MODULE: 4 (JavaScript Basic & DOM):-

1. What is JavaScript ?

Ans:- **JavaScript** is the Programming for the **Web**.
JavaScript can update and change both **HTML** and **CSS**.
JavaScript can **Calculate, manipulate** and **validate** data.

2. What is use of NaN Function?

Ans:- In **JavaScript , NaN** is short for **" Not-a-Number"**
In **JavaScript , NaN** is a number that is not legal Number.
The **Number.isNaN()** method returns true if the value is NaN and the type is Number.

3. What is Negative Infinity?

Ans:- **NEGATIVE_INFINITY** is a special numeric value that is returned when an arithmetic operation or mathematical function generates a negative value greater than the largest representable number in JavaScript (i.e., more negative than -Number. MAX_VALUE).

4. Which Company developed JavaScript?

Ans:- In **September 1995**, a **Netscape** programmer named **Brandan Eich** developed a new scripting language in just **10 days**. It was originally named **Mocha**, but quickly became known as **LiveScript** and, later, **JavaScript.**

5. What are undeclare and undefined variables ?

Ans:- **Undefined:** It occurs when a variable has been declared but has not been assigned with any value. Undefined is not a keyword.

Undeclared: It occurs when we try to access any variable that is not initialized or declared earlier using var or const keyword. If we use **'type of'** operator to get the value of an undeclared variable, we will face the runtime error with return value as "undefined". The scope of the undeclared variables is always global.

6. Write the code for adding new elements dynamically ?

Ans:-

7. What is difference between View State and Session State ?

Ans:-

View State	**Session State**
❖ Maintained at page level only .	❖ Maintained at session level
❖ View state can only be visible from a single page and not multiple page	❖ Session state value availability is across all pages available in a user session .
❖ It will retain values in the event of a postback operation occurring	❖ In session state ,user data remains in the server. Data is available to user until the browser is closed or there is session expiration.
❖ Information is stored on the client end only	❖ Information stored on the server.
❖ Used to allow the persistence of page instance specific data	❖ Used for the persistence of user specific data on the server end .
❖ View state values are lost/cleared when new page is loaded	❖ Session state can be cleared by programmer or user or in case of timeouts.

8. What is === Operator ?

Ans:- When we compare two variables of different type e.g. a Boolean with a string or a number with string using == operator, it automatically converts one type into another and return value based upon content equality.

While === operator is strict equality and only return true if both variable of same type and also contains same value.

```
console.log(1 === 1);

// expected output: true.


console.log('hello' === 'hello');

// expected output: true.


console.log('1' ===  1);

// expected output: false.


console.log(0 === false);

// expected output: false.
```

9. How can the Style/Class can be changed ?

Ans:-
```
<!DOCTYPE html>

<html>
    <head>
        <title>Change an element class with javascript</title>
        <style type="text/css">
            .default{
                background-color: red;
            }
            .changedClass{
                background-color: green;
            }
            #myPara{
                margin-top: 20px;
            }
            #myButton{
                padding: 10px;
            }
            body {
                text-align:center;
            }
            h1 {
                color:green;
            }
        </style>
        <script type="text/javascript">
            function changeClass() {
                document.getElementById('myButton').className =
"changedClass";
                var button_class =
document.getElementById('myButton').className;

            }
        </script>
    </head>
    <body>

        <button class="default" onclick="changeClass()"
                    id="myButton">Click Here!</button><br>

    </body>
</html>
```

10. How to read and write a file using JavaScript?

Ans:- **fs.readFile ()** and **rs.writeFile ()** methods are used to read and write of a file using javascript. The file is read using the fs.readFile() function, which is an inbuilt method.

Syntax :

fs.readFile(file_name, encoding, callback_function)

Parameters :

filename: It contains the filename to be read, or the whole path if the file is saved elsewhere.
encoding: It stores the file's encoding. 'utf8' is the default setting.
callback function: This is a function that is invoked after the file has been read. It requires two inputs:
err: If there was an error.
data: The file's content.
Return Value: It returns the contents contained in the file, as well as any errors that may have occurred.

fs.writeFile() function is used to write data to a file in an asynchronous manner. If the file already exists, it will be replaced.

