JAVA SCRIPT

Q)What is the difference between Java and JavaScript?

Java	JavaScript
It is a non-scriptling language.	It is a scripting language.
It is a strongly typed checking lang.	It is a weakly typed checking language.
It is a object oriented programming language.	It is a object based programming language.
It can run individually.	It can't run individually.
It does not required browser window for execution.	It requires browser window for execution.
It is complex language.	It is simple language compare to java.

History JavaScript

- Orginally name of javascript is LiveScript.
- Livescript is created Netscape Corporation in 1990's.
- Livescript is developed by using C language syntax's.
- Later 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 Europeon Company Manufacturer Association.

Advantages of Javascript

- It is used to develop intractive web pages.
- It is used to perform client side form validation.
- It is used to display dialog boxes and pop-up boxes.
- It is used to create dynamic web pages.
- It supports objects like Strings,Arrays,RegEx and etc.
- It supports Date and Time.
- It supports Cookies.
- It supports drop-down menu.
- It is loosely typed checking language.

syntax:

```
        <script type="text/javascript" language="javascript">
                stmt1;
                stmt2;
                stmt3;
        </script>
```

- Here "type" and "language" attribute is optional to declare.
- Here semicolon i.e ';' is optional to declare.

ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>IHUB TALENT</title>
        </head>
        <body>
                <script type="text/javascript" language="javascript">
                        document.write("Welcome to IHUB ");
                        document.write("This is JavaScript class");
                </script>
        </body>
</html>
```

ex:2

```
<!DOCTYPE html>
```

```
<html>
    <head>
        <title>IHUB TALENT</title>
    </head>
    <body>
        <script>
            document.write("Welcome to IHUB ")
            document.write("This is JavaScript class")
        </script>
    </body>
</html>
```

ex:3

```
<!DOCTYPE html>
<html>
    <head>
        <title>IHUB TALENT</title>
    </head>
    <body>
        <script>
            document.write("Welcome to IHUB ");
            <br>
            document.write("This is JavaScript class");
        </script>
    </body>
</html>
```

- Here no output because we can't place markup language in scripting language.
- To overcome this above limitation we need to use below example.

ex:5

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB TALENT</title>
	</head>
	<body>
		<script>
			document.write("Welcome to IHUB ");
			document.write("<br>");
			document.write("This is JavaScript class");
		</script>
	</body>
</html>
```

- In javascript we can declare html tags and css properties also.

ex:6

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

```
                <script>
                        document.write("<center>Welcome to IHUB</center>");
                        document.write("<br>");
        document.write("<h1 style='color:blue'>This is JavaScript
class</h1>");
                </script>
        </body>
</html>
```

Note:

- If a program contains html ,css and javascript code then it is called
- DHTML i.e Dynamic Hypertext Markup Language.

ex:7

```
<!DOCTYPE html>
<html>
        <head>
                <title>IHUB TALENT</title>
        </head>
        <body>
                <script>
                        document.writeln("Ihub");
                        document.writeln("Talent");
                </script>
        </body>
</html>
```

- A document.writeln() will give one space at the end.

JavaScript Engine

- It allow us to execute javascript code on browser window.
- It is responsible to convert user understandable scripting language to machine understandable scripting language.
- Bydefault every browser contains javascript engine.
- We have following list of javascript engine.

ex:

Browser	Engine
Chrome	V8 Engine
Mozilla	spider monkey
Edge	Chakra
Opera	Charakan
and etc.	

JavaScript comments

- Comments are created for documentation purpose.
- Javascript comments are used to improve readability of our code.
- Javascript comments will not display in output.

We have two types comments in javascript.

1)Single line comment

- It is used to comment a single line.

ex:

```
// comment here
```

2)Multiple line comment

- It is used to comment multiple lines.
- IT is more convenient when compare to single line comment because we can comment single line as well as multiple lines.

ex:

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

Output statement in javascript

- Output statements are used to display userdefined statements or data.

We have two output statements 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>IHUB TALENT</title>
    </head>
    <body>
        <script>
            document.writeln("<center>JavaScript Class</center>");
        </script>
    </body>
```

```
</html>
```

2)console.log()

- It is used to display the output on browser console.
- To see the browser console we need to press F12 function key.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB TALENT</title>
	</head>
	<body>
		<script>
			console.log("This is Javascript");
		</script>
	</body>
</html>
```

Javascript variables

- A name which is given to a memory location is called variable.
- Purpose of variable is used to store the data.
- In javascript , a variable is also known as identifier.
- A variable contains same rules as we have rules for an identifiers.

Rule1:

- A variable must and should starts with alphabet, underscore or dollar symbol.

ex:

```
var emp1;
```

```
        var _emp2;

        var $sal;
```

Rule2:

- A variable can't start with digits.

ex:

```
        var  a123; //valid

        var  1abcd; // invalid
```

Rule3:

- Every variable is a case sensitive.

ex:

```
        var number;

        var NUMBER;

        var NumBer;
```

There are two types of variables in javascript.

