**React** — это JavaScript-библиотека для разработки пользовательского интерфейса.

JSX — расширение языка JavaScript, После компиляции каждое JSX-выражение становится обычным вызовом JavaScript-функции, результат которого — объект JavaScript.

**Babel**

Транспайлер, преобразующий современный JavaScript (и JSX) в код, совместимый со старыми браузерами.

**SPA (Single Page Application)** – одностраничное приложение, где все происходит на одной HTML-странице, без полной перезагрузки страницы при навигации.

**Рендеринг** – процесс отображения компонентов на экране. В React это происходит с помощью ReactDOM.render() или автоматом при изменении **состояния**.

**Пропсы (props)**

Данные, передаваемые от родительского компонента к дочернему (только для чтения).

**Компоненты** – независимые, переиспользуемые части интерфейса (например, кнопка, форма). Бывают:

* **Классовые** (с жизненным циклом, используют this)
* **Функциональные** (современный подход, используют **хуки**)

**Состояние (state)**

Внутренние данные компонента, которые могут изменяться (через useState или this.setState). При изменении состояния компонент ре-рендерится.

**Жизненные циклы компонента**

Методы, вызываемые на разных этапах "жизни" компонента (монтирование, обновление, размонтирование). В функциональных компонентах заменены на хуки (

**Принцип единственной ответственности**

Каждый компонент должен решать одну задачу (например, отображать кнопку, а не управлять данными и UI).

**Односторонний поток данных**

Данные в React передаются только сверху вниз (от родителя к ребенку через пропсы). Дочерние компоненты не могут напрямую изменять пропсы.

**Иммутабельность**

Неизменяемость данных. В React состояние нельзя изменять напрямую – нужно создавать новую копию

**Чистая функция (Pure Function)**

Функция, которая:

Не изменяет входные данные.

Всегда возвращает одинаковый результат для одинаковых аргументов.

**Функциональная компонента**

Компонент, написанный как функция (вместо класса). С появлением хуков стал стандартом.

**Хук (Hook)**

Функция, позволяющая использовать состояние и другие возможности React в функциональных компонентах (начинается с use).

**useState**

Хук для управления состоянием в функциональных компонентах:

**useEffect**

Хук для выполнения побочных эффектов (запросы к API, подписки) в функциональных компонентах: