Training Day 7

27th June 2025

TOPICS COVERED

- **Function Calling**

A function is called by using its name followed by parentheses ().

Example:

```
function greet() {
  console.log("Hello!");
}
greet();  // Calling the function
```

- **Return Function**

Functions can return values using the return keyword.

Example:

```
function add(a, b) {
  return a + b;
}
let result = add(5, 3);  // result = 8
```

- **Block Scope: var vs let**

var is function-scoped, while let is block-scoped.

Example:

```
{
  var a = 10;
  let b = 20;
}
console.log(a);  // 10 (var is accessible outside the block)
console.log(b);  // Error (let is not accessible outside the block)
```

- **Arrow Function**

A more concise way to write functions using =>.

Syntax:

```
const greet = () => {
  console.log("Hello");
};

const add = (a, b) => a + b;
```

Arrow functions do not have their own this context and are often used in callbacks or concise logic.

- **Higher-Order Function**

A function that takes another function as an argument or returns a function.

Example:

```
function greet() {
  return function() {
    console.log("Hello from inner function");
  };
}
let inner = greet();
inner();
```

- **Callback Function**

A callback is a function passed as an argument to another function, and executed later.

Example:

```
function sayHello(name, callback) {
  console.log("Hello " + name);
  callback();
}
```

```
function sayBye() {
  console.log("Goodbye!");
}
sayHello("Navneet", sayBye);
```

TOOLS USED

Visual Studio Code (VS Code)

Chrome Browser (JavaScript Console)

TASK FOR TOMORROW

Read and understand the basics of:

DOM Manipulation

Asynchronous JavaScript

JavaScript Promises

API Calls