

Introduction to React & React Native

Welcome Back - Session 2 Overview

Quick recap of Session 1

What we'll cover today:

- What is JSX?
- Functional vs Class Components
- Understanding Props
- Hands-on project: Reusable Button Component

What is JSX?

JSX = JavaScript XML

Allows writing HTML in JavaScript

Makes UI more intuitive to build

Example:

```
const element = <h1>Hello JSX!</h1>;
```

- Must return a single parent element
- Use className instead of class
- Use {} to embed JS expressions

JSX in Practice

```
function GreetUser(props) {  
  return <h2>Hello, {props.name}!</h2>;  
}
```

- Can loop with `map()`
- Conditional rendering: `{condition ? <A /> : }`
- Comment syntax: `{/* comment */}`

Components - The Building Blocks

2 Types:

- Functional (preferred, with Hooks)
- Class (older, less common)

Examples:

```
function Welcome() {  
  return <h1>Welcome!</h1>;  
}  
  
class Welcome extends React.Component {  
  render() {  
    return <h1>Welcome!</h1>;  
  }  
}
```

Understanding Props

Props = Data passed to components

- Like arguments to a function
- Read-only

Example:

```
function Welcome(props) {  
  return <h1>Hello, {props.name}</h1>;  
}
```

```
<Welcome name="Omar" />
```

Props Demo in React Native

```
const Greeting = (props) => (  
  <Text>Hello {props.username}!</Text>  
)  
  
<Greeting username="John" />
```

- Text needs to be inside Text component
- Props work the same as React (Web)

Real-Life Example - Reusable Button

```
function MyButton({ label, onPress }) {  
  return <button onClick={onPress}>{label}</button>;  
}  
  
<MyButton label="Click Me" onPress={() => alert('Hello!')} />
```

React Native:

```
<TouchableOpacity onPress={onPress}>  
  <Text>{label}</Text>  
</TouchableOpacity>
```


Lab Exercise

Goal: Build a reusable button component

- Use props to change button text
- Add a function to handle clicks
- Build both web & mobile versions

Wrap-Up

- Recap: JSX syntax, Components, Props
- What we built: Reusable Button
- Next session: State & Hooks
- Homework: Create a custom card component with title, description, and image props