

JavaScript

=====

Q) What is the difference between Java and JavaScript?

Java

It is a non-scripting language.

It is a Object oriented programming language.

It is a strongly typed checking

language.

It does not required browser window
for execution.

We can run invidually.

It is a complex language.

JavaScript

It is a scripting language.

It is a object based programming language.

It is a weakly typed checking

language.

It requires browser window for execution.

We can't run invidually.

It is easy language.

History of Javascript

=====

Original name of JavaScript is LiveScript.

LiveScript was introduced by Netscape Corporation in late 1990's.

LiveScript was developed by using C language syntax's.

Late in 1995, Brenden Eich the popular scientist of Netscape Corporation renamed LiveScript to JavaScript.

The official name of JavaScript is ECMA script.

ECMA stands for European Computer Manufacturer Association.

Advantages of JavaScript

=====

- > It is used to create interactive web pages.
- > It is used to display dialog boxes and popup boxes.
- > It is used to add dynamic content to the web page.
- > It is used to perform form validation.
- > It is used to create response(dynamic) web page.
- > It supports objects i.e Arrays,Strings,RegExp and etc.
- > It supports Date and Time.

> It supports cookies.

> It supports dropdown menu and etc.

Javascript syntax

=====

```
<script type="text/javascript">
```

```
    stmt1;
```

```
    stmt2;
```

```
    stmt3;
```

```
</script>
```

Here semicolon is not mandatory because javascript is a loosely typed checking language.

Here "type" attribute is optional to declare.

ex:

```
<script>
```

```
    stmt1
```

```
    stmt2
```

```
    stmt3
```

```
</script>
```

Sublime Editor

=====

Download link : <https://www.sublimetext.com/download>

ex:1

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
    <script type="text/javascript">
```

```
        document.writeln("Welcome to JavaScript");
```

```
    </script>
```

```
</body>
```

```
</html>
```

ex:2

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
    <script>
```

```
document.writeln("Welcome to JavaScript")
```

```
</script>
```

```
</body>
```

```
</html>
```

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
    <script>
```

```
        document.writeln("Welcome to JavaScript");
```

```
        document.writeln("<br>");
```

```
        document.writeln("This is ihub talent management");
```

```
    </script>
```

```
</body>
```

```
</html>
```

ex:

```
<!DOCTYPE html>
```

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

        document.writeln("<h1 style='color:red'>Welcome to JavaScript</h1>");
        document.writeln("<br>");
        document.writeln("<p style='color:blue'>This is ihub talent
management</p>");

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

Note:

If a document contains HTML, CSS and javascript then it is called DHTML.

DHTML stands for Dynamic Hypertext Markup Language.

JavaScript Engine

=====

JavaScript engine is used to execute javascript code in browser window.

JavaScript engine converts user understandable scripting language to machine understandable scripting language.

By default every browser contains javascript engine.

We have following list of javascript engines.

ex:

Browser	Javascript Engine
-----	-----
Chrome	V8 Engine
Mozilla	Spider Monkey
Edge	Chakra
and etc.	

Comments in javascript

=====

Comments are created for documentation purpose.

Comments are used to improve readability of our code.

In javascript, we have two types of comments.

1) Single Line Comment

It is used to comment a single line.

ex:

// comment here

2) Multiple Line Comment

It is more convenient because we can comment a single and multiple lines.

ex:

```
/*  
-  
- comment here  
-  
*/
```

Note:

HTML - <!-- comment here -->

```
CSS - /*  
      comment here  
      */
```

Output statement in javascript

=====

Whenever we want to display any data or user defined statements then we need to use output statement.

We have two types of output statement in javascript.

1) document.writeln()

2) console.log()

1) document.writeln()

It is used to display the output on browser window.

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
    <script>
```

```
        document.writeln("Welcome to JavaScript");
```

```
    </script>
```

```
</body>
```

```
</html>
```

2) console.log()

It is used to display the output on browser console.

In order to see the browser console we need to press F12 Function key.

ex:

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
    <script>
        console.log("stmt executed");
    </script>
</body>
</html>
```

Types of Javascript

=====

We have two types of javascripts.

1) Internal javascript / Embedded javascript

2) External javascript / Seperate javascript

1) Internal javascript

In internal javascript, we will add html code and javascript code in a ".html" file.

Advantages:

- > There is no confusion of multiple files.
- > We can maintain HTML code and javascript code separately.

Disadvantage:

- > If code is increases then it will increase the complexity.

Note:

For practising level we need to use internal javascript.

2) External javascript

In external javascript, we will add html code in ".html" file and javascript code in ".js" file.

Advantages:

- > If code is increases then it will not increase the complexity.
- > We can maintain html code and javascript in a seperate files.

Disadvantage:

> There is a confusion of multiple files.

Note:

For company level we need to use external javascript.

Javascript variables

=====

A name which is given to a memory location is called variable.

Purpose of variable is used to store the data.

Javascript variables having same rules as we have for identifiers.

Rule1:

Javascript variable must and should starts with alphabet,dollar symbol and underscore but not with digits.

ex:

var _abcd;

var a123;

var \$abcd;

var 1abcd; //invalid

Rule2:

Every javascript variable is a case sensitive.

ex:

```
var number;  
var NUMBER;  
var NumBer;
```

We have two types of variables in javascript.

1) Local variable

2) Global variable

1) Local variable

A variable which is declared inside the functions and blocks is called local variable.

Local variable we can access with in the function but not outside the function.

ex:

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

```
        function f1()
        {
            //local variable
            var i=10;
            document.writeln(i+"");
        }
        //calling
        f1();
    </script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
    <script>
        function f1()
        {
            //local variable
            var i=10;
            document.writeln(i+"");
        }
    </script>
</body>
</html>
```

```

        function f2()
        {
            document.writeln(i+"<br>");
        }

        //calling
        f1();
        f2();
    </script>
</body>
</html>

```

2) Global variable

A variable which is declared outside the function or a block is called global variable.

Global variable we can access with in the function and outside the function.

ex:

```

<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
    <script>
        //global variable

```

```
var i=100;

function f1()
{
    document.writeln(i+"<br>");
}

//calling
f1();
```

```
</script>
```

```
</body>
```

```
</html>
```

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
    <script>
```

```
        //global variable
```

```
        var i=100;
```

```
        function f1()
```



```
        {
            document.writeln(i+"<br>");
        }
        function f2()
        {
            document.writeln(i+"<br>");
        }

        //calling
        f1();
        f2();

    </script>

</body>
</html>
```

Javascript Datatypes

=====

Javascript is a dynamically typed language it means we don't need to declare any datatype at the time of variable declaration.

