Javascript Datatypes

====================

Datatype describe what type of value we want to store inside a variable.
Javascript is a dynamically typed language.We don't need to use datatype at the time of variable declaration.
To declare a variable we need to use "var" keyword.
ex:

```
	var i = 10;
```

Internally,A javascript engine is used to determine corresponding datatype based on the value.
We have two types of datatypes in javascript.

1)Primitive Datatypes
2)Non-Primitive Datatypes

1)Primitive Datatypes

We have following list of primitive datatypes.

Datatype	Description
--------	------------
1)Number	It is used to represent numbers.
2)String	It is used to represent strings.
3)Boolean	It is used to represent boolean.
4)null	It is used to represent null.
5)undefined	It is used to represent undefined.

ex:1

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
		<script>
			var i=10;
			document.writeln(i+"<br>");

			var j=10.56;
			document.writeln(j+"<br>");
		</script>
</body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
```

```
</head>
<body>
        <script>
            var i='ihub';
            document.writeln(i+"<br>");

            var j="talent";
            document.writeln(j+"<br>");
        </script>
</body>
</html>

ex:3
-------
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        <script>
            var i=true;
            document.writeln(i+"<br>");
            var j=false;
            document.writeln(j+"<br>");
        </script>
</body>
</html>

ex:4
------
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        <script>
            var i=null;
            document.writeln(i+"<br>");
        </script>
</body>
</html>

ex:5
-----
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
```

```
<body>
                    <script>
                              var i;
                              document.writeln(i+"<br>");
                    </script>
</body>
</html>
```

2)Non-Primitive Datatypes

We have following list of non-primitive datatypes.

Datatype	Description
1)Object	It is used to represent an instance through which we can access members.
2)Arrays	It is used to represent of similar elements/data.
3)RegEx	It is used to represent regular expression.

Javascript Operators

====================

Javascript operators are the symbols which are used to perform some operations on oprands.

ex:

```
          var c = a + b;

          Here = and + are operators.
          Here a,b and c are operands.
```

We have following list of operators in javascript.

1)Arithmetic operators
2)Conditional operators
3)Bitwise operators
4)Logical operators
5)Assignment operators
6)Special operators

1)**Arithmetic operators**

We have following list of arithmetic operators.

operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modules
++	Incrementation
--	Decrementation

```
ex:1
------
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        <script>
            var a=10,b=5;
            document.writeln((a+b)+"<br>");//15
            document.writeln((a-b)+"<br>");//5
            document.writeln((a*b)+"<br>");//50
            document.writeln((a/b)+"<br>");//2
            document.writeln((a%b)+"<br>");//0
        </script>
</body>
</html>
ex:2
------
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        <script>
            var a=5,b=10;
            document.writeln((a/b)+"<br>");//0.5
            document.writeln((a%b)+"<br>");//5
        </script>
</body>
</html>

ex:3
-------
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        <script>
            var i=10;
            i++;
            document.writeln(i+"<br>");
        </script>
</body>
</html>
```

ex:4

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
                <script>
                        var i=10;
                        var j=i++;
                        document.writeln(i+" "+j+"<br>");
                </script>
</body>
</html>
```

ex:5

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
                <script>
                        var i=10;
                        var j=i++ + i--;
                        document.writeln(i+" "+j+"<br>");//10  21
                </script>
</body>
</html>
```

ex:6

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
                <script>
                        var i=10;
                        ++i;
                        document.writeln(i+"<br>");//11
                </script>
</body>
</html>
```

ex:7

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
```

```
</head>
<body>
        <script>
            var i=10;
            var j=++i;
            document.writeln(i+" "+j+"<br>");//11   11
        </script>
</body>
</html>
```

ex:8

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        <script>
            var i=10;
            document.writeln(i++ + ++i);//22
        </script>
</body>
</html>
```

2)Conditional operators

We have following list of operators.

operator	description
--------	-----------
>	Greater then
<	Less Then
>=	Greater then equals to
<=	Less then equals to
==	equals to
!=	not equals to

ex:

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        <script>
            document.writeln((10>5)+"<br>");//true
            document.writeln((10<5)+"<br>");//false
            document.writeln((10>=10)+"<br>");//true
            document.writeln((10<=50)+"<br>");//true
            document.writeln((10==10)+"<br>");//true
            document.writeln((10==20)+"<br>");//false
            document.writeln((10!=20)+"<br>");//true
            document.writeln((10!=10)+"<br>");//false
        </script>
</body>
</html>
```

3)Bitwise operators

We have following list of bitwise operators.

operator	description
--------	------------
&	Bitwise AND
|	Bitwise OR
^	Bitwise XOR
~	Bitwise NOT
>>	Right shift
<<	Left shift

Bitwise AND (&)

Bitwise AND operator deals with binary numbers.

Truth table

T	T	= T
T	F	= F
F	T	= F
F	F	= F

ex:

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        <script>
            var a=10,b=5;
            var c = a & b;
            document.writeln(c);//0
            /*
                10 - 1010
                5  - 0101
                ----------
                &  - 0000
            */
        </script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        <script>
            var a=10,b=15;
            var c = a & b;
            document.writeln(c);//10
            /*
                10 - 1010
                15 - 1111
                ----------
                &  - 1010
                                <---
                0*1  +  1*2  +  0*4  + 1*8
                0+2+0+8
                10
            */
        </script>
</body>
</html>
```

Bitwise OR (|)

Bitwise OR operator deals with binary number.

Truth table

T	T	= T
T	F	= T
F	T	= T
F	F	= F

ex:

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        <script>
            var a=10,b=15;
            var c = a | b;
            document.writeln(c);//15
            /*
                10 - 1010
                15 - 1111
                ----------
                |  - 1111
```

```
                        <---
                1*1  +  1*2  +  1*4  + 1*8
                1+2+4+8
                15
            */
        </script>
</body>
</html>
```

Bitwise XOR (^)

Bitwise XOR operator deals with binary numbers.

Truth table

T	T	= F
T	F	= T
F	T	= T
F	F	= F

ex:

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        <script>
            var a=10,b=15;
            var c = a ^ b;
            document.writeln(c);//5
            /*
                10 - 1010
                15 - 1111
                ----------
                ^  - 0101
                        <---
                1*1 +  0*2 +  1*4 +  0*8
                1+0+4+0
                5
            */
        </script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
```

```
<body>
        <script>
            var a=~10;
            document.writeln(a);//-11
        </script>
</body>
</html>
ex:
---
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        <script>
            var a=~(-45);
            document.writeln(a);//44
        </script>
</body>
</html>
```

Right shift (>>)

```
-------------------
10  >> 1 = 10/2
10  >> 2 = 10/4
10  >> 3 = 10/8
10  >> 4 = 10/16
ex:1
-------
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        <script>
            var a=10 >> 3;
            document.writeln(a);// 10/8 =1
        </script>
</body>
</html>

ex:2
-----
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
```

```
			<script>
				var a=10 >> 5;
				document.writeln(a);// 10/32 =0
			</script>
</body>
</html>
```

Left shift operator (<<)

```
10  <<  1 = 10*2
10  <<  2 = 10*4
10  <<  3 = 10*8
10  <<  4 = 10*16
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
			<script>
				var a=10 << 3;
				document.writeln(a);// 10 * 8  = 80
			</script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
			<script>
				var a=10 << 5;
				document.writeln(a);// 10 * 32  = 320
			</script>
</body>
</html>
```

Q) What is the difference between == and === equals?
==

It is used to check only values are same or not.
ex:

```
<!DOCTYPE html>
<html>
<head>
```

```
    <title>MyPage!</title>
</head>
<body>
        <script>
            document.writeln((1==1) +"<br>");//true
            document.writeln((1==true) +"<br>");//true

            document.writeln((0==0) +"<br>");//true
            document.writeln((false==0) +"<br>");//true

            document.writeln((10=="10") +"<br>");//true
        </script>
</body>
</html>
```

===

It is used to check values and datatypes are same or not.

ex:

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        <script>
            document.writeln((1===1) +"<br>");//true
            document.writeln((1===true) +"<br>");//false

            document.writeln((0===0) +"<br>");//true
            document.writeln((false===0) +"<br>");//false

            document.writeln((10==="10") +"<br>");//false
        </script>
</body>
</html>
```

4)Logical operators

We have following list of logical operators.

operator	description
&&	logical AND
||	logical OR
!	logical NOT

logical AND (&&)

```
<!DOCTYPE html>
<html>
<head>
```

```
    <title>MyPage!</title>
</head>
<body>
        <script>
            document.writeln((6>3) && (7<4) +"<br>"); //false
            document.writeln((6>3) && (7<40) +"<br>"); //true
        </script>
</body>
</html>
```

logical OR (||)

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        <script>
            document.writeln(((6>3) || (7<4))+"<br>"); //true
            document.writeln(((6>3) || (7<40)) +"<br>"); //true
            document.writeln(((6>30) || (7<4))+"<br>"); //false
        </script>
</body>
</html>
```

logical NOT

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        <script>
            document.writeln((!(6>3))+"<br>"); //false
            document.writeln((!(6>30)) +"<br>"); //true
        </script>
</body>
</html>
```

5)Assignment operator

We have following list of assignment operatores.

Operator	Description
-----------	---------------
+=	addition and equals to
-=	subtraction and equals to
*=	multiplication and equals to
/=	division and equals to
%=	modules and equals to

```
ex:
----
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
                <script>
                        var i=10;
                        i+=4;
                        document.writeln(i);//14
                </script>
</body>
</html>
ex
----
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
                <script>
                        var i=10;
                        i-=4;
                        document.writeln(i);//6
                </script>
</body>
</html>
ex:
---
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
                <script>
                        var i=10;
                        i*=4;
                        document.writeln(i);//40
                </script>
</body>
</html>
ex:
---
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
```

```
</head>
<body>
        <script>
            var i=10;
            i/=4;
            document.writeln(i);//2.5
        </script>
</body>
</html>
ex:
------
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        <script>
            var i=10;
            i%=4;
            document.writeln(i);//2.5
        </script>
</body>
</html>
```

6)Special operators

We have following list of special operators.

operator	description
----------	-------------
?:	conditional
new	It is used to create an instance.
typeof	It is used to identify type of an object.

conditional(?:)

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        <script>
            (5>2)?document.writeln("TRUE"):document.writeln("FALSE");
        </script>
</body>
</html>
ex:
----
<!DOCTYPE html>
<html>
```

```
<head>
    <title>MyPage!</title>
</head>
<body>
        <script>
            (!(5>2))?document.writeln("TRUE"):document.writeln("FALSE");
        </script>
</body>
</html>
```

Q)Write a javascript program to find out greatest of two numbers?

