Java Scripts

To design a webpage the users must learn 3 languages.

1. HTML : It is used to specify the content on a webpage.
2. CSS : It is used to specify the layout of the webpage.
3. Java Script : It is used to program the behavior of web page.

→ **Java Scripts :-**

- It is a client side scripting language which is used for validating the webpage.
- It is a object based programming language
- It inherits the concepts of java such as objects, functions, arrays, etc.,.
- It allows to create an interactive webpage

Syntax :- <script type="text/java script">

Java code

</script>

→ **Script tag :-**

Script tag allows to embed the java code in an html document. The script tag can be maintained in both head section & body section.

→ Variable :-

It is used to store the value.

Syntax :- 1. Var Variable name

the value can be over written

2. let Variable name

the value cannot be over written

3. const constant name

Examples for Var :

1.
```
<script>
Var message ="Hellow world!";
console.log (message);
</script>
```

- Example for let :

```
<script>
let x=5;
let y=10;
let sum=x+y;
console.log("The sum of"+x+"and"+y+"is"+"sum+".");
</script>
```

Example for const :

```
<script>
const PI = 3.14159;
const radius = 5;
const circumference = 2* PI*radius;
console.log ("The circumference of the circle is"
                                + circumference);

</script>
```

Operators :-

They are different types of operators in javascript.

1. Airthmetic operators
2. Assignment operators
3. Comparison operators
4. String operators
5. Logical operators
6. Ternary operators
7. Type operators
8. BitWise operators

Arithmetic Operators :

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

Assignment operators :-

Operators : =, +=, -=, *=, /=, %=, **=

Comparison Operators :-

Operators : ==, ===, !=, !==, >, <, >=, <=, ?

Logical Operators :-

Operators : &&, ||, !

→ Type Operators:-

Operators: typeof, instanceof

→ Bitwise Operators:-

Operators: &, |, ~, ^, <<, >>, >>>

→ Window Objects:

It supports three Methods...

1. Prompt Method:-

It ~~allo~~ allows to enter the values from Keyboard.

Syntax: window.prompt("Enter text");

(or)

Prompt("Enter text");

2. Alert Method:-

It allows to have an alert message on a webpage.

Syntax: alert("Message");

3. Confirm Method:-

It allows to have confirmation box for a webpage.

Syntax: window.confirm("continue");

→ Document Objects:-

It supports two methods write & writeln, get Element By Id.

Example:-
```
<script type="text/javascript">
Var x,y;
x=Integer.parseInt(prompt("Enter x"));
```

```
y=integer.parseInt(prompt("Enter y");
s=x+y;
doc.writeln("Sum:"+s);
confirm("continue");
</script>
```

→ Control Structures

Conditional Statements :- Java script supports all the conditional statements provided by Java

if

if else

if else if else

switch

Example :- (if-else)

```
<script>
Var x= Integer parseInt(Prompt("Enter x"));
Var y= Integer parseInt(Prompt("Enter y"));
if (x<y)
   doc write("x is small");
else
   doc.write("y is small");
</script>
```

Example :- (switch)

```
<script>
Var ch=prompt("Enter choice");
switch (ch)
{
case 'a';
case 'e';
case 'i';
case 'o';
case 'u'; doc.write("vowel");
break;
}
default:
   doc.write("constant")
```

```
break;
</script>
```

→ Control Statements: For
While
do while

Example for "for loop":-

```
<script>
var i;
var n=Integer parseInt(Prompt("Enter Max Value"));
for(i=1; i<=n; i++)
{
doc.write("<table border="1">");
doc.write("<tr><th>NO</th><th>Square</th>");
doc.write("<tr><td>"+ i + "</td><td>"+ i*i +"</td
></tr>");
}
</script>
```

→ Branching Statements:-

Java script supports different types of branching statements.

