Java Script

- JS was invented by Brendan Eich in 1995
- It was developed for Netscape 2, and became the ECMA-262 standard in 1997.
- European Comp Manufactures Association Ecma International is an organization that develops standards in computi & technology
- ES1 to ES5 (1997 to 2009)
- Js is a light weight object programming language
- Use in form submit
- in client side validation , - pop up / event on click.

Uses:

- client side executi / browser (JS query, React.JS, angular JS)
- website ServerSide (node.js, Expres.js).
- Mobile Development (Hybrid App) (Framework for mobile app react native, phone gap etc)
- Software Development (Eection js, Ex - vscode, Framework etc)

CODE

```
<script> alert ("hellow"); <script>
js file connect -> <script src = "script.js"> </script>
```

VARIABLE

```
Let a = 10
Let a = 20;  — error

var a = 10 }
var a = 20 }  no error

Console.log (a);   const abc;
a = "hina";        Cons pi = 3.142;
```

```
let name = "Hira"
name = "Sumaira"
Console.log (name);
```

```
let isPass = true;
Console.log (isPass);
```

x

- **Variable**: is just like a container, store information, reserve space in memory, its data can vary but memory location will alway remains same
- **Naming variable in JS**:
- variable's name can't be any keyword e.g alert, prompt
- case sensetive, name should be different not same. alphabet, name, ~~number~~, dollar sign, and underscore
- not start with digit, no space allowed

As a good programmer

- Variable name should be match with content
- 2 name variable join first start with small letter and 2nd capital letter (Camel case) fullName, rollNumber.
- Var (used before EcmaScript (declare again and again)
  After Es6 -> 2 keyword use for declaration variable
- let (its value can change any time in PL)
- Const -> example pi value

~~Variable~~ Block Scope Variable: - { }

Globle Scope Variable: used globally in whole Program.

Comment in JS: Single line //

Double line /* */

object hai or is lea sub object doc hai

Print / Display in JS (document.write ("world");)

on Browser: window.document.write ("hina")

in Console: Console.log ("hina");

popup : window.alert ("hina");

Taking Input from User in JS:

Prompt() function ask for the user for input

let answer = prompt ("Do u want to send payment y/n ?:");

Variable

- Var , let , Const
- Var a=6; let a=6 ; let a = 10;

not allowed again declare

- Const abc = "hello";

Datatypes Rules Data Types

- number , string , boolean , null , undefined
- Array , object , functions

let abc; let abc123; let $abc; let _abc;

int age = 55;

document.write (age);

console.log (type of age);

Primitive Data Types

String → let rollNO =56;

Boolean → let name = "Noor"; let ispass= true;

Undefined → let percentage;

Object (null) → let class= null;

Data Types in JS

Primitive Type

- Numbers
- Strings
- Boolean

- null
- undefined

} also known as Trivial / other data type

Composite / Non-Primitive Types

- Objects
- Arrays

Non-Primitive Data Types

Array: Store multiple value in single variable. value written square backets [ ]

```
let info = [5, 'hindi', comp];
Console.log (info)
```

Array	Object

let info = ['math', 'eng', 'comp'];
console.log(typeof info);
	let student {
	rollno =5,
	name: 'dania';
	sub : math ; }
	Console.log (typeof student);

— × ——

Class : 4 "Operators"

Arithmetic (+, -, *, /, %, Exponentation) ++a a++ Increment / Decrement --a

Assignment a = a+2 , a = a-2, a = a*2

Comparison a==b , a>=b , a<=b , a!=b

Logical && || ! not

Conditional

```
function welcome() {
 document.write ("welcome"); }
function sub(a,b) {
 let c=a+b;
 return c; }
```

— x — x — x —

Arithmetic Operator

```
let a = 10; b = 20;
consol.log (a+b);
   "      (a-b);
   "      (axb);
   "      (a/b);
   "      (a+b+a);
```

