

Managing State in Class Components



Cory House

REACT CONSULTANT AND TRAINER

@housecor reactjsconsulting.com



Agenda



Why prefer functions?

So why are classes important?

Convert Checkout and Detail to a class

- Declaring, deriving, and setting state
- Avoiding `this` binding issues
- Sharing logic and consuming hooks
- Consuming context



Class Components vs Function Components



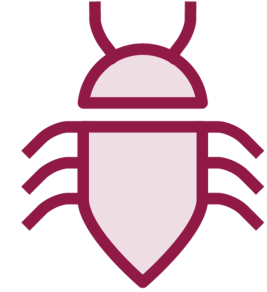
Why Prefer Function Components?



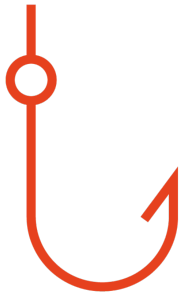
Less code



Simpler mental model



No this keyword



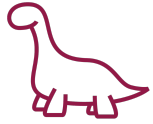
Hooks



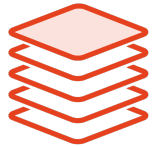
Functions are the
future of React



Why Learn React Class Components?



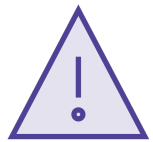
In React 16.7 or older, only classes support state



Existing classes in many apps



Existing blog posts and forum answers often use classes



Error Boundaries, `getSnapshotBeforeUpdate` require classes



Some prefer classes



Demo



Convert Checkout, Detail to class components

- Consume a hook in a class
- Sharing logic between classes
- Declare state
- Handle derived state
- Setting state
- Binding to `this`
- Consuming context



Ways to Consume a Hook with a Class



Convert class to a function 😎



Pass hook data/funcs down from parent via props



Wrap the Hook

1. Custom wrapper
2. Render prop
3. Function as child



Or Share Logic

Ways to Consume a Hook ^ with a Class



Convert class to a function 😎



Pass hook data/funcs down from parent via props



Wrap the Hook

1. Custom wrapper
2. Render prop
3. Function as child



Summary



Function component benefits

- Less code, simpler model, hooks

Classes are relevant too

- Many existing codebases, blog posts, and forums, and some prefer classes

Share logic and consume hooks

- Pass via props, custom wrapper, render props, function as child

Use class field to omit constructor

Arrow funcs avoid `this` binding issues

Directly consuming context

- Static for 1, Render prop for multiple

