* ***JavaScript***

**Q) What is the difference between Java and JavaScript?**

Java JavaScript

------------- --------------

It is a non-scripting language. It is a scripting language.

(We won't compile the program)

We can run individually. We can't run individually.

It does not required browser window It requires browser window for

For execution. execution.

It is a object oriented programming It is a object based programming language. language.

It is a strongly typed checking It is a loosely typed language.

language.

It is a complex language. It is easy language compare to java.

**JavaScript History**

**==================**

Original name of JavaScript is LiveScript.

LiveScript was developed in the year of 1990's.

In 1995, Brenden Eich popular scientist of Netscape Corporation renamed LiveScript to JavaScript.

JavaScript consider as weakly typed checking language.

The official name of JavaScript is ECMA Script.

ECMA means Europeon Computer Manufacturing Association.

**Advantages of JavaScript**

**==========================**

1) It is used to develop interactive web pages.

2) It is used to perform client side form validation.

3) It is used to display dialog boxes and popup boxes.

4) It supports drag and drop menu.

5) It supports objects like Date,Array,String,RegEx and etc.

6) It supports cookies.

7) It is used to add dynamic content in a web page.

and etc.

**Javascript program**

---------------------------------

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

stmt1;

stmt2;

stmt3;

</script>

Here type and language attributes are optional to declare.

In javascript, semicolon is not mandatory because it is a loosely typed checking language.

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

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

document.write("Welcome to JavaScript class");

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

document.write("Welcome to JavaScript class")

</script>

</body>

</html>

If we need space at the end of the statement then we need to use document.writeln().

ex:

--

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

document.writeln("Welcome to JavaScript")

document.writeln('This is IHUB Talent')

</script>

</body>

</html>

In order to get a new line we need to use <br> tag.

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')

</script>

</body>

</html>

We can write HTML and CSS styles in a javascript code.

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

document.writeln("<h1 style='color:blue'>Welcome to JavaScript</h1>")

document.writeln("<br>")

document.writeln("<p style='text-align:center;background-color:cyan;'>This is IHUB Talent</p>")

</script>

</body>

</html>

If any document contains HTML, CSS and JavaScript is called DHTML (Dynamic Hypertext Markup Language).

**JavaScript Engine**

**==================**

JavaScript engine allows javascript code to execute on a browser window.

It converts user understandable scripting language to machine understable scripting language.

Every browser contains javascript engine as given below.

Ex:

Browser JavaScript Engine

------------- ---------------------------------

Chrome V8 Engine

Mozilla SpiderMonkey

Edge Chakra

and etc.

**Comments in JavaScript**

**=======================**

Comments are create for documentation purpose.

Comments improve readability of javascript code.

JavaScript engine does not display javascript comments in output.

We have two types of comments in javascript.

1) Single line comment

--------------------

It is used to comment a single line.

ex:

//

2) Multiple line comment

--------------------

It is more convenient when compare to single line comment because we can comment a single

line and multiple lines.

ex:

/\*

-

-

-

\*/

**Types of output statement in JavaScript**

**==================================**

We have two types of 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>MyPage!</title>

</head>

<body>

<script>

document.writeln("<h1>Welcome to JavaScript</h1>");

</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 enable developer option.

To see the developer option we need to press F12 Function key.

ex:

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

console.log("Welcome to JavaScript");

</script>

</body>

</html>

**Types of variables in javascript**

**=================================**

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.

We have same rules for javascript variables as we have for identifiers.

Rule1:

--------

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

ex:

\_=10;

$=20;

abcd=30;

Rule2:

-----

After first alphabet it can have any number of digits.

ex:

a1234; //valid

Rule3:

-----

Every identifier is a case sensitive.

ex:

a=10;

A=10;

In javascript variables are divided into two types.

1)Local variable

2)Global variable

1)Local variable

-------------------------

If we declare any variable inside a block or a function is called local variable.

We can access local variable within the block or function only.

ex:1

-----

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

function f1()

{

var i=10;

document.writeln(i);

}

//call the function

f1();

</script>

</body>

</html>

ex:2

----

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

function f1()

{

var i=10;

document.writeln(i);

}

function f2()

{

document.writeln(i);

}

//call the function

f1();

f2();

</script>

</body>

</html>

2)Global variable

----------------------------

If we declare any variable outside a block or a function is called global variable.

We can access global variable within the block or function or outside of the block or function.

ex:1

------

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var i=100;

function f1()

{

document.writeln(i);

}

//call the function

f1();

</script>

</body>

</html>

ex:2

----

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var i=100;

function f1()

{

document.writeln(i);

}

function f2()

{

document.writeln(i);

}

//call the function

f1();

f2();

</script>

</body>

</html>

**JavaScript Datatypes**

**====================**

JavaScript is a dynamically typed language so we don't use any datatype at the time of variable declaration.

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

syntax:

var variable\_name=value;

ex:

var i = 10;

Internally , JavaScript engine determines type of 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.

ex:

Datatype Description

---------------- ----------------------

number It is used to represent numbers

String It is used to represent strings

boolean It is used to represent boolean

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=20.65;

document.writeln(j+"<br>");//20.65

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var i="Hi";

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

var j='Hello';

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

</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=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 represent instance through which we can

access the members.

Arrays It is used to represent similar elements.

RegEx It is used to represent regular expression

**Q) Difference between null and undefined?**

undefined

----------------

In JavaScript, undefined is a type.

It means a variable declared, but no value has been assigned.

Ex:

<script>

var x;

document.writeln(x); //undefined

</script>

null

-----

In JavaScript, null is an object.

A null in JavaScript is an assignment value.

Ex:

<script>

var x=null;

document.writeln(x); //null

</script>

**Q) 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 declare 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 seperately.

Disadvantages:

> If code increases then it will increase complexity of a programmer.

2) External JavaScript

-------------------------------------

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

We can't execute javascript file directly on a browser.

Advantages:

-----------------------

> If code increases then it won't increase the complexity of a programmer.

Disadvantages:

------------------------

> There is a confusion of multiple files.

**JavaScript operators**

**==================**

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.

We have following list of operators in javascript.

1) Arithmetic operators

2) Conditional operators

3) Bitwise operators

4) Logical operators

5) Assignment operators

6) Special operators

1) Arithmetic operators

---------------------------------------

We have following list of arithmetic operators.

ex:

operator Description

-------------- ----------------------

% modules

/ division

\* multiplication

+ addition

- subtraction

++ incrementation

-- decrementation

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

document.writeln((10%2)+"<br>");

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

document.writeln((10\*2)+"<br>");

document.writeln((10-2)+"<br>");

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

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

</script>

</body>

</html>

ex:

----

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var i=10;

var j = i++ + i++;

document.writeln(i+" "+j);

</script>

</body>

</html>

ex:

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var i=10;

var j = i-- + i--;

document.writeln(i+" "+j);// 8 19

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var i=10;

var j = ++i + ++i;

document.writeln(i+" "+j);// 12 23

</script>

</body>

</html>

ex:

----

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var i=10;

var j = --i + --i;

document.writeln(i+" "+j);// 8 17

</script>

</body>

</html>

2) Conditional operators

----------------------------------------

We have following list of conditional operators.

ex:

operator description

-------------- -------------------

> greater than

< less than

>= greatera than equals to

<= less than equals to

== equals to

!= not equals to

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

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

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

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

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

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

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

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

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

</script>

</body>

</html>

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

**==**

**----**

It is used to check values are same or not.

ex:

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

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

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

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

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

</script>

</body>

</html>

**===**

----

It is used to check values as well as datatype is same or not.