To declare any variable in javascript we will use "var" keyword.

Javascript internally uses javascript engine to determine particular 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.

ex:

datatype	description
-----	-----
number	It is used to represent numbers
boolean	It is used to represent boolean
String	It is used to represent string
null	It is used to represent null
undefined	It is used to represent undefined

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
    <script>
```

```
var i=10;  
document.writeln(i+"<br>");//10
```

```
var j=10.5;  
document.writeln(j+"<br>");//10.5
```

```
</script>
```

```
</body>
```

```
</html>
```

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
    <script>
```

```
var i=true;  
document.writeln(i+"<br>");//true
```

```
var j=false;  
document.writeln(j+"<br>");//false
```

```
        </script>

</body>
</html>
```

ex:

```
<!DOCTYPE html>

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

    <script>

        var i="ihub";
        document.writeln(i+"<br>");//ihub

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

    </script>

</body>
</html>
```

ex:

```
<!DOCTYPE html>
```

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

    var i=null;
    document.writeln(i+"<br>"); //null

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

ex:

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

    var i;
    document.writeln(i+"<br>"); //undefined
```

```
</script>

</body>

</html>
```

2) Non-Primitive datatypes

We have following list of non-primitive datatypes.

ex:

datatype	description
-----	-----
Object	It is used to create an instance through which we can access the members.
Array	It is used to represent similar elements.
RegEx	It is used to represent regular expression.

Operator

=====

Operator is a symbol which is used to perform some operations on operands.

ex:

$c = a + b ;$

Here = and + are operators

Here a,b and c are operands.

It can be arithmetic operation, logical operation, bitwise operation and etc.

We have following list of operators present in javascript.

1) Arithmetic operators

2) Assignment operators

3) Logical operators

4) Bitwise operators

5) Relational operators

6) Special operators

1) Arithmetic operators

We have following list of arithmetic operators.

ex:

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

-- Decrementation

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
    <script>
```

```
        var i=10;
```

```
        document.writeln((i%2)+"<br>");//0
```

```
        document.writeln((i%20)+"<br>");//0
```

```
    </script>
```

```
</body>
```

```
</html>
```

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```



```

</head>
<body>
    <script>

        var i=10;

        document.writeln((i/2)+"<br>");//5

        document.writeln((i/20)+"<br>");//0.5

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

```

ex:

```

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

        var i= 10+6%2+7/2+8*9-40;

        document.writeln((i)+"<br>");//

```

```
/*  
    10 + 6%2 + 7/2 + 8*9 - 40  
  
    10 + 0 + 3.5 + 72 - 40  
  
    85.5 - 40 = 45.5  
*/
```

```
</script>
```

```
</body>
```

```
</html>
```

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
    <script>
```

```
        var i=10;
```

```
        var j= i++ + ++i; //10 + 12
```

```
        document.writeln(i+" and "+j);//12  22
```

```
        </script>

</body>
</html>
```

ex:

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

        <script>

                var i=10;

                var j= i-- + --i; //10 + 8

                document.writeln(i+" and "+j);//8 18

        </script>

</body>
</html>
```

2) Assignment operators

We have following list of assignment operators.

ex:

operator	description
-----	-----
=	assignment operator
+=	addition and equals to
-=	subtraction and equals to
%=	modules and equals to
/=	division and equals to
==	equals to
!=	not equals to

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
    <script>
```

```
        var i=10;
```

```
        i+=5;
```

```
document.writeln(i);//15
```

```
</script>
```

```
</body>
```

```
</html>
```

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
    <script>
```

```
        var i=10;
```

```
        i-=5;
```

```
        document.writeln(i);//5
```

```
    </script>
```

```
</body>
```

```
</html>
```

ex:

```
--  
  
<!DOCTYPE html>  
  
<html>  
  
<head>  
    <title>MyPage!</title>  
</head>  
<body>  
    <script>  
  
        var i=10;  
  
        i*=5;  
  
        document.writeln(i);//50  
  
    </script>  
</body>  
</html>
```

ex:

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

```
<script>
```

```
var i=10;
```

```
i/=5;
```

```
document.writeln(i);//2
```

```
</script>
```

```
</body>
```

```
</html>
```

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
  <script>
```

```
var i=10;
```

```
i%=5;
```

```
document.writeln(i);//0
```

```
        </script>

</body>
</html>
```

ex:

```
<!DOCTYPE html>

<html>

<head>

    <title>MyPage!</title>

</head>

<body>

    <script>
```

```
        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) Logical operators

We have following list of logical operators.

ex:

operator	description
-----	-----
&&	Logical AND
	Logical OR
!	Logical NOT

Logical AND (&&)

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>

```
document.writeln(((10>5) && (10<15))+<br>");//true
document.writeln(((10>50) && (10<15))+<br>");//false
document.writeln(((10>5) && (10<6))+<br>");//false
document.writeln(((10>50) && (10<6))+<br>");//false
```

</script>

</body>

</html>

Logical OR (||)

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>

        document.writeln(((10>5) || (10<15))+ "<br>");//true
        document.writeln(((10>50) || (10<15))+ "<br>");//true
        document.writeln(((10>5) || (10<6))+ "<br>");//true
        document.writeln(((10>50) || (10<6))+ "<br>");//false

    </script>

</body>
</html>

```

Logical NOT (!)