Class: 5;

Comparision:

```
if (a==b) { document.write (a equal to b); }
else { doc.write (a not equal to b)};

let num=40
let result = num %2 ==0? "Even" : "odd";)
alert (result);
```

Class: 6:

Create Marksheet (Solve Assignment 01)

Table Create in JS

```
document.write ("<table border = 4>");
   "         ("<tr><th> Subject </th></th> Marks
             </th><th> Out of </th></tr>");
```

Class : 7 Condition Operator

if , else , if else if Statement

```
if (a==b) { doc.write ("equal"),
~~if~~ else if (a>=b) { doc.write ("a greater than b")
  else { doc write ("b greater than a);
~~else~~
```

Switch let reply = prompt ("Do You want to continue --");

```
Switch (reply)
{ case "y"; doc.write ("Continue"); break;
  case "yes";
  document.write ("continue"); break;
  case 'N' ; doc.write ("end");
  default { doc.write ("wrong input"); }
```

String Creation & Manipulation

```
let str1 = "JavaScript";
 " Str2 = 'Java Script';
   Str3 = 'String template'; // template literal
                     (adjacent to 1 key in keyboard)
```

Template literal
introduce ES6

- for next line \n
- for tab \t
- for print \ in string \
- for write Variable in String ${var name};
- for double quote 'hello \" hello'

Str.length length of String

String Properties & methods.

str.length

To Join Strings → doc.write(str1+" "+str2)/(str1," ",str2)

by concat → let str4.concat(str2)/str.concat(str2,str3)
doc.write(str4);

Concat - join to string

str1 = Java str2 = "Script"

new = str1.concat(str2)

output → Javascript → (str2, str3)
add more string.

str.trim() remove space start & end -
trimStart() trimEnd()
str.toUppercase) uppercase
str.toLowercase( ) lowercase.
str.replace(r) replace
(str.replace("reading","learning"));

str.includes("is") //search word is/if not found
return -1

str = "she is learning JS"

str.include("is");
str.include("JS"); return True
str.include("we"); return false

Class . 8 ; Switch /break /string Method /
For loop.

Loop: To execute a piece of code
finite loop & infinite loop
finite loop (ending point)
infinite (not end) memory full/
Computer hang.

for

```
for (let i=1 ; i>=10 ; i++)
    { Console log (i); }
```

for of:

```
let arr = [34, 'abc', 'xyz']
for (let value of arr) {
    doc.write (value);}
```

for in

```
const student = { name:"fatima",
                  rollno:55,
                  sub: "comp"}

for (let key in Student)
  {
    document.write (Key,": ", student{k}, "<br>")
  }
```

While

```
while (i<=5) {doc.write ("Hello",i); i++}
```

Do-while

```
Do{
  doc.write ("Hello", <br>); i++ }
while (i<=5);
```

Class: 9:

Array.

Store multiple value in single Variable
Value written in square [ ]

Value Sepeated by comma.
each position is called index
arr[0] , arr[1)
arr=['a', 'b', 'c', 'd'];
arr.length() // length of arr
arr.push("e"); // add word in array in the end
arr.unshift("e"); // add in Start
arr.shift( ); // remove a word from start
arr.pop() // remove word from last & return array
arr.toString() // convert array in string
arr.concat(arr2) // join 2 or 3 string

FUNCTION

A JS function is a block of code designed to perform a particular task.
A JS function is executed when "something" invokes it {calls it}

func name.
function abc() { doc.write("Noor"); }
" 5 time
abc();

Function definition (define)	function invoke (call)
function fname()	fname()
{ block of codes }	fname(arg1, arg2) or
function fname(p1, p2)	fnam(2,3)
{ doc.write(p1+p2) }	let ans = sum(2,3)
function sum(p1, p2)	console.log(ans);
{ ans = p1+ p2; return ans; }	

Arrow function

```
Const fname= () => { block of code }
```