1)Local variable

2)Global variable

1)Local variable

- If we declare a variable inside the functions is called local variable.
- We can access local variable with inthe function.

ex:1

```
<!DOCTYPE html>

<html>

    <head>

        <title>IHUB TALENT</title>

    </head>
    <body>
        <script>
            function f1()
            {
                //local variable
                var a=10;
                document.writeln(a);//10
            }

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

ex:2

```
<!DOCTYPE html>
<html>
    <head>
        <title>IHUB TALENT</title>
    </head>
    <body>
        <script>
            function f1()
            {
                //local variable
                var a=10;
                document.writeln(a);//10
            }
            function f2()
            {
                document.writeln(a); // Error
            }

            //call the function
            f1();
            f2();
        </script>
    </body>
</html>
```

2)Global variable

- If we declare a variable outside the function is called global variable.
- Global variable can access within the function and outside the function.

ex:1

```
<!DOCTYPE html>
<html>
    <head>
        <title>IHUB TALENT</title>
    </head>
    <body>
        <script>
```

```
            //global variable
            var a=100;

            function f1()
            {
                document.writeln(a);//100
            }

            //call the function
            f1();

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

ex:2

```
<!DOCTYPE html>
<html>
    <head>
        <title>IHUB TALENT</title>
    </head>
    <body>
        <script>
            //global variable
            var a=100;
```

```
                    function f1()
                    {
                            document.writeln(a);//100
                    }
                    function f2()
                    {
                            document.writeln(a);//100
                    }

                    //call the function
                    f1();
                    f2();

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

Types of Javascript

We have two types of javascript.

1)Internal Javascript / Embedded JavaScript

2)External Javascript / Seperate JavaScript

1)Internal Javascript

- **In internal javascript, we will mix javascript code and html code in ".html" file.**

Advantages:

- There is no confusion of multiple files.
- We can undstand html code and javascript code easily.

Disadvantage:

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

Note:

- Internal javascript is best suitable for practicing level.

2)External Javascript

- In external javascript, we will maintain html code in ".html" file and javascript code in ".js" file sperately.
- A javascript file can't execute directly on a browser window.We need to load or attach with .html file.

Advantages:

- We can maintain html code and javascript code seperately.
- If code is increases then it will not increase the complexity.

Disadvantages:

- There is a confusion of multiple files.

JavaScript Datatypes

- Datatype describe what type of value we want to store inside a variable.
- Datatype also tells how much memory has to be created for a variable.
- Javascript is a dynamically typed language it means we should not declare any datatype during the variable declaration.
- To declare any variable we need to use "var" keyword.
- Internally javascript uses javascript engine to determine perticular datatype based on the value.
- In javascript, we have two types of datatypes.

1)Primitive Datatypes

2)Non-Primitive Datatypes

1)Primitive Datatypes

We have following list of primitive datatypes.

Datatype	Description
Number	It is used to represent numbers.
Boolean	It is used to represent boolean values.
String	It is used to represent string.
null	It is used to represent null.
undefined	It is used to represent undefined.

Number

```
<!DOCTYPE html>

<html>

<head>

        <title>IHUB TALENT</title>

</head>

<body>

        <script>

                var a=10;

                document.writeln(a+"<br>");

                var b=10.56;

                document.writeln(b+"<br>");

        </script>

</body>
```

```
</html>
```

Boolean

```
<!DOCTYPE html>
<html>
<head>
        <title>IHUB TALENT</title>
</head>
<body>
        <script>
                var a=true;
                document.writeln(a+"<br>");
                var b=false;
                document.writeln(b+"<br>");
        </script>
</body>
</html>
```

string

```
<!DOCTYPE html>
<html>
<head>
        <title>IHUB TALENT</title>
</head>
<body>
        <script>
        var a="ihub";
        document.writeln(a+"<br>");
        var b='talent';
        document.writeln(b+"<br>");
    </script>
</body>
</html>
```

null

```
<!DOCTYPE html>
<html>
<head>
    <title>IHUB TALENT</title>
</head>
<body>
    <script>
        var a=null;
        document.writeln(a+"<br>");

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

undefined

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

```
        <title>IHUB TALENT</title>
</head>
<body>
        <script>
                var a;
                document.writeln(a+"<br>");


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

2)Non-Primitive Datatypes

We have following list of non-primitive datatypes.

Datatype	Description
Object	It is used to represent an instance through which we canaccess members.
Arrays	It is used to represent similar elements.
RegEx	It is used to represent regular expression.

JavaScript Operators

- **Operator is a symbol which is used to perform some operations on operands.**
- **It can be arithmetic operation, logical operation, conditional operation and etc.**

ex:

```
        var c=a+b;
```

Here + and = are operators.

Here a,b and c are operands.

We have following list of operations in javascript.

1)Arithmetic operators

2)Conditional operators

3)Bitwise operators

4)Logical operators

5)Assignment operators

6)Special operators

1)Arithmetic operators

We have following list of arithmetic operators.

Operator	Description
+	addition
-	subtraction
++	Incrementation
--	Decrementation
%	modules
/	division
*	multiplication

ex:1

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

```
</head>
<body>
    <script>
        document.writeln((10%5)+"<br>");//0
        document.writeln((10/2)+"<br>");//5
        document.writeln((10*3)+"<br>");//30
        document.writeln((20-10)+"<br>");//10
        document.writeln((20+10)+"<br>");//30
    </script>
</body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
<head>
    <title>IHUB TALENT</title>
</head>
<body>
    <script>
        var a=10,b=20;

        document.writeln((a++ + b++)+"<br>");//30

        document.writeln((a-- - b--)+"<br>");//-10
    </script>
```

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

ex:3

```
<!DOCTYPE html>
<html>
<head>
	<title>IHUB TALENT</title>
</head>
<body>
	<script>
		var a=10,b=20;

		document.writeln((++a + ++b)+"<br>");//32

		document.writeln((--a - --b)+"<br>");//-10
	</script>
</body>
</html>
```

2)Conditional operators

We have following list of conditional operators.

Operator	Description
>	greater then
<	less then
>=	greater then equals to
<=	less then equals to
==	equals to
!=	not equals to

ex:1

```
<!DOCTYPE html>
<html>
<head>
        <title>IHUB TALENT</title>
</head>
<body>
        <script>

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

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

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

                document.writeln((20>=20)+"<br>");//true
        </script>
</body>
</html>
```

ex:2

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

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

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

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

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

                document.writeln((20!=20)+"<br>");//false
        </script>
</body>
</html>
```

3)Bitwise operators

We have following list of bitwise operators.

Operator	Description
&	Bitwise AND
|	Bitwise OR
^	Bitwise XOR
~	Bitwise NOT
>>	Right Shift
<<	Left Shift

Bitwise AND (&)

```
<!DOCTYPE html>
<html>
<head>
    <title>IHUB TALENT</title>
</head>
<body>
    <script>

        var a=10,b=15;
        var c= a & b;
        document.writeln(c);//10
        /*
            10 - 1010
            15 - 1111
            ----------
            &  - 1010
                     <---
            0*1 +  1*2  +  0*4 +  1*8

            0 + 2 + 0 + 8

            10
        */
        </script>
</body>
</html>
```

Bitwise OR (|)

```
<!DOCTYPE html>
<html>
<head>
        <title>IHUB TALENT</title>
</head>
<body>
        <script>

                var a=10,b=15;
                var c= a | b;
                document.writeln(c);//15
                /*
                        10 - 1010
                        15 - 1111
                        ----------
                        |  - 1111
                                        <--
                        1*1 + 1*2 + 1*4 + 1*8
                        1 + 2 + 4 + 8
                        15
        */

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

Bitwise XOR (^)

```
<!DOCTYPE html>
<html>
<head>
    <title>IHUB TALENT</title>
</head>
<body>
    <script>

        var a=10,b=15;
        var c= a ^ b;
        document.writeln(c);//5
        /*
            10 - 1010
            15 - 1111
            ----------
            ^  - 0101
                    <--
            1*1 + 0*2 + 1*4 + 0*8
            1 + 0 + 4 + 0
            5
        */

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

Bitwise NOT (~)

```
<!DOCTYPE html>
<html>
<head>
    <title>IHUB TALENT</title>
</head>
<body>
    <script>

        var a=~10;
        document.writeln(a+"<br>");//-11

        var b=~(-30);
        document.writeln(b+"<br>");//29


    </script>
</body>
```

```
</html>
```

Right Shift (>>)

10 >> 1 = 10/2

10 >> 2 = 10/4

10 >> 3 = 10/8

10 >> 4 = 10/16

ex:

```
<!DOCTYPE html>
<html>
<head>
        <title>IHUB TALENT</title>
</head>
<body>
        <script>

                var a = 200 >> 7;
                document.writeln(a);// 200/128



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

Left Shift (<<)

10 << 1 = 10*2

10 << 2 = 10*4

10 << 3 = 10*8

10 << 4 = 10*16

ex:

```
<!DOCTYPE html>
<html>
<head>
        <title>IHUB TALENT</title>
</head>
<body>
        <script>


                var a = 100 << 4;
                document.writeln(a);// 100 * 16



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

4) logical operators

We have following list of logical operators.