```

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

```

```
<script>
```

```
var i=!(5>10);
```

```
document.writeln(i);//true
```

```
</script>
```

```
</body>
```

```
</html>
```

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
    <script>
```

```
var i=!(5>2);
```

```
document.writeln(i);//false
```

```
        </script>

</body>
</html>
```

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

==

It is used to check both values are same or not.

ex:

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

        document.writeln((10==10)+"<br>");//true
        document.writeln((10=="10"+"<br>");//true
        document.writeln((false==0)+"<br>");//true
        document.writeln((true==1)+"<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((10===10)+"<br>");//true
```

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

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

```
        document.writeln((true===1)+"<br>");//false
```

```
    </script>
```

```
</body>
```

```
</html>
```

4) Bitwise Operators

We have following list of bitwise operators.

ex:

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

Bitwise AND operator (&)

```
<!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
*/

```

```

</script>

```

```

</body>

```

```

</html>

```

Bitwise OR operator

```

<!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
    */

```

```

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

```

Bitwise XOR operator

```

-----
<!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

*/

</script>

</body>

</html>

Bitwise NOT

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var i=10;

var j=~i;

document.writeln(j); // -11

```
        </script>

</body>
</html>
```

ex:

```
<!DOCTYPE html>

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

```
    <script>
```

```
        var i=23;
```

```
        var j=~i;
```

```
        document.writeln(j); // -24
```

```
    </script>

</body>
</html>
```

ex:

```
<!DOCTYPE html>
```

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

        var i=-19;

        var j=~i;

        document.writeln(j); //18

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

ex:

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

```
document.writeln((10 >> 2)+"<br>"); //10/4= 2
```

```
document.writeln((20 >> 4)+"<br>"); // 20/16= 1
```

```
document.writeln((10 << 2)+"<br>"); //10*4= 40
```

```
document.writeln((20 << 4)+"<br>"); // 20*16= 320
```

```
</script>
```

```
</body>
```

```
</html>
```

5) Relational operators

We have following list of relational operators.

ex:

operators	description
-----	-----
>	greater then equals to
>=	greater then equals to
<	less then
<=	less then equals to

ex:

```

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

        document.writeln((10 < 20)+"<br>"); //true

        document.writeln((10 <= 10)+"<br>"); //true

        document.writeln((10 > 20)+"<br>"); //false

        document.writeln((10 >= 4)+"<br>"); // true

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

```

6) Special operators

We have following list of special operators.

ex:

operator	description
-----	-----
?:	It is used to write the condition
typeof	It is used to check type of an object
new	IT is used to create an object.

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
    <script>
```

```
        (5>2)?document.writeln("TRUE"):document.writeln("FALSE");
```

```
        document.writeln("<br>");
```

```
        (5>20)?document.writeln("TRUE"):document.writeln("FALSE");
```

```
    </script>
```

```
</body>
```



```
</html>
```

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
    <script>
```

```
        var i=10;
```

```
        document.writeln(typeof(i)+"<br>");
```

```
        var j=true;
```

```
        document.writeln(typeof(j)+"<br>");
```

```
        var k="hi";
```

```
        document.writeln(typeof(k)+"<br>");
```

```
        var l=null;
```

```
        document.writeln(typeof(l)+"<br>");
```

```
        var m;
```

```
        document.writeln(typeof(m)+"<br>");
```

```
</script>
```

```
</body>
```

```
</html>
```

JavaScript IF ELSE Statement

=====

We have three forms of javascript if else statement.

1) IF STMT

2) IF ELSE STMT

3) IF ELSE IF STMT

1) IF STMT

It will 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(0,1,2,3,4,5)
```

```
        {
```

```
            document.writeln("Hi!!!");
```

```
        }
```

```
    </script>
```

```
</body>
```

```
</html>
```

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
<script>
```

```
    if(5,4,3,2,1,0)
```

```
    {
```

```
        document.writeln("Hi!!!");
```

```
    }
```

```
</script>
```

```
</body>
```

```
</html>
```

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
    <script>
```

```
        if(-1)
```

```
        {
```

```
            document.writeln("Hi!!!");
```

```
        }
```

```
</script>
```

```
</body>
```

```
</html>
```

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

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
    <script>
```

```
        var a=10,b=20;
```

```
        if(a>b)
```

```
            document.writeln(a+" is greatest");
```

```
        if(b>a)
```

```
            document.writeln(b+" is greatest");
```

```
    </script>
```

```
</body>
```

```
</html>
```

2) IF ELSE STMT

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

syntax:

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

ex:

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

```
        if(5>20)
        {
            document.writeln("TRUE");
        }
        else
        {
            document.writeln("FALSE");
        }
```

```
    </script>
```

```
</body>
```

```
</html>
```

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
    <script>
```

```
        if(5>2)
        {
            document.writeln("TRUE");
        }
```

```

        else
        {
            document.writeln("FALSE");
        }

    </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 sno=prompt("Enter the number :");

        var n=parseInt(sno); //5

        if((n & 1)==0)
            document.writeln("It is even number");
        else

```



```
document.writeln("It is odd number ");
```

```
/*  
5 - 0101  
1 - 0001  
-----  
& - 0001  
*/
```

```
</script>
```

```
</body>
```

```
</html>
```

3) IF ELSE IF STMT

It will evaluate the code based on multiple conditions.

syntax:

```
if(cond1)  
{  
    - //code to be evaluate  
}  
else if(cond2)  
{  
    - //code to be evaluate  
}  
else if(cond3)
```

```
{  
    - //code to be evaluate  
}  
else  
{  
    - //code to be evaluate if all conditions are false  
}
```

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
    <script>
```

```
        var sno=prompt("Enter the option :");
```

```
        var option=parseInt(sno); //
```

```
        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 will evaluate the code based on multiple conditions.

