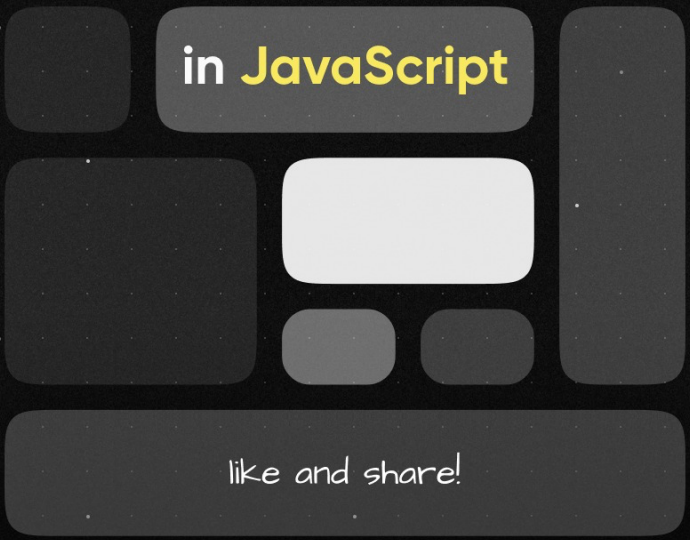
JS

Design Patterns

in JavaScript



JS

Design patterns are established, well-documented solutions to common programming problems that can be applied in various contexts.

Creational Patterns:
Singleton
Factory Method
Abstract Factory
Builder
Prototype

Structural Patterns:
Adapter
Decorator
Facade
Proxy

Swipe →

JS

Creational Patterns

Focus: Object creation

Goal: Provide mechanisms to control and manage how objects are instantiated and initialized.

Benefits:

- Decouple object creation from usage.
- Promote reusability of object creation logic.
- Improve code flexibility for handling different object types.

Use case Scenario: E-commerce site uses a Factory Pattern to create product objects (clothes, electronics) based on category, ensuring flexibility for future product types.

Swipe ⟶

JS

Structural Patterns

Focus: Object composition

Goal: Define ways to compose classes and objects to form larger structures and relationships.

Benefits:

- Structuring objects clearly: Improves code readability and maintainability.
- Enabling dynamic composition: Makes code adaptable to changing needs.
- Promoting loose coupling: Enhances code flexibility and reduces dependencies.

Use case Scenario: Video player uses an Adapter Pattern to play various video formats (MP4, AVI), allowing seamless integration of new formats.

Swipe →


Subscribe on YouTube to learn more
Code with Sloba
@CodewithSloba • 21.8K subscribers
Hi, my name is Slobodan (Sloba) Gajic, and I'm a JavaScript software developer. Building a
Subscribed