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

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

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

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

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

</script>

</body>

</html>

3) Bitwise Operators

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

We have following list of bitwise operators.

ex:

Operator Description

-------- ------------

& Bitwise AND

| Bitwise OR

^ Bitwise XOR

~ Bitwise NOT

>> Right Shift

<< Left Shift

Bitwise AND (&)

--------------------------

Bitwise AND operator deals with binary number.

ex:

truth table

-------------------

T T = T

T F = F

F T = F

F F = F

ex:

----

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var a = 10;

var b = 15;

var c = a & b;

document.writeln(c);

/\*

10 - 1010

15 - 1111

----------

& - 1010 <----

0\*1 + 1\*2 + 0\*4 + 1\*8

0 + 2 + 0 + 8 = 10

\*/

</script>

</body>

</html>

Bitwise OR

-------------------

Bitwise OR operator deals with binary numbers.

ex:

Truth table

------------------

T T = T

T F = T

F T = T

F F = F

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var a = 10;

var b = 15;

var c = a | b;

document.writeln(c);

/\*

10 - 1010

15 - 1111

----------

| - 1111 <----

1\*1 + 1\*2 + 1\*4 + 1\*8

1 + 2 + 4 + 8 = 15

\*/

</script>

</body>

</html>

Bitwise XOR (^)

---------------------------

Bitwise XOR operator deals with binary number.

ex:

Truth table

---------------------

T T = F

T F = T

F T = T

F F = F

ex:

----

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var a = 10;

var b = 15;

var c = a ^ b;

document.writeln(c);

/\*

10 - 1010

15 - 1111

----------

^ - 0101 <----

1\*1 + 0\*2 + 1\*4 + 0\*8

1 + 0 + 4 + 0 = 5

\*/

</script>

</body>

</html>

Bitwise NOT (~)

----------------

ex:

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var i=~10;

document.writeln(i);//-11

/\*

-(n+1)

-(10+1)

\*/

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var i=~(-10);

document.writeln(i);//9

/\*

-(n+1)

-(-10+1)

- (-9)

9

\*/

</script>

</body>

</html>

Right Shift (>>)

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

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var i = 10 >> 3;

document.writeln(i); // 10/8 (2\*2\*2)

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var i = 20 >> 4;

document.writeln(i); // 20 / 16

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var i = 10 >> 4;

document.writeln(i); // 10 / 16

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var i = 10 << 2;

document.writeln(i); // 10 \* (2\*2) = 40

</script>

</body>

</html>

ex:

--

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var i = 100 << 3;

document.writeln(i); // 100 \* (2\*2\*2) = 800

</script>

</body>

</html>

4) Logical operators

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

We have following list of logical operators.

ex:

operator description

------------- ---------------------

&& Logical AND

|| Logical OR

! Logical NOT

Logical AND (&&)

-----------------------------

Logical AND operator deals with boolean values either true or false.

truth table

------------------

T T = T

T F = F

F T = F

F F = F

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var i = (10 == 10) && (100 === "100");

document.writeln(i); //false

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var i = (5>2) && (6<10);

document.writeln(i); //true

</script>

</body>

</html>

Logical OR (||)

----------------------

Logical OR operator deals with boolean values either true or false.

Truth table

-------------------

T T = T

T F = T

F T = T

F F = F

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var i = true || false;

document.writeln(i); //true

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var i = 0 || false;

document.writeln(i); //false

</script>

</body>

</html>

Logical NOT (!)

-------------------------

Logical NOT operator deals with boolean values either true or false

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var i = !(5>2);

document.writeln(i); //false

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var i = !(5>20);

document.writeln(i); //true

</script>

</body>

</html>

5) Assignment operators

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

We have following list of assignment operators.

ex:

operator description

-------- ------------

+= addition and equals to

-= subtraciton and equals to

\*= multiplication and equals to

/= Division and equals to

%= Modules and equals to

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var i =10;

i+=3; // i = 10 + 3

document.writeln(i); //13

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var i =10;

i-=3; // i = 10 - 3

document.writeln(i); //7

</script>

</body>

</html>

ex:

----

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var i =10;

i\*=3; // i = 10 \* 3

document.writeln(i); //30

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var i =10;

i/=3; // i = 10 / 3

document.writeln(i); //3.33

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var i =10;

i%=3; // i = 10 % 3

document.writeln(i); //1

</script>

</body>

</html>

6) Special operators

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

We have following list of special operators.

ex:

operator description

------- -----------

?: conditional

new It is used to create an instance.

typeof It is used to check type of an object.

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

(10>20)?document.writeln("TRUE"):document.writeln("FALSE");

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

(!(10>20))?document.writeln("TRUE"):document.writeln("FALSE");

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var val=prompt("Enter the Age : ");

var age=parseInt(val);

(age>=18)?document.writeln("Eligible"):document.writeln("Not Eligible");

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var i;

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

var j=null;

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

var k=10;

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

var l="Hi";

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

var m=true;

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

</script>

</body>

</html>

**JavaScript IF ELSE stmt**

**=======================**

We have three forms of JavaScript IF ELSE STMT.

1) IF STMT

2) IF ELSE STMT

3) IF ELSE IF STMT

1) IF STMT

------------------

It evaluates the code only if our condition is true.

ex:

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var i=10;

if(i>5)

{

document.writeln("TRUE");

}

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

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

{

document.writeln("TRUE");

}

</script>

</body>

</html>

ex:

---

<!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>

ii) IF ELSE STMT

----------------------------

It evaluates the code either our condition is true or false.

ex:

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

if(true)

document.writeln('welcome');

else

document.writeln('thankyou');

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

if(!true)

document.writeln('welcome');

else

document.writeln('thankyou');

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

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

var n = parseInt(val);

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 evaluates the code based on multiple conditions.

ex:

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var opt=10;

if(opt==100)

document.writeln("It is a police number");

else if(opt==103)

document.writeln("It is a enquiry number");

else if(opt==108)

document.writeln("It is a emergency number");

else

document.writeln("Invalid option");

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var a=5,b=100,c=179;

if((a>b) && (a>c))

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

else if((b>a) && (b>c))

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

else if((c>a) && (c>b))

document.writeln(c+" is greatest");

</script>

</body>

</html>

**JavaScript Switch**

**=================**

It is similar to if else if statement.

It is more convenient then if else if statement because we can declare numbers,characters and strings.

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>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

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

var opt=parseInt(ch);

switch(opt)

{

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"); break;

default : document.writeln('Invalid option');

}

</script>

</body>

</html>

**LOOPS**

**======**

It evalutes the code for multiple times.

We have four types of loops.