It is similar to if else if statement.

syntax:

```
switch(condition)
{
    case value1: //code to be evaluate
        break stmt;

    case value2: //code to be evaluate
        break stmt;
```

-

```
-  
    default: //code to be evaluate if all cases are false  
}
```

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
    <script>
```

```
        var sno=prompt("Enter the option :");
```

```
        var option=parseInt(sno); //
```

```
        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:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
    <script>
```

```
        var ch=prompt("Enter the Alphabet :");
```

```
        switch(ch)
```

```
        {
```

```
            case 'a': document.writeln("It is a vowel"); break;
```

```
            case 'e': document.writeln("It is a vowel"); break;
```

```
case 'i': document.writeln("It is a vowel"); break;  
case 'o': document.writeln("It is a vowel"); break;  
case 'u': document.writeln("It is a vowel"); break;  
default: document.writeln("It is a consonent");
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```

Javascript loops

=====

LOOPS are used to iterate 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 only if our condition is true.

syntax:

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

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
```

```
            i++;
```

```
}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
```

```
            i--;
```

```
        }while(i>=1)
```

```
    </script>
```

```
</body>
```



```
</html>
```

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
    <script>
```

```
        var i=1,n=5;
```

```
        do
```

```
        {
```

```
            document.writeln(n+"*"+i+"="+n*i+"<br>");//
```

```
            i++;
```

```
        }while(i<=10)
```

```
    </script>
```

```
</body>
```

```
</html>
```

2) while loop

=====

It will evaluate the code only if our condition is true.

syntax:

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

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
    <script>
```

```
        var i=10;
```

```
        while(i>=1)
```

```
        {
```

```
            document.writeln(i);//10 9 8 7 6 5 4 3 2 1
```

```
            i--;
```

```
        }
```

```
        </script>

</body>
</html>
```

ex:

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

    <script>

        var i=1,sum=0;

        while(i<=10)
        {
            sum+=i;

            i++;
        }
        document.writeln(sum);

    </script>
```

```
</body>
```

```
</html>
```

3) for loop

=====

It will evaluate the code only if our condition is true.

syntax:

```
for(initialization;condition;incrementation/decremenation)
```

```
{
```

```
-
```

```
- //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);//infinite 1
```

```
i--;
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
    <script>
```

```
        var n=5;
```

```
        for(var i=1;i<=10;i++)
```

```
        {
```

```
            document.writeln(n+" * "+i+" = "+n*i+"<br>");
```

```
        }
```

```
        </script>

</body>
</html>
```

ex:

```
<!DOCTYPE html>

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

    <script>

        var n=5,result=1;

        for(var i=n;i>=1;i--)
        {
            result*=i;
        }

        document.writeln(result);

    </script>

</body>
</html>
```

4) for in loop

=====

It is used to iterate the elements 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'];

    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=["hi","hello","bye"];

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

    </script>

</body>
</html>
```

Note:

If number of iterations are known by the user then we need to use for loop.

If number of iterations are not known by the user then we need to use while loop.

If number of iterations are not known by the user but code must execute atleast for one time then we need to use do while loop.

Interview Question

=====

Q) Write a java program to display longest/largest common subsequence?

Input:

ABCAB

AECB

Output:

3

class Test

{

public static void main(String[] args)

{

String firstStr="ABCAB";

String secondStr="AECB";

//caller method

System.out.println(longestCommonSubsequence(firstStr,secondStr));

}

public static int longestCommonSubsequence(String str1,String str2)

{

return solve(str1,str2,0,0);

}

public static int solve(String s1,String s2,int i,int j)

{

if(i==s1.length())

return 0;

if(j==s2.length())

```

        return 0;

    int ans=0;

    if(s1.charAt(i) == s2.charAt(j))
    {
        ans=1+solve(s1,s2,i+1,j+1);
    }
    else
    {
        ans= Math.max(solve(s1,s2,i+1,j),solve(s1,s2,i,j+1));
    }

    return ans;

}

}

```

Javascript Functions

=====

Function is a block of scope which is used to perform perticular task.

Functions are designed to maintain business logic.

Using functions we can achieve reusability.

To declare a function in javascript we will use function keyword.

We can declare a function in javascript followed by a function name, and parenthesis().

Javascript function parenthesis contains list of arguments seperated with comma.

syntax:

```
function <function_name>()
{
    -
    - //code to be evaluate
    -
}
```

Functions are executed at the time when they we have requested or when event occur.

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
    <script>
```

```
        function f1()
```

```
        {  
            document.bgColor="red";  
        }  
  
        //calling  
        f1();  
  
    </script>  
</body>  
</html>
```

ex:

```
<!DOCTYPE html>  
<html>  
<head>  
    <title>MyPage!</title>  
</head>  
<body>  
    <script>  
  
        function f1()  
        {  
            for(var i=1;i<=4;i++)  
            {  
                for(var j=1;j<=4;j++)  
                {
```

```

        document.writeln("*");
    }
    //new line
    document.writeln("<br>");
}
}

//calling
f1();

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

```

ex:

```

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

        function f1()
        {
            for(var i=1;i<=4;i++)

```

```

        {
            for(var j=1;j<=i;j++)
            {
                document.writeln("*");
            }
            //new line
            document.writeln("<br>");
        }
    }

    //calling
    f1();
    f1();

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

```

In javascript ,every function is a case sensitive.

ex:

```

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

```

```
<body>
  <script>

    function f1()
    {
      document.writeln("F1 Function");
    }

    //calling
    F1();

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

ex:

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

  <button onclick="f1()"> Click Here </button>

  <script>
```



```
        function f1()
        {
            document.bgColor="blue";
        }
    </script>
</body>
</html>
```

Q) What is JavaScript Closure?

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

In JavaScript, closures are created every time when functions are created.

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

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()
        {
            var c=30;

            document.writeln(a+" "+b+" "+c);
        }
        //calling
        f2();
    }

    //calling
    f1();

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

=====

A javascript object is an entity which is having states 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>
```

Q)Write a javascript program to perform sum of array elements?

```
<!DOCTYPE html>
<html>
<head>
    <title>My Page!</title>
</head>
<body>
    <script>

        var arr=[10,20,30,40];

        var sum=0;
```

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

```
</script>
```

```
</body>
```

```
</html>
```

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
    <script>
```

```
        var str="racar";
```

```
        var carr=str.split("");// h e l l o
```

```
        var rev="";
```

```
        //reading reverse
        for(var i=carr.length-1;i>=0;i--)
        {
            rev+=carr[i];
        }

        if(str==rev)
            document.writeln("It is palindrome ");
        else
            document.writeln("it is not palindrome");

    </script>
</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 automatically 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>
```

2)confirm()

It will display 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", "_blank", "width=200px,height=200px");

            }
        </script>
    </body>
</html>

```

```
        <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>
```

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 f1()
      {
        setTimeout(function f1()
          {
            alert("Hello World")
          },4000);
      }
    </script>

    <button onclick="f1()">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 continuously function until 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>
  </body>
</html>

