

# Conditional Rendering and Shared State

---



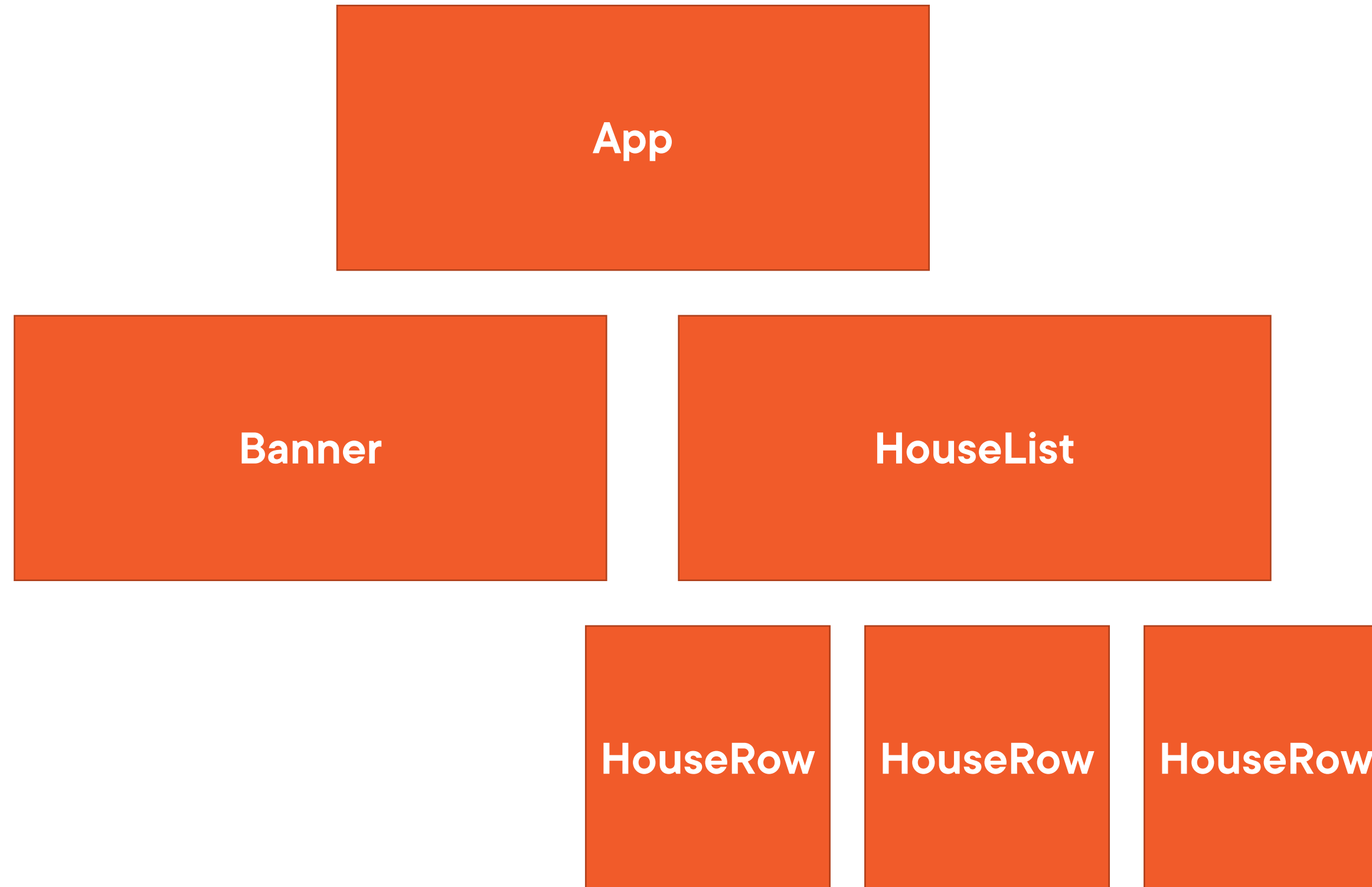
