Q.) What is JavaScript ?
 -> JavaScript is an interpreted, client-side, event-based, object-oriented scripting language.
 -> JavaScript Developed for 1995 at Netscape Corporation (LiveScript).
 -> JavaScript Programs are run by an interpreter built into the user's web browser
 -> It is a case-sensitive language
 -> JavaScript is not Java

Q.) What can do JavaScript
    1. JavaScript can dynamically modify an HTML page
    2. JavaScript can validate user input JavaScript can validate user input
    3. JavaScript can validate user input
    4. JavaScript can be used to crate cookies
    5. JavaScript is a full-featured programming language
    6. JavaScript user interaction does not require any communication with the server

Q.) JavaScript Framework and Libraries
    1. Angular
    2. React
    3. Vue.js
    4. Node.js

Q.) Types of JavaScript
    1. Internel JavaScript
        A. JavaScript can be inserted into documents by using the SCRIPT tag.
        B. The SCRIPT tag provides a block to write the JavaScript programs.
        Examples:-
            <script type="text/javascript">
                // JS code goes here
            </script>

    2. External JavaScript
        A. To use the pre-defined programs of any JavaScript Libraries
        Examples:-
            <script src="myscript.js"></script>

Q.) Important Function of JavaScript
    1. alert()
        It is used to alert the user that something has happened.
        Examples :-
            <script type="text/javascript">
                alert("Hello Students...");
            </script>

    2. confirm()
        Opens up a Confirm/Cancel dialog and returns true/false depending on user's click.
        Examples :-
            <script type="text/javascript">
                confirm("Login to WebSite..");
            </script>

    3. console.log()
        Writes information to the browser console, good for debugging purpose
        Examples :-
            <script type="text/javascript">
                console.log("JavaScript Tutorials..");
            </script>

```
    4. document.write()
        Write directly to the HTML document
        Examples :-
            <script type="text/javascript">
                document.write("Simple Page..");
            </script>

    5. prompt(msg.default)
        Creates an dialogue for user input
        Examples :-
            <script type="text/javascript">
                prompt("Hello","Students");
            </script>

Q.) What is variable?
 -> Variable are containers which hold reusable data.
 -> It is the basic unit of storage in a program.
 -> The value stored in a variable can be changed during program execution.

Q.) Scope and Life time of a variable
    1. Scope of a Variable :-
        -> variables declared within a function are local to that function.
        -> Variables declared outside of any function are global variable.

    2. Life Time of Variable :-
        -> Local variable's life time is within the block of its declaration.
        -> Global variable's life time is throughout the program.

Q.) Javascript DOM Function
    1. document.getElementById()
    2. document.getElementByClassName()
    3. innerHTML()

Q.) What is an Operator And Types of Operators in JS
    1) What is an Operator?
        -> An operator is a symbol that tell the compiler which arithmetic or logical operation
to be performed between the respective operands.

    2) Types of Operators
        1. Arithmetic Operator
            a. (+) Addition         ->   Adds Two Numbers
            b. (-) Subtraction      ->   Subtracts the second operand from the first
            c. (*) Multiplication   ->   Multiply both operands
            d. (/) Division         ->   Divide the numerator by the denominator
            e. (%) Modules          ->   Output the remainder of an integer division
            f. (++) Increment       ->   Increases an integer value by one
            g. (--) Decrement       ->   Decreases an integer value by one

        2. Logical Operator
            a. (&&) Logical And
            b. (||) Logical OR

        3. Ternary Operator
            <condition> ? <true> : <false> -> Ex:- 2>1 ? "Yes" : "No"
                                -> if 2 is greater than 1, yes will be printed else no will
be printed
```

4. Assignment Operator
    a. (=) Simple Assignment
    b. (+=) Add and Assignment
    c. (-=) Subtract and Assignment
    d. (*=) Multiply and Assignment
    e. (/=) Division and Assignment
    f. (%=) Modules and Assignment

5. Comparison Operator
    a. (==) Equal
    b. (!=) Not Equal
    c. (>) Greater than
    d. (<) Less than
    e. (>=) Greater than or Equal to
    f. (<=) Less than or Equal to

Q.) Function in JavaScript
-> Function are group of code or program which is used more often.
-> It leads programming to code reusability and clean code.

=> Types of Function
1. pre defined -> Such function are defined at the time of making of any language.
2. user defined -> Such function defined by users according to their needs.