Syntax:

fs.writeFile (file_name, data, options, callback)

Parameters

file_name: It's a string, a buffer, a URL, or a file description integer that specifies the location of the file to be written. When you use a file descriptor, it will function similarly to the fs. write() method.
data: The data that will be sent to the file is a string, Buffer, Typed Array, or **DataView.**
options: It's a string or object that may be used to indicate optional output options. It includes three more parameters that may be selected.
encoding: It's a string value that indicates the file's encoding. 'utf8' is the default setting.
mode: The file mode is specified by an integer number called mode. 0o666 is the default value.
flag: This is a string that indicates the file-writing flag. 'w' is the default value.
callback: This function gets invoked when the method is run.
err: If the process fails, this is the error that will be thrown.

11. What are all Looping structure in JavaScript ?
    Ans:-

    Different Types of loops:

    For : loops through a block of code a number of times

    For/in : loops through the properties of an object

    For/of : loops through the values of an iterable object

    While : loops through a block of code while a specified condition is true

    Do/while : also loops through a block of code while a specified condition is true

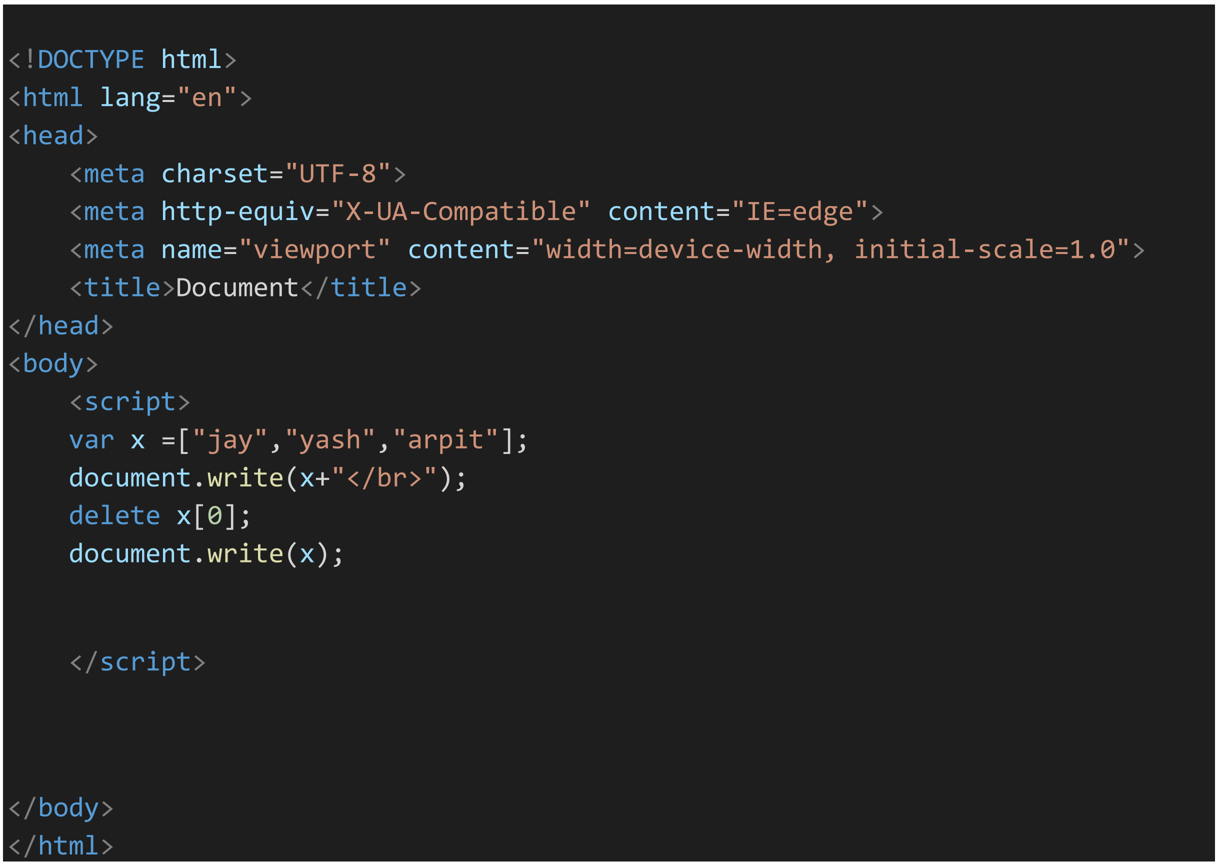
12. How can you convert the string of any base to an integer in JavaScript?
    Ans:-

```
<script>
    let stringConversion = (string_value) => {
    console.log("Initial Type: " + typeof string_value);
    let integer_value = parseInt(string_value);
    console.log("Final Type: " + typeof integer_value);
    console.log(integer_value);
    };

    stringConversion("512000");
    stringConversion("126410");
    stringConversion("0x8975");
</script>
```