1) do while loop

2) while loop

3) for loop

4) for in loop

1) do while loop

------------------------

It evalutes the code how long 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.write(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=1,sum=0;

do

{

sum += i;

i++;

}while(i<=10);

document.writeln(sum);

</script>

</body>

</html>

2) while loop

--------------------

It evaluates the code how long our condition is true.

syntax:

-----

while(condition)

{

-

- //code to be execute

-

}

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var i=11;

while(i<=10)

{

document.writeln(i);

i++;

}

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var n=5;

var i=n,fact=1;

while(i>=1)

{

fact\*=i;

i--;

}

document.writeln(fact);

</script>

</body>

</html>

3) for loop

------------

It evaluates the code how long our condition is true.

syntax:

for(initialization;condition;incrementation/decrementation)

{

-

- //code to be evaluate

-

}

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

for(var i=10;i>=1;i--)

{

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

}

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var n=6;

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

{

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

}

</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];

for(var i in arr)

{

document.writeln(arr[i]);

}

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var arr=['a','b','c','d'];

for(var i in arr)

{

document.writeln(arr[i]);

}

</script>

</body>

</html>

**Interview Questions**

**===================**

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

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

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

var rem,rev=0;

while(n>0)

{

rem=n%10;

rev = rev \* 10 + rem;

n = parseInt(n / 10);

}

document.writeln(rev);

</script>

</body>

</html>

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

\* \* \* \*

\* \* \* \*

\* \* \* \*

\* \* \* \*

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

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

{

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

{

document.writeln("\*");

}

document.writeln("<br>");

}

</script>

</body>

</html>

**Q) Write a program to display below loop patterns?**

\*

\* \*

\* \* \*

\* \* \* \*

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

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

{

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

{

document.writeln("\*");

}

document.writeln("<br>");

}

</script>

</body>

</html>

**Assignment**

**============**

Q) Write a program to display below loop pattern?

1 1

1 2 2 1

1 2 3 3 2 1

1 2 3 4 4 3 2 1

**JavaScript Functions**

**==================**

JavaScript function is a block of code which is used to perform particular task.

JavaScript function we can declare by using function keyword followed by name and followed by parentheses i.e '()'.

JavaScript function contains letters, digits, underscore and dollar symbol. Having same rules as variables.

JavaScript parentheses contains arguments and each argument must separated with comma.

syntax

-----------

function <function\_name>(arg1,arg2,.....,argN)

{

-

- //block of code

-

}

JavaScript function will execute in following ways.

1) When we call or invoke a function

2) When event is occur

3) Self Invocation

1) When we call or invoke a function

--------------------------------------------------------------

ex:

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

function f1()

{

document.writeln("Function Example");

}

//calling

f1();

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

function f1()

{

document.bgColor="#FF00FF";

}

//calling

f1();

</script>

</body>

</html>

2) When event is occur

--------------------------------------

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

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

<script>

function f1()

{

document.writeln("Function Example");

}

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

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

<script>

function f1()

{

document.bgColor="#FFFF00";

}

</script>

</body>

</html>

In javascript, one function can call to another function.

ex:

----

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

function f1()

{

document.writeln("F1-Function");

}

function f2()

{

f1();

document.writeln("F2-Function");

}

f2();

</script>

</body>

</html>

In javascript, every function name is a case sensitive.

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

function f1()

{

document.writeln("F1-Function");

}

//F1(); // Error F1 is not defined

f1();

</script>

</body>

</html>

**No return type with no argument method**

**-----------------------------------------------------------------------**

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

function display()

{

document.writeln("Function1");

}

display();

</script>

</body>

</html>

**No return type with argument method**

**------------------------------------------------------------------**

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

function display(a,b)

{

document.writeln(a+b);

}

display(10,20);

</script>

</body>

</html>

**With return type with no argument method**

**----------------------------------------------------------------------------**

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

function display()

{

return "Hello World";

}

document.writeln(display());

</script>

</body>

</html>

**With return type with argument method**

**----------------------------------------------------------------------**

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

function display(a,b)

{

return a+b;

}

document.writeln(display(10,20));

</script>

</body>

</html>

**Types of functions in javascript**

**===========================**

We have three functions in javscript.

1) Named function

2) Anonymous function

3) Arrow function

1) Named function

--------------------------------

This type of function contains name at the time of declaration.

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

function f1()

{

document.writeln("Named Function");

}

f1();

</script>

</body>

</html>

2) Anonymous function

-------------------------------------

Anonymous function does not have any name at the time of declaration.

Anonymous function used 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 ES6 standards we need to use arrow function.

Arrow function is more secure then 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>

**JavaScript closures**

**=================**

JavaScript closure is functions bundled together along with lexical scope.

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

In Javascript closure , inner function can access the data of outer function.

ex:

--

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

//lexical scope

var i=10;

function f1()

{

var j=20;

function f2()

{

var k=30;

document.writeln(i+" "+j+" "+k);

}

//calling

f2();

}

//calling

f1();

</script>

</body>

</html>

**JavaScript Recursion**

**==================**

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

<!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 JavaScript hoisting?**

Hoisting is a default behavior of JavaScript where all the variable and function declarations are moved on top.

ex: ex:

<script> <script>

i=1; var i;

document.writeln(i); ===> i = 1;

var i; document.writeln(i);

</script> </script>

ex: ex:

<script> <script>

f1(); function f1()

function f1() {

{ document.writeln("F1-Function");

document.writeln("F1-Function"); ==> }

} f1();

</script> </script>

**JavaScript objects**

**===============**

A javascript object is an entity which is having state and behaviours.

In general, javascript object is a collection of properties and functions.

Javascript is a object based language because everything is present in objects.

Javascript is a template based but not class based.We don't need to create a class to get the object.We can create object directly.

There are three ways to create javascript objects.

1)By using Object literal

2)By creating instance of an Object i.e using new keyword.

3)By using Object constructor i.e using new keyword.

1) By using Object literal

---------------------------

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

emp={

"empId":101,

"empName":"Alan", "empSal":1000

}

document.writeln("Employee Id :"+emp.empId+"<br>");

document.writeln("Employee Name :"+emp.empName+"<br>") document.writeln("Employee Salary :"+emp.empSal+"<br>");

</script>

</body>

</html>

2) By creating instance of an Object

-----------------------------------------------------------

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var emp=new Object();

emp.empId=201;

emp.empName="Jose";

emp.empSal=2000;

document.writeln("Employee Id :"+emp.empId+"<br>"); document.writeln("Employee Name :"+emp.empName+"<br>") document.writeln("Employee Salary :"+emp.empSal+"<br>");

</script>

</body>

</html>

3) By using Object constructor

----------------------------

In this we need to create a function with parameter and each parameter we need to assign in a current object by using this keyword.

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

function emp(empId,empName,empSal)

{

this.empId=empId;

this.empName=empName;

this.empSal=empSal;

}

var e=new emp(301,"Lisa",3000);

document.writeln("Employee Id :"+e.empId+"<br>");

document.writeln("Employee Name :"+e.empName+"<br>");

document.writeln("Employee Salary :"+e.empSal+"<br>");

</script>

</body>

</html>

**JavaScript Arrays**

**=================**

JavaScript Array is an object which is used to hold similar elements.

Array index always starts with '0' because it is logical process.

There are three ways to create javascript arrays.

1) By using Array literal

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

3) By using Array constructor i.e using new operator

1) By using Array literal

-----------------------------------------

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var arr=[10,20,30];

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

{

document.writeln(arr[i]);

}

</script>

</body>

</html>

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>

2) By creating instance of an Array

------------------------------------------------------------

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var arr=new Array();

arr[0]=10;

arr[1]=20;

arr[3]=30;

for(var i in arr)

{

document.writeln(arr[i]);

}

</script>

</body>

</html>

3) By using Array constructor

------------------------------------------------

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var arr=new Array(10,20,30,40,50);

for(var i in arr)

{

document.writeln(arr[i]);

}

</script>

</body>

</html>

**Interview Question**

**===================**

**Q) Write a javascript program to insert and delete array elements?**

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var arr=[];

arr.push(10);

arr.push(20);

arr.push(30);

arr.pop();

for(var i in arr)

{

document.writeln(arr[i]);//10 20

}

</script>

</body>

</html>

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

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var arr=[10,20,30];

var sum=0;

for(var i in arr)

{

sum+=arr[i];

}

document.writeln(sum);

</script>

</body>

</html>

**JavaScript String**

**=================**

JavaScript String is an object which holds collection of characters.

