Day1 – JavaScript: Values, Types, and Variables

1. Values & Types

JavaScript values are divided into two categories:

- Primitive Types: Number, String, Boolean, Null, Undefined, BigInt, Symbol (Immutable).
- Objects: arrays, functions, dates, etc. (Mutable).

Primitives are copied by value, while objects are copied by reference.

2. Type Coercion

- JavaScript automatically converts values between types depending on context.
- + performs string concatenation if one operand is a string; otherwise numeric addition.
- == performs loose equality (with coercion); === performs strict equality (no coercion).
- Falsy values: false, 0, -0, 0n, “”, null, undefined, Nan.
- Everything else is truthy.

Examples:

[] == "" // true → array becomes "" by toString()

[] == 0 // true → array becomes "" → then Number("") = 0

0 == false // true → false becomes 0 → 0 == 0

"" == 0 // true → "" becomes 0 → 0 == 0

3. var / let / const

- var: function-scoped, hoisted with value undefined.
- let: block-scoped, hoisted but uninitialized (TDZ).
- const: block-scoped, cannot reassign reference (but objects inside are mutable).

Prefer const by default, and use let only when reassignment is necessary.

4. Hoisting

JavaScript hoists declarations to the top of their scope:

- Function declarations are fully hoisted (can be called before definition).
- var is hoisted with undefined.
- let and const are hoisted but in the Temporal Dead Zone until initialized.

5. Keys:

- Use const whenever possible.
- Understand how coercion works to avoid bugs.
- Prefer explicit conversions: Number(), String(), Boolean().
- Use === for comparison unless you intentionally need == behavior.