

Training Day 4

24th June 2025

Topics Covered – JavaScript Hoisting, Type Coercion

Hoisting

We learned that **Hoisting** is JavaScript's default behavior of moving declarations (not initializations) to the top of their scope during compilation.

Example:

```
console.log(a); // undefined  
var a = 5;
```

In the background, JavaScript treats it like:

```
var a;  
console.log(a);  
a = 5;
```

Only declarations (var, function) are hoisted, not the actual assignments.

Type Coercion

Coercion refers to JavaScript's ability to automatically or implicitly convert data types during operations.

Examples:

```
console.log("5" + 2); // "52" (string concatenation)  
console.log("5" - 2); // 3 (string converted to number)
```

We also discussed **strict comparison (===)** which checks both value and type.

URN:2302665

Code Task – Loops Implementation

We practiced creating loops to handle repetitive tasks. Focused on three main types:

- **For Loop**

```
for (let i = 1; i <= 5; i++) {  
  console.log("Count:", i);  
}
```

- **While Loop**

```
let i = 1;  
while (i <= 5) {  
  console.log(i);  
  i++;  
}
```

- **Do-While Loop**

```
let j = 1;  
do {  
  console.log(j);  
  j++;  
} while (j <= 5);
```

Task Objective: Use loops to print multiplication tables, iterate over arrays, and build simple logic like number checking.

Task for Tomorrow

- Practice problems on nested loops (e.g., pattern printing, tables).
- Explore break and continue statements.
- Begin introduction to **functions** in JavaScript: definition, usage, and types.