There are two ways to create JavaScript String.

1) By using String literal

2) By creating instance of a String i.e using new operator.

1) By using String literal

--------------------------

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var str="ihubtalent";

document.writeln(str);

</script>

</body>

</html>

2) By creating instance of a String

---------------------------------------------------------

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var str=new String("qualitythought");

document.writeln(str);

</script>

</body>

</html>

ex:

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var arr=[10,20,30];

document.writeln(arr.length+"<br>");

var str="ihubtalent";

document.writeln(str.length+"<br>");

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var str='qualitythought';

document.writeln(str.toUpperCase());

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

var str='hello';

for(var i=str.length-1;i>=0;i--)

{

document.writeln(str.charAt(i));

}

</script>

</body>

</html>

**BOM (Browser Object Model)**

**==========================**

Browser Object Model is used to interact with browser.

A default object for every browser is window object. It means we can call all the properties directly or by using window.

ex:

alert("This is alert box");

or

window.alert("This is alert box");

**window object**

**=============**

A window object is created automically by the browser.

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

Javascript objects are String,Array,Date and etc.

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

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

1)It display dialog boxes and pop boxes.

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

3)We can move or resize the browser.

4)Scroll to the browser.

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

6)We can get javascript history.

1)alert()

========

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

ex:

<!DOCTYPE html>

<html>

<head>

<title>IHUB Talent</title>

</head>

<body>

<script type="text/javascript">

function f1()

{

alert("Welcome to JavaScript");

}

</script>

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

</body>

</html>

ex:2

------

<!DOCTYPE html>

<html>

<head>

<title>IHUB Talent</title>

</head>

<body>

<button onclick="alert('This is IHUB Talent')">click</button>

</body>

</html>

2)confirm()

----------------

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

ex:

<!DOCTYPE html>

<html>

<head>

<title>IHUB Talent</title>

</head>

<body>

<script type="text/javascript">

function f1()

{

var v=confirm("Do you wants to delete ?");

if(v==true)

{

alert("ok");

}

else

{

alert("cancel");

}

}

</script>

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

</body>

</html>

3)prompt()

---------------

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

ex:

<!DOCTYPE html>

<html>

<head>

<title>IHUB Talent</title>

</head>

<body>

<script type="text/javascript">

function f1()

{

var v=prompt("Who are you?");

alert("Welcome :"+v);

}

</script>

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

</body>

</html>

innerWidth and innerHeight

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

<!DOCTYPE html>

<html>

<head>

<title>IHUB Talent</title>

</head>

<body>

<script type="text/javascript">

var w=window.innerWidth;

var h=window.innerHeight;

document.writeln("Width :"+w+"<br>");

document.writeln("Height :"+h+"<br>");

</script>

</body>

</html>

Note:

----

Press "CTRL + +" for zoomin.

Press "CTRL + -" for zoomout.

window.open()

--------------

ex:1

-----

<!DOCTYPE html>

<html>

<head>

<title>IHUB Talent</title>

</head>

<body>

<script type="text/javascript">

function openWindow()

{

window.open("http://www.google.com");

}

</script>

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

</body>

</html>

ex:2

-----

<!DOCTYPE html>

<html>

<head>

<title>IHUB Talent</title>

</head>

<body>

<script type="text/javascript">

function openWindow()

{

window.open("http://www.google.com","\_blank");

}

</script>

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

</body>

</html>

ex:3

------

<!DOCTYPE html>

<html>

<head>

<title>IHUB Talent</title>

</head>

<body>

<script type="text/javascript">

function openWindow()

{

window.open("http://www.google.com","\_parent");

}

</script>

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

</body>

</html>

ex:4

-------

<!DOCTYPE html>

<html>

<head>

<title>IHUB Talent</title>

</head>

<body>

<script type="text/javascript">

function openWindow()

{

window.open("http://www.google.com","\_blank","width=200px,height=200px");

}

</script>

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

</body>

</html>

close()

-------------------

<!DOCTYPE html>

<html>

<head>

<title>IHUB Talent</title>

</head>

<body>

<script type="text/javascript">

var myWindow;

function openWindow()

{

myWindow=window.open("http://www.google.com","","width=300px,height=300px");

}

function closeWindow()

{

myWindow.close();

}

</script>

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

<button onclick="closeWindow()">close a window</button>

</body>

</html>

close()

-------------------

<!DOCTYPE html>

<html>

<head>

<title>IHUB Talent</title>

</head>

<body>

<script type="text/javascript">

var myWindow;

function openWindow()

{

myWindow=window.open("http://www.google.com","","width=300px,height=300px");

}

function closeWindow()

{

myWindow.close();

}

</script>

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

<button onclick="closeWindow()">close a window</button>

</body>

</html>

Whenever we open a new window , it takes left top alignment.

In order to move the window we need to use moveTo() or moveBy() function.

ex:

<!DOCTYPE html>

<html>

<head>

<title>IHUB Talent</title>

</head>

<body>

<script type="text/javascript">

var myWindow;

function openWindow()

{

myWindow=window.open("http://www.google.com","","width=300px,height=300px");

}

function moveWindow()

{

myWindow.moveTo(100,100);

}

</script>

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

<button onclick="moveWindow()">move window</button>

</body>

</html>

Note: Here we can't move window because in browser console we will get one

error.

To over come this limitation we need to use custom window.

ex:

<!DOCTYPE html>

<html>

<head>

<title>IHUB Talent</title>

</head>

<body>

<script type="text/javascript">

var myWindow;

function openWindow()

{

myWindow=window.open("","\_blank","width=300px,height=300px");

}

function moveWindow()

{

myWindow.moveTo(100,100);

}

</script>

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

<button onclick="moveWindow()">move window</button>

</body>

</html>

Note:

-----

MoveTo() function will move from absolute position.

MoveBy() function will move from relative position.

setTimeout()

-----------------

The setTimeout() is executed only once.

If you need repeated executions, use setInterval() instead.

ex:

<!DOCTYPE html>

<html>

<head>

<title>IHUB Talent</title>

</head>

<body>

<script type="text/javascript">

function setTimeOut()

{

setTimeout(function f1()

{

alert("Hello World")

},4000);

}

</script>

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

</body>

</html>

ex:

<!DOCTYPE html>

<html>

<head>

<title>IHUB Talent</title>

</head>

<body>

<script type="text/javascript">

function setTimeOut()

{

setTimeout(Anim,4000);

}

function Anim()

{

alert("Yahoo! this is javascript");

}

</script>

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

</body>

</html>

clearTimeout()

-----------------------

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

ex:

<!DOCTYPE html>

<html>

<head>

<title>IHUB Talent</title>

</head>

<body>

<script type="text/javascript">

var myId;

function setTimeOut()

{

myId=setTimeout(Anim,4000);

}

function Anim()

{

alert("Yahoo! this is javascript");

}

function removeTimeOut()

{

clearTimeout(myId);

}

</script>

<button onclick="setTimeOut()">set time</button>

<button onclick="removeTimeOut()">remove time</button>

</body>

</html>

setInterval()

==========

A setInterval() method calls a function to evaluate the expression at specified interval(milliseconds).

