j<=i; j++)
  {
    if(i%j ==0)
    {
      count++;
    }
  }
  if(count ==2)
    {
     document.writeln(i);
    }
 }
</script>
</head>
</html>
```