```

```
        }

        </script>

        <div id="myId"></div>

    </body>
</html>
```

Javascript History

=====

```
<!DOCTYPE html>

<html>

<head>

    <title>MyPage!</title>

    <style>

        body

        {

            height: 100vh;

            width: 100%;

            display: flex;

            justify-content: center;

            align-items: center;

        }

        a

        {

            font-size: 30px;

            text-decoration: none;

            margin: 0 50px;
```



```

        }
    </style>
</head>
<body>

    <a href="javascript:history.back()"> &laquo;Previous </a>

    <a href="javascript:history.forward()"> Next&raquo; </a>

</body>
</html>

```

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 particular 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>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
    <script>
```

```
        localStorage.setItem("name","Alan");
```

```
        localStorage.setItem("age",30);
```

```
        document.writeln(localStorage.getItem("name")+"<br>");
```

```
        document.writeln(localStorage.getItem("age")+"<br>");
```

```
        localStorage.removeItem("age");
```

```
        document.writeln(localStorage.getItem("name")+"<br>");
```

```
        document.writeln(localStorage.getItem("age")+"<br>");
```

```
        localStorage.clear();
```

```
        document.writeln(localStorage.getItem("name")+"<br>");
```

```
        document.writeln(localStorage.getItem("age")+"<br>");
```

```
        </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 particular 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>
```

```
<title>MyPage!</title>

</head>

<body>


    <script>


        sessionStorage.setItem("name","Alan");
        sessionStorage.setItem("age",30);


        document.writeln(sessionStorage.getItem("name")+"<br>");
        document.writeln(sessionStorage.getItem("age")+"<br>");


        sessionStorage.removeItem("age");
        document.writeln(sessionStorage.getItem("name")+"<br>");
        document.writeln(sessionStorage.getItem("age")+"<br>");


        sessionStorage.clear();
        document.writeln(sessionStorage.getItem("name")+"<br>");
        document.writeln(sessionStorage.getItem("age")+"<br>");


    </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 hierarchy.

A document object is a root node for entire html document.

DOM always looks for three nodes.

1)Element node

2)Attribute node

3)Text node

Diagram: class22.1

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

`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"/>

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

```
<script>
```

```
    function f1()
```

```
    {
```

```
        var val=document.getElementById('t1').value;
```

```
        document.writeln("Welcome :"+val);
```

```
    }
```

```
</script>
```

```
</body>
```

```
</html>
```

ex:2

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

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

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

```
    <br>
```



```

        <h1 id="result"></h1>

        <script>
            function f1()
            {
                var val=document.getElementById('t1').value;

                document.getElementById('result').innerHTML="Welcome :"+val;

            }
        </script>
    </body>
</html>

```

Adding two text fields and display the result using javascript

=====

```

<!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>Result:</td>
            <td><input type="text" id="t3"/></td>
        </tr>
        <tr>
            <td><input type="reset" value="reset"/></td>
            <td><button onclick="f1()"> ADD </button></td>
        </tr>
    </table>

    <script>
        function f1()
        {
            var a=document.getElementById("t1").value;
            var b=document.getElementById("t2").value;
            var c=parseFloat(a)+parseFloat(b);
            document.getElementById('t3').value=c;
        }
    </script>

</body>
</html>

```

Hide and show the portion of a form page using javascript

=====

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
    <fieldset id="per_id">
        <legend>Permanent Address</legend>
        <table align="center">
            <tr>
                <td>Locality:</td>
                <td><input type="text" id="t1"/></td>
            </tr>
            <tr>
                <td>State:</td>
                <td><input type="text" id="t2"/></td>
            </tr>
            <tr>
                <td>City:</td>
                <td><input type="text" id="t3"/></td>
            </tr>
        </table>
    </fieldset>
```

```

<br>
<input type="checkbox" id="box" value="checked" onclick="f1()" />
Permanent address same as current address?
<br>
<br>

<fieldset id="cur_id">
<legend>Current Address</legend>
<table align="center">
  <tr>
    <td>Locality:</td>
    <td><input type="text" id="t1"/></td>
  </tr>
  <tr>
    <td>State:</td>
    <td><input type="text" id="t2"/></td>
  </tr>
  <tr>
    <td>City:</td>
    <td><input type="text" id="t3"/></td>
  </tr>
</table>
</fieldset>

<script>
  function f1()
  {

```

```

        if(document.getElementById('box').checked)
        {
            document.getElementById('cur_id').style.display="none";
        }
        else
        {
            document.getElementById('cur_id').style.display="block";
        }
    }
</script>

```

```
</body>
```

```
</html>
```

```
document.getElementsByName()
```

```
=====
```

It is used to return collection of elements with specified name.

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

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

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

```
<p id="result"> </p>
```

```
<script>
```

```
    function f1()
```

```
    {
```

```
        var name=document.getElementsByName("t1")[0].value;
```

```
        document.getElementById('result').innerHTML="Welcome
```

```
:"+name;
```

```
    }
```

```
</script>
```

```
</body>
```

```
</html>
```

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

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

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

```
<p id="result"> </p>
```

```
<script>
```

```
    function f1()
```

```
    {
```

```
        var name=document.getElementsByName("t1")[0].tagName;
```

```
        document.getElementById('result').innerHTML=name;
```

```
    }
```

```
</script>
```

```
</body>
```

```
</html>
```

ex:

```
<!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()"> click </button>
```

```
<p id="result"> </p>
```

```

        <script>
            function f1()
            {
                var name=document.getElementsByName("t1").length;
                document.getElementById('result').innerHTML=name;
            }
        </script>
    </body>
</html>

```

ex:

```

<!DOCTYPE html>

<html>

<head>
    <title>MyPage!</title>
</head>
<body>

    Courses Completed ?