1. Break
2. continue

Example for Break :-

```
<script>
var i;
var n=Integer parseInt(prompt("Enter Max
Values"));
for(i=1; i<=n; i++)
```

```
{
  if (i==5)
  {
   var x=i;
    Break;
  }
  doc.write(i);
}
doc.write(x);
</script>
```

Example for continue:-

```
<script>
var i;
Var n=Integer parseInt(prompt("Enter Max Value"));
for (i=1; i<=n; i++)
{
  if (i==5)
  {
   Var x=i;
    continue;
  }
  doc.write(i);
}
doc.write(x);
</script>
```

→ Built-in Objects (or) Built-in functions:-

Math Objects: It supports different Methods abs, pow, sqrt, log, cos, sin, tan, etc.,.

Example:

```
<script>
Var x=-5
doc.write ("Abs.write:" + Math.abs(x));
Var p= Math.pow (x,2);
doc.write ("i x value is i"+ p);
Var s= Math.sqrt (p);
doc.write ("sqrt:" +s);
</script>
```

→ String Objects:-

Java script supports different types of String Methods

1. charAt
2. concat
3. substring
4. length
5. to uppercase
6. to lowercase

Example:-

```
<script>
S1 = " Hi" ;
S2 = "Hello" ;
S3 = S1.concat (S2)
doc.write ("concat str:" +S3);
doc.write ("length of S3:" +S3.length);
doc.write ("substring:" +S3.substring(2,4));
doc.write ("charAt:" +S1.charAt(1));
</script>
```

Date Objects:-

Java Script allows date objects with different methods.

toString : Convert data into String format

getMonth: returns the month number

getDay : returns the day of the week.

getYear : returns the full year.

getMinutes: returns the minutes from the time

getSeconds: returns the seconds from the time

getHours : returns the hours from the type.

Example:-

```
<script>
var dt=new Date()
doc.write("Month:"+dt.getMonth());
doc.write("Date:"+dt.toString());
doc.write("Year:"+dt.getYear());
doc.write("Hour:"+dt.getHour());
</script>
```

Arrays:

An array is a single variable that is used to store elements of different data types. Javascript arrays are zero-indexed.

Syntax: Var a=new Array(5);
Var b=new Array(10, 20, 30);
Var c = ['a', 'b', 'c', 'd'];

Example:-

```
<script>
Var a=new Array(5);
Var b=new Array(10,20,30);
Var c=['a', 'b', 'c', 'd'];
for (i=0; i<5; i++)
{
a[i]=i;
doc.write(a[i]+"<br>");
}
for (i=0; i<3; i++)
{
doc.write(" B Array is:");
doc.write(b[i]+"<br>");
}
for (i=0; i<c.length; i++)
{
doc.write(" C.Array is");
doc.write(" c[i]+"<br>");
}
</script>
```

Example-2:-

```
<script>
var days=new Array();
days[0]="sun";
days[1]="Mon";
        ⋮
days[6]="sat";
for(i in days)
{
doc.write("Array is:");
doc.write("days[i]+"<br>");
}
</script>
```

→ Array Objects:-

Java script allows different methods. They are...

Join():- used to join array elements using delimiter.

Syntax: Arrayname.join(delimiter())

reverse():- To reverse an array

Syntax: Arrayname.reverse()

Sort():- To sort the array elements

Syntax: Arrayname.sort()

Push():- It will insert an element at the end of the array.

Syntax: Arrayname.push(Item)

Pop():- used to delete an element from the end of the array.

Syntax: Arrayname.pop()

Shift():- It is used to delete an element from the begining of array.

Syntax: array.shift()

unshift():- It is used to delete an element at begining of array.

Syntax: arrayname.unshift(Item)

Slice():- used to return sub array

Syntax: Arrayname.slice(starting index, Ending index)

Example:-

```
<script>
Var ar=['Mango','banana','apple','kiwi'];
doc.write("Array:" ar.join(":"));
doc.write("Reverse.array:"+ ar.reverse());
doc.write("sort is :" ar.sort());
doc.write("push:"+ar.push('orange'));
doc.write("After push:"+ar);
doc.write("pop:"+ar.pop());
doc.write("After pop:"+ar);
doc.write("unshift:"+ar.unshift('dragon'));
doc.write("subarray:"+ar.slice(1,3));
</script>
```

Functions:-

A Javascript function is a block of code designed to perform a particular task.

