<https://learn.javascript.ru/courses/jsarchitect>

1. Роутинг SPA-приложений (Single Page Application) и "ленивая" загрузка (lazy-loading).
2. Событийная архитектура, паттерны взаимодействия компонент.
3. Жизненный цикл компонент, обработка событий и утечки памяти.
4. Хранение и обновление общего состояния системы, способы организации.
5. Структура API проекта для расширяемости и тестируемости.
6. Эффективное взаимодействие с браузерным DOM в проекте без фреймворков.
7. Prerendering и server-side rendering.
8. Наследование, область применения в современных проектах.

Рассмотрим базовые концепции применяемые в React-проектах:

* Понятие и реализация реактивности.
* Однонаправленный поток данных.
* Виртуальный DOM.
  + vdom - каждый раз рендерим компонент с нуля в памяти, а затем умный алгоритм reconciliation обновляет дом
* Функциональные и классовые компоненты.
* Хранение и изменение состояния компонента.
* Иммутабельность данных.
* Хуки.
* JSX (JavaScript syntax extension).

Разберём архитектурные решения, концепции и подходы, применяемые в Angular-проектах:

* Базовые структурные единицы: модули, компоненты, сервисы, директивы.
* DI (Dependency Injection) - внедрение зависимостей.
* Двунаправленный поток данных.
* Реализация реактивности, библиотека RxJS.
* Работа с асинхронным кодом.
* Управление глобальным состоянием приложения.
* Роутинг и "ленивая" загрузка.

Разберём архитектурные решения, применяемые во Vue-проектах:

* Реактивность, вычисляемые свойства, отслеживание.
* Однофайловые компоненты.
* Хуки жизненного цикла.
* Двунаправленный поток данных.
* Роутинг и "ленивая" загрузка.
* Входные параметры и события компонентов.
* Слоты и именованные слоты компонентов.
* Управление глобальным состоянием приложения.