    <br>
    <button onclick="f1()"> Select All </button>

    <br>
    <input type="checkbox" name="t1" value="html"/> HTML <br>
    <input type="checkbox" name="t1" value="css"/> CSS <br>
    <input type="checkbox" name="t1" value="js"/> JAVASCRIPT <br>

```



```
<input type="checkbox" name="t1" value="bootstrap"/> Bootstrap <br>
```

```
<script>
```

```
    function f1()
```

```
    {
```

```
        var x=document.getElementsByName("t1");
```

```
        for(var i=0;i<x.length;i++)
```

```
        {
```

```
            if(x[i].type=="checkbox")
```

```
            {
```

```
                x[i].checked="true";
```

```
            }
```

```
        }
```

```
    }
```

```
</script>
```

```
</body>
```

```
</html>
```

document.getElementsByTagName()

=====

It will return collection of elements specified by using tag name.

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```

<head>
    <title>MyPage!</title>
</head>
<body>

    Name: <input type="text" /> <br>
    <button onclick="f1()"> click </button>

    <div id="result"></div>
    <script>
        function f1()
        {
            var name=document.getElementsByTagName('INPUT')[0].value;
            document.getElementById('result').innerHTML=name;
        }
    </script>
</html>

```

ex:

```

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

    Name: <input type="text" /> <br>
    <button onclick="f1()"> click </button>

```

```

    <div id="result"></div>

    <script>
        function f1()
        {
            var name=document.getElementsByTagName('INPUT').length;
            document.getElementById('result').innerHTML=name;
        }
    </script>

</html>

```

ex:

```

<!DOCTYPE html>

<html>

<head>
    <title>MyPage!</title>
</head>

<body>

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

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

    <div id="result"></div>

    <script>
        function f1()
        {

```

```
        var
name=document.getElementsByTagName('INPUT')[0].tagName;

        document.getElementById('result').innerHTML=name;

    }
</script>

</html>
```

ex:

```
<!DOCTYPE html>

<html>

<head>

    <title>MyPage!</title>

</head>

<body>

    <div> This is division 1 tag </div>

    <div> This is division 2 tag </div>

    <div> This is division 3 tag </div>

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

    <script>

        function f1()

        {

            var x=document.getElementsByTagName('div');

            x[0].innerHTML="This is my division tag 1";

            x[1].innerHTML="This is my division tag 2";

            x[2].innerHTML="This is my division tag 3";
```

```
        x[0].style.color="red";
        x[0].style.textAlign="center";
        x[1].style.color="blue";
        x[2].style.color="green";
    }
</script>
```

</html>

document.getElementsByClassName()

It will return the collection of elements by using specified class name.

ex:

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

<style>

.para-style

{

color: blue;

}

</style>

</head>

```

<body>

    <div class="para-style">

        This is javascript document object example

    </div>

    <button onclick="f1()"> click Here </button>

    <script>
        function f1()
        {
            var x=document.getElementsByClassName('para-style');
            x[0].style.backgroundColor="cyan";
            x[0].style.textAlign="center";
        }
    </script>

</body>
</html>

```

addEventListener()

=====

It is used to attaches an event handler to a document.

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

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

    <h1> Click here </h1>

    <script>

        addEventListener("click",f1);

        function f1()
        {
            alert("Yahoo! you have clicked ");
        }

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

ex:

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

```
<h1 id="hover"> Mouse Over Here </h1>
```

```
<div id="result"></div>
```

```
<script>
```

```
var x=document.getElementById('hover');
```

```
x.addEventListener("mouseover",f1);
```

```
function f1()
```

```
{
```

```
    document.getElementById('result').innerHTML="Move is over!";
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```



```
<h1 id="hover"> Mouse Over and Out Here </h1>
```

```
<div id="result"></div>
```

```
<script>
```

```
var x=document.getElementById('hover');
```

```
x.addEventListener("mouseout",f1);
```

```
function f1()
```

```
{
```

```
    document.getElementById('result').innerHTML="Move is Out!";
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```

removeEventListener

=====

It is used to remove the handler from the document.

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```

<title>MyPage!</title>

</head>

<body>

    <h1 id="hover"> Mouse Over Here </h1>

    <br>

    <button onclick="f2()"> stop </button>

    <div id="result"></div>

    <script>

        var x=document.getElementById('hover');

        x.addEventListener("mouseover",f1);

        function f1()
        {
            document.getElementById('result').innerHTML+="<p>Move is
Out!</p>";
        }
        function f2()
        {
            x.removeEventListener("mouseover",f1);
            document.getElementById('result').innerHTML+="<p>Event is
stop!!</p>"

```

```
        }  
    </script>
```

```
</body>
```

```
</html>
```

How can convert Feet to Inches using addEventListener()

=====

```
<!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>
```

```

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>

Date

=====

Date is a javascript object which is used to display date and time.

Using Date object we can display timer on the web page.

ex:

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
    <script>
        var date=new Date();
        document.writeln(date);
    </script>

</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
    <script>
        var date=new Date();
```

```
        var d=date.getDate();
        var m=date.getMonth()+1;
        var y=date.getFullYear();
        document.writeln(d+"/"+m+"/"+y);
    </script>
```

```
</body>
```

```
</html>
```

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
    <script>
```

```
        var date=new Date();
```

```
        var h=date.getHours();
```

```
        var m=date.getMinutes();
```

```
        var s=date.getSeconds();
```

```
        document.writeln(h+":"+m+": "+s);
```

```
    </script>
```

```
</body>
```

```
</html>
```

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <title>MyPage!</title>
```

```
  <style>
```

```
    body
```

```
    {
```

```
      background: linear-gradient(yellow,red);
```

```
      height:100vh;
```

```
      width:100%;
```

```
      display: flex;
```

```
      justify-content: center;
```

```
      align-items: center;
```

```
    }
```

```
    span
```

```
    {
```

```
      font-size:40px;
```

```
      box-shadow:2px 2px 19px 3px #FFF;
```

```
      color:#FFF;
```

```
      font-weight: bold;
```

```
    }
```

```
</style>
```

```
</head>
```

```
<body>
```

```
    <span id="result"> </span>
```

```
    <script>
```

```
        window.onload=function(){getTime();}
```

```
        function getTime()
```

```
        {
```

```
            var d=new Date();
```

```
            var h=d.getHours();
```

```
            var m=d.getMinutes();
```

```
            var s=d.getSeconds();
```

```
            m=check(m);
```

```
            s=check(s);
```

```
            document.getElementById('result').innerHTML=h+":"+m+":s";
```

```
            setInterval(getTime,1000);
```

```
        }
```

```
        function check(i)
```

```
        {
```

```
            if(i<10)
```

```
            {
```

```
                i= "0"+i;
```

```
            }
```

```
            return i;
```



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

Q) What is the difference between innerText vs innerHTML?

innerText

=====

It is used to write simple text using javascript dynamically.

ex:

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