Syntax:
```
Function name()
{
Stmts;
}
```

Example:-
```
<head>
<script>
function.myfn()
{
document.Write("Function called");
}
</script>
</head>
<body>
<script>
document.Write("Statement before
                function call"+"<br>");
myfn();
</script>
```

Example using Arrays:-
```
<head>
<script>
function display(a)
{
document.Write("contents of array are:"+"<br>");
i=0;
for (i in a)
```

```
{
document.write (a[i] + "<br>");
i++;
}
}
</script>
</head>
<body>
<script>
Var ar = new Array [10];
for (i=0; i<10; i++)
{
ar[i] = i;
}
display(ar);
</script>
</body>
```

→ **Event Handling:-**

Whenever an event is generated on a webpage to handle it, javascript provides event handlers.

Events	Event Handlers
1. click	1. onclick
2. dblclick	2. ondblclick
3. focus	3. onfocus
4. blur	4. onblur
5. mouseOut	5. onmouseOut
6. mouseOver	6. onmouseOver

7. Keyup
8. Keydown
9. Keyreleased
10. Keypressed
11. mousedragged
12. mousereleased
13. change
14. load
15. unload

7. onkeyup
8. onkeydown
9. onkeyreleased
10. onkeypressed
11. onmousedragged
12. onmousereleased
13. onchange
14. onload
15. onunload.

Example for event Handlers:-

```
<html>
<head>
<script languages="Javascript">
function loadd()
{
alert("page is loaded");
}
function unloadd()
{
alert("page is unloaded");
}
function clicked()
{
confirm("button is clicked");
}
function mouse()
{
alert("mouse over");
}
```

```
function focus()
{
alert("focus on field");
}
function blur()
{
alert("performed blur");
}
</script>
</head>
<body onload="loadd()" onunload="unloadd()">
<form>
Name: <input type="text" Value=" " onfocus="focusss()"
><br>
Password: <input type="password" Value=" "
onBlur="blurr()"><br>
<input type="Submit" Value="Click me" onclick=
"clicked()">
</form>
<a href="F1.html" on mouseover="mouse()">
link </a>
</body>
</html>
```

Example

```
<!DOCTYPE HTML>
<head>
<title> chainging colors </title>
<script type = "text/Javascript" src="dyncolors.js">
</script>
</head>
<body>
<p. style="font-family: Times ; font-style: italic;
                font-size: 24pt">
This small page illustrate dynamic setting of the
foreground and background colors for a document
</p>
<form>
<p>
<label>
Background color:
<input type="text" name="background" size="10"
    onchange="SetColor("background", this.value)">
</label>
<br>
<label>
Foreground color:
<input type="text" name="foreground" size="10"
   onchange="SetColor("foreground", this.value)">
</label>
<br>
</p>
</form>
</body>
```

Example JS file

```
Function SetColor(where, newcolor)
{
if (where=="background")
document.body.style.backgroundcolor=new color;
Else
document.body.style.color=new color;
}
```

Example

```
<!DOCTYPE HTML>
<html>
<head>
<title> Dynamic Values </title>
<script type="text/JavaScript" src="dynvalue.js">
</script>
</head>
<body>
<form>
<p Style="font-weight:bold">
<span Style="font-style:italic">
Student information
</span>
<br>
<label> Name: <input type="text" onmouseover=
    "messages(0)" onmouseout="messages(4)"></label>
<br>
<span Style="font-style: italic">
To create an account, Provide the following:
```

```
</span>
<br>
<label> User Id : <input type="text" onmouseover=
    "messages(2)" onmouseout="messages(4)"></label>
<br>
<label> password: <input type="text" onmouseover=
    "messages(3)" onmouseout="messages(4)"></label>
<br>
<textarea id="advice Box" rows="3" cols="50"
    style="position: absolute; left: 250px; top=20px;
                color: red">
This box provides advice on filling out the form
on this page. put them input field to get
advice.
</text area>
<br>
<input type="submit" value="submit">
<input type="reset" value="reset">
<p>
</form>
</body>
```