Operator	Description
&&	Logical AND
||	Logical OR
!	Logical NOT

Logical AND (&&)

```
<!DOCTYPE html>
<html>
<head>
        <title>IHUB TALENT</title>
</head>
<body>
        <script>

                document.writeln(((5>6) && (6<9))+"<br>");//false
                document.writeln(((50>6) && (6<9))+"<br>");//true
        </script>
</body>
</html>
```

Logical OR

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

```
        <title>IHUB TALENT</title>
</head>
<body>
        <script>


                document.writeln(((5>6) || (6<9))+"<br>");//true
                document.writeln(((5>6) || (60<9))+"<br>");//false


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

Logical Not (!)

```
<!DOCTYPE html>
<html>
<head>
        <title>IHUB TALENT</title>
</head>
<body>
        <script>


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


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

5)Assignment operators

We have following list of assignment operators.

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

ex:

```
<!DOCTYPE html>
<html>
<head>
        <title>IHUB TALENT</title>
</head>
<body>
        <script>

                var a=10;
        a+=20;

        document.writeln(a);// a = a+ 20

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

ex:

```
<!DOCTYPE html>
<html>
<head>
    <title>IHUB TALENT</title>
</head>
<body>
    <script>

        var a=10;

        a-=20;

        document.writeln(a);// a = a - 20

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

ex:

```
<!DOCTYPE html>
<html>
<head>
        <title>IHUB TALENT</title>
</head>
<body>
        <script>

                var a=10;

                a*=20;

                document.writeln(a);// a = a * 20

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

ex:

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

```
        <script>

                var a=10;

                a/=20;

                document.writeln(a);// a = a / 20

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

ex:

```
<!DOCTYPE html>
<html>
<head>
        <title>IHUB TALENT</title>
</head>
<body>
        <script>

                var a=10;

                a%=20;

                document.writeln(a);// a = a % 20

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

6)Special operators

We have following list of special operators.

Operator	Description
?:	It is used to represent conditional statements.
new	It is used to create an instance.
typeof	It is used to check type of an object.

ex:1

```
<!DOCTYPE html>
<html>
<head>
        <title>IHUB TALENT</title>
</head>
<body>
        <script>

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

        </script>
</body>
```

```
</html>
```

ex:2

```
<!DOCTYPE html>

<html>

<head>

        <title>IHUB TALENT</title>

</head>

<body>

        <script>


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


        </script>

</body>

</html>
```

ex:3

```
<!DOCTYPE html>

<html>

<head>

        <title>IHUB TALENT</title>

</head>

<body>

        <script>


                var a=10;
                document.writeln(typeof(a)+"<br>");//number

                var b="hi";
                document.writeln(typeof(b)+"<br>");//string

                var c=true;
                document.writeln(typeof(c)+"<br>");//boolean

                var d=null;
                document.writeln(typeof(d)+"<br>");//object

                var e;
                document.writeln(typeof(e)+"<br>");//undefined

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

Interview Questions

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

==

- **It is used to check values are same or not.**

ex:

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

```
<head>
        <title>IHUB TALENT</title>
</head>
<body>
        <script>

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

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

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

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

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

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

===

- **It will check values and datatypes are same or not.**

ex:

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

```
    <title>IHUB TALENT</title>
</head>
<body>
    <script>

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

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

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

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

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

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

Q)Write a javascript program to display sum of two numbers?

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

```
        <script>
            var val1=prompt("Enter the First Number :");
            var val2=prompt("Enter the Second Number :");

            var a=parseInt(val1);
            var b=parseInt(val2);

            var c=a+b;

            document.writeln("sum of two numbers is ="+c);

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

Q)Write a javascript program to perform swapping of two numbers?

```
<!DOCTYPE html>
<html>
    <head>
        <title>IHUB TALENT</title>
    </head>
    <body>
        <script>

            var a=10,b=20;
            document.writeln("Before swapping A="+a+" and
B="+b+"<br>");

            //logic
            var temp=a;
            a=b;
            b=temp;

            document.writeln("After swapping A="+a+" and
B="+b+"<br>");
        </script>
    </body>
</html>
```

JavaScript IF ELSE STMT

- It is used to evaluate the code either our condition is true or false.
- We have three forms of javascript if else stmt.

i) IF STMT

ii) IF ELSE STMT

iii) IF ELSE IF STMT

i) IF STMT

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

syntax:

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

ex:1

```
<!DOCTYPE html>
<html>
    <head>
        <title>IHUB TALENT</title>
    </head>
    <body>
        <script>

            if(0,1,2,3,4,5)
            {
                document.writeln("Hello"); //Hello
            }

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

ex:

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

```
    <head>
        <title>IHUB TALENT</title>
    </head>
    <body>
        <script>

            if(5,4,3,2,1,0)
            {
                document.writeln("Hello"); //nothing
            }

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

ex:

```
<!DOCTYPE html>
<html>
    <head>
        <title>IHUB TALENT</title>
    </head>
    <body>
        <script>

            if((6>>4) && (10<<2))
            {
                document.writeln("Hello"); //nothing
            }

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

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

```
<!DOCTYPE html>
<html>
        <head>
            <title>IHUB TALENT</title>
        </head>
        <body>
            <script>

                var val1=prompt("Enter the first value :");
                var val2=prompt("Enter the second value :");
                var a=parseInt(val1);
                var b=parseInt(val2);
                if(a>b)
                    document.writeln(a+" is greatest");
                if(b>a)
                    document.writeln(b+" is greatest");
                </script>
        </body>
</html>
```

ii) IF ELSE STMT

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

syntax:

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

ex:

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

```
    <body>
        <script>

            if(~0)
            {
                document.writeln("TRUE");
            }
            else
            {
                document.writeln("FALSE");
            }

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

o/p:

```
    TRUE
```

ex:

```
<!DOCTYPE html>
<html>
    <head>
        <title>IHUB TALENT</title>
    </head>
    <body>
        <script>
```

```
                if(5>10)
                {
                        document.writeln("TRUE");
                }
                else
                {
                        document.writeln("FALSE");
                }

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

o/p: FALSE

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

```
<!DOCTYPE html>
<html>
        <head>
                <title>IHUB TALENT</title>
        </head>
        <body>
                <script>

                        var n=parseInt(prompt("Enter the number :"));
```

```
                    if(n%2==0)
                              document.writeln("It is even number");
                    else
                              document.writeln("It is odd number");


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

iii) IF ELSE IF STMT

- It will evaluate the code based on the multiple conditions.

syntax:

```
     if(condition1)
     {
          - //code to be evaluate if cond1 is true
     }
     else if(condition2)
     {
          - //code to be evaluate if cond2 is true
     }
     else if(condition3)
     {
          - //code to be evaluate if cond3 is true
     }
     else
     {
        - //code to be evaluate if all conditions are false
    }
```

ex:

```
<!DOCTYPE html>
<html>
    <head>
        <title>IHUB TALENT</title>
    </head>
    <body>
        <script>

            var option=parseInt(prompt("Enter the option  :"));

            if(option==100)
                document.writeln("It is a police number");
            else if(option==103)
                document.writeln("It is a enquiry number");
            else if(option==108)
                document.writeln("It is a emergency number");
            else
                document.writeln("It is odd number");

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

JavaScript Switch case

- It is used to evalute the code based on multiple conditions.
- It is similar to if else if stmt.
- Javascript switch case is more convenient when compare to javascript if else if stmt because we can declare numbers, characters, strings and etc.

syntax:

```
    switch(condition)

    {

        case val1:  //code to be evaluate

                break stmt;


        case val2:  //code to be evaluate

                break stmt;

        -

        -

        default:    //code to be evaluate if all cases are false.

    }
```

ex:

```
<!DOCTYPE html>

<html>

    <head>

        <title>IHUB TALENT</title>

    </head>

    <body>
```

```
        <script>

            var option=parseInt(prompt("Enter the option  :"));

            switch(option)
            {
                case 100: document.writeln("It is a police number");
                         break;
                case 103: document.writeln("It is a enquiry number");
                         break;
                case 108: document.writeln("It is a emergency number");
                         break;
                default: document.writeln("Invalid option");
            }



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

ex:

```
<!DOCTYPE html>
<html>
    <head>
        <title>IHUB TALENT</title>
    </head>
    <body>
        <script>

            var al=prompt("Enter the  alphabet  :");

            switch(al)
            {
                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 not a vowel");
            }
        </script>
    </body>
</html>
```

ex:

```
<!DOCTYPE html>
```

```
<html>
    <head>
        <title>IHUB TALENT</title>
    </head>
    <body>
        <script>

            var val=prompt("Enter the  month number in word  :");

            switch(val)
            {
                case "one": document.writeln("It is a january");
                         break;
                case "two": document.writeln("It is a February");
                         break;
                case "three": document.writeln("It is a march");
                         break;
                case "four": document.writeln("It is a April");
                         break;
                case "five": document.writeln("It is a May");
                         break;
                default: document.writeln("Coming soon");
            }

        </script>
```

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

JavaScript LOOPS

- ➢ LOOPS are used to iteration the code for multiple times.

Javascript contains four types of loops.

1) do while loop

2) while loop

3) for loop

4) for in loop

1) do while loop

- ➢ It will evaluate the code untill our condition is true.

syntax:

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

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB TALENT</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>IHUB TALENT</title>
    </head>
    <body>
        <script>

            var i=1,sum=0;
            do
```

```
                    {

                              sum+=i;
                              i++;

                    }while(i<=10)

                    document.writeln("sum of 10 natural numbers is ="+sum);

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

2) while loop

- It will evaluate the code untill our condition is true.

syntax:

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

ex:

