DAY – 1 INTRO

Front End Devloper - HTML,CSS,Bootstrap,Tailwindcss And Javascript, React and NextJs and Typescript

Database : SQL and NOSQL

Static and Dynamic webpage:-

1. **Static webpage:-**

A static webpage is a page that always shows fixed content.

The content does NOT change unless a developer manually edits the file.

2. **Dynamic webpage:-**

A dynamic webpage is a page that changes content automatically based on any triggers.

React - client side rendering process.

Server sends HTML → Browser runs JS → JS builds UI → Browser shows the final page.

1. **BLINK (Browser Rendering Engine)**

The part of the browser that converts your HTML & CSS into actual visual webpage.

2. **CSSOM (CSS Object Model)**

The browser **parses your CSS file** and converts it into a structured tree called **CSSOM**.

CSSOM = Browser's internal map of all CSS styling.

3. **DOM (Document Object Model)**

DOM = HTML converted into objects so JavaScript can control the webpage.

4. **V8 Engine (JavaScript Engine of Chrome)**

What is V8?

V8 is Google's JavaScript Engine used in Chrome and Node.js.It executes JavaScript code

V8 = The brain that runs JavaScript inside Chrome and Node.js.

Expression is **any code that produces a value**.

Ex : var a= "Hello"

S**tatement** is a complete instruction that tells the program what to do.

Ex : if,for,while

Variable:-

Declaration : Creating a variable (reserving memory) **without assigning any value.**

Initialization : **Assigning a value** to a declared variable **for the first time**.

Feature	var	let	const
Scope	Function	Block	Block
Redeclare	✅Yes	❌No	❌No
Reassign	✅Yes	✅Yes	❌No
Must initialize	❌No	❌No	✅Yes

What is Hoisting?

Hoisting is JavaScript's default behavior.You can access a variable before initialization execution time javascript is declaration part only move to a top of the scope.

JavaScript Data Types :-

1. Primitive Data Types – Number,String,Boolean,undefined,null
2. Non Primitive Data Types – Object,Array,Function

Operators :-

1. **Arithmatic Operator**
2. **Assignment Operator**
3. **Comparison Operator**
4. **Logical Operator**

1. Arithmatic Operator:-

Operator	Meaning	Example
+	Addition	5 + 3 = 8
-	Subtraction	5 - 3 = 2
*	Multiplication	5 * 3 = 15
/	Division	10 / 2 = 5
%	Modulus (remainder)	10 % 3 = 1
**	Exponent (power)	2 ** 3 = 8
++	Increment	x++
--	Decrement	x--

2. Assignment Operator:-

Operator	Meaning	Example
=	Assign	x = 10
+=	Add & assign	x += 5 → x = x + 5
-=	Subtract & assign	x -= 3
*=	Multiply & assign	x *= 2
/=	Divide & assign	x /= 4
%=	Modulus & assign	x %= 2

3. Comparison Operator:-

Operator	Meaning	Example
==	Equal (loose)	5 == "5" → true
===	Strict equal	5 === "5" → false
!=	Not equal	5 != 3
!==	Strict not equal	5 !== "5"
>	Greater than	6 > 3
<	Less than	3 < 6
>=	Greater or equal	5 >= 5
<=	Less or equal	4 <= 6

4. Logical Operator:-

Operator	Meaning	Example
&&	AND	true && true → true
||	OR	True ||false → true
!	NOT	!true → false

2. NON PRIMITIVE DATA TYPES

1. Object

2. Array

3. Array of Object

4. Function

DAY 1 TASKS:-

1. **Implicit Conversion & Explicit Conversion ?**

Implicit Conversion **:-**

JavaScript automatically converts one data type to another **without you writing code.**

Ex

one datatype to another datatype

let a=”jagath”;

let a=5;

and console.log(“5”==5)

Explicit Conversion :-

You convert type manually using functions.

Ex

Function(){

let a=5

console.log(String(5)) -> convert to string

}

2. **Difference between return and console.log ?**

   return → Sends **output from a function** to the place where the function was called. Ends the function immediately.

   Console.log → Prints message to the **browser console**. Used only for **debugging**.

3. **Lexical scope and Closures ?**

Lexical scope means a function can access variables from its parent (outer) scope. Outer function variables are available inside inner functions.

Example:-

```
let a = 10;

function parent() {
  let b = 20;
  function child() {
    console.log(a); // 10
    console.log(b); // 20
  }
  child();
}

parent();
```

Closures :-

Closure = Function + its outer scope variables stored together even after the outer function is finished.

Example

```
function outer() {
  let count = 0;
  function inner() {
    count++;
    console.log(count);
  }
  return inner;
}
const counter = outer(); // outer finished
counter();  // 1
counter();  // 2
```

4. **Differences between null and undefined ?**

Null - A value that represents **nothing**, **empty**, or **no data** — but intentionally set

Undefined - A variable is declared but **not given a value**

5. **Array ,Object , Array of Object and JSON ?**

Array - An **ordered list** of values stored in **index format**.

Access → console.log(fruits[1]); // banana

Array Of Object : An **array** where each element is an **object**.

Access → console.log(students[1].name); // Kamal

JSON → **J**ava**S**cript **O**bject **N**otation. A **data format** used to send/receive data between backend and frontend.

Access →fetch method