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        <script>
            var val1=prompt("Enter the First Number :");
            var a=parseInt(val1);

            var val2=prompt("Enter the Second Number :");
            var b=parseInt(val2);
(a>b)?document.writeln(a+" is greatest"):document.writeln(b+" is greatest");
        </script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        <script>
            var val1=prompt("Enter the First Number :");
            document.writeln(typeof(val1)+"<br>");

            var a=parseInt(val1);
            document.writeln(typeof(a)+"<br>");
        </script>
</body>
</html>
```

JavaScript IF ELSE STMT
==========================

It is used to evaluate the code either our condition is true or false.
We have three forms for Javascript IF ELSE STMT .
1)IF STMT
2)IF ELSE STMT
3)IF ELSE IF STMT

1)IF STMT

It is used to evaluate the code only if our condition is true.

```
syntax:
            if(condition)
            {
                        -
                        - //code to be evaluate
                        -
            }
ex:
-----
<!DOCTYPE html>
<html>
<head>
            <title>MyPage!</title>
</head>
<body>
                        <script>
                                    if(1,2,3,4,5)
                        {
                                                document.writeln("Hello"); // Hello
                        }
                        </script>
</body>
</html>
ex:
----
<!DOCTYPE html>
<html>
<head>
            <title>MyPage!</title>
</head>
<body>
                        <script>
                                    if(1,2,3,4,5,0)
                                    {
                                                document.writeln("Hello"); //nothing
                                    }
                        </script>
</body>
</html>
ex:
----
<!DOCTYPE html>
<html>
<head>
            <title>MyPage!</title>
</head>
<body>
```

```
                <script>
                        if(~0)
                        {
                                document.writeln("Hello");//Hello
                        }
                </script>
</body>
</html>
```

2)IF ELSE STMT

It will evaluate the code either our condition is true or false.

syntax:

```
        if(condition)
        {
                - //code to be evaluate if cond is true
        }
        else
        {
                - // code to be evaluate if cond is false
        }
```

ex:

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
                <script>
                        if((2>5)  &&  (5<10))
                        {
                                document.writeln("WELCOME");
                        }
                        else
                        {
                                document.writeln("Thankyou");
                        }
                </script>
</body>
</html>
```

o/p:
thank you

ex:

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
```

```
</head>
<body>
        <script>
            if( (10>>20) || (10<<2))
            {
                document.writeln("WELCOME");
            }
            else
            {
                document.writeln("Thank you");
            }
        </script>
</body>
</html>
```

Q)Write a javascript program to find out given number is even or odd?

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        <script>
            var n=parseInt(prompt("Enter the number :"));
            if(n%2==0)
                document.writeln("It is even number ");
            else
                document.writeln("It is odd number ");
        </script>
</body>
</html>
```

Q)write a javascript program to find out given alphabet is a vowel or not?

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        <script>
            var val=prompt("Enter the character :");
            var c=val.charAt(0);
            if(c=='a' || c=='e' || c=='i'|| c=='o' || c=='u')
                document.writeln("It is a vowel");
            else
                document.writeln("It is not a vowel");
        </script>
</body>
</html>
```

iii)IF ELSE IF STMT

It will evaluate the code based on multiple conditions.

syntax:

```
        if(cond1)
        {
                - //code to be execute if cond1 is true
        }
        else if(cond2)
        {
                - //code to be execute if cond2 is true
        }
        else if(cond3)
        {
                - //code to be execute if cond3 is true
        }
        else
        {
                - //code to be execute if all conditions are false.
        }
```

ex:

------

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
                <script>
                        var option=parseInt(prompt("Enter the Option :"));

                        if(option==100)
                                document.writeln("It is police number ");
                        else if(option==103)
                                document.writeln("It is enquiry number ");
                        else if(option==108)
                                document.writeln("It is Emergency number ");
                        else
                                document.writeln("Invalid option");
                </script>
</body>
</html>
```

Javascript Switch case

==================

It is used to evaluate the code based on multiple conditions.

It is similar to if else if stmt.

Javsacript switch case is more convenient when compare to javascript if else if stmt because we can declare numbers,characters,strings and decimals.

syntax:

```
        switch(condition)
        {
                case val1: //code to be execute
                           //break stmt

                case val2: //code to be execute
                           //break stmt
                -
                -
        }
```

ex:1

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
                <script>
                        var option=parseInt(prompt("Enter the Option :"));
                        switch(option)
                        {
                                case 100: document.writeln("It is police number"); break;
                                case 103: document.writeln("It is enquiry number"); break;
                                case 108: document.writeln("It is emergency number"); break;
                                default: document.writeln("Invalid option ");
                        }
                </script>
</body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
                <script>
                        var str=prompt("Enter the String :");
                        switch(str)
                        {
                                case "one": document.writeln("January"); break;
                                case "two": document.writeln("February"); break;
                                case "three": document.writeln("March"); break;
                                default: document.writeln(" Coming soon...");
                        }
```

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

ex:3

```
<!DOCTYPE html>
<html>
<head>
          <title>MyPage!</title>
</head>
<body>
                    <script>
                              var val=parseFloat(prompt("Enter the decimal value :"));
                              switch(val)
                              {
                                        case 10.0: document.writeln("stmt1"); break;
                                        case 10.1: document.writeln("stmt2"); break;
                                        case 10.2: document.writeln("stmt3"); break;
                                        default: document.writeln("Not Avaiable");
                              }
                    </script>
</body>
</html>
```

Javascript LOOPS

================

LOOPS are used to evaluate the code for multiple times.
We have four types of LOOPS in javascript.
1)do while loop
2)while loop
3)for loop
4)for IN loop

1)do while loop

It will evaluate the code untill our condition is true.
In do while loop our code will execute atleast for one time. either our condition is true or false.
syntax:

```
          do
          {
                    -
                    - //code to evaluate
                    -
          }while(condition);
```

ex:

```
<!DOCTYPE html>
<html>
<head>
```

```
	<title>MyPage!</title>
</head>
<body>
		<script>
			var i=1;
			do
			{
				document.writeln(i);//infinite 1

			}while(i<=10);
		</script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
		<script>
			var i=1;
			do
			{
				document.writeln(i++);//1 2 3 4 5 6 7 8 9 10
			}while(i<=10);
		</script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
		<script>
			var i=10;
			do
			{
				document.writeln(i--);//10 9 8 7 6 5 4 3 2 1

			}while(i>=1);

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

2)while loop

It is used to evaluate the code untill our condition is true.

syntax:

```
            while(condition)
            {
                        -
                        - //code to be evaluate
                        -
            }
```

ex:

```
<!DOCTYPE html>
<html>
<head>
            <title>MyPage!</title>
</head>
<body>
                        <script>
                                    var i=1;
                                    while(i<=10)
                                    {
                                                            document.writeln(i);//1 2 3 4 5 6 7 8 9 10
                                                            i++;
                                    }
                        </script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
            <title>MyPage!</title>
</head>
<body>
                        <script>
                                    var i=11;
                                    while(i<=10)
                                    {
                                                            document.writeln(i);//nothing
                                                            i++;
                                    }
                        </script>
</body>
</html>
```

3)for loop

It is used to evaluate the code untill our condition is true.

syntax:

```
-------
        for(initialization;condition;incrementation/decrementation)
        {
                -
                - //code to be evaluate
                -
        }
```

ex:

```
----
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
                <script>
                        for(var i=1;i<=10;i++)
                        {
                                document.writeln(i);//1 2 3 4 5 6 7 8 9 10
                        }
                </script>
</body>
</html>
```

ex:

```
----
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
                <script>
                        var sum=0;
                        for(var i=1;i<=10;i++)
                        {
                                sum+=i;
                        }
                        document.writeln("sum of 10 natural numbers is ="+sum);
                </script>
</body>
</html>
```

4)for IN loop

================

It is used to iterate the data from array.

ex:

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
                <script>
                        var arr=[10,20,30,40];

                        for(var i in arr)
                        {
                                document.writeln(arr[i]+" ");
                        }

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

ex:

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
                <script>
                        var arr=['a','b','c','d'];

                        for(var i in arr)
                        {
                                document.writeln(arr[i]+" ");
                        }

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

Q)Write a javascript program to find out given number is palindrome or not?

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
                <script>
                        var n=parseInt(prompt("Enter the number :"));
                        var temp=n;
                        var rem,rev=0;
                        while(n>0)
                        {
                                rem=n%10;
                                rev=rev*10+rem;
                                n=parseInt(n/10);
                        }
                        if(rev==temp)
                                document.writeln("It is a palindrome number");
                        else
                                document.writeln("It is not a palindorm number");
                </script>
</body>
</html>
```

Q)Write a javascript program to display below loop pattern?

```
* * * *
* * * *
* * * *
* * * *
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
                <script>
                        for(var i=1;i<=4;i++)
                        {
                                for(var j=1;j<=4;j++)
                                {
                                        document.writeln("* ");
                                }
                                //new line
                                document.writeln("<br>");
                        }
                </script>
</body>
</html>
```

Q)Write a javascript program to display below loop pattern?

