What is Abstraction?

✅ Definition:

Abstraction means **hiding complex internal details** and showing **only the essential features** of an object.It helps reduce complexity and makes code easier to use and maintain.

✅ Key Goals of Abstraction:

1. Show **what an object does**, not **how** it does it.
2. Provide a **simple interface** to interact with complex logic.
3. Improve code readability and reduce errors.

✅ Example in JavaScript:

```
class Car {
  // Public method (interface)
  start() {
    this.#injectFuel();
    this.#igniteEngine();
    console.log("Car started.");
  }

  // Private methods (implementation details)
  #injectFuel() {
    console.log("Fuel injected.");
  }

  #igniteEngine() {
    console.log("Engine ignited.");
  }
}

// Using the class
const myCar = new Car();
myCar.start();
// Output:
// Fuel injected.
// Engine ignited.
// Car started.

myCar.#injectFuel(); // ❌ Error: Private method can't be accessed directly
```

✅ Explanation (Point-by-Point):

1. The user only calls myCar.start() — that's the **abstracted interface**.
2. The inner working methods (#injectFuel() and #igniteEngine()) are **hidden/private**.
3. Abstraction **hides complexity** from the user and shows only what is necessary.

✅ Real-Life Analogy:

When you **press a button on a TV remote**, you don't need to know the electronics inside — you just need the **interface**. That's **abstraction**.