**Roland Guijt**

Freelance consultant and trainer

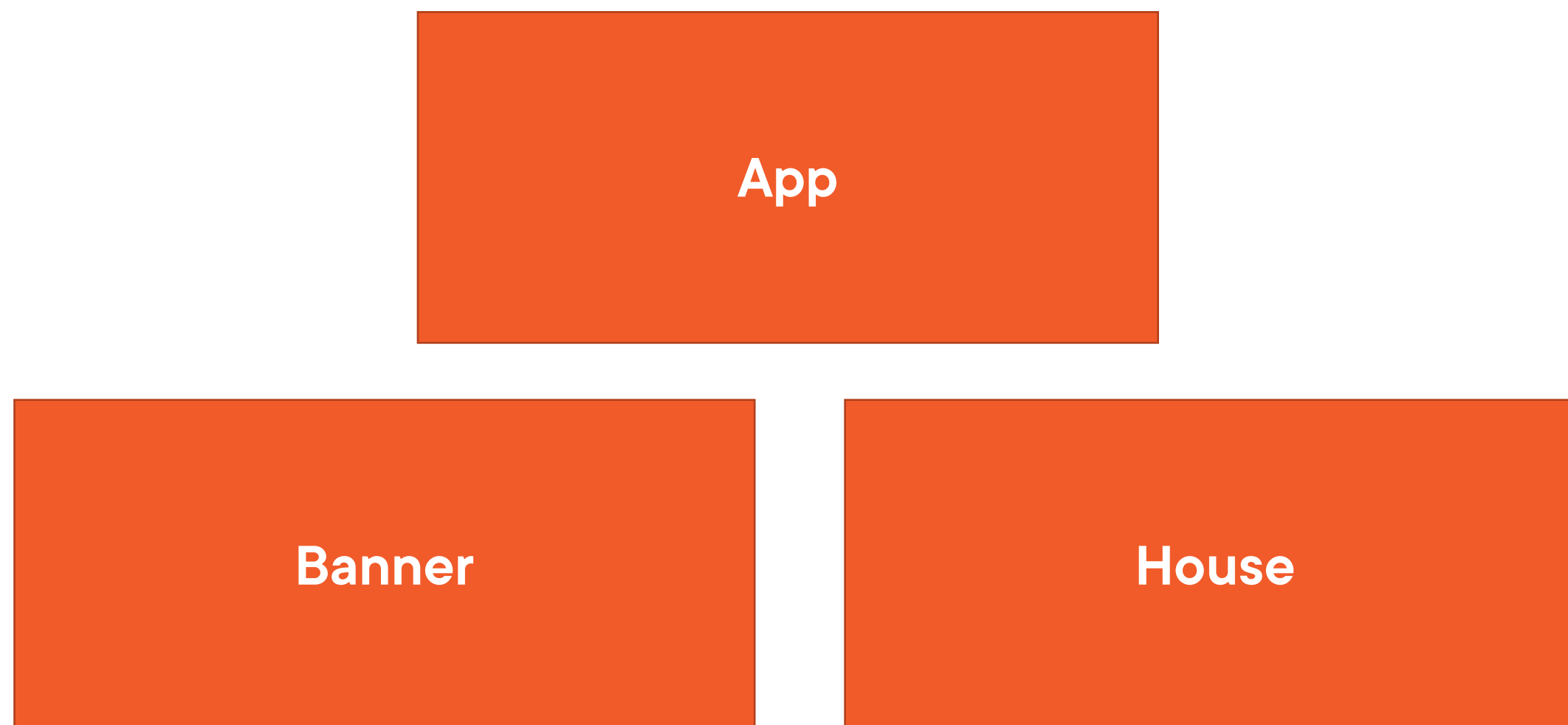
@rolandguijt roland.guijt@gmail.com