```
*
* *
* * *
* * * *
<!DOCTYPE html>
<html>
<head>
```

```
        <title>MyPage!</title>
</head>
<body>
                <script>
                        for(var i=1;i<=4;i++)
                        {
                                for(var j=1;j<=i;j++)
                                {
                                        document.writeln("* ");
                                }
                                //new line
                                document.writeln("<br>");
                        }
                </script>
</body>
</html>
```

Javascript Functions
========================

Javascript function is a block of code which is used to perform perticular task.
Javascript function can be declare by using function keywords , followed by name and followed by paranetheses i.e '()'.
Javascript function contains letters,digits,underscore and dollar.same rule as variables.
Javascript paranetheses contains parameters/arguments and each argument seperated with comma (,).

syntax:

```
        function  fun_name(parameter1,parameter2,...,parameterN)
        {
                -
                -//code to be execute
                -
        }
```

JavaScript functions are executed at the time when they are invoke/call.

1)When it is called from javascript code.
2)When event is occur (when user click on button).
3)Self invocation.

1)When it is called from javascript code

--

```
<!DOCTYPE html>
<html>
        <head>
                <title>IHUB Talent</title>
        </head>
        <body>
                <script type="text/javascript">
                        function f1()
                        {
```

```
                document.writeln("This is Function");
            }

            //call
            f1();
        </script>
    </body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
    <head>
        <title>IHUB Talent</title>
    </head>
    <body>
        <script type="text/javascript">

            function f1()
            {
                document.bgColor="#FFFF00";
            }

            //call
            f1();
        </script>
    </body>
</html>
```

2)When event is occur

```
<!DOCTYPE html>
<html>
    <head>
        <title>IHUB Talent</title>
    </head>
    <body>
        <script type="text/javascript">

            function f1()
            {
                document.writeln("This is Function");
            }
        </script>

        <button onclick="f1()">click</button>
    </body>
</html>
```

ex:

```
<!DOCTYPE html>
```

```
<html>
        <head>
                <title>IHUB Talent</title>
        </head>
        <body>
                <script type="text/javascript">

                        function f1()
                        {
                                document.bgColor="#FF0000";
                        }
                </script>
                <button onclick="f1()">click</button>
        </body>
</html>
```

No returntype with No argument function

--

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
                <script>
                function f1()
                {
                        document.writeln("Javascript class");
                }
                f1();
                </script>
</body>
</html>
```

No returntype with Argument function

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
                <script>
                function f1(a,b)
                {
                        var c=a+b;
                        document.writeln(c);
                }
                f1(10,20);
                </script>
```

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

With returntype with No argument function

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
		<script>

		function f1()
		{
			return "Hello Javascript people";
		}

		document.writeln(f1());

		</script>


</body>
</html>
```

With returntype with Argument function

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
		<script>
		function f1(a,b)
		{
			return a+b;
		}
		document.writeln(f1(10,20));
		</script>
</body>
</html>
```

Interview Questions

=====================

Q)What is JavaScript?

JavaScript is a scripting language.
It is a case sensitive language.
It is an object based language.

It is a loosely typed checking language.
It was developed by Brendan Eich in 1995.
The original name of JavaScript is LiveScript.

Q)Advantages of javascript?

Speed/Faster
Simplicity
Interoperability
Versatility
Rich interfaces
Reduce Server Load
No compiler and No interpreter
Weakly typed language
Platform independent
Client Side validation

Q)Disadvantages of JavaScript?

Client-Side Security
Browser Support
Stop Rendering
Slow Bitwise Operation
Single Inheritance

Q)Types of Functions in javascript?

We have three types of functions in javascript.
1)Named Function
2)Anonymous Function
3)Arrow Function

1)Named Function

These types of functions contains name at the time of definition.
ex:

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
                <script>
                function f1()
                {
                        document.writeln("Named Function");
                }
                f1();
                </script>
</body>
</html>
```

2)Anonymous Function

These types of functions don’t contain any name.
They are declared dynamically at runtime.
ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
		<script>
		var f1=function()
				{
				document.writeln("Anonymous Function");
				}
		f1();
		</script>
</body>
</html>
```

3)Arrow function

According to ES6 standard we need to use arrow function.

Arrow functions are more secured when compare to named function and anonymous function.

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
		<script>

		var f1=()=>
				{
				document.writeln("Arrow Function");
				}
		f1();
		</script>


</body>
</html>
```

Q) What is JavaScript Closure?

A closure is the combination of a function bundled together along lexical scope.

In other words, a closure gives you access to an outer function's scope from an inner function.

In JavaScript, closures are created every time when function is created.

Ex:

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
                <script>
                        //lexical scope
                        var a=10;
                        function f1()
                        {
                                //lexical scope
                                var b=20;
                                function f2()
                                {
                                        document.writeln(a+" "+b);
                                }
                                f2();
                        }
                        f1();

                </script>


</body>
</html>
```

Javascript Object

====================

A javascript object is an entity which is having state and behaviours.
In general, javascript object is a collection of properties and functions.
Javascript is a object based language because everything is present in objects.
Javascript is a template based but not class based.We don't need to create a class to get the object.We can create object directly.

There are three ways to create javascript objects

1)By using Object literal
2)By creating instance of an Object i.e using new keyword.
3)By using Object constructor i.e using new keyword.

1)By using Object literal

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB Talent</title>

	</head>
	<body>
		<script type="text/javascript">
				emp={
						eid:101,
						ename:"Alan Morries",
						esal:10000
				};
				document.writeln("Employee Id:"+emp.eid+"<br>");
				document.writeln("Employee Name:"+emp.ename+"<br>");
				document.writeln("Employee Salary:"+emp.esal+"<br>");
		</script>
	</body>
</html>
```

2)By creating instance of an Object
--

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB Talent</title>
	</head>
	<body>
		<script type="text/javascript">
				var emp=new Object();
				emp.eid=102;
				emp.ename="Erick Anderson";
				emp.esal=20000;
				document.writeln("Employee Id:"+emp.eid+"<br>");
				document.writeln("Employee Name:"+emp.ename+"<br>");
				document.writeln("Employee Salary:"+emp.esal+"<br>");
		</script>
	</body>
</html>
```

3)By using Object constructor
--

Here we need to create a function with parameters and each parameter must assign in the current object by using this keyword.

ex:

```
<!DOCTYPE html>
<html>
	<head>
```

```
        <title>IHUB Talent</title>
    </head>
    <body>
        <script type="text/javascript">
                function emp(eid,ename,esal)
                {
                    this.eid=eid;
                    this.ename=ename;
                    this.esal=esal;
                }
                e=new emp(103,"Ana Julie",30000);
                document.writeln("Employee Id :"+e.eid+"<br>");
                document.writeln("Employee Name :"+e.ename+"<br>");
                document.writeln("Employee Sal :"+e.esal+"<br>");
        </script>
    </body>
</html>
```

Javascript Array

==================

In javascript , Array is an object which contains similar elements.
Array index always starts with '0' because it is a logical process.
There are three ways to create an array in javascript.

1)By using array literal
2)By creating instance of an array i.e using new operator.
3)By creating array constructor i.e using new operator.

1)By using array literal

```
<!DOCTYPE html>
<html>
    <head>
        <title>IHUB Talent</title>
    </head>
    <body>
        <script type="text/javascript">
            var arr=[10,20,30,40];
            for(var i=0;i<arr.length;i++)
            {
                document.writeln(arr[i]+" ");
            }
        </script>
    </body>
    </body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
```

```
    <head>
        <title>IHUB Talent</title>
    </head>
    <body>
        <script type="text/javascript">
            var arr=[10,20,30,40];
            for(var i in arr)
            {
                document.writeln(arr[i]+" ");
            }
        </script>
    </body>
    </body>
</html>
```

ex:3

```
<!DOCTYPE html>
<html>
    <head>
        <title>IHUB Talent</title>
    </head>
    <body>
        <script type="text/javascript">
            var arr=["html","css","js"];
            for(var i in arr)
            {
                document.writeln(arr[i]+" ");
            }
        </script>
    </body>
    </body>
</html>
```

2)By creating instance of an array i.e using new operator

```
<!DOCTYPE html>
<html>
    <head>
        <title>IHUB Talent</title>
    </head>
    <body>
        <script type="text/javascript">
            var arr=new Array();
            arr[0]=10;
            arr[1]=20;
            arr[2]=30;
            for(var i in arr)
            {
                document.writeln(arr[i]+" ");
            }
```

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

3)By creating array constructor i.e using new operator

```
===================================================
<!DOCTYPE html>
<html>
		<head>
			<title>IHUB Talent</title>
		</head>
		<body>
			<script type="text/javascript">
				var arr=new Array(10,20,30,40,50);
				for(var i in arr)
				{
					document.writeln(arr[i]+" ");
				}
			</script>
		</body>
		</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
		<title>IHUB TALENT</title>
</head>
<body>
		<script type="text/javascript">
			var arr=[];
			arr.push(10);
			arr.push(20);
			arr.push(30);
			for (i in arr)
			{
				document.write(arr[i]+" ");
			}
		</script>
</body>
</html>
```

ex:

```
-----
<!DOCTYPE html>
<html>
<head>
		<title>IHUB TALENT</title>
</head>
```

```
<body>
    <script type="text/javascript">
        var arr=[];
        arr.push(10);
        arr.push(20);
        arr.push(30);
        arr.pop();
        for (i in arr)
        {
            document.write(arr[i]+" ");
        }
    </script>
</body>
</html>
```

Javascript String
===================

In javascript , string is an object which contains collection of characters.
There are two ways to create a string in javascript.

1)By using string literal
2)By creating instance of a string.

1)By using string literal

```
<!DOCTYPE html>
<html>
    <head>
        <title>IHUB Talent</title>
    </head>
    <body>
        <script type="text/javascript">
            var str1="bhaskar";
            document.writeln(str1+"<br>");
            var str2='solution';
            document.writeln(str2+"<br>");
        </script>
    </body>
    </body>
</html>
```

2)By creating instance of a string.

```
<!DOCTYPE html>
<html>
    <head>
        <title>IHUB Talent</title>
    </head>
    <body>
        <script type="text/javascript">
            var str=new String("bhaskar");
				document.writeln(str);
			</script>
		</body>
		</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
		<head>
			<title>IHUB Talent</title>
		</head>
		<body>
			<script type="text/javascript">
				var str="bhaskar";
				document.writeln(str.length);
			</script>
		</body>
		</body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
		<head>
			<title>IHUB Talent</title>
		</head>
		<body>
			<script type="text/javascript">
				var str="bhaskar";
				document.writeln(str.toUpperCase());
				var str2="BHASKAR";
				document.writeln(str.toLowerCase());
			</script>
		</body>
		</body>
</html>
```

ex:3

```
<!DOCTYPE html>
<html>
		<head>
			<title>IHUB Talent</title>
		</head>
		<body>
			<script type="text/javascript">
				var str1="ihub";
				var str2="talent";
```

```
                                document.writeln(str1.concat(str2));
                        </script>
                </body>
                </body>
</html>

ex:4
------
<!DOCTYPE html>
<html>
        <head>
                <title>IHUB Talent</title>
        </head>
        <body>
                <script type="text/javascript">
                        var str1="ihub";
                        document.writeln(str1.charAt(2));
                </script>
        </body>
        </body>
</html>

ex:
------
<!DOCTYPE html>
<html>
        <head>
                <title>IHUB Talent</title>
        </head>
        <body>
                <script type="text/javascript">
                        var str="ihub";
                        var arr=str.split(");
                        for(var i in arr)
                        {
                                document.writeln(arr[i]+"<br>");
                        }
                </script>
        </body>
        </body>
</html>

ex:
-----
<!DOCTYPE html>
<html>
        <head>
                <title>IHUB Talent</title>
        </head>
        <body>
                <script type="text/javascript">
```

