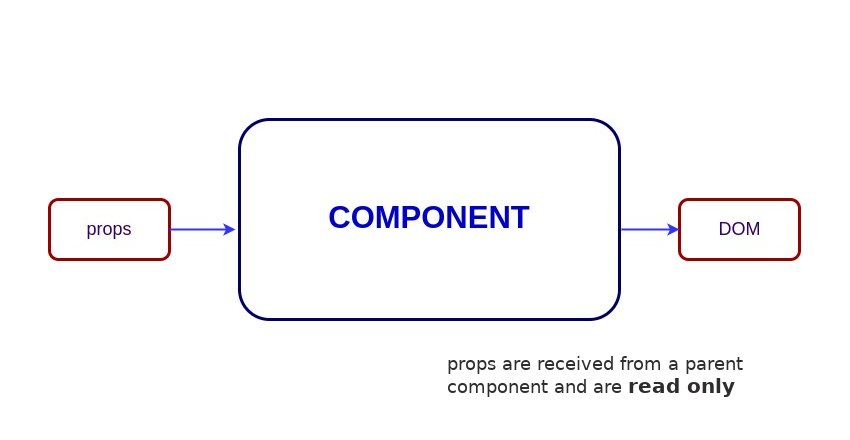
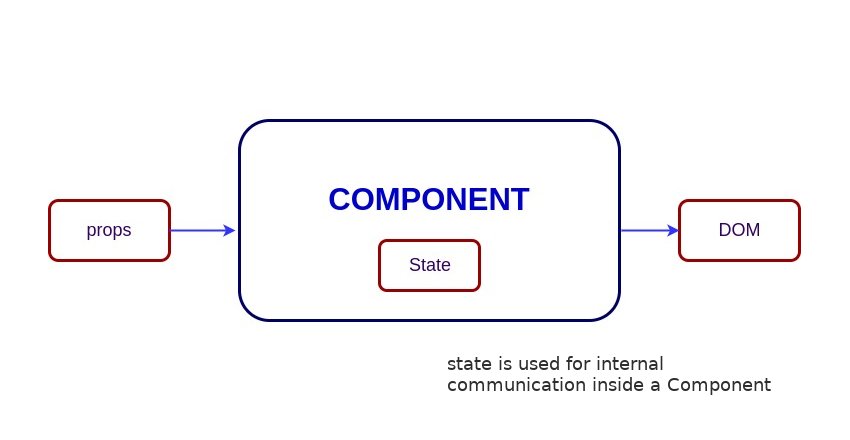
**~ Про стэйтфул и стейтлэсс**

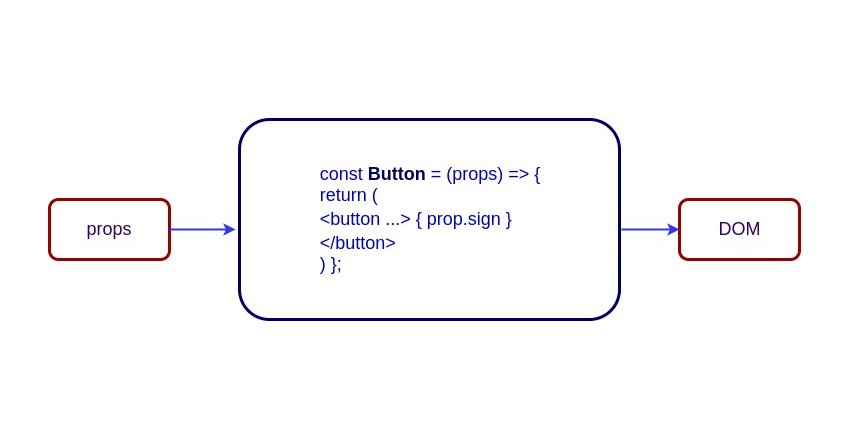
Если компоненты это простые JavaScript функции, то props это входные параметры функции. Исходя из этой аналогии, компонент принимает вход (то, что мы называем props), обрабатывает его, а затем рендерит некоторый JSX код .

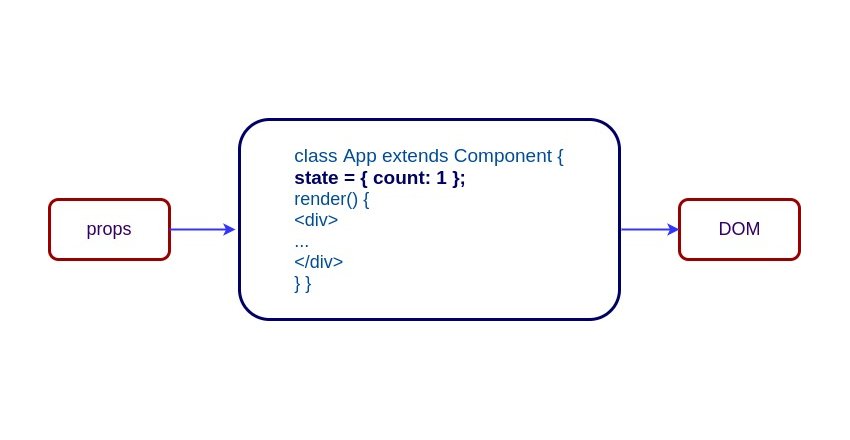


С другой стороны, State является объектом, принадлежащим компоненту, где он объявлен. Его область ограничена текущим компонентом. Компонент может инициализировать свое состояние и обновлять его, когда это необходимо. Состояние родительского компонента обычно становится props дочернего компонента. Когда состояние передается из текущей области, мы называем его prop.



Функциональные компоненты - это просто JavaScript функции. Они берут необязательный входной параметр, который, как я упоминал ранее, является тем, что мы называем props.

Компоненты в виде классов предлагают больше возможностей, и с большим количеством функций появляется больше багажа. Основная причина выбора компонентов класса над функциональными компонентами заключается в том, что они могут иметь state.



**Компоненты имеющие состояние**

Компоненты имеющие состояние всегда являются компонентами в виде класса. Как упоминалось ранее, компоненты с состоянием имеют состояние, которое инициализируется в конструкторе.

### Компоненты без состояния

Вы можете использовать либо функцию, либо класс для создания компонентов без состояния. Но если вам не нужно использовать хуки жизненного цикла в ваших компонентах, желательно перейти на функциональные компоненты без состояния.

**~ Про синтетик ивент**

Synthetic Event – кроссбраузерная оболочка вокруг нативного события браузера. Назначение: события работают одинаково во всех браузерах.

**~ Про индексы**

Ключи помогают React идентифицировать элементы, которые подверглись изменению, добавились или удалились. Ключи должны быть присвоены элементам, заключенным в массивы, чтобы у них была стабильная идентифицируемость:

const numbers = [1, 2, 3, 4, 5];

const listItems = numbers.map((number) =>

<li key={number.toString()}>

{number}

</li>

);

Лучший способ подобрать ключ — использовать строку, которая однозначно идентифицирует элемент списка. Чаще всего, в качестве ключей вы будете использовать ID ваших данных:

const todoItems = todos.map((todo) =>

<li key={todo.id}>

{todo.text}

</li>

);

Если у вас нет стабильных ID для созданных значений, то вы можете использовать индекс значения в качестве ключа (как крайний случай).