-> Name: Just like a variable where we call "var" and give that a name, to create a function, we call "function" and give it a name in the same way

Syntax :- function <functionName>(){
    // Function Body
}

-> Arguments: These are the inputs that the function will manipulate-separate multiple inputs with commas.

Syntax :- function <functionName>(a,b){
    // Function Body
}

Q.) Coditional Statements for JavaScript
-> conditional Statements (control statements) used to change the flow of the program's execution.
-> Two Possible cases, either true or false.

=> Types of conditional Statements in javascript
1. If
    - If provide true, performs a function or displays informarion.
    -> Syntax:-
        if (condition){
            // block of code to be executed
            // if the condition is true
        }

2. If-else
    - If provide true, performs a task.
    - If false performs any one task.
    -> Syntax:-

```
            if (condition){
                // block of code to be executed
                // if the condition is true
            }
            else{
                // block of code to be executed
                // if the condition is false
            }

    3. If-elseif
        - If proved true, performs a task.
        - If false checks another if condition.
        -> Syntax:-
            if (condition1){
                // block of code to be executed
                // if the condition is true
            }
            else if(condition2){
                // block of code to be executed if
                // the condition is false and condition2 is true
            }
            else{
                // block of code to be executed if
                // the condition is false and condition2 is false
            }

Q.) Array in JavaScript
 -> Array is a group of continuous memory location.
 -> It is used to store multiple value into a Single variable.

 => Declaration of Array in JS
 Method-1)
    var house=[];

    // initializing while declaring
    var house=["1BHK","2BHK","3BHK","4BHK"];
    // initializing after declaring
    house[0]="1BHK";
    house[1]="2BHK";
    house[2]="3BHK";
    house[3]="4BHK";

 Method-2)
    var house=new array()

    // Create an array having elements - 10,20,30,40,50
    var house=new Array(10,20,30,40,50);
    // Storing Number, string in array
    var house=["1BHK",32500,"2BHK",45000,"Sales"];

Q.) What is Loop ?
 -> A loop is used to repeat a block of code until the specifed condition is Method
 -> When similar task is needed to be done again and again.
 -> Save time and leads to code reusability.
    1. ForLoop
        Syntax:-
```

```
            for(initialization;test condition;+ement/-ement)
            {
                // Statement(s) to be executed if test condition is true
            }

        Examples:-
            for (var count=0;count<10;count++)
            {
                document.write(count+"Welcome JavaScript <br>");
            }
```

2. While Loop

-> A While loop is to execute a statement or code block repeatedly as long as Condition is true.

-> Once the Condition becomes false, the loop terminates.

-> It is also called entry control loop.

=> Syntax:-

```
            while (expression)
            {
                // Statemet(s) to be executed if expression is true
            }
```

=> Examples:-

```
            var count=0;
            while(count>10)
            {
                document.write(count+"<br>");
                count++;
            }
```

3. ForEach loop

-> A ForEach loop (method) is used to get data from JS array or object.

-> The method calls a function once for each element in an array, in order.

=> Syntax And Examples

```
            var cars=["Car1","Car2","Car3"];        -> item - (here "Car1") required parameter
            cars.ForEach(myfunction);               -> index - (here 0 for "Car1") optional
                                                    -> array - (here cars) optional
            function myfunction(item,index,array){
                // body of the function
            }
```

Q.) Continue Statement ?

-> The continue statement is used to skip the current iteration and go to next iteration.

-> Syntax :-

```
            continue;    --> It is also a keyword.
```

Q.) Break Statement ?

-> The Break statement is used to break the execution of current loop or switch case.

-> As soon as it is encountered the flow of the program goes to next statement of loop or switch case.