```
                        var str="ihub";
                        var arr=str.split(");
                        for(var i=arr.length-1;i>=0;i--)
                        {
                                document.writeln(arr[i]);
                        }
                </script>
        </body>
        </body>
</html>
```

BOM (Browser Object Model)

============================

The Browser Object Model is used to interact with browser.

The default object for a browser is window object.It means we can call all the functions by using window or directly.

ex:

```
        window.alert("Welcome to JavaScript");
        or
        alert("Welcome to JavaScript");
```

window object

=====================

It is used to create a window on a browser.

A window object is created automically by the browser.

A "window" is a object of browser but not javascript.

Javascript objects are String,Array,Date and etc.

A "window" object is used to write programming related to browser.

With the help of window object we can perform following activities very easily.

1)It display dialog boxes and pop boxes.
2)We can find width and height of a browser.
3)We can move or resize the browser.
4)Scroll to the browser.
5)Get URL,hostname,protocol and etc of a browser.
6)We can get javascript history.

1)alert()

==============

It will display alert dialog box.It has message with ok button.

ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>IHUB Talent</title>
        </head>
        <body>
                        <script type="text/javascript">
                                function f1()
                                {
```

```
					alert("Welcome to JavaScript");
				}
			</script>
			<button onclick="f1()">click</button>
	</body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB Talent</title>
	</head>
	<body>
	<button onclick="alert('This is IHUB Talent')">click</button>
	</body>
</html>
```

2)confirm()

It will dispaly confirm dialog box.It has message with ok button and cancel button.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB Talent</title>
	</head>
	<body>
			<script type="text/javascript">
				function f1()
				{
					var v=confirm("Do you wants to delete ?");
					if(v==true)
					{
						alert("ok");
					}
					else
					{
						alert("cancel");
					}
				}
			</script>
			<button onclick="f1()">delete</button>
	</body>
</html>
```

3)prompt()

It will display prompt dialog box.It contains message with textfield.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB Talent</title>
	</head>
	<body>
			<script type="text/javascript">
				function f1()
				{
					var v=prompt("Who are you?");
					alert("Welcome :"+v);
				}
			</script>
			<button onclick="f1()">click</button>
	</body>
</html>
```

innerWidth and innerHeight
==============================

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB Talent</title>
	</head>
	<body>
			<script type="text/javascript">
				var w=window.innerWidth;
				var h=window.innerHeight;
				document.writeln("Width :"+w+"<br>");
				document.writeln("Height :"+h+"<br>");
			</script>
	</body>
</html>
```

Note:

Press "CTRL + +" for zoomin.
Press "CTRL + -" for zoomout.

window.open()

----

ex:1

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB Talent</title>
	</head>
	<body>
```

```
            <script type="text/javascript">
                function openWindow()
                {
                    window.open("http://www.google.com");
                }
            </script>
            <button onclick="openWindow()">open a new window</button>
    </body>
</html>

ex:2
-----
<!DOCTYPE html>
<html>
    <head>
        <title>IHUB Talent</title>
    </head>
    <body>
            <script type="text/javascript">
                function openWindow()
                {
                    window.open("http://www.google.com","_blank");
                }
            </script>
            <button onclick="openWindow()">open a new window</button>
    </body>
</html>

ex:3
------
<!DOCTYPE html>
<html>
    <head>
        <title>IHUB Talent</title>
    </head>
    <body>
            <script type="text/javascript">
                function openWindow()
                {
                    window.open("http://www.google.com","_parent");
                }
            </script>
            <button onclick="openWindow()">open a new window</button>
    </body>
</html>

ex:4
-------
<!DOCTYPE html>
<html>
    <head>
```

```
        <title>IHUB Talent</title>
    </head>
    <body>
            <script type="text/javascript">
                function openWindow()
                {
        window.open("http://www.google.com","_blank","width=200px,height=200px");
                }
            </script>
            <button onclick="openWindow()">open a new window</button>
    </body>
</html>
```

close()

```
<!DOCTYPE html>
<html>
    <head>
        <title>IHUB Talent</title>
    </head>
    <body>
            <script type="text/javascript">
                var myWindow;
                function openWindow()
                {
myWindow=window.open("http://www.google.com","","width=300px,height=300px");
                }
                function closeWindow()
                {
                    myWindow.close();
                }
            </script>
            <button onclick="openWindow()">open a new window</button>
            <button onclick="closeWindow()">close a window</button>
    </body>
</html>
```

close()

```
<!DOCTYPE html>
<html>
    <head>
        <title>IHUB Talent</title>
    </head>
    <body>
            <script type="text/javascript">
                var myWindow;
                function openWindow()
                {
myWindow=window.open("http://www.google.com","","width=300px,height=300px");
                }
```

```
					function closeWindow()
					{
						myWindow.close();
					}
				</script>
				<button onclick="openWindow()">open a new window</button>
				<button onclick="closeWindow()">close a window</button>
	</body>
</html>
```

Whenever we open a new window , it takes left top alignment.
In order to move the window we need to use moveTo() or moveBy() function.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB Talent</title>
	</head>
	<body>
				<script type="text/javascript">
					var myWindow;
					function openWindow()
					{
myWindow=window.open("http://www.google.com","","width=300px,height=300px");
					}
					function moveWindow()
					{
						myWindow.moveTo(100,100);
					}
				</script>
				<button onclick="openWindow()">open a new window</button>
				<button onclick="moveWindow()">move window</button>
	</body>
</html>
```

Note: Here we can't move window because in browser console we will get one error.
To over come this limitation we need to use custom window.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB Talent</title>
	</head>
	<body>
				<script type="text/javascript">
					var myWindow;
					function openWindow()
					{
```

```
                         myWindow=window.open("","_blank","width=300px,height=300px");
                              }
                              function moveWindow()
                              {
                                   myWindow.moveTo(100,100);
                              }
                         </script>
                         <button onclick="openWindow()">open a new window</button>
                         <button onclick="moveWindow()">move window</button>
     </body>
</html>
```

Note:

MoveTo() function will move from absolute position.
MoveBy() function will move from relative position.

setTimeout()

The setTimeout() is executed only once.
If you need repeated executions, use setInterval() instead.
ex:

```
<!DOCTYPE html>
<html>
     <head>
          <title>IHUB Talent</title>
     </head>
     <body>
               <script type="text/javascript">
                    function setTimeOut()
                    {
                         setTimeout(function f1()
                                        {
                                        alert("Hello World")
                                        },4000);
                    }
               </script>
               <button onclick="setTimeOut()">click</button>
     </body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
     <head>
          <title>IHUB Talent</title>
     </head>
     <body>
               <script type="text/javascript">
					function setTimeOut()
					{
						setTimeout(Anim,4000);
					}
					function Anim()
					{
						alert("Yahoo! this is javascript");
					}
				</script>
				<button onclick="setTimeOut()">click</button>
	</body>
</html>
```

clearTimeout()

The clearTimeout() method clears a timer set with the setTimeout() method.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB Talent</title>
	</head>
	<body>
				<script type="text/javascript">
					var myId;
					function setTimeOut()
					{
						myId=setTimeout(Anim,4000);
					}
					function Anim()
					{
						alert("Yahoo! this is javascript");
					}
					function removeTimeOut()
					{
						clearTimeout(myId);
					}
				</script>
				<button onclick="setTimeOut()">set time</button>
				<button onclick="removeTimeOut()">remove time</button>
	</body>
</html>
```

setInterval()

==================

A setInterval() method calls a function to evaluate the expression at specified interval(milliseconds).
A setInterval() method calls continously function untill we call clearInterval() method or window is closed.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB Talent</title>
		<style>
			div
			{
				width:150px;
				height: 150px;
				background-color: #FF0000;
			}
		</style>
	</head>
	<body>
			<script type="text/javascript">
				var a=0;
				setInterval(Anim,1000);

				function Anim()
				{
					a = a + 10;
			var target=document.getElementById("myId");
					target.style.marginLeft= a + 'px';
				}
			</script>
			<div id="myId"></div>
	</body>
</html>
```

clearInterval()
===================
A clearInterval() function is used to clear the timer set on setInterval() function.
An id which is return from setInterval() function will use as parameter to clearInterval().
ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB Talent</title>
		<style>
			div
			{
				width:150px;
				height: 150px;
				background-color: #FF0000;
			}
		</style>
	</head>
	<body>
			<script type="text/javascript">
				var a=0;
```

```
                                        var id=setInterval(Anim,1000);
                                        function Anim()
                                        {
                                                a = a + 10;
                                                if(a==100)
                                                {
                                                        clearInterval(id);
                                                }
                                                var target=document.getElementById("myId");
                                                target.style.marginLeft= a + 'px';
                                        }
                        </script>
                        <div id="myId"></div>
        </body>
</html>
```

window history

```
=================
<!DOCTYPE html>
<html>
        <head>
                <title>IHUB Talent</title>
                <style>
                        a
                        {
                                text-decoration: none;
                                color:blue;
                        }
                </style>
        </head>
        <body>
                        <a href="javascript:history.back()">&laquo; previous
                        </a>
                        &nbsp; &nbsp;
                        <a href="javascript:history.forward()">
                                        next &raquo;
                        </a>
        </body>
</html>
```

Note:
----
www.ihubtalent.com
file:///D:/IHUB-TRAINING-BATCHES/ReactAngularBatch/practicals/index.html
www.qualitythought.in

localStorage

================

A localStorage properties allows us to save key/value pairs in a browser window.
A localStorage allows us to store the data with no-expiry.It means our data will not be delete even if we close the browser.It will be present for next day.

A localStorage is a read-only.
To add the data in a localStorage we need to use setItem(key,value) function.
To read the data from localStorage we need to use getItem(key) function.
To remove perticular data from localStorage we need to use removeItem(key) function.
To remove all the data from localStorage we need to use clear() function.
ex:

```
<!DOCTYPE html>
<html>
<head>
	<!-- page title -->
	<title>IHUB TALENT</title>
</head>
<body>
		<script type="text/javascript">
			//set the items
			localStorage.setItem("FirstName","Alan");
			localStorage.setItem("LastName","Morries");

			//reading the items
			document.writeln(localStorage.getItem("FirstName")+"<br>");
			document.writeln(localStorage.getItem("LastName")+"<br>");

			//remove perticular item
			localStorage.removeItem("LastName");