A setInterval() method calls continously function untill we call clearInterval() method or window is closed.

ex:

<!DOCTYPE html>

<html>

<head>

<title>IHUB Talent</title>

<style>

div

{

width:150px;

height: 150px;

background-color: #FF0000;

}

</style>

</head>

<body>

<script type="text/javascript">

var a=0;

setInterval(Anim,1000);

function Anim()

{

a = a + 10;

var target=document.getElementById("myId");

target.style.marginLeft= a + 'px';

}

</script>

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

</body>

</html>

clearInterval()

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

A clearInterval() function is used to clear the timer set on setInterval() function.

An id which is return from setInterval() function will use as parameter to clearInterval().

ex:

<!DOCTYPE html>

<html>

<head>

<title>IHUB Talent</title>

<style>

div

{

width:150px;

height: 150px;

background-color: #FF0000;

}

</style>

</head>

<body>

<script type="text/javascript">

var a=0;

var id=setInterval(Anim,1000);

function Anim()

{

a = a + 10;

if(a==100)

{

clearInterval(id);

}

var target=document.getElementById("myId");

target.style.marginLeft= a + 'px';

}

</script>

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

</body>

</html>

window history

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

<!DOCTYPE html>

<html>

<head>

<title>IHUB Talent</title>

<style>

a

{

text-decoration: none;

color:blue;

}

</style>

</head>

<body>

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

&laquo; previous

</a>

&nbsp; &nbsp;

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

next &raquo;

</a>

</body>

</html>

Note:

-------

www.ihubtalent.com

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

[www.qualitythought.in](http://www.qualitythought.in)

localStorage

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

A localStorage properties allows us to save key/value pairs in a browser window.

A localStorage allows us to store the data with no-expiry.It means our data will not be

delete even if we close the browser.It will be present for next day.

A localStorage is a read-only.

To add the data in a localStorage we need to use setItem(key,value) function.

To read the data from localStorage we need to use getItem(key) function.

To remove perticular data from localStorage we need to use removeItem(key) function.

To remove all the data from localStorage we need to use clear() function.

ex:

------

<!DOCTYPE html>

<html>

<head>

<!-- page title -->

<title>IHUB TALENT</title>

</head>

<body>

<script type="text/javascript">

//set the items

localStorage.setItem("FirstName","Alan");

localStorage.setItem("LastName","Morries");

//reading the items

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

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

//remove perticular item

localStorage.removeItem("LastName");

//remove all items

localStorage.clear();

//reading the items

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

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

</script>

</body>

</html>

ex:2

------

<!DOCTYPE html>

<html>

<head>

<!-- page title -->

<title>IHUB TALENT</title>

</head>

<body>

<script type="text/javascript">

//set the items

localStorage.setItem("FirstName","Alan");

localStorage.setItem("FirstName","Morries");

//reading the items

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

</script>

</body>

</html>

sessionStorage

-------------------

A sessionStorage properties allows us to save key/value pair in a browser window.

A sessionStorage store the data with respect to one session.It means our data will be deleted

once if we close the browser window.

To add the data in a sessionStorage we need to use setItem(key,value) function.

To read the data from sessionStorage we need to use getItem(key) function.

To remove perticular data from sessionStorage we need to use removeItem(key) function.

To remove all the data from sessionStorage we need to use clear() function.

ex:

---

<!DOCTYPE html>

<html>

<head>

<!-- page title -->

<title>IHUB TALENT</title>

</head>

<body>

<script type="text/javascript">

//set the items

sessionStorage.setItem("Name","Alan");

sessionStorage.setItem("Age",29);

//reading the items

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

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

//remove perticular item

sessionStorage.removeItem("Age");

//remove all items

sessionStorage.clear();

//reading the items

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

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

</script>

</body>

</html>

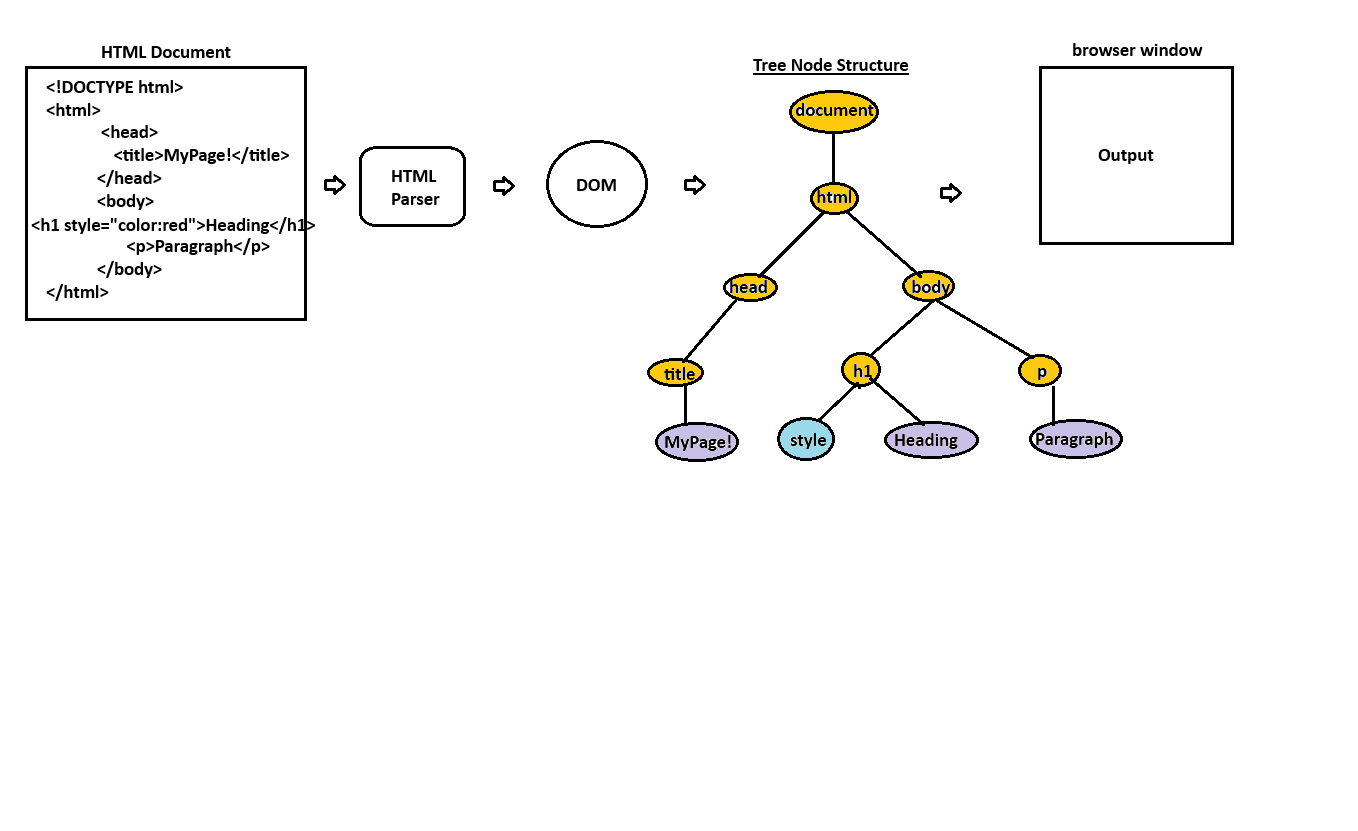
**DOM**

**=====**

DOM stands for Document Object Model.