=> Syntax :-

```
        break;      --> It is also a keyword.

Q.) What is Switch Statement ?
 -> Switch statement is used to choose in between many cases.
 -> It works same as nested if-elseif-else statement.

=> Syntax:-

    switch(x)
    {
        case 'value1':
            statement1;
            [break]
        case 'value2':
            statement2;
            [break]
        case 'value3':
            statement3;
            [break]
        default:
            statementDefault;
            [break]
    }

=> Examples :-
    <script type="text/javascript">
        let i=1;
        switch(i)
        {
            case 0:
                document.write("i is equal to Zero.");
                break;
            case 1:
                document.write("i is equal to One.");
                break;
            case 2:
                document.write("i is equal to Two.");
                break;
            default:
                document.write("i is Greater Than Two.");
        }
        document.write("out Of Switch.");
    </script>

Q.) List of Methods
    1. valueOf()    ->  To print the data for array
    2. isArray()    ->  Check if the variable is an array or not ?
    3. concat()     ->  Join several array into one
    4. indexOf()    ->  Returns the primitive value of the specified object
    5. join()       ->  Combine elements of an array into a single string and return the string

    6. pop()        ->  Removes the last element of an array
    7. push()       ->  Add a new element at the end
    8. reverse()    ->  Sort elements in descending order
    9. shift()      ->  Remove the first element of an array
   10. slice()      ->  Pulls a copy of a portion of an array into a new array
```

Q.) Date Methods in JavaScript

1. Date() -> To get the current date and time details according to the brower.
2. getFullYear() -> Get the YEAR as a four digit number (yyyy).
3. getMonth() -> Get the MONTH as a number (0-11).
4. getDate() -> Get the DAY as a number (0-31).
5. getHours() -> Get the HOUR (0-23).
6. getMinutes() -> Get the MINUTE (0-59).
7. getSeconds() -> Get the SECOND (0-59).
8. getMilliseconds()-> Get the MILLISECONDS (0-999).
9. getTime() -> Get the time (milliseconds since January 1,1970).
10. getDay() -> Get the Weekday as a number (0-6).

Q.) Events in JavaScript ?

-> In Programming, event is an occurrence of any action.
-> When we use javascript in HTML pages, JavaScript can "react" with events.
-> Eg. When user click the mouse button, or press any key on the keyboard.

Q.) List of Mouse event in JS

1. onclick() -> The Event occurs when the user clicks on an element
2. oncontentmenu() -> User right-clicks on an element to open a context menu
3. ondblclick() -> The user double-clicks on an element
4. onmousedown() -> User presses a mouse button over an element
5. onmouseenter() -> The pointer moves onto an element
6. onmouseleave() -> Pointer moves out of an element
7. onmousemove() -> The pointer is moving while it is over an element
8. onmouseover() -> When the pointer is moved onto an element or one of its children
9. onmouseout() -> User moves the mouse pointer out of an element or one of its children
10. onmouseup() -> The user releases a mouse button while over an element

Q.) List of keyboard event in JS

1. onkeydown() -> When the user is pressing a key down
2. onkeypress() -> The moment the user starts pressing a key
3. onkeyup() -> The user releases a key

Q.) List of Form Event in JS

1. onblur -> When an element loses focus
2. onchange -> The content of a form element changes
   (for <input>, <select>and <textarea>)
3. onfocus -> An element gets focus
4. oninput -> User input on an element
5. oninvalid -> An element is invalid
6. onreset -> A form is reset
7. onsearch -> The user writes something in a search
   (for <input="search">)
8. onselect -> The user selects some text
   (for <input> and <textarea>)
9. onsubmit -> A form is submitted
10. onload -> When document is fully loded

Q.) DOM Two Methods ?

1. getElementsByName()
2. getElementsByTagName()

Q.) setInterval() in JavaScript ?

-> The setInterval() method reports a given function at every given time-interval.
=> Syntax:- windows.setInterval(function,milliseconds);
-> funcion:- first parameter is the function to be executed
-> milliseconds:- indicates the length of the time-interval between each execution

Q.) setTimeout() in JavaScript ?
-> The setTimeout() method executes a funcion, after waiting a specifed number of milliseconds.
=> Syntax:- windows.setTimeout(funcion,milliseconds);
-> funcion:- first parameter is the funcion to be executed
-> milliseconds:- indicates the number of milliseconds before execution takes place. 1 sec=1000ms.

Q.) clearTimeout method
-> The clearTimeout() method clears a timer set with the setTimeout() method.
-> The ID value returned by setTimeout() is used as the parameter for the clearTimeout() method.

=> myVar = setTimeout("function",milliseconds);
=> clearTimeout(myVar);

Q.) clearInterval method
-> The clearInterval() method clears a timer set with the setInterval() method.
-> The ID value returned by setInterval() is used as the parameter for the clearInterval() method.

=> myVar=setInterval("funcion",milliseconds);
=> clearInterval(myVar);

Q.) What is History Object ?
-> The History object contains the URLs visited by the user (Within a browser window).
-> The history object is part of the window object and is accessed through the window.history property

=> All the major brower support it.

Q.) History object methods

Methods	Description	Syntax
1. back()	Loads the previous URL in the history list	window.history.back()
2. forward()	Loads the next URL in the history list	window.history.forward()
3. go()	Loads a specific URL from the history list	window.history.go()

Q.) Window Screen object ?
-> The Screen object contains information about the visitor's screen.

Property	Description	Syntax
1. availHeight	Returns the height of the screen (execuding the Window Taskbar)	screen.availHeight
2. availWidth	Returns the width of the screen (execuding the Window Taskbar)	screen.availWidth
3. colorDepth	Returns the bit depth of the color palette for displaying images	screen.colorDepth
4. height	Returns the total height of the screen	screen.height
5. pixelDepth	Returns the color resolution(in bits per pixel) of the screen	screen.pixelDepth

6. width Returns the total width of the screen screen.width

Q.) What is Math object ?
-> The math object contains some predefined mathematical methods
-> These methods are used to perform mathematical calculation on numerical values
-> To access these methods (Math.) keyword is used

=> Methods for Math objects

Methods	Description
1. Math.PI	Return the value of PI (3.141592653589793)
2. Math.round(x)	Return the rounded value of x to its nearest integer
3. Math.pow(x,y)	Return the value of x to the power y
4. Math.sqrt(x)	Return the square root of x
5. Math.abs(-x)	Return the positive value of x
6. Math.ceil(x)	Returns the value of x rounded -up- to its nearest integer
7. Math.floor(x)	Returns the value of x rounded -down- to its nearest integer
8. Math.mix(x,y,z)	Return the minimum value among x,y and z
9. Math.max(x,y,z)	Return the maximum value among x,y and z
10. Math.random	Return random decimal value in between 0 and 1

Q.) What is this keyword ?
-> this keyword in javaScript is used to refer the object it belongs to.
-> In a method, this refers to the owner object.
-> Alone, this refers to the global object.
-> In a function, this refers to the global object.
-> In a function, in strict mode, this is undefined.
-> In an event, this refers to the element that received the event.

Q.) Event Listenet in JavaScript
-> Event Listener is used to perform any task on an event of any element
-> Using event listener one can add and remove task on the occurrence of any event for an element.
-> It is a part of JS DOM

=> Types of EventListener
1. addEventListener method in JavaScript
-> The addEventListener() method attaches an event handler to the specified element.
-> The addEventListener() method attaches an event handler to an element without overWriting existing event handler.
-> One can add many existing event handlers.
-> SYNTAX:- element.addEventListener(event,funcion);

Q.) What is an Exception ?
-> An Exception signifies the presence of an abnormal condition which requires special operable techniques.
-> In programming terms, an exception is the anomalous code that breaks the normal flow of the code.
-> Such exception require specialized programming constructs for its execution.

Q.) what is exception handling ?
-> Exception handling is a process or method used for handling the abnormal statements in the code and executing them.

=> Try , Catch , throw AND finally are used.

Q.) JavaScript try...Catch Statement
=> try{} statement :- Here, the code which needs possible error testing is kept within the try block. In case any occur, it passes to the catch{} block for taking suitable actions and handle

the error. Otherwise, it executes the code written within.

=> catch{} statement :- This block handles the error of the code by executing the set of statements written within the block. This contains either the user-defined execution handler or the built-in Otherwise, the catch block is skipped.

=> Syntax :-

```
try{
    expression;
}
catch(error){
    expression;
}
```

Q.) JavaScript try...Catch...throw statement

-> Try..Catch block can only handle the predefined error. Syntax error and any other run time error.

-> In order to handle user defined error throw is used. Which is triggered on the basic of any condition.

=> Syntax :-

```
try{
    if(<condition>){
        throw;
    }
}
catch(<error_variable>){
    // use err here
}
```

Q.) JavaScript try...Catch...finally Statement

-> Try catch block can only handle the predefined error, Syntax error and any other run time error.

-> The finally statement lets you execute code, after try and catch, regardless of the result.

=> Syntax :-

```
try{
    //  Block of code to try
}
catch(err){
    //  Block of code to handle error
}
finally{
    //  Block of code to be executed regardless of the try / catch result
}
```