			//remove all items
			localStorage.clear();

			//reading the items
			document.writeln(localStorage.getItem("FirstName")+"<br>");
			document.writeln(localStorage.getItem("LastName")+"<br>");
		</script>
</body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
<head>
	<!-- page title -->
	<title>IHUB TALENT</title>
</head>
<body>
		<script type="text/javascript">
			//set the items
			localStorage.setItem("FirstName","Alan");
			localStorage.setItem("FirstName","Morries");

			//reading the items
			document.writeln(localStorage.getItem("FirstName")+"<br>"); //Morries
			</script>
</body>
</html>
```

sessionStorage

A sessionStorage properties allows us to save key/value pair in a browser window.
A sessionStorage store the data with respect to one session.It means our data will be deleted once if we close the browser window.
To add the data in a sessionStorage we need to use setItem(key,value) function.
To read the data from sessionStorage we need to use getItem(key) function.
To remove perticular data from sessionStorage we need to use removeItem(key) function.
To remove all the data from sessionStorage we need to use clear() function.
ex:

```
<!DOCTYPE html>
<html>
<head>
	<!-- page title -->
	<title>IHUB TALENT</title>
</head>
<body>
			<script type="text/javascript">
				//set the items
				sessionStorage.setItem("Name","Alan");
				sessionStorage.setItem("Age",29);

				//reading the items
				document.writeln(sessionStorage.getItem("Name")+"<br>");
				document.writeln(sessionStorage.getItem("Age")+"<br>");

				//remove perticular item
				sessionStorage.removeItem("Age");

				//remove all items
				sessionStorage.clear();

				//reading the items
				document.writeln(sessionStorage.getItem("Name")+"<br>");
				document.writeln(sessionStorage.getItem("Age")+"<br>");s
			</script>
</body>
</html
```

DOM
======

The document object represent whole HTML document.
When HTML document is loaded in a browser it represent document object.
Here HTML document is represented in a tree node hierarcy.
A document object is a root node for entire html document.

DOM always looks for three nodes.

1)Element node
2)Atribute node
3)Text node
Using document object we can add dynamic content to the web page.
A document object is a property of window.It means we can call document object directory or by using window.
ex:

```
	window.document
	or
	document
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<script type="text/javascript">
		window.document.writeln("This is document object"+"<br>");
		document.writeln("This is document object"+"<br>");
		</script>
	</body>
</html>
```

document.write()
===================
It is used to display data or custom messages without space.
ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<script type="text/javascript">
		document.write("This is First Stmt");
		document.write("This is Second Stmt");
		</script>
	</body>
</html>
```

document.writeln()
=====================
It will display the output with space at last.
ex:

```
<!DOCTYPE html>
```

```
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<script type="text/javascript">
		document.writeln("This is First Stmt");
		document.writeln("This is Second Stmt");
		</script>
	</body>
</html>
```

document.getElementById()
==========================
It is used to read the elements based on id.

ex:1

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>

		Name : <input type="text" id="t1"/> <br>

		<button onclick="f1()">submit</button>

		<script type="text/javascript">
			function f1()
			{
				var name=document.getElementById('t1').value;
				document.writeln("Welcome :"+name);
			}
		</script>
</body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>

		Name : <input type="text" id="t1"/> <br>
		<button onclick="f1()">submit</button>
		<script type="text/javascript">
```

```
            function f1()
            {
                var name=document.getElementById('t1').tagName;
                document.writeln(name);
            }
        </script>
</body>
</html>
```

ex:3

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        Name : <input type="text" id="t1"/> <br>
        <button onclick="f1()">submit</button>
        <br><br>
        <h1 id="result"></h1>
        <script type="text/javascript">
            function f1()
            {
                var name=document.getElementById('t1').value;
                document.getElementById('result').innerHTML=name;
            }
        </script>
</body>
</html>
```

Javascript program to add two text field data

==

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        <table align="center">
            <tr>
                <td>No1:</td>
                <td><input type="text" id="t1"/></td>
            </tr>
            <tr>
                <td>No2:</td>
                <td><input type="text" id="t2"/></td>
            </tr>
            <tr>
                <td><button onclick="f1()">ADD</button></td>
                <td><span id="result"></span></td>
            </tr>
        </table>

        <script type="text/javascript">
            function f1()
            {
                var val1=document.getElementById('t1').value;
                var val2=document.getElementById('t2').value;
                var c=parseFloat(val1)+parseFloat(val2);
                document.getElementById('result').innerHTML=c;
            }
        </script>
</body>
</html>
```

Javascript program to hide and show a portion of a form page

```
===============================================================
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
    <fieldset id="curr_id">
        <legend>Current Address</legend>
        <table align="center">
            <tr>
                <td>House No:</td>
                <td><input type="text" id="t1"/></td>
            </tr>
            <tr>
                <td>Locality:</td>
                <td><input type="text" id="t2"/></td>
            </tr>
            <tr>
                <td>City:</td>
                <td><input type="text" id="t3"/></td>
            </tr>
            <tr>
                <td>State:</td>
                <td><input type="text" id="t4"/></td>
            </tr>
        </table>
    </fieldset>
    <br>
    <input type="checkbox" id="box" onclick="f1()" />Current Address same as permanent
address?
    <br>
    <br>
    <fieldset id="per_id">
        <legend>Permanent Address</legend>
        <table align="center">
            <tr>
                <td>House No:</td>
                <td><input type="text" id="t1"/></td>
            </tr>
            <tr>
                <td>Locality:</td>
                <td><input type="text" id="t2"/></td>
            </tr>
            <tr>
                <td>City:</td>
                <td><input type="text" id="t3"/></td>
            </tr>
            <tr>
                <td>State:</td>
                <td><input type="text" id="t4"/></td>
            </tr>
        </table>
    </fieldset>
    <script type="text/javascript">
        function f1()
        {
            if(document.getElementById('box').checked)
            {
                document.getElementById('per_id').style.display="none";
            }
            else
            {
                document.getElementById('per_id').style.display="block";
            }
        }
    </script>
</body>
</html>
```

document.getElementsByName()
===========================

It is used to read the elements by a specified name.

ex:1

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        Name: <input type="text" name="t1"/> <br>
        <button onclick="f1()">submit</button>
        <script type="text/javascript">
            function f1()
            {
                var name=document.getElementsByName('t1')[0].value;
                document.writeln(name);
            }
        </script>
</body>
</html>
ex:2
------
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        Name: <input type="text" name="t1"/> <br>
        <button onclick="f1()">submit</button>
        <script type="text/javascript">
            function f1()
            {
                var name=document.getElementsByName('t1')[0].tagName;
                document.writeln(name);
            }
        </script>
</body>
</html>
ex:3
-----
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        Name: <input type="text" name="t1"/> <br>
        <button onclick="f1()">submit</button>
        <script type="text/javascript">
            function f1()
            {
                var name=document.getElementsByName('t1').length;
                document.writeln(name);
            }
        </script>
</body>
</html>
```

Javascript program to add first name and last name

```
=========================================
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
```

```
</head>
<body>
        First Name: <input type="text" name="t1"/> <br>
        Last Name: <input type="text" name="t2"/> <br>
        <button onclick="f1()">submit</button>
        <script type="text/javascript">
            function f1()
            {
                var fname=document.getElementsByName('t1')[0].value;
                var lname=document.getElementsByName('t2')[0].value;
                document.writeln(fname+lname);
            }
        </script>
</body>
</html>
```

```
ex:2
------
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        First Name: <input type="text" name="t1"/> <br>
        Last Name: <input type="text" name="t1"/> <br>
        <button onclick="f1()">submit</button>
        <script type="text/javascript">
            function f1()
            {
                var fname=document.getElementsByName('t1')[0].value;
                var lname=document.getElementsByName('t1')[1].value;
                document.writeln(fname+lname);
            }
        </script>
</body>
</html>
```

Javascript program to select multiple checkboxes

```
======================================
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        <h1>Courses Completed</h1>
        <button onclick="f1()">All</button>
        <br><br>
        <input type="checkbox" name="c1" value="html">HTML <br>
        <input type="checkbox" name="c1" value="css">CSS <br>
        <input type="checkbox" name="c1" value="js">JavaScript <br>
        <input type="checkbox" name="c1" value="bootstrap">Bootstrap <br>
        <script type="text/javascript">
            function f1()
            {
                var x=document.getElementsByName('c1');

                for(var i=0;i<x.length;i++)
                {
                    if(x[i].type="checkbox")
                    {
                        x[i].checked=true;
                    }
                }
            }
        </script>
</body>
</html>
```

document.getElementsByTagName()

===================================

It is used to read the elements by using a specified tag name.

ex:1

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        Name : <input type="text" /> <br>
        <button onclick="f1()">submit</button>
        <script type="text/javascript">
            function f1()
            {
                var name=document.getElementsByTagName('input')[0].value;
                document.writeln(name);
            }
        </script>
</body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        Name : <input type="text" /> <br>
        <button onclick="f1()">submit</button>
        <script type="text/javascript">
```

```
            function f1()
            {
                var name=document.getElementsByTagName('input')[0].tagName;
                document.writeln(name);
            }
        </script>
</body>
</html>
ex:3
-------
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        Name : <input type="text" /> <br>
        <button onclick="f1()">submit</button>
        <script type="text/javascript">
            function f1()
            {
                var name=document.getElementsByTagName('input').length;
                document.writeln(name);
            }
        </script>
</body>
</html>
ex:4
------
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        <div>This is div1</div>
        <div>This is div2</div>
        <div>This is div3</div>
        <button onclick="f1()">change</button>
        <script type="text/javascript">
            function f1()
            {
                 var x=document.getElementsByTagName('div');
                 x[0].innerHTML="This is javascript class";
                 x[1].innerHTML="This is DOM topic";
                 x[2].innerHTML="This is ihub talent";
            }
        </script>
</body>
</html>
```

```
ex:5
-----
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        <div>This is div1</div>
        <div>This is div2</div>
        <div>This is div3</div>
        <button onclick="f1()">change</button>
        <script type="text/javascript">
            function f1()
            {
                var x=document.getElementsByTagName('div');
                x[0].style.color="red";
                x[1].style.backgroundColor="yellow";
                x[2].style.textAlign="center";
            }
        </script>
</body>
</html>
```

document.getElementsByClassName()

--

It is used to read the elements by using a specified class name.