Output:

```
Initial Type: string

Final Type: number

512000

Initial Type: string

Final Type: number

126410

Initial Type: string

Final Type: number

35189
```

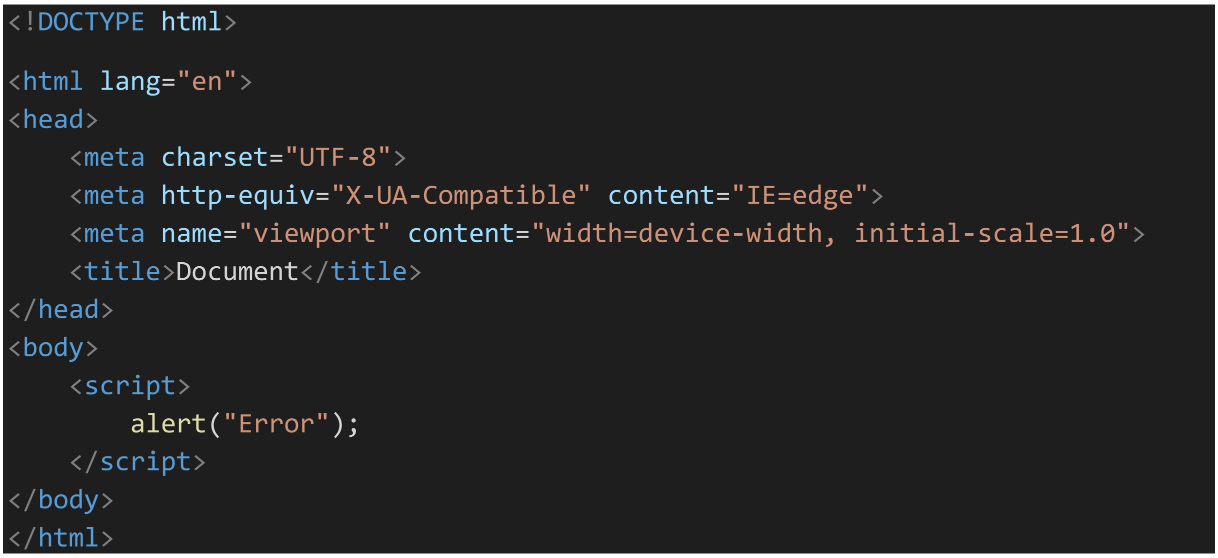
13. What is the function of the delete operator?
Ans.



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
    var x =["jay","yash","arpit"];
    document.write(x+"</br>");
    delete x[0];
    document.write(x);