Whenever HTML document loaded in a browser, it represent document object.

Diagram: frontend23.1



Internally, DOM is a tree node structure.

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

Using document object we can add dynamic content to the web page.

A document is a property of window. Means we can call document methods or by using window.

ex:

<script>

document.write("Welcome to JavaScript");

window.document.write("Welcome to JavaScript");

<script>

We have following list of functions in document object.

1) document.write()

2) document.writeln()

3) document.getElementById()

4) document.getElementsByName()

5) document.getElementsByTagName()

6) document.getElementsByClassName()

7) addEventListener()

8) removeEventListener()

1) document.write()

-------------------------------

It is used to display data or custom messages without space.

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

document.write("This is stmt1");

document.write("This is stmt2");

</script>

</body>

</html>

2) document.writeln()

-----------------------------------

It will display the output with space at last.

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

document.writeln("This is stmt1");

document.writeln("This is stmt2");

</script>

</body>

</html>

3) document.getElementById();

-----------------------------------------------------

It is used to read the elements based on id.

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

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

<br>

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

<script>

function f1()

{

var name=document.getElementById('t1').value;

document.writeln(name);

}

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

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

<br>

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

<script>

function f1()

{

var name=document.getElementById('t1').tagName;

document.writeln(name);

}

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

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

<br>

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

<br>

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

<script>

function f1()

{

var name=document.getElementById('t1').value;

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

}

</script>

</body>

</html>

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

<!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>&nbsp;</td>

<td><button onclick="f1()">ADD</button></td>

</tr>

<tr>

<td>Result:</td>

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

</tr>

</table>

<script type="text/javascript">

function f1()

{

var val1=document.getElementById("t1").value;

var val2=document.getElementById("t2").value;

var sum = parseFloat(val1)+parseFloat(val2);

document.getElementById("result").innerHTML=sum;

}

</script>

</body>

</html>

**JavaScript Hide and show portion of a form page**

**================================================**

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<fieldset id="curr\_id">

<legend>Current Address</legend>

<table align="center">

<tr>

<td>Locality:</td>

<td><input type="text" id="t1"/></td>

</tr>

<tr>

<td>City:</td>

<td><input type="text" id="t2"/></td>

</tr>

<tr>

<td>State:</td>

<td><input type="text" id="t3"/></td>

</tr>

</table>

</fieldset>

<br>

<input type="checkbox" id="box" onclick="f1()" />

Current Address is same as Permanent Address?

<br><br>

<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>City:</td>

<td><input type="text" id="t2"/></td>

</tr>

<tr>

<td>State:</td>

<td><input type="text" id="t3"/></td>

</tr>

</table>

</fieldset>

<script>

function f1()

{

if(document.getElementById('box').checked)

{

document.getElementById('per\_id').style.display="none";

}

else

{

document.getElementById('per\_id').style.display="block";

}

}

</script>

</body>

</html>

4) document.getElementsByName()

------------------------------------------------------------

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

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

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

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

<script>

function f1()

{

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

document.writeln(name);

}

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

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

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

<script>

function f1()

{

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

document.writeln(name);

}

</script>

</body>

</html>

ex;

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

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

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

<script>

function f1()

{

var name=document.getElementsByName('t1').length;

document.writeln(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="t2"/> <br>

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

<br>

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

<script type="text/javascript">

function f1()

{

var fname=document.getElementsByName('t1')[0].value;

var lname=document.getElementsByName('t2')[0].value;

document.getElementById('result').innerHTML=fname+lname;

}

</script>

</body>

</html>

**JavaScript program to select multiple checkbox**

**========================================**

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

Course Completed : <br>

<button onclick="f1()">select all </button>

<br>

<input type="checkbox" name="c1" value="html"/> HTML <br>

<input type="checkbox" name="c1" value="css"/> CSS <br>

<input type="checkbox" name="c1" value="js"/> JavaScript

<script>

function f1()

{

var x=document.getElementsByName('c1');

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

{

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

{

x[i].checked="true";

}

}

}

</script>

</body>

</html>

**document.getElementsByTagName()**

**================================**

The getElementsByTagName() method returns a collection of all elements with a specified tag name.

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

Name: <input type="text" />

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

<script>

function f1()

{

var val=document.getElementsByTagName('input').length;

document.writeln(val);

}

</script>

</body>

</html>

ex:

--

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

Name: <input type="text" />

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

<script>

function f1()

{

var val=document.getElementsByTagName('input')[0].value;

document.writeln(val);

}

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

Name: <input type="text" />

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

<script>

function f1()

{

var val=document.getElementsByTagName('input')[0].tagName;

document.writeln(val);

}

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<div>Division Tag 1</div>

<div>Division Tag 2</div>

<div>Division Tag 3</div>

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

<script>

function f1()

{

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

x[0].innerHTML="First Change";

x[1].innerHTML="Second Change";

x[2].innerHTML="Third Change";

}

</script>

</body>

</html>

ex;

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<div>Division Tag 1</div>

<div>Division Tag 2</div>

<div>Division Tag 3</div>

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

<script>

function f1()

{

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

x[0].innerHTML="First Change";

x[1].innerHTML="Second Change";

x[2].innerHTML="Third Change";

x[0].style.color="blue";

x[0].style.backgroundColor="cyan";

x[1].style.color="red";

x[2].style.color="green";

}

</script>

</body>

</html>

**document.getElementsByClassName()**

**================================**

The getElementsByClassName() method returns a collection of all elements with a specified class name.

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

<style>

.myStyle

{

color:violet;

text-align: center;

}

</style>

</head>

<body>

<div class="myStyle">Division Tag</div>

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

<script>

function f1()

{

var name=document.getElementsByClassName('myStyle')[0].tagName;

document.writeln(name);

}

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

<style>

.myStyle

{

color:violet;

text-align: center;

}

</style>

</head>

<body>

<div class="myStyle">Division Tag</div>

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

<script>

function f1()

{

var name=document.getElementsByClassName('myStyle').length;

document.writeln(name);

}

</script>

</body>

</html>

**document.addEventListener()**

**===========================**

It is used to attach the handler to the HTML elements.

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<h1> Please Click Anywhere</h1>

<script>

document.addEventListener("click",function(){

alert("You have clicked");

});

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<h1> Please Click Anywhere</h1>

<script>

document.addEventListener("click",f1);

function f1()

{

alert("You have clicked once again");

}

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

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

<br>

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

<script>

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

x.addEventListener('mouseover',f1);

function f1()

{

document.getElementById('result').innerHTML="Mouse Over";

}

</script>

</body>

</html>

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

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

<br>

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

<script>

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

x.addEventListener('mouseout',f1);

function f1()

{

document.getElementById('result').innerHTML="Mouse Out";

}

</script>

</body>

</html>

**JavaScript program to convert Feet to Inches**

**=======================================**

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<center>

<h3>Conversion Application</h3>

<table>

<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>

</center>

<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>

document.removeEventListener()

-------------------------------------------------------

It is used to remove the handler from the HTML elements.

ex:

---

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

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

<br>

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

<br>

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

<script>

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

x.addEventListener('mouseover',f1);

function f1()

{

document.getElementById('result').innerHTML+="<p>Mouse Over</p>";

}

function f2()

{

x.removeEventListener('mouseover',f1);

document.getElementById('result').innerHTML+="<p>Event Stop!!!</p>";

}

</script>

</body>

</html>

**JavaScript Date**

**===============**

JavaScript Date is used to display date and time.