```
<!DOCTYPE html>
```

```
<html>
	<head>
		<title>IHUB TALENT</title>
	</head>
	<body>
		<script>

			var i=1;

			while(i<=10)
			{
				document.writeln(i);//1 2 3 4 5 6 7 8 9 10

				i++;
			}

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

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB TALENT</title>
	</head>
	<body>
		<script>

			var i=1,n=5;

			while(i<=10)
			{
					document.writeln(n+" * "+i+" = "+n*i+"<br>");
					i++;
			}

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

3) for loop

- It will evaluate the code untill our condition is true.

syntax:

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

ex:

```
<!DOCTYPE html>
```

```
<html>
	<head>
		<title>IHUB TALENT</title>
	</head>
	<body>
		<script>

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

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB TALENT</title>
	</head>
	<body>
		<script>

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

```
                        for(var i=n;i>=1;i--)
                        {
                                fact=fact*i;
                        }
                        document.writeln("factorial of a given number is ="+fact);
                </script>
        </body>
</html>
```

4) for in loop

- It will iterate the elements from array.

syntax:

```
        for(variable in  array_variable)
        {
                -
                - //code to be evaluate
                -
        }
```

ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>IHUB TALENT</title>
        </head>
        <body>
        <script>

            var arr=[10,20,30];

            for(var i in arr)
            {
                    document.writeln(arr[i]);
            }
        </script>
    </body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
    <head>
        <title>IHUB TALENT</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>IHUB TALENT</title>
    </head>
    <body>
        <script>

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

            for(var i in arr)
            {
                    document.writeln(arr[i]);
            }
        </script>
    </body>
</html>
```

Loop Patterns

1)

```
* * * *

* * * *

* * * *

* * * *
```

```
<!DOCTYPE html>

<html>

        <head>

                <title>IHUB TALENT</title>

        </head>

        <body>

                <script>


                        for(var i=1;i<=4;i++)

                        {

                                for(var j=1;j<=4;j++)

                                {

                                        document.writeln("*");

                                }

                                //new line

                                document.writeln("<br>");

                        }

                </script>
```

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

2)

```
*
* *
* * *
* * * *
```

```
<!DOCTYPE html>
<html>
    <head>
        <title>IHUB TALENT</title>
    </head>
    <body>
        <script>

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

Q)Write a javascript program to display reverse of a given number?

input:

```
    123
```

output:

```
    321
```

ex:

```
<!DOCTYPE html>
<html>
    <head>
        <title>IHUB TALENT</title>
    </head>
    <body>
        <script>
            var n=123;
            var rem,rev=0;

            while(n>0)
            {
                rem=n%10;
                rev=rev*10+rem;
```

```
                        n=parseInt(n/10);
                   }
                   document.writeln(rev);



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

Javascript Functions

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

syntax:

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

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

ex:1

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

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

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

ex:2

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

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

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

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

ex:3

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

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

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

ex:4

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

            function f1()
            {
                document.bgColor="#FF0000";
            }
        </script>
```

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

- One function can call to another function .

ex:

```
<!DOCTYPE html>
<html>
<head>
        <title>IHUB TALENT</title>
</head>
<body>
        <script>
                function f1()
                {
                        document.writeln("f1 function <br>");
                        f2();
                }
                function f2()
                {
                        document.writeln("f2 function <br>");

                }

                //calling
```

```
            f1();
        </script>


</body>
</html>
```

Interview Questions

Q)Disadvantages of JavaScript?

Client-Side Security

Browser Support

Stop Rendering

Slow Bitwise Operation

Single level Inheritance

Q)Types of Functions in javascript?

We have three types of functions in javascript.

1)Named Function

2)Anonymous Function

3)Arrow Function

1)Named Function

- These types of functions contains name at the time of definition.

ex:

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

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


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



</body>
</html>
```

2)Anonymous Function

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

ex:

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

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

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



</body>
</html>
```

3)Arrow function

- According to ES6 standard we need to use arrow function.
- Arrow functions are more secured when compare to named function and anonymous function.

ex:

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

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



</body>
</html>
```

Q) What is JavaScript Closure?

- A closure is the combination of a 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()
                {
                    document.writeln(a+" "+b);
                }
                f2();
            }
            f1();

        </script>


</body>
</html>
```

Javascript Object

- 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>
```

BOM (Browser Object Model)

- **The Browser Object Model is used to interact with browser.**
- **The default object for a browser is window object.It means we can call all the functions by using window or directly.**

ex:

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

Window Object

- It is used to create a window on a browser.
- A window object is created automically by the browser.
- A "window" is a object of browser but not javascript.
- Javascript objects are String,Array,Date and etc.
- A "window" object is used to write programming related to browser.
- With the help of window object we can perform following activities very easily.

