

4. Component Composition in React

Last updated by | Subramanya Dixit | May 5, 2025 at 7:59 PM GMT+5:30

Why Component Composition is Important

Component Composition allows developers to build UIs by assembling small, reusable components. It aligns with React's core philosophy of modularity and declarative design, making large applications easier to develop and maintain.

Key Concepts

Children Prop

Used to pass elements/components between opening and closing tags.

```
function Wrapper({ children }) {  
  return <div className="box">{children}</div>;  
}  
  
<Wrapper>  
  <p>Hello from inside!</p>  
</Wrapper>
```



Props Composition

You can customize component behavior by passing props.

```
function Button({ text, onClick }) {  
  return <button onClick={onClick}>{text}</button>;  
}  
  
<Button text="Click Me" onClick={() => alert('Clicked!')} />
```



Component Nesting

Combine multiple components to build complex UIs.

```
function Header() {  
  return <h1>Welcome</h1>;  
}  
  
function Layout() {  
  return (  
    <div>  
      <Header />  
      <Content />  
    </div>  
  );  
}
```



Higher-Order Components (HOC)

A function that takes a component and returns a new component.

```
function withLogger(WrappedComponent) {
  return function Logger(props) {
    console.log('Rendering', WrappedComponent.name);
    return <WrappedComponent {...props} />;
  };
}
```



✓ Render Props

Share code using a function as a child.

```
function MouseTracker({ render }) {
  const [position, setPosition] = useState({ x: 0, y: 0 });
  return <div onMouseMove={(e) => setPosition({ x: e.clientX, y: e.clientY })}>{render(position)}</div>;
}

<MouseTracker render={({ x, y }) => <p>{x}, {y}</p>} />
```



Visual Overview

```
flowchart TD
  A[App Component] --> B[Layout Component]
  B --> C[Header Component]
  B --> D[Content Component]
  D --> E[Button Component with Props]
```



Guidelines

- Design small, single-purpose components.
- Use composition instead of inheritance.
- Favor children props for layout flexibility.
- Use HOCs and render props sparingly and clearly.

Practice Exercises

1. Create a `Card` component that wraps `children` in a styled `div`.
2. Compose a `Header`, `Body`, and `Footer` into a single `Layout` component.
3. Write a Higher-Order Component that logs when a component mounts.
4. Use the render prop pattern to build a flexible mouse position tracker.

Quiz Questions

1. What is the `children` prop used for?

- a) To track mouse position
- b) To apply global styles

☒ **c) To pass nested components**

d) To manage context

2. What does component composition help with?

a) Avoiding JSX

b) Global variables

☒ **c) Reusability and modular design**

d) Styling only

3. Which concept lets you pass behavior into a component?

a) useEffect

b) children prop

☒ **c) Render props**

d) JSX only

4. A Higher-Order Component is...

a) A custom React Hook

b) A JSX element

c) A way to style components

☒ **d) A function that returns a component**