```
ex:1
-----
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
    <style type="text/css">
        .myClass
        {
            width: 100px;
            height: 100px;
            background-color: red;
        }
    </style>
</head>
<body>
        <div class="myClass"></div>
        <button onclick="f1()">submit</button>
        <script type="text/javascript">
            function f1()
            {
                var len=document.getElementsByClassName('myClass').length;
                document.writeln(len);
            }
        </script>
</body>
</html>
ex:2
------
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
    <style type="text/css">
        .myClass
        {
            width: 100px;
            height: 100px;
            background-color: red;
        }
    </style>
</head>
<body>
        <div class="myClass"></div>
        <button onclick="f1()">submit</button>
        <script type="text/javascript">
            function f1()
            {

    Varname=document.getElementsByClassName('myClass')[0].tagName;
                document.writeln(name);
            }
        </script>
</body>
</html>
```

document.addEventListener()

========================

It is used to add the handler to a function.

```
ex:1
-----
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
    <style type="text/css">
        .myClass
        {
            width: 100px;
            height: 100px;
            background-color: red;
        }
    </style>
</head>
<body>
```

```
        <h1>Click Anywhere </h1>
        <script type="text/javascript">
            document.addEventListener("click",function(){
                alert("You have clicked!");
            })
        </script>
</body>
</html>
ex:2
------
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
    <style type="text/css">
        .myClass
        {
            width: 100px;
            height: 100px;
            background-color: red;
        }
    </style>
</head>
<body>
        <h1>Click Anywhere </h1>
        <script type="text/javascript">
            document.addEventListener("click",f1);
            function f1()
            {
                alert("Yahoo! you clicked");
            }
        </script>
</body>
</html>
ex:3
-----
<!DOCTYPE html>
<html>
<head>
<title>MyPage!</title>
    <style type="text/css">
        .myClass
        {
            width: 100px;
            height: 100px;
            background-color: red;
        }
    </style>
</head>
<body>
        <h1>Mouse over here !!! </h1>
        <br>
        <span id="result"></span>
        <script type="text/javascript">
            document.addEventListener("mouseover",f1);
            function f1()
            {
                document.getElementById('result').innerHTML="mouse over";
            }
        </script>
</body>
</html>
ex:4
-----
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
    <style type="text/css">
        .myClass
        {
            width: 100px;
            height: 100px;
            background-color: red;
        }
    </style>
</head>
<body>
        <h1 id="hover">Mouse out here !!! </h1>
        <br>
        <span id="result"></span>
        <script type="text/javascript
            var y=document.getElementById('hover');
            y.addEventListener("mouseout",f1);
            function f1()
            {
                document.getElementById('result').innerHTML="mouse out";
            }
        </script>
</body>
</html>
```

Javascript program to convert Feet to Inches

====================================

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        <table align="center">
            <tr>
```

```
                <td>Feet</td>
                <td>&nbsp;</td>
                <td>Inches</td>
            </tr>
            <tr>
                <td><input type="text" id="feet"/></td>
                <td><big>=</big></td>
                <td><input type="text" id="inches"/></td>
            </tr>
        </table>
        <script type="text/javascript">
            var feet=document.getElementById('feet');
            var inches=document.getElementById('inches');
            feet.addEventListener('input',function(){
                var f=this.value;
                var i=f*12;
                inches.value=i;

            })
            inches.addEventListener('input',function(){
                var i=this.value;
                var f=i/12;
                if(!Number.isInteger(f))
                {
                    f=f.toFixed(2);
                }
                feet.value=f;
            })
        </script>
</body>
</html>
```

Javascript program on money conversion application

==

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
    <table align="center">
        <tr>
            <td>USD</td>
            <td><input type="text" id="usd"/></td>
            <td><big>=</big></td>
            <td>INR</td>
            <td><input type="text" id="inr"/></td>
        </tr>
    </table>
    <script type="text/javascript">
        var usd=document.getElementById('usd');
        var inr=document.getElementById('inr');
        usd.addEventListener('input',function(e){
                var dollar=e.target.value;
                var rupees=dollar*81.85;
                inr.value=rupees;
        })
    </script>
</body>
</html>
```

removeEventListener()

==========================

It is used to remove the handler from the function.
ex:

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
    <h1 id="hover">Mouse Over Here !</h1>
    <button onclick="stop()">stop event</button>
    <div id="result"></div>
    <script type="text/javascript">
        var y=document.getElementById('hover');
        y.addEventListener('mouseover',mouseOver);
        function mouseOver()
        {
            document.getElementById('result').innerHTML+="<p>Mouse is over</p>";
        }
        function stop()
        {
            y.removeEventListener('mouseover',mouseOver);
        document.getElementById('result').innerHTML+="<p>Mouse is stopped</p>";
        }
    </script>
</body>
</html>
```

JavaScript Date object

===================

JavaScript Date is used to display date and time.
Using javascript Date we can display timer on the web page.
ex:1

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
```

```
    <script type="text/javascript">
        var d=new Date();
        var h=d.getHours();
        var m=d.getMinutes();
        var s=d.getSeconds();
        document.writeln(h+":"+m+":"+s);
    </script>
</body>
</html>
ex:2
-----
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
    <script type="text/javascript">
        var d=new Date();
        var dd=d.getDate();
        var mm=d.getMonth()+1;
        var yy=d.getFullYear();
        document.writeln(dd+"/"+mm+"/"+yy);
    </script>
</body>
</html>

ex:3
-----
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>

    <!-- add external css -->
    <link rel="stylesheet" type="text/css" href="css/mystyles.css">
</head>
<body>
        <span id="result"></span>
        <script type="text/javascript">
            window.onload=function(){getTime();}
            function getTime()
            {
                var date=new Date();
                var h=date.getHours();
                var m=date.getMinutes();
                var s=date.getSeconds();
                m=check(m);
                s=check(s);
                document.getElementById('result').innerHTML=h+":"+m+":"+s;
            }
            function check(i)
            {
                if(i<10)
                {
                    i = "0"+i;
                }
                return i;
            }
            setInterval(getTime,1000);
        </script>
</body>
</html>
```

mystyles.css

```
*
{
    margin: 0;
    padding: 0;
}
body
{
    height: 100vh;
    display: flex;
    justify-content: center;
    align-items: center;
    background:linear-gradient(yellow,red);
}
span
{
    padding:10px;
    width:80px;
    box-sizing: border-box;
    box-shadow: 2px 2px 9px 5px #FFF;
}
```

Q)What is the difference between innerHTML and innerText ?

innerText:

==========

The innerText property is used to write the simple text using JavaScript dynamically.

ex:1

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        <h1 id="result"></h1>
        <button onclick="f1()"> clickHere </button>
        <script type="text/javascript">
            function f1()
            {
                document.getElementById('result').innerText="This is javascript";
            }
        </script>
</body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        <h1 id="result"></h1>
        <button onclick="f1()"> clickHere </button>
        <script type="text/javascript">
            function f1()
            {
                document.getElementById('result').innerText="<font
color='red'>This is javascript</font>";
            }
        </script>
</body>
</html>
```

innerHTML:

The innerHTML property is used to write the HTML code using JavaScript
dynamically.

ex:1

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        <h1 id="result"></h1>
        <button onclick="f1()"> clickHere </button>
        <script type="text/javascript">
            function f1()
            {
                document.getElementById('result').innerHTML="This is javascript";
            }
        </script>
</body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
<head>
		<title>MyPage!</title>
</head>
<body>
				<h1 id="result"></h1>
				<button onclick="f1()"> clickHere </button>
				<script type="text/javascript">
						function f1()
						{
								document.getElementById('result').innerHTML="<font
color='red'>This is javascript</font>";
						}
				</script>
</body>
</html>
```

Q) What is JavaScript hoisting?

Hoisting is the default behavior of JavaScript where all the variable and function declarations are moved on top.
This means that irrespective of where the variables and functions are declared, they are moved on top of the scope.
The scope can be both local and global.

Ex1:

```
	i=100;                          var i;
	document.writeln(i);  ==>       i=100;
	var i;                          document.writeln(i);
```

Ex2:

```
	//calling                                   //declaring
	f1();                                       function f1()
	                                            {
	//declaring                                         document.writeln("Hello");
	function f1()                               }
	{
		document.writeln("Hello"); =>       //calling
	}                                           f1();
```

Q)What are the types of errors in JavaScript?

There are two types of errors in JavaScript.

1)Syntax error:

Syntax errors are mistakes or spelling problems in the code that cause the program to not execute at all or to stop running halfway through.

2)Logical error:

Reasoning mistakes occur when the syntax is proper but the logic or program is incorrect. The application executes without problems in this case. However, the output findings are inaccurate.

How to hide and show password in a text field

==

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
    <!-- fontawesome icon cdn link -->
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css" />
    <style type="text/css">
        .myClass
        {
            padding-right: 25px;
        }
        #myId
        {
            position: relative;
            right: 25px;
        }
    </style>
</head>
<body>
        Password: <input type="password" id="t1" class="myClass"/>
        <span class="fa fa-eye" id="myId" onclick="f1()"></span>
        <script type="text/javascript">
            function f1()
            {
                var x=document.getElementById('t1');
                if(x.type="password")
                {
                    x.type="text";
                }
                else
                {
                    x.type="password";
                }
            }
        </script>
</body>
</html>
```

JavaScript Form validation

=====================

The process of checking format and pattern of form data is called form validation. There are two ways to perform form validation.

1)**Client side form validation**

Validation which is performed at client side is called client side form validation.
To perform client side form validation we need to use javascript.

