



React Components & Props

What are Components?

- ▶ Building blocks of a React application
- ▶ Small, reusable UI pieces (Button, Navbar, etc.)
- ▶ Components return JSX (UI)
- ▶ Think of them like Lego blocks

Function Components (Modern & Preferred)

- ▶ • Simple JavaScript functions
- ▶ • Return JSX
- ▶ • Can use Hooks (useState, useEffect)

▶ Example:

```
▶ function Welcome() {  
  ▶   return <h1>Hello, I am a Function  
    Component!</h1>;  
  ▶ }
```

Class Components (Older Method)

- ▶ • ES6 class extending `React.Component`
- ▶ • Use `this.state` and `this.props`
- ▶ • Rarely used in modern projects

- ▶ Example:
- ▶ `class Welcome extends React.Component {`
- ▶ `render() {`
- ▶ `return <h1>Hello, I am a Class Component!</h1>;`
- ▶ `}`
- ▶ `}`

What are Props?

- ▶ • Props = Properties
 - ▶ • Allow data to pass from Parent → Child
 - ▶ • Like function parameters
 - ▶ • Read-only
-
- ▶ Example:
 - ▶ `<Greeting name='Mitesh' />`

Props Example

▶ Parent (App.js):

```
▶ function App() {  
▶   return (  
▶     <div>  
▶       <Greeting  
▶         name='Mitesh' />  
▶       <Greeting  
▶         name='John' />  
▶     </div>  
▶   );  
▶ }
```

▶ Child (Greeting.js):

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

▶ Output:

```
▶ Hello, Mitesh!  
▶ Hello, John!
```

Props with Destructuring

- ▶ Instead of `props.name`, extract directly
- ▶ Example:
- ▶

```
function Greeting({ name }) {
```
- ▶

```
  return <h2>Hello, {name}!</h2>;
```
- ▶

```
}
```

Props with Multiple Data

- ▶ Parent (App.js):

- ▶ `<Student name='Aarav' age={20} course='React' />`
- ▶ `<Student name='Sita' age={22} course='Node.js' />`

- ▶ Child (Student.js):

- ▶ `function Student({ name, age, course }) {`
- ▶ `return (`
- ▶ `<div>`
- ▶ `<h3>Name: {name}</h3>`
- ▶ `<p>Age: {age}</p>`
- ▶ `<p>Course: {course}</p>`
- ▶ `</div>`
- ▶ `);`
- ▶ `}`

- ▶ Output:

- ▶ Name: Aarav, Age: 20, Course: React
- ▶ Name: Sita, Age: 22, Course: Node.js

Summary

- ▶ • Components = Reusable UI blocks
- ▶ • Function Components → modern, preferred
- ▶ • Class Components → older, less used
- ▶ • Props → pass data from Parent to Child (read-only)