Using JavaScript Date we can display timer on a web page.

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>

<!-- add external css -->

<link rel="stylesheet" type="text/css" href="css/mystyles.css">

</head>

<body>

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

<script>

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

function getTime()

{

var date=new Date();

var h = date.getHours();

var m = date.getMinutes();

var s = date.getSeconds();

m=check(m);

s=check(s);

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

}

function check(i)

{

if(i<10)

{

i = "0"+i;

}

return i;

}

setInterval(getTime,1000);

</script>

</body>

</html>

mystyles.css

--------------------

\*

{

margin:0;

padding:0;

}

body

{

height: 100vh;

background: linear-gradient(yellow,red);

display: flex;

justify-content: center;

align-items: center;

}

span

{

font-size:40px;

font-weight:bold;

color:#FFF;

font-style: italic;

}

**Q) What is the difference between innerText and innerHTML ?**

innerText

==========

A innerText property is used to display simple text in javascript dynamically.

ex:

--

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

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

<script>

document.getElementById('result').innerText="<p style='color:blue'> Inner Text Property</p>"

</script>

</body>

</html>

innerHTML

---------

A innerHTML property is used to display HTML code in javascript dynamically.

ex:

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

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

<script>

document.getElementById('result').innerHTML="<p style='color:blue'> Inner Text Property</p>"

</script>

</body>

</html>

**Hide and show password in a form page**

**======================================**

<!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>

#icon

{

position:relative;

right:25px;

}

.mystyle

{

padding-right:25px;

}

</style>

</head>

<body>

Password: <input type="password" id="t1" class="mystyle"/>

<span class="fa fa-eye" id="icon" onclick="f1()"></span>

<script>

function f1()

{

var x = document.getElementById('t1');

if(x.type=="password")

{

x.type="text";

}

else

{

x.type="password";

}

}

</script>

</body>

</html>

**Q) Explain High order functions in JavaScript?**

Functions that operate on other functions, either by taking them as arguments or by returning them, are called higher-order functions.

Ex:

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

function f1(f2)

{

f2();

}

f1(function(){document.writeln("F2-Function")});

</script>

</body>

</html>

Ex:

<!DOCTYPE html>

<html>

<head>

<title>MyPage!</title>

</head>

<body>

<script>

function f1()

{

return function(){

return "Another Function";

}

}

var f2=f1();

document.writeln(f2());

</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 to below url.

ex:

https://regex101.com/

ex:

<!DOCTYPE html>

<html>

<style>

input[type=text],input[type=password], select {

width: 100%;

padding: 12px 20px;

margin: 8px 0;

display: inline-block;

border: 1px solid #ccc;

border-radius: 4px;

box-sizing: border-box;

}

input[type=submit] {

width: 100%;

background-color: #4CAF50;

color: white;

padding: 14px 20px;

margin: 8px 0;

border: none;

border-radius: 4px;

cursor: pointer;

}

input[type=submit]:hover {

background-color: #45a049;

}

div {

border-radius: 5px;

background-color: #f2f2f2;

padding: 20px;

width:500px;

position: relative;

left:200px;

top:20px;

}

</style>

<script type="text/javascript">

function validate()

{

var name=document.getElementById('name').value;

var pwd=document.getElementById('pwd').value;

var phone=document.getElementById('phone').value;

var email=document.getElementById('email').value;

var country=document.getElementById('country').value;

var namecheck=/[A-Za-z. ]{6,20}$/;

var pwdcheck=/(?=.\*[0-9])(?=.\*[!@#$%^&\*])(?=.\*[A-Z])[a-zA-Z0-9!@#$%^&\*]{10,30}$/;

var phonecheck=/[789][0-9]{9}$/;

var emailcheck=/[A-Za-z.]{1,}@[A-Za-z]{2,15}[.][A-Za-z]{3,}$/;

if(!(namecheck.test(name)))

{

alert("UserName must be 6 characters");

document.getElementById('name').value="";

document.getElementById('name').focus();

return false;

}

if(!(pwdcheck.test(pwd)))

{

alert("password must have 1 uppercase, 1 special symbol and 1 digit");

document.getElementById('pwd').value="";

document.getElementById('pwd').focus();

return false;

}

if(!(phonecheck.test(phone)))

{

alert("Phone must start with 7,8,9 series with 10 digits");

document.getElementById('phone').value="";

document.getElementById('phone').focus();

return false;

}

if(!(emailcheck.test(email)))

{

alert("Please insert valid email");

document.getElementById('email').value="";

document.getElementById('email').focus();

return false;

}

if(country=="")

{

alert("Please select the country option ");

return false;

}

return true;

}

</script>

<body>

<div>

<form action="/action\_page.php" onsubmit="validate()">

<label for="name">UserName</label>

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

<label for="pwd">Password</label>

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

<label for="phone">Phone</label>

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

<label for="email">Email</label>

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

<label for="country">Country</label>

<select id="country" name="country">

<option value="">none</option>

<option value="australia">Australia</option>

<option value="canada">Canada</option>

<option value="usa">USA</option>

</select>

<input type="submit" value="Submit">

</form>

</div>

</body>

</html>

**Synchronous and Asynchronous in JavaScript**

**=======================================**

Synchronous JavaScript:

---------------------------------------

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

earlier statement to get executed.

ex:

<script>

document.write("Hi"); // First

document.write("<br>");

document.write("IHUB TALENT") ;// Second

document.write("<br>");

document.write("How are you"); // Third

</script>

Asynchronous JavaScript:

------------------------------------------

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

ex:

<script>

document.write("Hi");

document.write("<br>");

setTimeout(function() {

document.write("Let us see what happens");

}, 2000);

document.write("<br>");

document.write("End");

document.write("<br>");

</script>

ex:

<script>

document.write("Hi");

document.write("<br>");

setTimeout(() => {

document.write("Let us see what happens");

}, 2000);

document.write("<br>");

document.write("End");

document.write("<br>");

</script>

**Javascript promises**

**=================**

Promises are used to handle asynchronous operations in JavaScript.

They can handle multiple asynchronous operations easily and provide better

error handling than callbacks and events.

A Promise has four states:

1)fulfilled: Action related to the promise succeeded

2)rejected: Action related to the promise failed

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

4)settled: Promise has fulfilled or rejected

A promise can be created using Promise constructor.

Syntax:

var promise = new Promise(function(resolve, reject){

//do something

});

ex:1

--------

<script>

var promise = new Promise(function(resolve, reject) {

resolve('IHub Talent');

})

promise

.then(function(successMessage) {

//success handler function is invoked

console.log(successMessage);

}, function(errorMessage) {

console.log(errorMessage);

})

</script>

ex:2

--------

<script>

var promise = new Promise(function(resolve, reject) {

reject('Error occured');

})

promise

.then(function(successMessage) {

//success handler function is invoked

console.log(successMessage);

}, function(errorMessage) {

console.log(errorMessage);

})

</script>

ex:3

----

<script>

var promise = new Promise(function(resolve, reject) {

const x = "ihubtalent";

const y = "ihubtalent1";

if(x === y) {

resolve();

} else {

reject();

}

});

promise.

then(function () {

console.log('Success, You are a GEEK');

}).

catch(function () {

console.log('Some error has occurred');

});

</script>

**Q)Differences between var , let and const?**

var let const

----- ----- -------

It is a functional scope. It is a block scope. It is a block scope.