2)Server side form validation

Validation which is performed at server side is called server side form validation.
To perform server side form validation we need to use php,nodejs,expressjs and etc.

ex:1

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
	<script type="text/javascript">
		function validate()
		{
			var name=document.getElementById('t1').value;
			if(name=="")
			{
				alert("Name is mandatory");
				document.getElementById('t1').focus();
				return false;
			}
			return true;
		}
	</script>
</head>
<body>
		<form action="#"  onsubmit="return validate()">
			Name : <input type="text" id="t1"/> <br>
			<input type="submit" value="submit"/>
		</form>
</body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
	<script type="text/javascript">
		function validate()
		{
			var name=document.getElementById('t1').value;
			if(name=="")
			{
				alert("Name is mandatory");
				document.getElementById('t1').focus();
				return false;
```

```
            }
            if(name.length<6)
            {
                alert("Name must have 6 characters");
                document.getElementById('t1').value="";
                document.getElementById('t1').focus();
                return false;
            }
            return true;
        }
    </script>
</head>
<body>
        <form action="#"  onsubmit="return validate()">
            Name : <input type="text" id="t1"/> <br>
            <input type="submit" value="submit"/>
        </form>
</body>
</html>
ex:3
------
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
    <script type="text/javascript">
        function validate()
        {
            var age=document.getElementById('t1').value;
            if(age=="")
            {
                alert("Age is mandatory");
                document.getElementById('t1').focus();
                return false;
            }
            if(isNaN(age))
            {
                alert("Age must be numeric");
                document.getElementById('t1').value="";
                document.getElementById('t1').focus();
                return false;
            }
            return true;
        }
    </script>
</head>
<body>
        <form action="#"  onsubmit="return validate()">
            Age : <input type="text" id="t1"/> <br>
            <input type="submit" value="submit"/>
        </form>
```

```
</body>
</html>

ex:4
------
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
    <script type="text/javascript">
        function validate()
        {
            var pwd=document.getElementById('t1').value;
            var cpwd=document.getElementById('t2').value;
            if(pwd=="")
            {
                alert("password is mandatory");
                document.getElementById('t1').focus();
                return false;
            }
            if(cpwd=="")
            {
                alert("Confirm password is mandatory");
                document.getElementById('t2').focus();
                return false;
            }
            if(pwd!=cpwd)
            {
                alert("Both password must be same");
                document.getElementById('t1').value="";
                document.getElementById('t2').value="";
                document.getElementById('t1').focus();
                return false;
            }
            return true;
        }
    </script>
</head>
<body>
        <form action="#"  onsubmit="return validate()">
            Password : <input type="password" id="t1"/> <br>
            Confirm Password : <input type="password" id="t2"/> <br>
            <input type="submit" value="submit"/>
        </form>
</body>
</html>

ex:5
---------
<!DOCTYPE html>
<html>
```

```
<head>
        <title>MyPage!</title>
        <script type="text/javascript">
                function validate()
                {
                        var email=document.getElementById('t1').value;
                        var atPosition=email.indexOf('@');
                        var dotPosition=email.lastIndexOf('.');
                        if(atPosition<1 || dotPosition<atPosition+1)
                        {
                                alert("Enter valid mail id");
                                document.getElementById('t1').focus();
                                return false;
                        }
                        return true;
                }
        </script>
</head>
<body>
                <form action="#"  onsubmit="return validate()">
                        Email : <input type="text" id="t1"/> <br>
                        <input type="submit" value="submit"/>
                </form>
</body>
</html>
```

JavaScript Regular Expression

========================

Regular expressions are patterns used to match character combinations in strings.
In JavaScript, regular expressions are also objects.

JavaScript Form validation using RegularExpression

==

To generate proper regular expression we can login below url.
ex:
        https://regex101.com/
ex:

```
<!DOCTYPE html>
<html>
<style>
input[type=text],input[type=password], select {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  border-radius: 4px;
  box-sizing: border-box;
}
input[type=submit] {
  width: 100%;
  background-color: #4CAF50;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}
input[type=submit]:hover {
  background-color: #45a049;
}
div {
  border-radius: 5px;
  background-color: #f2f2f2;
  padding: 20px;
  width:500px;
  position: relative;
  left:200px;
  top:20px;
}
</style>
<script type="text/javascript">
  function validate()
  {
    var name=document.getElementById('name').value;
    var pwd=document.getElementById('pwd').value;
    var phone=document.getElementById('phone').value;
    var email=document.getElementById('email').value;
    var country=document.getElementById('country').value;

    var namecheck=/[A-Za-z. ]{6,20}$/;
    var pwdcheck=/(?=.*[0-9])(?=.*[!@#$%^&*])(?=.*[A-Z])[a-zA-Z0-9!@#$%^&*]{10,30}$/;
    var phonecheck=/[789][0-9]{9}$/;
    var emailcheck=/[A-Za-z.]{1,}@[A-Za-z]{2,15}[.][A-Za-z]{3,}$/;

    if(!(namecheck.test(name)))
    {
        alert("UserName must be 6 characters");
        document.getElementById('name').value="";
        document.getElementById('name').focus();
        return false;
    }
    if(!(pwdcheck.test(pwd)))
    {
      alert("password must have 1 uppercase, 1 special symbol and 1 digit");
      document.getElementById('pwd').value="";
      document.getElementById('pwd').focus();
       return false;
    }
    if(!(phonecheck.test(phone)))
    {
      alert("Phone must start with 7,8,9 series with  10 digits");
      document.getElementById('phone').value="";
      document.getElementById('phone').focus();
       return false;
    }
    if(!(emailcheck.test(email)))
    {
      alert("Please insert valid email");
      document.getElementById('email').value="";
      document.getElementById('email').focus();
       return false;
    }
    if(country=="")
    {
      alert("Please select the country option ");
      return false;
    }
    return true;
}
</script>
<body>
<div>
  <form action="/action_page.php" onsubmit="validate()">

    <label for="name">UserName</label>
    <input type="text" id="name" name="name" placeholder="Your username.."/>

    <label for="pwd">Password</label>
    <input type="text" id="pwd" name="pwd" placeholder="Your password.."/>

    <label for="phone">Phone</label>
    <input type="text" id="phone" name="phone" placeholder="Your phone.."/>

     <label for="email">Email</label>
    <input type="text" id="email" name="email" placeholder="Your email.."/>

    <label for="country">Country</label>
    <select id="country" name="country">
      <option value="">none</option>
      <option value="australia">Australia</option>
      <option value="canada">Canada</option>
      <option value="usa">USA</option>
    </select>

    <input type="submit" value="Submit">
  </form>
</div>

</body>
</html>
```

Synchronous and Asynchronous in JavaScript
==

Synchronous JavaScript:

As the name suggests synchronous means to be in a sequence, i.e. every statement of the code gets executed one by one. So, basically a statement has to wait for the
earlier statement to get executed.

ex:

```
<script>
   document.write("Hi"); // First
   document.write("<br>");

   document.write("IHUB TALENT") ;// Second
   document.write("<br>");

   document.write("How are you"); // Third
</script>
```

Asynchronous JavaScript:

Asynchronous code allows the program to be executed immediately where the synchronous code will block further execution of the remaining code until it finishes the current one. This may not look like a big problem but when you see it in a bigger picture you realize that it may lead to delaying the User Interface.

ex:

```
<script>
   document.write("Hi");
   document.write("<br>");

   setTimeout(function() {
      document.write("Let us see what happens");
   }, 2000);
   document.write("<br>");
   document.write("End");
   document.write("<br>");
</script>
```

ex:

```
<script>
   document.write("Hi");
   document.write("<br>");

   setTimeout(() => {
      document.write("Let us see what happens");
   }, 2000);
   document.write("<br>");
   document.write("End");
   document.write("<br>");
</script>
```

Javascript promises
=================

Promises are used to handle asynchronous operations in JavaScript.
They can handle multiple asynchronous operations easily and provide better error handling than callbacks and events.

A Promise has four states:

1)fulfilled: Action related to the promise succeeded
2)rejected: Action related to the promise failed
3)pending: Promise is still pending i.e. not fulfilled or rejected yet
4)settled: Promise has fulfilled or rejected

A promise can be created using Promise constructor.
Syntax:

```
        var promise = new Promise(function(resolve, reject){
                //do something
        });
```

ex:1

```
<script>
var promise = new Promise(function(resolve, reject) {
        resolve('IHub Talent');
})
promise
        .then(function(successMessage) {
        //success handler function is invoked
                console.log(successMessage);
        }, function(errorMessage) {
                console.log(errorMessage);
        })
</script>
```

ex:2

```
<script>
var promise = new Promise(function(resolve, reject) {
        reject('Error occured');
})
promise
        .then(function(successMessage) {
        //success handler function is invoked
                console.log(successMessage);
        }, function(errorMessage) {
                console.log(errorMessage);
        })
</script>
```

ex:3

```
<script>
var promise = new Promise(function(resolve, reject) {
const x = "ihubtalent";
const y = "ihubtalent1";
if(x === y) {
        resolve();
} else {
        reject();
}
});
promise.
        then(function () {
                console.log('Success, You are a GEEK');
        }).
        catch(function () {
                console.log('Some error has occurred');
        });
</script>
```

Q)Differences between var , let and const?

var	**let**	**const**
-----	-----	--------
It is a functional scope.	It is a block scope.	It is a block scope.
It can be declare without initialization.	It can be declare without initialization.	It can't be declare without initialization.
It can be updated.	It can be updated.	It can't be updated.
It can be redeclared.	It can't be redeclared.	It can't be redeclared.
It can be access without initialization as it default value is undefined.	It can be access without initialization as it default value is undefined.	It can’t be access without initialization.

Intialization
----

ex:1

```
<!DOCTYPE html>
<html>
        <head>
                <title>mypage!</title>
        </head>
        <body>
                <script type="text/javascript">
                        var i;
                        document.writeln(i);//undefined
                </script>
        </body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
        <head>
                <title>mypage!</title>
        </head>
        <body>
                <script type="text/javascript">
                        let i;
                        document.writeln(i);//undefined
                </script>
        </body>
</html>
```

ex:3

```
<!DOCTYPE html>
<html>
        <head>
                <title>mypage!</title>
        </head>
        <body>
                <script type="text/javascript">
                        const i;
                        document.writeln(i);//invalid

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

Update
----

ex:1

```
<!DOCTYPE html>
<html>
        <head>
                <title>mypage!</title>
        </head>
        <body>
                <script type="text/javascript">
                        var i=10;
                        i=20;
                        document.writeln(i);//20
                </script>
        </body>
</html>
```

ex:2

```
<!DOCTYPE html>
```

```
<html>
	<head>
		<title>mypage!</title>
	</head>
	<body>
		<script type="text/javascript">

			let i=10;
			i=20;
			document.writeln(i);//20
		</script>
	</body>
</html>
```

ex:3

```
<!DOCTYPE html>
<html>
	<head>
		<title>mypage!</title>
	</head>
	<body>
		<script type="text/javascript">
			const i=10;
			i=20;
			document.writeln(i);//invalid
		</script>
	</body>
</html>
```

Redeclared
----

ex:1

```
<!DOCTYPE html>
<html>
	<head>
		<title>mypage!</title>
	</head>
	<body>
		<script type="text/javascript">

			var i=10;
			var i=20;
			document.writeln(i);//20

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