```
</body>
```

```
</html>
```

innerHTML

It is used to write HTML code using javascript dynamically.

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

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

```
    <div id="result"></div>
```

```
    <script type="text/javascript">
```

```
        function f1()
```

```
        {
```

```
            document.getElementById('result').innerHTML="<p style='color:red'>This is HTML Code  
</p>";
```

```
        }
```

```
    </script>
```

```
</body>
```

```
</html>
```

Javascript program to hide and show the password in a textfield

```
=====
```

```
<!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>
```

```
        .pwdstyle
```

```
        {
```

```
            padding-left: 10px;
```

```
            padding-right: 25px;
```

```
        }
```

```
        #iconId
```

```
        {
```

```
            position: relative;
```

```
            right: 25px;
```

```
    }

</style>

</head>
<body>

    Password: <input type="password" id="myId" class="pwdstyle"/>
    <span class="fa fa-eye" id="iconId" onclick="f1()"></span>

    <script>
        function f1()
        {
            var x=document.getElementById('myId');
            if(x.type=="password")
            {
                x.type="text";
            }
            else
            {
                x.type="password";
            }
        }
    </script>

</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.."/>
    </form>
</div>
</body>
</html>
```

```
<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>
```

JavaScript Interview Questions

=====

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>

```

Q)What is the difference between var , let and const ?

var	let	const
-----	-----	-----
It is a functional scope.	It is a block scope.	It is a block scope.
We can declare without initialization.	We can declare without initialization.	We can't 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 accessible without declaration and default value without declaration.	It can't be accessible without declaration.	It can't be accessible without declaration.

is undefined.

initialization

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
  <script>
```

```
    var i;
```

```
    document.writeln(i); //undefined
```

```
  </script>
```

```
</body>
```

```
</html>
```

ex:

```
<!DOCTYPE html>
```

```
<html>
<head>
  <title>MyPage!</title>
</head>
<body>
  <script>
    let i;
    document.writeln(i); //undefined
  </script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
  <title>MyPage!</title>
</head>
<body>
  <script>
    const i;
    document.writeln(i); //invalid
  </script>
</body>
</html>
```

update

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
  <script>
```

```
    var i=10;
```

```
    i=20;
```

```
    document.writeln(i); //20
```

```
  </script>
```

```
</body>
```

```
</html>
```

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

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

ex:

```
---
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
    <script>
        const i=10;
        i=20;
        document.writeln(i); //invalid
    </script>
</body>
</html>
```

Redeclared

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
  <script>
```

```
    var i=10;
```

```
    var i=20;
```

```
    document.writeln(i); //20
```

```
  </script>
```

```
</body>
```

```
</html>
```

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
  <script>
```

```
    let i=10;

    let i=20;

    document.writeln(i); //invalid
</script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
  <title>MyPage!</title>
</head>
<body>
  <script>
    const i=10;

    const i=20;

    document.writeln(i); //invalid
  </script>
</body>
</html>
```

Accessible

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <title>MyPage!</title>
```

```
</head>
```

```
<body>
```

```
  <script>
```

```
    document.writeln(i); //undefined
```

```
    var i=10;
```

```
  </script>
```

```
</body>
```

```
</html>
```

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>MyPage!</title>
</head>
<body>
  <script>
    document.writeln(i); //invalid
    let i=10;
  </script>
</body>
</html>
```

ex:

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

ex:1

```
    i=10;          var i;
document.writeln(i); ==>    i=10;
var i;             document.writeln(i);
```

ex:2

```
    f1();
function f1()
{
    document.writeln("Hello");
}

function f1()
{
    document.writeln("Hello"); ==>
}

                                f1();
```

Q) What is spread operator in JavaScript?

A spread operator is used to spreading an array.

Ex:

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

    var sum=0;
    function f1(num1,num2,num3,num4)
    {
      sum=sum+num1+num2+num3+num4;
      document.writeln(sum);
    }

    let arr=[10,20,30,40];

    f1(...arr);

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

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>MyPage!</title>
```

```
  </head>
```

```
<body>
```

```
  <script>
```

```
    var arr=[10,20,30];
```

```
    function sum(arg1,arg2,arg3)
```

```
    {
```

```
      return arg1+arg2+arg3;
```

```
    }
```

```
    //calling
```

```
    document.writeln(sum(...arr));
```

```
  </script>
```

```
</body>
```

```
</html>
```

Q)What is javascript recursion?

A function which call itself for many number of times is called recursion.

ex:

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

    function f1(i)
    {
      if(i<=10)
      {
        document.writeln(i);// 1 2 3 4 5 6 7 8 9 10
        f1(i+1);
      }
    }

    f1(1);
  </script>
</body>
</html>
```


Q)What is map() method in javascript?

A map() creates a new array from calling a function for every array element.

ex:

```
<!DOCTYPE html>

<html>

<head>

  <title>MyPage!</title>

</head>

<body>

  <script>

    var arr=[10,20,30];

    var newArr=arr.map(f1);

    function f1(ele)

    {

      return ele;

    }

    document.writeln(newArr);

  </script>

</body>

</html>
```

ex:

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

      var newArr=arr.map(f1);

      function f1(ele)
      {
        return ele+100;
      }

      document.writeln(newArr);
    </script>

  </body>
</html>
```

Q)What is reduce function in javascript?

The reduce() method executes a reducer function for array element.

The reduce() method returns a single value.

ex:

```
<!DOCTYPE html>
<html>
<head>
  <title>MyPage!</title>
</head>
<body>
  <script>
    var arr=[100,20,30];

    var res=arr.reduce(f1);

    function f1(total,ele)
    {
      return total+ele;
    }

    document.writeln(res);
  </script>
```

</body>

</html>

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

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