JS file

```
Var helpers=["yourname must be in the form:
                firstname, Middle, lastname",
"Your email address must have the form: \n
                user@domain",
"Your user ID must have the form atleast six
                characters",
```

" Your password must have atleast six charactors and it must include one digit",

" This box provides advice on fillingout \n the form on this page. put the mouseover function messages (advice Number)

```
{
document.getElement ById("advice Box");
value=helpers[advice Number];
}
```

Example for dynamic fonts:-

```
<!DOCTYPE HTML>
<head>
<title> changing Fonts </title>
<style type="text/css">
.regText
{
font: Times;
font-size: 16pt;
}
</style>
</head>
<body>
<p class="regtext">
The state of
<a style="color: blue;" onmouseover="this.style.
    color='red'; this.style.font="italic 16pt Times";"
onmouseout="this.style.color='blue'; this.style.
    font='normal 16pt Times';"
Washington </a> produces menu of our nation's
                                    apples.
</p>
```

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

Example for Moving Elements

```
<!DOCTYPE HTML>
<head>
<title>Moving Elements</title>
<script type="text/Javascript" src="mover.js">
</script>
</head>
<body>
<form>
<p>
<label> x coordinate:
<input type="text" id="leftcord" size="3">
</label>
<br>
<label> y coordinate:
<input type="text" id="topcord" size="3">
</label>
<br>
<input type="button" value="Move It" onClick=
 "moveIt('logo', document.getElementById
    ('topcord').value, document.getElementById
      ('leftcord').value)"> </p>
</form>
<div id="logo" style="position: absolute;
          top: 115px; left: 0;">
<img src="NEC.jpg">
```

```
</div>
<body>
</html>
```

Js file

```
Function moveIt (a, newTop, newLeft)
{
dom=document.getElementById(a).style;
dom.top=newTop+"px";
dom.left=newLeft+"px";
}
```

Example for positional elements

```
<html>
<head>
<style>
#myDIV {
  border:1px solid black;
  background-color: lightblue;
  width: 300px;
  height: 300px;
  Position: relative;
  top: 50px;
}
</style>
</head>
<body>
<p>Click the "Try it" button to change the position
    Property of the DIV Element: </p>
<button onClick="myFunction()">Try it</button>
<div id="myDIV">
```

```
<p>This DIV element is placed 50 pixels from
the top of its original position.</p>
<p> Click the button to set the position
Property to "absolute" and see what happens.
</p>
<p>It will then be placed 50 pixels from the top
    the page.</p>
</div>
<script>
function myFunction() {
  document.getElementById("myDIV").style.
        Position="absolute";
}
</script>
</body>
</html>
```

Regular Expression:

Javascript allows regular expression to Validate the form and for pattern matching. It Provides metacharacters [quantifiers] RegExp method and String Methods.

	Metacharacters	Description
1.	\w	Find a Word character
2.	\W	Find a Non Word character
3.	\d	Find a digit
4.	\D	Finds a Non-digit

5.	\s	Finds a whitespace character
6.	\S	Finds a Non whitespace character
7.	\n	Finds a newline character
8.	\t	Finds a tab character
9.	[abc]	Finds a word character btw brackets
10.	[^abc]	Finds a non word character btw brackets
11.	[0 to 9]	Finds a digits btw the brackets
12.	[^0 to 9]	Finds a non digits btw the brackets
13.	^n	Finds a pattern which begins with the specified character.
14.	n$	Find a pattern which ends with specified character.
15.	n?	'0' & '1' occurance of a pattern
16.	n+	Atleast '1' Occurance of a pattern
17.	n*	'0' & more occurances of a pattern
18.	[x/x]	Finds a pattern based on the character specified in the brackets.
19.	n{x}	Finds a pattern with sequence of characters.

→ **Regexp Methods:-**

It allows to find Matching pattern using two methods test() and exec() methods.

