Javascript is programming language
used to make the static webpages as dynamic
LiveScript -> by Netscape -> Javascript
Javascript versions will be referred as ES
Latest ES9. Mostly used version ES6
Most add on features are introduced in ES5 and ES6
ES-> ECMAScript
Javascript is a interpreter software
Javascript is a loosely coupled language

Ways to work with Javascript
1. Internal scripting -> In the same HTML document, script tags
2. External scripting -> Separate JS file, linked with script tag src attribute

Ways to display the output
1. alert("Message");
   The output will be display in dialog box
2. document.write("Message");
   The output will be display in new page
3. document.getElementById("idName").innerHTML = "Message";
   The output will be display in same page
4. console.log("Message");
   The output will be display in console window

Javascript comments
1. Single line comment - //
2. Multi-line comments - /* */

Javascript variables
Syntax for JS variables
keyword varName = value;

varName are user-defined starts with letters

Keywords
1. var -> Global -> accessible throughout the coding
2. let -> Local -> accessible only inside the specific block
3. const -> constant -> It cannot be modified

Javascript datatypes
1. String -> includes characters -> " or "" or ``[Template literals]
2. Numbers -> includes both whole numbers and decimal numbers
3. Boolean -> true or false
4. null -> Empty [valid value]
5. undefined -> Not defined the value yet

Javascript functions

Syntax
function functionName(arguments)
{

}

By default, every function will return the undefined.
So that we can write two types of functions
1. With arguments
2. Without arguments

ACCESSING ELEMENTS

1. getElementById()

```
let para = document.getElementById('intro');
console.log(para);
```

2. querySelector()

```
Accessing class - use (dot). before
let list = document.querySelector('.lists');
console.log(list);

Accessging id - use # before
let heading = document.querySelector('#heading');
console.log(heading)

Accessing by Tag name
let img = document.querySelector('img');
console.log(img);
```

READ ELEMENTS

1. textContent - read contents of webpage

```
let para = document.getElementById('intro');
let p = para.textContent;
console.log(p);
```

2. innerHTML - read contents of html

```
let lists = document.querySelector('.lists');
let l = lists.innerHTML;
console.log(l);
```

3. password

```
var pwd =
document.getElementById("i1").value;
```

MODIFY ELEMENTS

1.modifying contents of webpage

```
document.getElementById('dynamicContent').t
extContent = 'This is a dynamically generated
paragraph';
```

2.modifying contents of html

```
document.getElementById("output").innerHTM
L = `<p class="alert alert-danger">Invalid
username or password<p>`;
```

EVENTS

1. onclick -> normal button is clicked

```
<form onclick="abc()">
    <button>click</button>
</form>
```

2. onsubmit -> submit button is clicked

```
<form onsubmit="abc()">
  <button type="submit">click</button>
</form>
```

3. onfocus -> when you keep the I beam in the input field

```
<input onfocus="abc()" />
```

4. onblur -> when you click somewhere on the screen except input field

```
<input onblur="abc()" />
```

5. onmouseover -> on hovering

6. onchange -> When there is a change in input field

7. onkeyup -> When an key is pressed keyup

JAVASCRIPT OPERATORS

a. Unary operators
1. Increment operators -> pre[++a] and post[a++]
2. Decrement operators -> pre[--a] and post[a--]

b. Binary operators
1. Arithmetic operators -> +, - , *, /, %
2. Assignment operators -> =, +=, -= , *=, /= , %=
3. Relational operators -> <, >, <=, >=, ==, !=, ===
4. Logical operators -> &&, ||, !
5. Bitwise operators -> &, |, ~, ^
6. Shift operators -> <<, >>

c. Ternary operators -> ?:

JAVASCRIPT CONDITIONAL STATEMENTS

1. Simple if
2. if else
3. else if
4. if ladder
5. nested if
6. switch

JAVASCRIPT LOOPING STATEMENTS

1. for loops
2. for of loops
3. for in loops
4. forEach loops
5. while loops
6. do while loops

JAVASCRIPT LOOP CONTROL STATEMENTS

1. break
2. continue

JAVASCRIPT OBJECTS

key and value pair.

1. Object literal
2. new keyword

NORMAL FUNCTION

```
function abc()
{

}

//Storing a function as a variable
const abc = function () {

}
```

ARROW FUNCTION

```
const abc = () => {}
```

eg: normal function

```
function lmn(a){
   console.log(a)
}

const lmn = function(a){
   console.log(a)
}
```

eg: arrow function

```
const lmn = (a) => {console.log(a)}
```

CALLBACK FUNCTIONS
A function which accepts another function as an arguments

JAVASCRIPT ARRAYS

1. Array literal
2. new keyword
3. Array constructor

ARRAY METHODS

1. concat -> arr1.concat(arr2); -> It doesn't changes the original array
2. every -> arr1.every(callbackfn)
3. filter -> arr1.filter(callbackfn)
4. find -> arr1.find(callbackfn)
5. findIndex -> arr1.findIndex(callbackfn)
6. indexOf -> arr1.indexOf(element);
7. lastIndexOf -> arr1.lastIndexOf(element)
8. some -> arr1.some(callbackfn)
9. map -> arr1.map((val)=>{})

```
<body>
    <center>
        <form action="https://www.ethnus.com" class="form m-5"
onsubmit="return validate()">
            <table>
                <tr>
                    <td class="text-center"><h1>Gmail
Login</h1></td>
                </tr>
                <tr>
                    <td><p id="output"></p></td>
                </tr>
                <tr>
                    <td><input id="i1" class="form-control my-2"
type="text" placeholder="Username" required/></td>
                </tr>
                <tr>
                    <td><input id="i2" class="form-control my-2"
type="password" placeholder="Password" required/></td>
                </tr>
                <tr>
                    <td class="text-center"><button type="submit"
class="btn btn-primary">Login</button></td>
                </tr>
            </table>
        </form>
    </center>

    <script>
        function validate(){
            var un = document.getElementById("i1").value;
            var pwd = document.getElementById("i2").value;
            if(un.length < 5 || pwd.length < 5)
            {
                document.getElementById("output").innerHTML = `<p
class="alert alert-danger">Invalid username or password<p>`;
                return false;
            }
            return true;
        }
    </script>


</body>
```

10. push -> arr1.push(element);
11. slice -> arr1.slice(fromIndex, toIndex)

BOM -> Browser Object Model
(navigating to forward page and backward page)