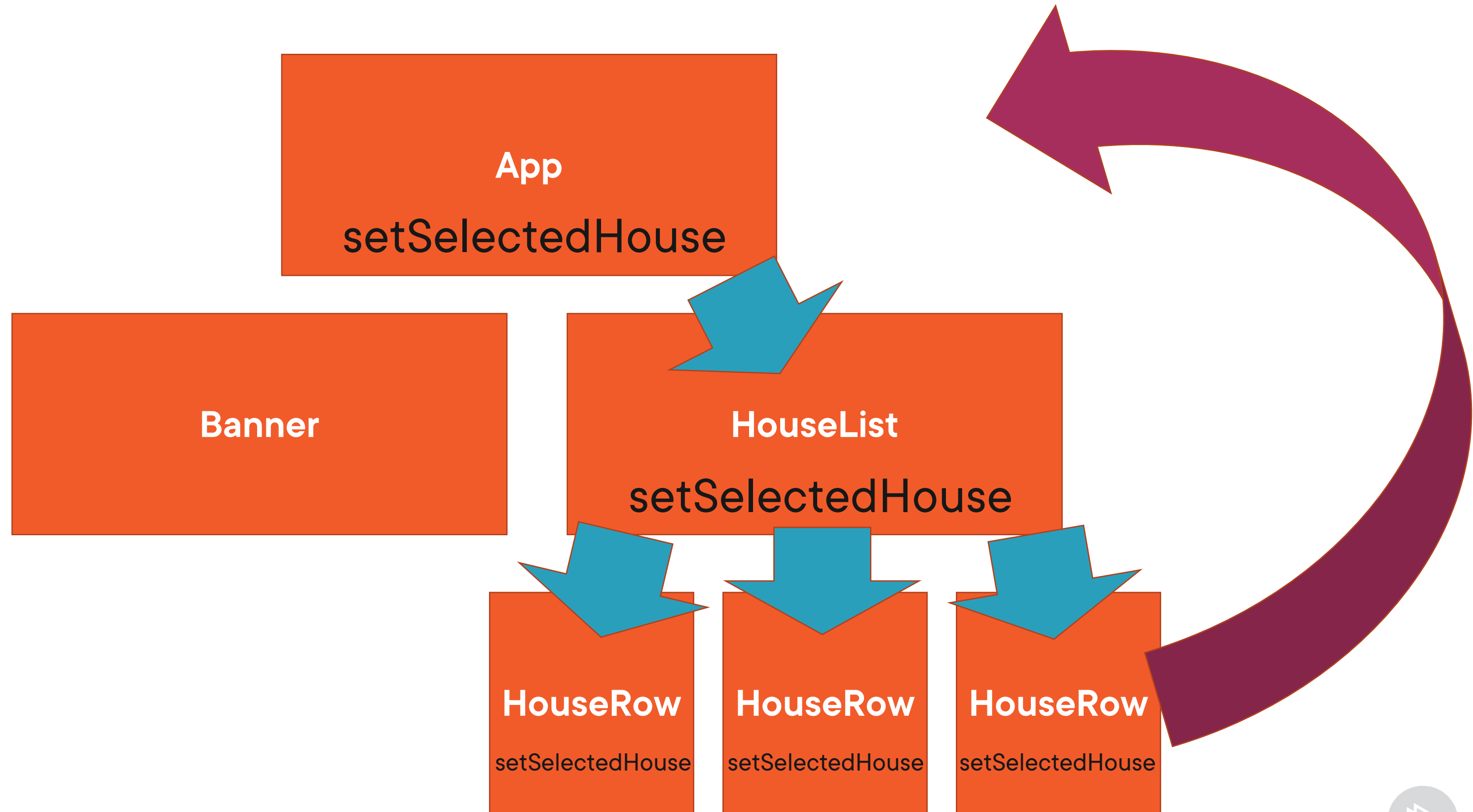
# Composition 1



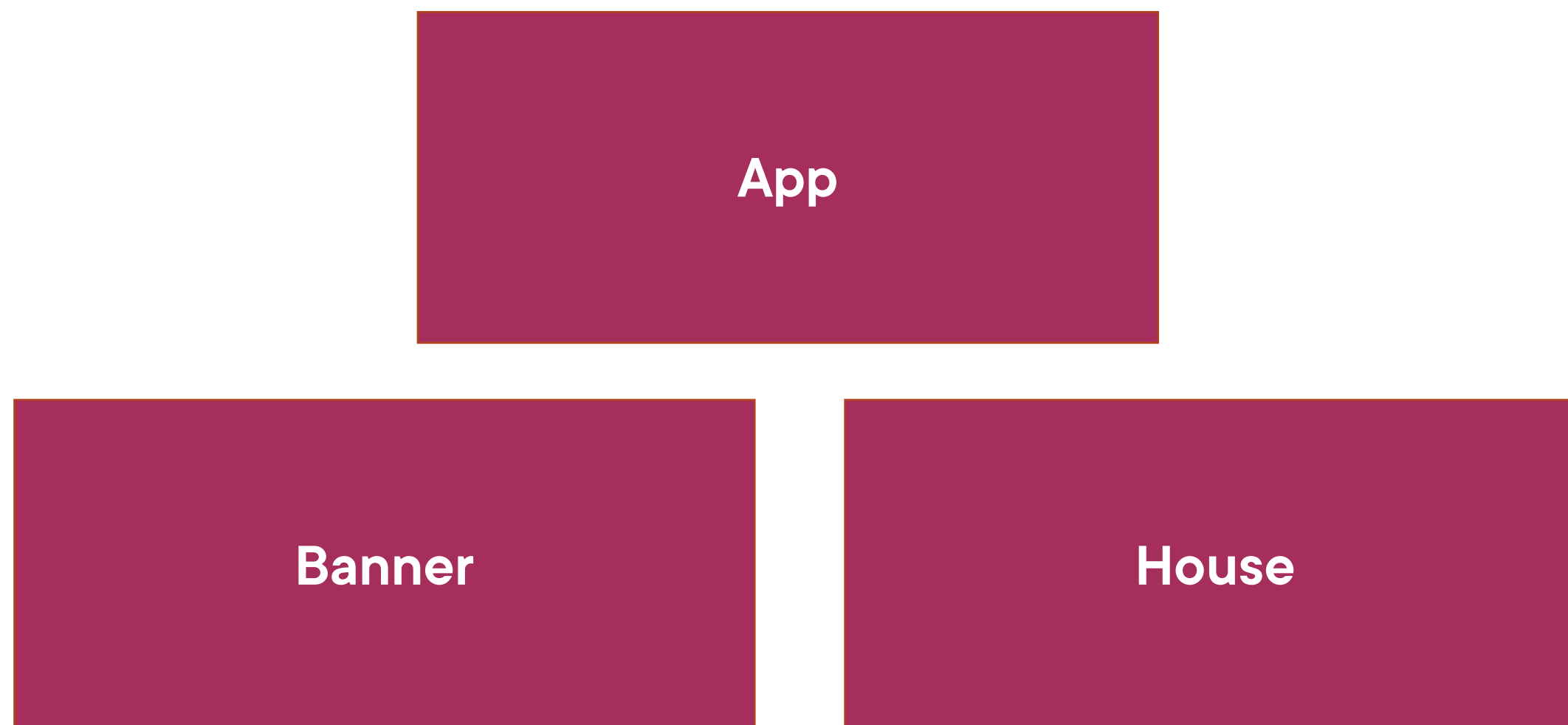