```
ex:2
------
<!DOCTYPE html>
<html>
    <head>
        <title>mypage!</title>
    </head>
    <body>
        <script type="text/javascript">

            let i=10;
            let i=20;
            document.writeln(i);//

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

```
ex:3
----
<!DOCTYPE html>
<html>
    <head>
        <title>mypage!</title>
    </head>
    <body>
        <script type="text/javascript">

            const i=10;
            const i=20;
            document.writeln(i);//

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

How to store form data in a localStorage

```
======================================
<!DOCTYPE html>
<html>
    <head>
        <title>mypage!</title>
    </head>
    <body>
            <form>
                <table align="center">
                    <tr>
                        <td>No:</td>
                        <td><input type="text" id="t1"/></td>
                    </tr>
                    <tr>
```

```
                                        <td>Name:</td>
                                        <td><input type="text" id="t2"/></td>
                                </tr>
                                <tr>
                                        <td>Address:</td>
                                        <td><input type="text" id="t3"/></td>
                                </tr>
                                <tr>
                                        <td><input type="reset" value="reset"/></td>
                        <td><input type="submit" value="submit" onclick="f1()"/></td>
                                </tr>
                        </table>
                </form>

                <script type="text/javascript">
                        function f1()
                        {
                                //reading form data
                                var no=document.getElementById('t1').value;
                                var name=document.getElementById('t2').value;
                                var add=document.getElementById('t3').value;

                                //store the items to localStorage

                                localStorage.setItem("studNo",no);
                                localStorage.setItem("studName",name);
                                localStorage.setItem("studAdd",add);
                        }
                </script>
        </body>
</html>
```

Javascript Set
===============

A JavaScript Set is a collection of unique values.
Each value can only occur once in a Set.
A Set can hold any value of any data type.

ex:1

```
<!DOCTYPE html>
<html>
<head>
        <title>IHUB TALENT</title>
</head>
<body>
                <script type="text/javascript">
                        var letters=new Set();
                        letters.add(6);
                        letters.add(1);
                        letters.add(1);
```

```
                    letters.add(5);
                    letters.add(9);
                    letters.forEach(function(value){
                            document.writeln(value);
                    })

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

ex:2

```
<!DOCTYPE html>
<html>
<head>
        <title>IHUB TALENT</title>
</head>
<body>
            <script type="text/javascript">
                    var letters=new Set([6,1,5,9,1,6]);
                    letters.forEach(function(value){
                            document.writeln(value);
                    })

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

ex:3

```
<!DOCTYPE html>
<html>
<head>
        <title>IHUB TALENT</title>
</head>
<body>
            <script type="text/javascript">
                    var letters=new Set([6,1,5,9]);

                    // Create an Iterator
                    const myIterator = letters.values();

                    // List all Values
                    for (const entry of myIterator)
                    {
                            document.writeln(entry+"<br>");
                    }

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

```
ex:4
------
<!DOCTYPE html>
<html>
<head>
    <title>IHUB TALENT</title>
</head>
<body>
        <script type="text/javascript">
            var letters=new Set(["a","b","c","d"]);

            // Create an Iterator
            const myIterator = letters.values();

            // List all Values
            for (const entry of myIterator)
            {
                document.writeln(entry+"<br>");
            }

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

JavaScript Maps

=================

A Map holds key-value pairs where the keys can be any datatype.
A Map remembers the original insertion order of the keys.

```
ex:1
-------
<!DOCTYPE html>
<html>
<head>
    <title>IHUB TALENT</title>
</head>
<body>
        <script type="text/javascript">
            // Create a Map
        const myMap = new Map([
                ["one", 1],
                ["two", 2],
                ["three", 3]
        ]);
        document.writeln(myMap.size); //3
        document.writeln(myMap.get("one"));//1
        myMap.delete("three");
        myMap.clear();
        </script>
</body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
<head>
        <title>IHUB TALENT</title>
</head>
<body>
                <script type="text/javascript">
                        // Create a Map
                const myMap = new Map([
                                ["one", 1],
                                ["two", 2],
                                ["three", 3]
                ]);
                myMap.forEach(function(value, key)
                {
                document.writeln(value+" "+key+"<br>");
                })


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

ex:3

```
<!DOCTYPE html>
<html>
<head>
        <title>IHUB TALENT</title>
</head>
<body>
                <script type="text/javascript">
                        // Create a Map
                const myMap = new Map([
                                ["one", 1],
                                ["two", 2],
                                ["three", 3]
                ]);
                for (const x of myMap.keys()) {
                 document.writeln(x);
        }


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

ex:4

```
<!DOCTYPE html>
<html>
<head>
	<title>IHUB TALENT</title>
</head>
<body>
		<script type="text/javascript">
			// Create a Map
		const myMap = new Map([
				["one", 1],
				["two", 2],
				["three", 3]
		]);
		for (const x of myMap.values()) {
		 document.writeln(x);
	}


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

Q)What is JavaScript Math object?

The JavaScript Math object allows you to perform mathematical tasks on numbers.
ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>IHUB TALENT</title>
</head>
<body>
		<script type="text/javascript">

		document.writeln(Math.ceil(10.6));
		document.writeln(Math.floor(10.6));
		document.writeln(Math.round(10.6));
		document.writeln(Math.trunc(10.56));
		</script>
</body>
</html>
```

Object Oriented Programming System / Structure (OOPS)
===

A technology or language said to be object oriented if it supports following features.
ex:

	class
	object
	abstraction

encapsulation
inheritance
and
polymorphism

Q)What is class in JavaScript?

A JavaScript class is not an object.
It is a template for JavaScript objects.
Use the class keyword to create a class.
A class keyword is used to declare a class with any particular name.
According to JavaScript naming conventions, the name of the class always starts with an uppercase letter.
Ex:

```
	<script>
	class Example
	{
		-
		-//code to be declare
		-
	}
</script>
```

Q)What is Constructor in JavaScript?

A JavaScript constructor is a special type of method which is used to initialize and create an object.
It is called when memory is allocated for an object.
The constructor keyword is used to declare a constructor method.
The class can contain one constructor method only.
JavaScript allows us to use parent class constructor through super keyword.
Ex:

```
	class Example
	{
		constructor()
		{
			-
			-// code to be declare
			-
		}
	}
```

Q)What is object in JavaScript?

A JavaScript object is an entity having state and behavior (properties and method).
Syntax:

```
	var objectname =new Object();
```

ex:1

```
<!DOCTYPE html>
<html>
<head>
```

```
    <title>IHUB TALENT</title>
</head>
<body>
        <script type="text/javascript">
            class  Example
            {
                constructor()
                {
                    document.writeln("Hello World");
                }
            }
            var e=new Example();
        </script>
</body>
</html>

ex:2
------
<!DOCTYPE html>
<html>
<head>
    <title>IHUB TALENT</title>
</head>
<body>
        <script type="text/javascript">
            class  Example
            {
                constructor()
                {
                    document.writeln("Hello World");
                }
            }
            var e1=new Example();
            var e2=new Example();
        </script>
</body>
</html>

ex:3
-----
<!DOCTYPE html>
<html>
<head>
    <title>IHUB TALENT</title>
</head>
<body>
        <script type="text/javascript">
            class  Example
            {
                constructor(id)
                {
                    document.writeln(id+"<br>");
                }
            }
            var e1=new Example(101);
            var e2=new Example(201);
        </script>
</body>
</html>
```

Q)What is Abstraction in JavaScript?

Hiding internal implementation and highlighting the set of services is called Abstraction.

The best example of Abstraction is GUI(Graphical User Interface) ATM machine where bank people will hide internal implementation and highlights the set of services like banking, withdrawal, mini statement, balance enquiry and etc.

Q)What is Encapsulation in JavaScript?

The process of wrapping property and function within a single unit is known as encapsulation.

To achieve an encapsulation in JavaScript we need to do following things.
> Use var keyword to make data members private.
> Use setter methods to set the data and getter methods to get that data.

ex:

```
<!DOCTYPE html>
<html>
<head>
    <title>IHUB TALENT</title>
</head>
<body>
        <script type="text/javascript">
            class  Example
            {
                //setter
                constructor(id,name,sal)
                {
                    this.eid=id;
                    this.ename=name;
                    this.esal=sal;
                }

                //getter
                getId()
                {
                    return this.eid;
                }
                getName()
                {
                    return this.ename;
                }
                getSal()
                {
                    return this.esal;
                }

            }
            var e=new Example(101,'Alan',1000.0);
            document.writeln(e.getId()+"<br>");
            document.writeln(e.getName()+"<br>");
            document.writeln(e.getSal()+"<br>");
        </script>
</body>
</html>

ex:2
------
<!DOCTYPE html>
<html>
<head>
    <title>IHUB TALENT</title>
</head>
<body>
        <script type="text/javascript">
            class  Example
            {
                //setter
                setId(id)
                {
                    this.eid=id;
                }
                setName(name)
                {
                    this.ename=name;
                }
                setSal(sal)
                {
                    this.esal=sal;
                }

                //getter
                getId()
                {
                    return this.eid;
                }
                getName()
                {
                    return this.ename;
                }
                getSal()
                {
                    return this.esal;
                }

            }
            var e=new Example(101,'Alan',1000.0);
            e.setId(501);
            e.setName("Jose");
            e.setSal(2000.0);
            document.writeln(e.getId()+"<br>");
            document.writeln(e.getName()+"<br>");
            document.writeln(e.getSal()+"<br>");
        </script>
</body>
</html>
```

Q)What is Inheritance in JavaScript?

The JavaScript inheritance is a mechanism that allows us to create new classes on the basis of already existing classes.

It provides flexibility to the child class to reuse the methods and variables of a parent class.

The JavaScript extends keyword is used to create a child class on the basis of a parent class.

ex:

```
<!DOCTYPE html>
<html>
<head>
    <title>IHUB TALENT</title>
</head>
<body>
        <script type="text/javascript">
            class A
            {
                f1()
                {
                    document.writeln("A-class <br>");
                }
            }
            class B extends A
            {
                f2()
                {
                    document.writeln("B-class <br>");
                }
            }
            var a=new A();
            a.f1();
            var b=new B();
```

```
                b.f1();
                b.f2();
        </script>
</body>
</html>
```

Q)What is polymorphism in JavaScript?

The ability to represent in a different forms is called polymorphism.

Ex:

```
    class A
    {
        display()
        {
            document.writeln("A is invoked<br>");
        }
    }
    class B extends A
    {
        display()
        {
            document.writeln("B is invoked");
        }
    }
    A a=new A();
    a.display(); // A is invoked
    B b=new B();
    b.display(); // B is invoked
```