

React функционирует как слой представления приложения, преобразуя состояние и данные в пользовательский интерфейс. Компоненты описывают, как должен выглядеть интерфейс для определённого состояния, а React обеспечивает соответствие DOM этому описанию.

В компоненте **Counter** хук `useState` хранит текущее значение счётчика. Когда пользователь нажимает кнопки **Increment** или **Decrement**, состояние обновляется. Это изменение состояния вызывает повторный рендер компонента, и React обновляет только те части DOM, которые действительно изменились.

Для оптимизации производительности React использует виртуальный DOM. Вместо прямого обновления реального DOM React сравнивает предыдущее состояние виртуального DOM с новым с помощью процесса, называемого **diffing**. После этого к реальному DOM применяются минимальные изменения, известные как **patching**.

Как объясняется в главе 1, React не занимается получением данных или маршрутизацией; он сосредоточен исключительно на эффективном рендеринге пользовательского интерфейса. Такое разделение ответственности делает приложения более модульными и удобными для поддержки.