# Composition 2



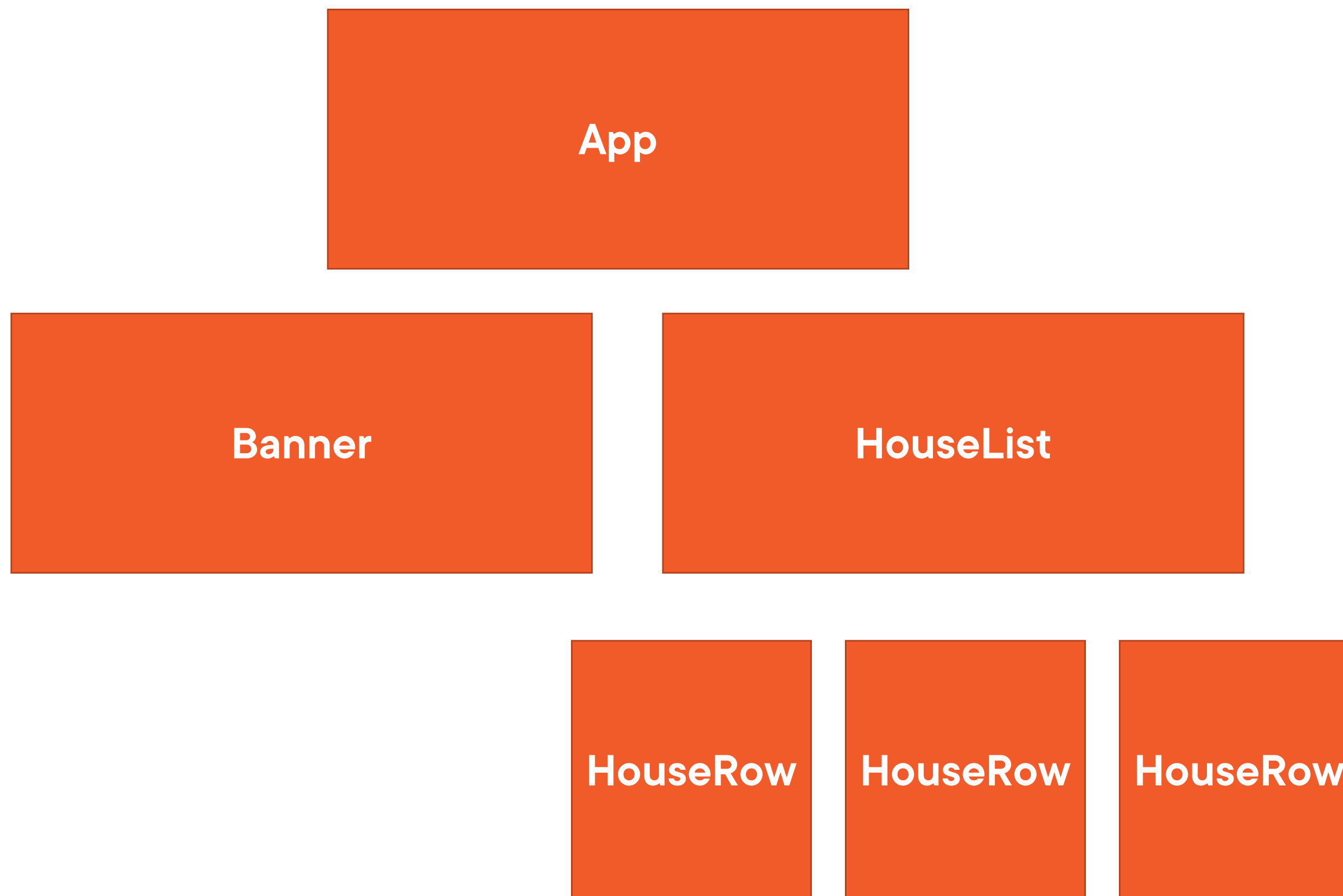
# Passing In