1)It display dialog boxes and pop boxes.

2)We can find width and height of a browser.

3)We can move or resize the browser.

4)Scroll to the browser.

5)Get URL,hostname,protocol and etc of a browser.

6)We can get javascript history.

1)alert()

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

ex:

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

```
                    }

                </script>

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

2)confirm()

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

ex:

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

```
                        alert("ok");
                    }
                    else
                    {
                        alert("cancel");
                    }
                }
            </script>

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

3)prompt()

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

ex:

```
<!DOCTYPE html>
<html>
    <head>
        <title>IHUB Talent</title>
    </head>
    <body>
            <script type="text/javascript">
                function f1()
                {
                    var v=prompt("Who are you?");
```

```
                                        alert("Welcome :"+v);
                              }
                    </script>

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

innerWidth and innerHeight

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



          </body>
</html>
```

Note:

Press "CTRL + +" for zoomin.

Press "CTRL + -" for zoomout.

window.open()

ex:1

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

                        <button onclick="openWindow()">open a new
window</button>
        </body>
</html>
```

ex:2

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

        window.open("http://www.google.com","_blank");
                                }
                        </script>


                        <button onclick="openWindow()">open a new
window</button>
        </body>
</html>
```

ex:3

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

            <button onclick="openWindow()">open a new window</button>
    </body>
</html>
```

close()

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

```
myWindow=window.open("http://www.google.com","","width=300px,height=300px");
                }
                function closeWindow()
                {
                    myWindow.close();
                }
            </script>

            <button onclick="openWindow()">open a new window</button>
            <button onclick="closeWindow()">close a window</button>
    </body>
</html>
```

- 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 continously function untill we call clearInterval() method or window is closed.

ex:

```
<!DOCTYPE html>
<html>
    <head>
        <title>IHUB Talent</title>
        <style>
            div
            {
                width:150px;
                height: 150px;
                background-color: #FF0000;
            }
```

```
			</style>
		</head>
		<body>
				<script type="text/javascript">
						var a=0;
						setInterval(Anim,1000);

						function Anim()
						{
							a = a + 10;
				var target=document.getElementById("myId");
							target.style.marginLeft= a + 'px';
						}

				</script>

				<div id="myId"></div>
		</body>
</html>
```

clearInterval()

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

ex:

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

                    function Anim()
                    {
                        a = a + 10;
                        if(a==100)
                        {
                            clearInterval(id);
                        }
                        var
target=document.getElementById("myId");
                        target.style.marginLeft= a + 'px';
                    }

            </script>

            <div id="myId"></div>
    </body>
</html>
```

window history

```
<!DOCTYPE html>
<html>
    <head>
        <title>IHUB Talent</title>
        <style>
            a
            {
                text-decoration: none;
                color:blue;
            }
        </style>
    </head>
    <body>
```

```
            <a href="javascript:history.back()">



                    &laquo; previous
            </a>
            &nbsp; &nbsp;
            <a href="javascript:history.forward()">
                    next &raquo;
            </a>

    </body>
</html>
```

Note:

www.ihubtalent.com

file:///D:/IHUB-TRAINING-BATCHES/ReactAngularBatch/practicals/index.html

www.qualitythought.in

localStorage

- A localStorage properties allows us to save key/value pairs in a browser window.
- A localStorage allows us to store the data with no-expiry.It means our data will not be
- delete even if we close the browser.It will be present for next day.
- A localStorage is a read-only.
- To add the data in a localStorage we need to use setItem(key,value) function.
- To read the data from localStorage we need to use getItem(key) function.
- To remove perticular data from localStorage we need to use removeItem(key) function.
- To remove all the data from localStorage we need to use clear() function.

ex:

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

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

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

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

                //remove all items
                localStorage.clear();

                //reading the items

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

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

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

ex:2

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

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

                        //reading the items

        document.writeln(localStorage.getItem("FirstName")+"<br>");
//Morries

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

Sessionstorage

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

ex:

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

```
                    //reading the items

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

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

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

                    //remove all items
                    sessionStorage.clear();

                    //reading the items

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

          document.writeln(sessionStorage.getItem("Age")+"<br>");s

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

Javascript promises

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

A Promise has four states:

1)fulfilled : Action related to the promise succeeded

2)rejected : Action related to the promise failed

3)pending : Promise is still pending i.e. not fulfilled or rejected yet

4)settled : Promise has fulfilled or rejected

- A promise can be created using Promise constructor.

Syntax:

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

ex:1

```
<script>
var promise = new Promise(function(resolve, reject) {
        resolve('IHub Talent');
})


promise
        .then(function(successMessage) {
        //success handler function is invoked
                console.log(successMessage);
        }, function(errorMessage) {
```