    </script>



</body>
</html>
```

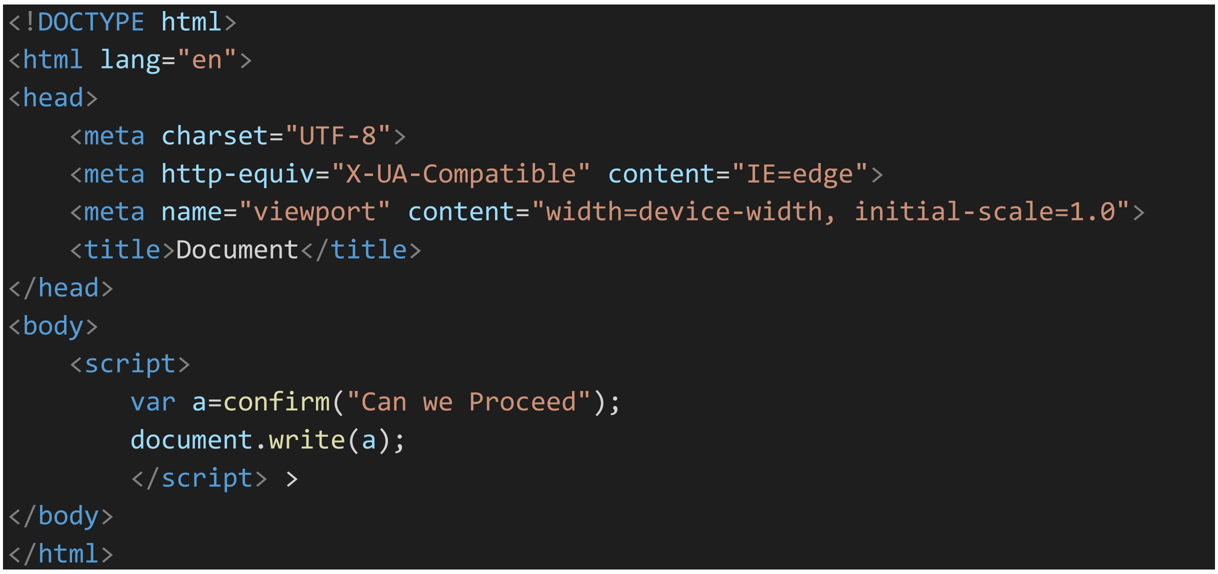
14. What are the all type of pop pup boxes are available in JavaScript ?
Ans:-

ALERT BOX:-



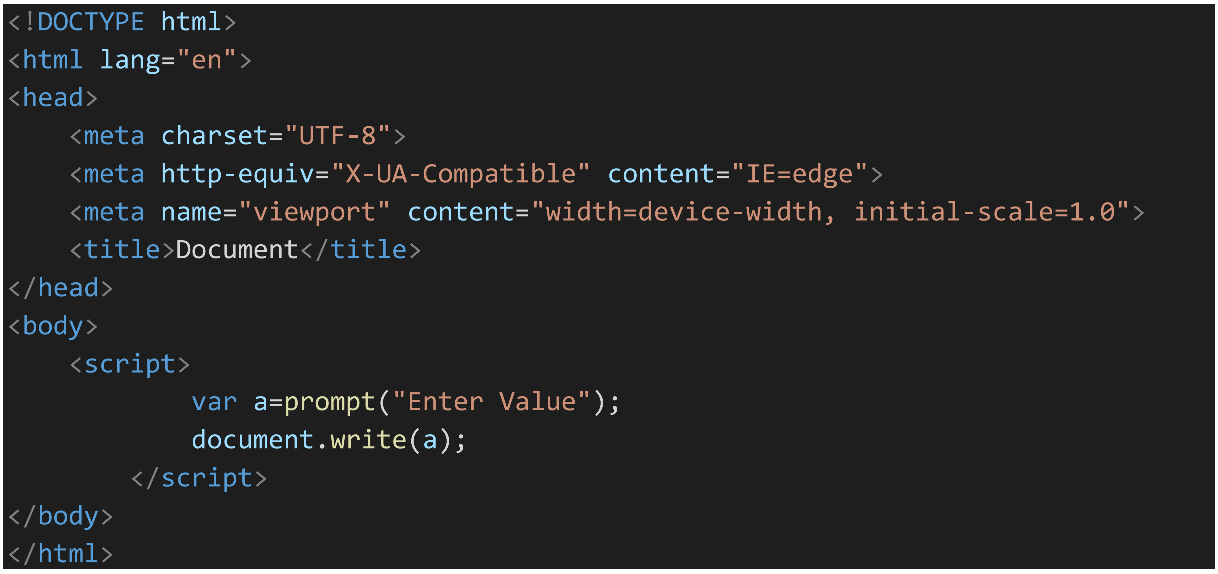
```
<!DOCTYPE html>

<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
        alert("Error");
    </script>
</body>
</html>
```

CONFIRM BOX:-



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
        var a=confirm("Can we Proceed");
        document.write(a);
        </script> >
</body>
</html>
```