```

        var promise= new Promise(function(resolve,reject){
            reject("Reject");
        })

        promise
        .then(function(successMessage){
            document.writeln(successMessage);

        },function(errorMessage)
        {
            document.writeln(errorMessage);
        })

    </script>

</body>

</html>

```

Types of functions in javascript

=====

We have Three types of functions in javascript.

1) Named Function

```

function f1()
{
    document.writeln("Named Function");
}

```

```
}
```

2) Anonymous Function

```
const f1=function()  
{  
    document.writeln("Named Function");  
}
```

3) Arrow Function - ES6 standards

```
const f1=()=>>  
{  
    document.writeln("Named Function");  
}
```

Javascript OOPS

=====

OOPS stands for Object oriented Programming System/Structure.

A technology is said to be object oriented if it support following features.

ex:

class

object

abstraction

encapsulation

inheritance
and
polymorphism.

Javascript is not a object oriented programming language. It is a object based programming language because javascript contains objects like Arrays,Strings,RegEx, Date and etc.

class

=====

It is a template for an object.

In javascript, class will not consider as an object.

To declare a class we need to use class keyword following by any perticular name.

As per naming conventions in javascript , A class name must starts with uppercase letter.

ex:

```
class Example
{
    -
    -
    -
}
```


constructor

=====

A constructor is a special function which is used to initialize or create an object.

A constructor will be called when memory is allocated.

To declare a constructor we will use constructor keyword.

A class can have only constructor.

ex:

```
class Example
{
    constructor()
    {
        -
        -
        -
    }
}
```

object

Object is an entity which is having state and behaviours (properties and functions).

To create object in javascript we will use new keyword.

syntax:

```
var ref_var=new Object_name();
```

ex:

```
var e = new Example();
```

ex:1

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>MyPage!</title>
```

```
  </head>
```

```
  <body>
```

```
    <script>
```

```
      class Example
```

```
      {
```

```
        constructor()
```

```
        {
```

```
          document.writeln("constructor <br>");
```

```
        }
```

```
      }
```

```
      var e1=new Example();
```

```
      var e2=new Example();
```

```
    </script>
```

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

ex:2

```
<!DOCTYPE html>
```

```
<html>
```

```
    <head>
```

```
        <title>MyPage!</title>
```

```
    </head>
```

```
    <body>
```

```
        <script>
```

```
            class Example
```

```
            {
```

```
                constructor(id,name,sal)
```

```
                {
```

```
                    document.writeln("Employee Id :"+id+"<br>");
```

```
                    document.writeln("Employee Name
```

```
:""+name+"<br>");
```

```
                    document.writeln("Employee Salary :"+sal+"<br>");
```

```
                }
```

```
            }
```

```
            var e1=new Example(101,'Alan',10000);
```

```
            var e2=new Example(102,'Jose',20000);
```

```
        </script>
```

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

Abstraction

=====

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

ex:

The best example of abstraction is GUI ATM machine where bank people will hide internal

implementation and highlights the set of services like banking,withdrawl,mini stmt and etc.

Encapsulation

=====

The process of encapsulating or grouping properties and it's associate functions is called encapsulation.

In encapsulation for every property we need to declare setter and getter methods.

ex:1

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>MyPage!</title>
```

```
  </head>
```

```
<body>
```

```
<script>
```

```
class Example
```

```
{
```

```
    //setter method
```

```
    setId(id)
```

```
    {
```

```
        this.id=id;
```

```
    }
```

```
    setName(name)
```

```
    {
```

```
        this.name=name;
```

```
    }
```

```
    setSalary(sal)
```

```
    {
```

```
        this.sal=sal
```

```
    }
```

```
    //getter methods
```

```
    getId()
```

```
    {
```

```
        return this.id;
```

```
    }
```

```
    getName()
```

```
    {
```

```
        return this.name;
```

```
    }
```

```

        getSalary()
        {
            return this.sal;
        }
    }

    var e=new Example();
    e.setId(201);
    e.setName("Lara");
    e.setSalary(10000);

    document.writeln("Employee Id :"+e.getId()+"<br>");
    document.writeln("Employee Name :"+e.getName()+"<br>");
    document.writeln("Employee Salary :"+e.getSalary()+"<br>");

```

```

</script>

```

```

    </body>
</html>

```

ex:2

```

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

```

<body>

<script>

```
class Example
{
    constructor(id,name,sal)
    {
        this.id=id;
        this.name=name;
        this.sal=sal;
    }

    //getter methods
    getId()
    {
        return this.id;
    }
    getName()
    {
        return this.name;
    }
    getSalary()
    {
        return this.sal;
    }
}

var e=new Example(301,"Nelson",20000);
```

```
document.writeln("Employee Id :"+e.getId()+"<br>");  
document.writeln("Employee Name :"+e.getName()+"<br>");  
document.writeln("Employee Salary :"+e.getSalary()+"<br>");
```

```
</script>
```

```
</body>
```

```
</html>
```

Inheritance

=====

Inheritance is a mechanism where one class will inherit the properties of another class.

or

Inheritance is a mechanism where one class will derive in the presence of another class.

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
    <head>
```

```
        <title>MyPage!</title>
```

```
    </head>
```

```
    <body>
```



```
<script>
```

```
    class A
```

```
    {
```

```
        f1()
```

```
        {
```

```
            document.writeln("A-F1 Function <br>");
```

```
        }
```

```
    }
```

```
    class B extends A
```

```
    {
```

```
        f2()
```

```
        {
```

```
            document.writeln("B-F2 Function <br>");
```

```
        }
```

```
    }
```

```
    var a=new A();
```

```
    a.f1();
```

```
    var b=new B();
```

```
    b.f1();
```

```
    b.f2();
```

```
</script>
```

```
</body>
```

</html>

Polymorphism

=====

Polymorphism has take from Greek word.

Poly means many and morphism means forms.

The ability to represent in different forms is called polymorphism.

ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>MyPage!</title>
```

```
  </head>
```

```
  <body>
```

```
    <script>
```

```
      class A
```

```
      {
```

```
        display()
```

```
        {
```

```
          document.writeln("A-class <br>");
```

```
        }
```

```
      }
```

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

var b=new B();
b.display(); // B-class

</script>

</body>

</html>
```