Exec() :-

It allows to find a matching pattern if it is found it returns the result as an array else it returns null

Syntax:- re.exec(String)

Example:-

```
<script>
str=prompt("Enter a String");
re= prompt("Enter a pattern");
var res =re.exec(str, 'i');
if (res)
{
doc.writeln("found:"+res[0]);
}
else
{
doc.writeln("Not found");
}
</script>
```

String Methods:-

Javascript Supports regular Expression string methods to find a pattern. It provides 4 methods 1) Search()

2). Match()

3). MatchAll()

3). Replace()

3) ii. replaceAll()

4) Split()

→ Search():-

It finds a matching pattern if match is found, it returns the index position of the match else it returns -1.

Example:-

```
<script>
str=prompt("Enter a String");
re= prompt(" Enter pattern");
var res = str.search(re)
if (res)
{
doc.writeln("found at :"+ res);
}
else
{
doc.writeln(" Not found");
}
</script>
```

→ Match():-

It allows to find a pattern if match is found it returns the result array & else it returns Null.

Example:-

```
<script>
str=prompt("Enter String");
re=prompt("Enter pattern");
```

```
Var res = Str.Match(re)
if (res)
{
doc.writeln("found at:"+res);
}
Else
{
doc.writeln("Not found");
}
</script>
```

→ replace():-

It is used to replace one pattern with another pattern.

Example:-

```
<script>
Str= prompt("Enter string");
res1= prompt("Enter pattern1);
res2= prompt("Enter pattern2);
Var rep=Str.replace(re1,re2);
if (rep)
{
doc.writeln("Before replace:"+Str);
}
Else
{
doc.writeln("After replace:"+rep);
}
</script>
```

→ Split():-

It allows to split the string based on the pattern.

Example:-

```
<script>
str=prompt("Enter String");
re1=prompt("Enter pattern");
res=str.split(re1);
for (i=0; i<res.length;i++)
    doc.writeln("splitted String"+ i+ ":"+res[i]);
```

→ Validating form:-

```
<html>
<script>
function fun1()
v1=/\w{6}/;
a2=v1.test(f.n.value);
if(!a2)
{
alert("Name should be mimum 6 characters!");
Exit(0);
}}
function fun2()
{
v1=/\w{6}\d{2}/;
a2=v1.test(f.p.value);
if(!a2)
{
alert("password should be minimum 6 characters!")
Exit(0);
}}
```

```
function fun 3 ()
{
v2 = /\w{1}@\D{5}.\D{3}/;
a2 = v2.test(f.e.value);
if (!a2)
{
alert("check E-mail id!");
}}
function fun 4 ()
{
v3 = /\d{10}/;
a2 = v3.test(f.ph.value);
if (!a2)
{
alert("Enter correct phone number")
Exit(0);
}}
</script>
</head>
<body>
<form name = "f">
<input type = "text" name = "n" onBlur = "fun1()">
<input type = "password" name = "p" onBlur = "fun2()">
<input type = "text" name = "e" onBlur = "fun3()">
<input type = "tel" name = "phone" onBlur = "fun4()">
<input type = "radio" name = "r" > F
<input type = "radio" name = "r" > M
Country:
```

```
<Select>
<option>India</option>
<option> US</option>
<option> UK </option>
</select>
ADDRESS: <textarea rows="2" cols="5"></textarea>
<input type="button" name="s" Value="submit">
<input type="button" name="r" Value="reset">
</form>
</body>
</html>
```

→ Modifiers

Modifier	Description
i	Matches case insensitive
g	Matches all the pattern in a doc. i.e doesn't stop at 1st Match
m	Multiline match

Example for Visibility control

```
<! DOCTYPE HTML>
<head>
<title> Visibility Control </title>
<script type="text/javascript" src="hide.js">
</script>
</head>
<body>
<form>
<div id="logo" style="position: relative;
                      visibility: visible;">
<img src="NEC.jpg">
</div>
<p>
<input type="button" value="Toggle" onclick=
                                  "flipImgag()">
</p>
</form>
</body>
</html>
```