It can be declare without It can be declare It can't be declare

initialization. without initialization. without initialization.

It can be updated. It can be updated. It can't be updated.

It can be redeclared. It can't be redeclared. It can't be redeclared.

It can be access without It can be access without It can’t be access without initialization as it initialization as it default initialization.

defaultvalue is value is undefined.

undefined.

Intialization

---------------

ex:1

-----

<!DOCTYPE html>

<html>

<head>

<title>mypage!</title>

</head>

<body>

<script type="text/javascript">

var i;

document.writeln(i);//undefined

</script>

</body>

</html>

ex:2

-----

<!DOCTYPE html>

<html>

<head>

<title>mypage!</title>

</head>

<body>

<script type="text/javascript">

let i;

document.writeln(i);//undefined

</script>

</body>

</html>

ex:3

----

<!DOCTYPE html>

<html>

<head>

<title>mypage!</title>

</head>

<body>

<script type="text/javascript">

const i;

document.writeln(i);//invalid

</script>

</body>

</html>

Update

----------

ex:1

-----

<!DOCTYPE html>

<html>

<head>

<title>mypage!</title>

</head>

<body>

<script type="text/javascript">

var i=10;

i=20;

document.writeln(i);//20

</script>

</body>

</html>

ex:2

-----

<!DOCTYPE html>

<html>

<head>

<title>mypage!</title>

</head>

<body>

<script type="text/javascript">

let i=10;

i=20;

document.writeln(i);//20

</script>

</body>

</html>

ex:3

-----

<!DOCTYPE html>

<html>

<head>

<title>mypage!</title>

</head>

<body>

<script type="text/javascript">

const i=10;

i=20;

document.writeln(i);//invalid

</script>

</body>

</html>

Redeclared

--------------

ex:1

-----

<!DOCTYPE html>

<html>

<head>

<title>mypage!</title>

</head>

<body>

<script type="text/javascript">

var i=10;

var i=20;

document.writeln(i);//20

</script>

</body>

</html>

ex:2

------

<!DOCTYPE html>

<html>

<head>

<title>mypage!</title>

</head>

<body>

<script type="text/javascript">

let i=10;

let i=20;

document.writeln(i);//

</script>

</body>

</html>

ex:3

----

<!DOCTYPE html>

<html>

<head>

<title>mypage!</title>

</head>

<body>

<script type="text/javascript">

const i=10;

const i=20;

document.writeln(i);//

</script>

</body>

</html>

How to store form data in a localStorage

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

<!DOCTYPE html>

<html>

<head>

<title>mypage!</title>

</head>

<body>

<form>

<table align="center">

<tr>

<td>No:</td>

<td><input type="text" id="t1"/></td>

</tr>

<tr>

<td>Name:</td>

<td><input type="text" id="t2"/></td>

</tr>

<tr>

<td>Address:</td>

<td><input type="text" id="t3"/></td>

</tr>

<tr>

<td><input type="reset" value="reset"/></td>

<td><input type="submit" value="submit" onclick="f1()"/></td>

</tr>

</table>

</form>

<script type="text/javascript">

function f1()

{

//reading form data

var no=document.getElementById('t1').value;

var name=document.getElementById('t2').value;

var add=document.getElementById('t3').value;

//store the items to localStorage

localStorage.setItem("studNo",no);

localStorage.setItem("studName",name);

localStorage.setItem("studAdd",add);

}

</script>

</body>

</html>

**Javascript Set**

**============**

A JavaScript Set is a collection of unique values.

Each value can only occur once in a Set.

A Set can hold any value of any data type.

ex:1

-----

<!DOCTYPE html>

<html>

<head>

<title>IHUB TALENT</title>

</head>

<body>

<script type="text/javascript">

var letters=new Set();

letters.add(6);

letters.add(1);

letters.add(1);

letters.add(5);

letters.add(9);

letters.forEach(function(value){

document.writeln(value);

})

</script>

</body>

</html>

ex:2

-----

<!DOCTYPE html>

<html>

<head>

<title>IHUB TALENT</title>

</head>

<body>

<script type="text/javascript">

var letters=new Set([6,1,5,9,1,6]);

letters.forEach(function(value){

document.writeln(value);

})

</script>

</body>

</html>

ex:3

-----

<!DOCTYPE html>

<html>

<head>

<title>IHUB TALENT</title>

</head>

<body>

<script type="text/javascript">

var letters=new Set([6,1,5,9]);

// Create an Iterator

const myIterator = letters.values();

// List all Values

for (const entry of myIterator)

{

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

}

</script>

</body>

</html>

ex:4

------

<!DOCTYPE html>

<html>

<head>

<title>IHUB TALENT</title>

</head>

<body>

<script type="text/javascript">

var letters=new Set(["a","b","c","d"]);

// Create an Iterator

const myIterator = letters.values();

// List all Values

for (const entry of myIterator)

{

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

}

</script>

</body>

</html>

**JavaScript Maps**

**==============**

A Map holds key-value pairs where the keys can be any datatype.

A Map remembers the original insertion order of the keys.

ex:1

-------

<!DOCTYPE html>

<html>

<head>

<title>IHUB TALENT</title>

</head>

<body>

<script type="text/javascript">

// Create a Map

const myMap = new Map([

["one", 1],

["two", 2],

["three", 3]

]);

document.writeln(myMap.size); //3

document.writeln(myMap.get("one"));//1

myMap.delete("three");

myMap.clear();

</script>

</body>

</html>

ex:2

-------

<!DOCTYPE html>

<html>

<head>

<title>IHUB TALENT</title>

</head>

<body>

<script type="text/javascript">

// Create a Map

const myMap = new Map([

["one", 1],

["two", 2],

["three", 3]

]);

myMap.forEach(function(value, key)

{

document.writeln(value+" "+key+"<br>");

})

</script>

</body>

</html>

ex:3

-------

<!DOCTYPE html>

<html>

<head>

<title>IHUB TALENT</title>

</head>

<body>

<script type="text/javascript">

// Create a Map

const myMap = new Map([

["one", 1],

["two", 2],

["three", 3]

]);

for (const x of myMap.keys()) {

document.writeln(x);

}

</script>

</body>

</html>

ex:4

-------

<!DOCTYPE html>

<html>

<head>

<title>IHUB TALENT</title>

</head>

<body>

<script type="text/javascript">

// Create a Map

const myMap = new Map([

["one", 1],

["two", 2],

["three", 3]

]);

for (const x of myMap.values()) {

document.writeln(x);

}

</script>

</body>

</html>

**Q)What is JavaScript Math object?**

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

ex:

<!DOCTYPE html>

<html>

<head>

<title>IHUB TALENT</title>

</head>

<body>

<script type="text/javascript">

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

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

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

document.writeln(Math.trunc(10.56));

</script>

</body>

</html>

Object Oriented Programming System / Structure (OOPS)

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

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

ex:

class

object

abstraction

encapsulation

inheritance

and

polymorphism

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 state and behavior (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

**Q) What is spread operator in JavaScript?**

The spread operator is used to spreading an array.

Ex:

<script>

function addFourNumbers(num1,num2,num3,num4)

{

document.writeln (num1+" "+num2+" "+num3+" "+num4);

}

let fourNumbers = [5, 6, 7, 8];

addFourNumbers(...fourNumbers);

</script