What is Encapsulation?

✅ Definition:

Encapsulation is one of the core principles of OOP (Object-Oriented Programming).It means **wrapping data (properties) and methods (functions)** into a single unit (usually a class), and **restricting direct access** to some parts of the object.

✅ Key Goals of Encapsulation:

1. **Protect** data from unwanted changes.
2. **Hide internal details** (implementation).
3. Allow access to data **only through methods** (getters/setters).

✅ Example in JavaScript:

```
class BankAccount {
  // Private property (using # symbol for true encapsulation)
  #balance = 0;

  constructor(initialAmount) {
    if (initialAmount >= 0) {
      this.#balance = initialAmount;
    }
  }

  // Getter method to access balance
  getBalance() {
    return this.#balance;
  }

  // Setter-like method to deposit money
  deposit(amount) {
    if (amount > 0) {
      this.#balance += amount;
    }
  }
}

// Creating an object
const account = new BankAccount(1000);
account.deposit(500);
console.log(account.getBalance()); // Output: 1500
console.log(account.#balance); // ❌ Error: Cannot access private field
```

✅ Explanation (Point-by-Point):

1. #balance is a **private field** – it cannot be accessed directly from outside the class.
2. The deposit() method is the **only way** to change the balance.
3. The getBalance() method is used to **safely read** the balance.
4. Trying to do account.#balance from outside will cause an error – that's **encapsulation**.

✅ Real-Life Analogy:

Think of a **vending machine**:You can press buttons (methods), but you can't directly access the internal parts (data) – it's **encapsulated**.