```
        console.log(errorMessage);
    })
</script>
```

ex:2

```
<script>
var promise = new Promise(function(resolve, reject) {
    reject('Error occured');
})

promise
    .then(function(successMessage) {
    //success handler function is invoked
        console.log(successMessage);
    }, function(errorMessage) {
        console.log(errorMessage);
    })


</script>
```

ex:3

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

</script>
```

Q)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 is undefined.	It can't be accessible without declaration.	It can't be accessible without declaration.

initialization

ex:1

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

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

ex:2

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

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

```
        let i;
        document.writeln(i);
    </script>

</body>
</html>
```

ex:3

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

    <script type="text/javascript">
        const i;
        document.writeln(i);
    </script>

</body>
</html>
```

update:

ex:1

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

        <script type="text/javascript">
                var i=10;
                i=20;
                document.writeln(i);
        </script>

</body>
</html>
```

ex:2

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

```
    <script type="text/javascript">
        let i=10;
        i=20;
        document.writeln(i);
    </script>

</body>
</html>
```

ex:3

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

    <script type="text/javascript">
        const i=10;
        i=20;
        document.writeln(i);
    </script>

</body>
</html>
```

redeclared

ex:1

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

	<script type="text/javascript">
		var i=10;
		var i=20;
		document.writeln(i);//
	</script>

</body>
</html>
```

ex:2

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

```
        <script type="text/javascript">
                let i=10;
                let i=20;
                document.writeln(i);//C.T.E
        </script>

</body>
</html>
```

ex:3

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

        <script type="text/javascript">
                const i=10;
                const i=20;
                document.writeln(i);//C.T.E
        </script>

</body>
</html>
```

accessible

ex:1

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

        <script type="text/javascript">

                document.writeln(i);
                i=10;
                var i;
        </script>

</body>
</html>
```

ex:2

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

```
<body>


        <script type="text/javascript">
                document.writeln(i);
                let i=10;
                document.writeln(i);
        </script>


</body>
</html>
```

ex:3

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


        <script type="text/javascript">
                document.writeln(i);
                const i=10;
                document.writeln(i);
        </script>


</body>
```

```
</html>
```

Q)What is JavaScript Math object?

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

ex:

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

                document.writeln(Math.ceil(10.6));

                document.writeln(Math.floor(10.6));

                document.writeln(Math.round(10.6));

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

Object Oriented Programming System / Structure (OOPS)

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

ex:

class

object

abstraction

encapsulation

inheritance

and

polymorphism

- JavaScript is not a object oriented programming language . It is a object based programming language.

Q)What is class in JavaScript?

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

Ex:

```
<script>
class Example
{
        -
```

```
                -//code to be declare

                -

        }

</script>
```

Q)What is Constructor in JavaScript?

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

Ex:

```
        class Example

        {

                constructor()

                {

                        -

                        -// code to be declare

                        -

                }

        }
```

Q)What is object in JavaScript?

- A JavaScript object is an entity having states and behaviors (properties and methods).

Syntax:

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

ex:1

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

            var e=new Example();


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

ex:2

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

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

                        var e1=new Example();
                        var e2=new Example();
                </script>
</body>
</html>
```

ex:3

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

```
<body>
          <script type="text/javascript">
               class  Example
               {
                    constructor(id)
                    {
                         document.writeln(id+"<br>");
                    }
               }

               var e1=new Example(101);
               var e2=new Example(201);



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

Q)What is Abstraction in JavaScript?

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

Q)What is Encapsulation in JavaScript?

- The process of wrapping property and function within a single unit is known as encapsulation.
- To achieve an encapsulation in JavaScript we need to do following things.
- Use var keyword to make data members private.
- Use setter methods to set the data and getter methods to get that data.

ex:

```
<!DOCTYPE html>

<html>

<head>

        <title>IHUB TALENT</title>

</head>

<body>

                <script type="text/javascript">

                        class  Example

                        {

                                //setter

                                constructor(id,name,sal)

                                {

                                        this.eid=id;

                                        this.ename=name;

                                        this.esal=sal;

                                }


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

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

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

ex:2

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

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

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

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

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

Q)What is Inheritance in JavaScript?

- The JavaScript inheritance is a mechanism that allows us to create new classes on the basis of already existing classes.
- It provides flexibility to the child class to reuse the methods and variables of a parent class.
- The JavaScript extends keyword is used to create a child class on the basis of a parent class.

ex:

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

```
                    document.writeln("B-class <br>");
                }
            }
            var a=new A();
            a.f1();
            var b=new B();
            b.f1();
            b.f2();
        </script>
</body>
</html>
```

Q)What is polymorphism in JavaScript?

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

Ex:

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

```
        document.writeln("B is invoked");
    }
}
A a=new A();
a.display(); // A is invoked
B b=new B();
b.display(); // B is invoked
```