Lesson 3

Branch: stateButton

Props (Passing Data to Components)

Props = properties

They allow you to pass data from parent to child components

Props are read-only

Props with Destructuring (Recommended)

```
function Greeting({ name }) {
  return <Text>Hello {name}</Text>;
}

function Greeting({ name, course }) {
  return <Text>{name} studies {course}</Text>;
}
<Greeting name="Nasirdin" course="React Native" />
```

Props Rules

X Props cannot be changed

X Child cannot modify props

Passed from parent → child

State (Dynamic Data)

State is data that:

- Changes over time
- Affects what is shown on the screen

Examples:

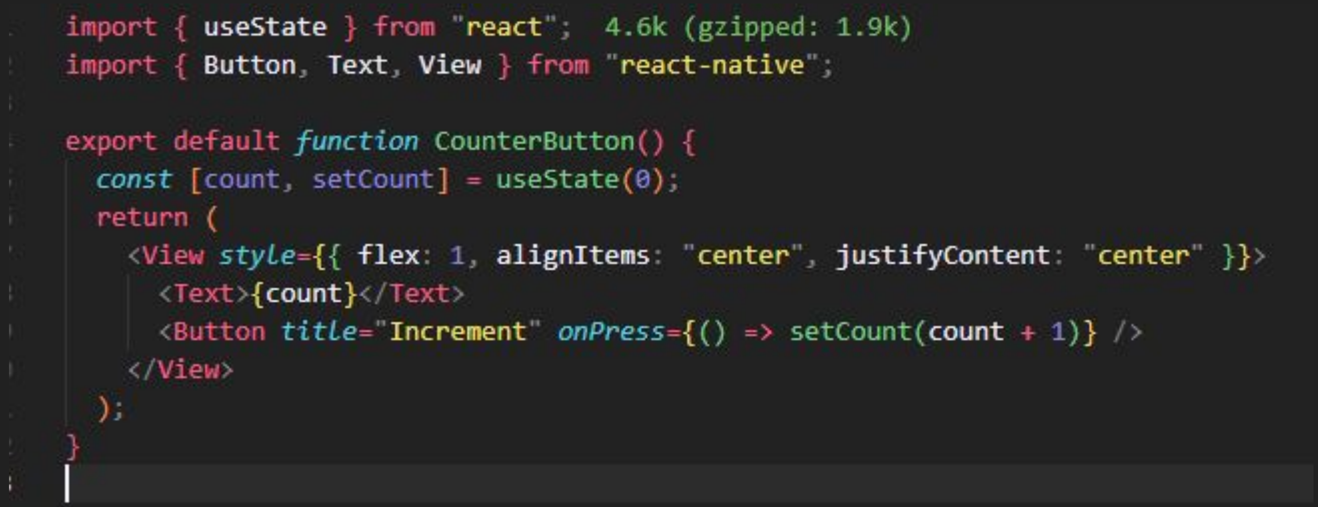
- Counter value
- Input text
- Logged-in user

useState Hook

```
import { useState } from 'react';


const [value, setValue] = useState(initialValue);
```

Simple State Example



```
import { useState } from "react";  4.6k (gzipped: 1.9k)
import { Button, Text, View } from "react-native";

export default function CounterButton() {
  const [count, setCount] = useState(0);
  return (
    <View style={{ flex: 1, alignItems: "center", justifyContent: "center" }}>
      <Text>{count}</Text>
      <Button title="Increment" onPress={() => setCount(count + 1)} />
    </View>
  );
}
```

How it Works?

User clicks button

setCount() updates state

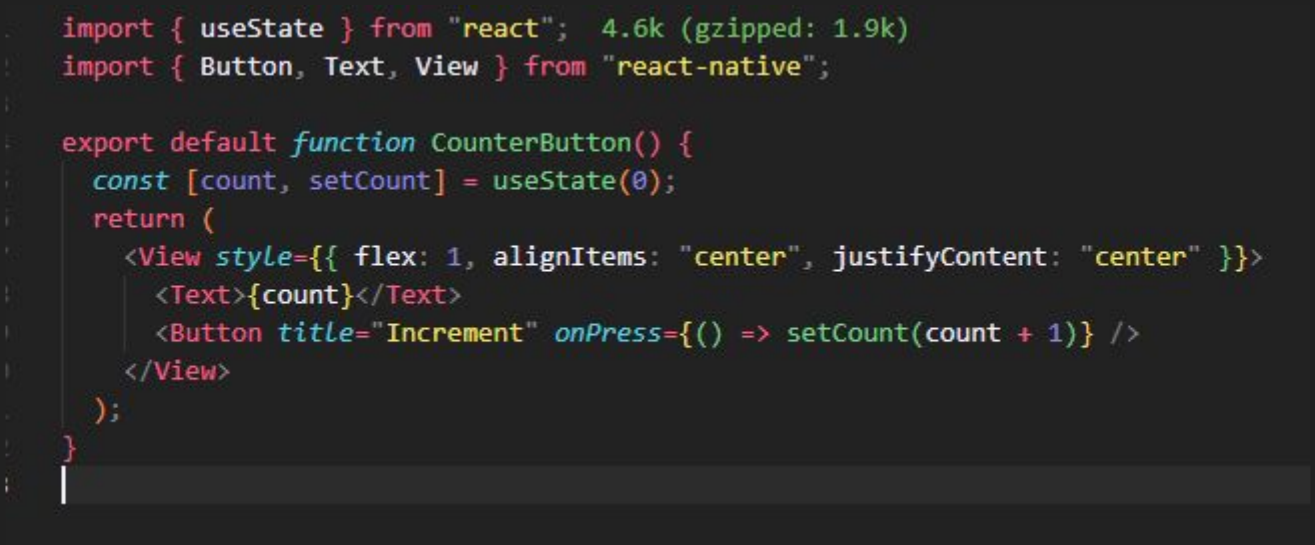
Component re-renders

UI updates automatically

Props vs State (Very Important Slide)

Props	State
Passed from parent	Managed inside component
Read-only	Can be changed
Static	Dynamic
External	Internal

Example: Reusable Counter Component (Convert the following to reusable component)



```
import { useState } from "react";  4.6k (gzipped: 1.9k)
import { Button, Text, View } from "react-native";

export default function CounterButton() {
  const [count, setCount] = useState(0);
  return (
    <View style={{ flex: 1, alignItems: "center", justifyContent: "center" }}>
      <Text>{count}</Text>
      <Button title="Increment" onPress={() => setCount(count + 1)} />
    </View>
  );
}
```

Mini Class Exercise

Task

Create a component FoodCard

Pass props:

- name
- price

Add state:

- button to increase Cart count

Homework Assignment

Create a new component

1. Pass at least 2 props
2. Use useState
3. Add a button that updates the state

Push to GitHub