PROMPT BOX:-



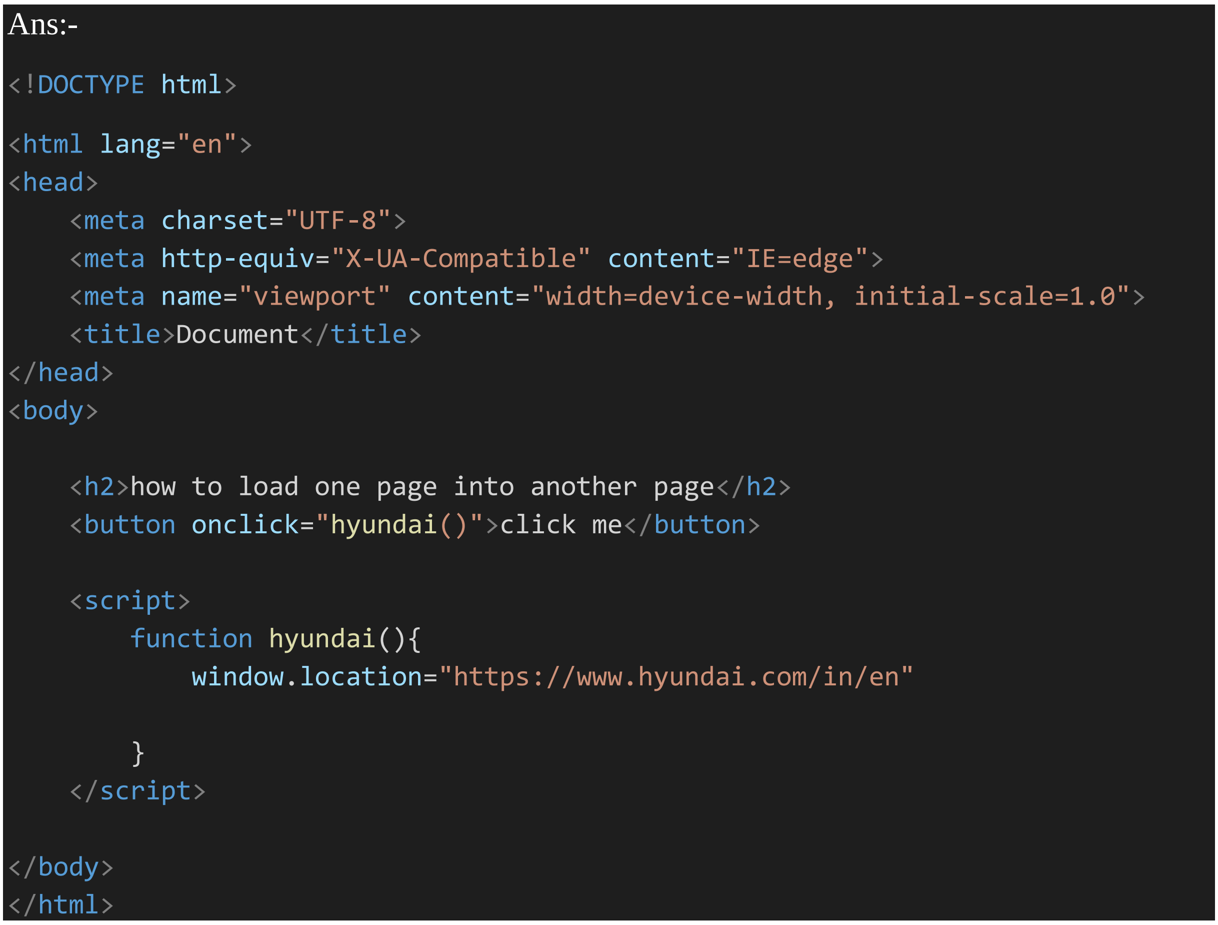
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
            var a=prompt("Enter Value");
            document.write(a);
        </script>
</body>
</html>
```

15. What is use of Void(0)?
Ans:-

If inserting an expression into a web page results in an unwanted effect, then use JavaScript void to remove it. Adding “JavaScript: void (0)”, returns the undefined primitive value.

```
<!DOCTYPE html>
<html>

<head>
    <title>JavaScript:void(0)</title>
</head>

<body>
    <center>
        <h3>JavaScript:void(0)</h3>
        <a href="javascript:void(0);"
        ondblclick="alert('Welcome to TOPS TECHNOLOGY')">
Double click on me </a>
    </center>
</body>

</html>
```

16. How can a page be forced to load another page in JavaScript?



Ans:-

```html
<!DOCTYPE html>

<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>

    <h2>how to load one page into another page</h2>
    <button onclick="hyundai()">click me</button>

    <script>
        function hyundai(){
            window.location="https://www.hyundai.com/in/en"

        }
    </script>

</body>
</html>
```

17. What are the disadvantages of using innerHTML in JavaScript?

Ans:- **The use of innerHTML very slow:** The process of using innerHTML is much slower as its contents as slowly built, also already parsed contents and elements are also re-parsed which takes time.

Content is replaced everywhere: Either you add, append, delete or modify contents on a webpage using innerHTML, all contents is replaced, also all the DOM nodes inside that element are reparsed and recreated.

Appending to innerHTML is not supported: Usually, += is used for appending in JavaScript. But on appending to an Html tag using innerHTML, the whole tag is re-parsed.

Can break the document: There is no proper validation provided by innerHTML, so any valid HTML code can be used. This may break the document of JavaScript. Even broken HTML can be used, which may lead to unexpected problems.