Functions as Props



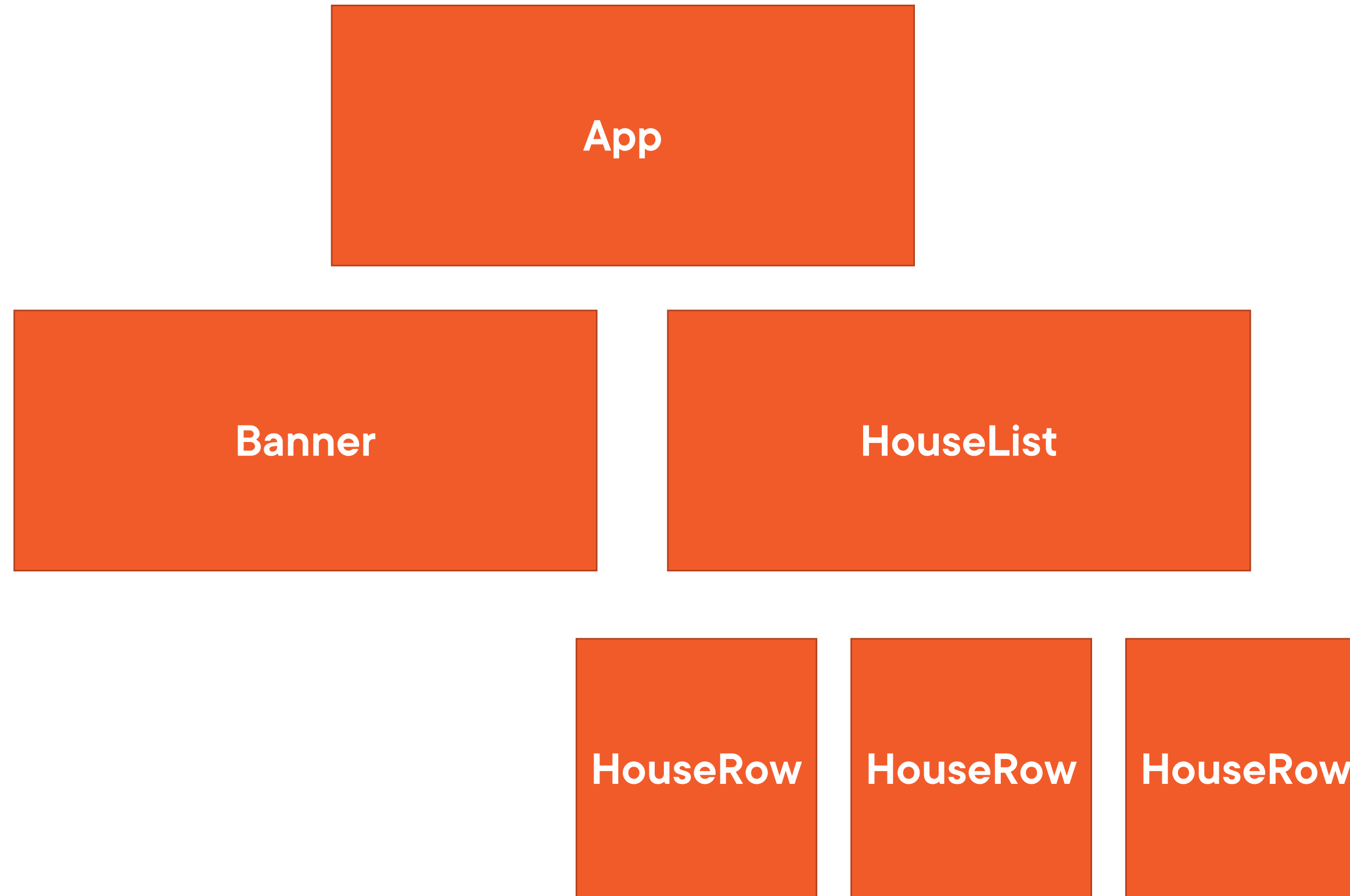
# Composition 2



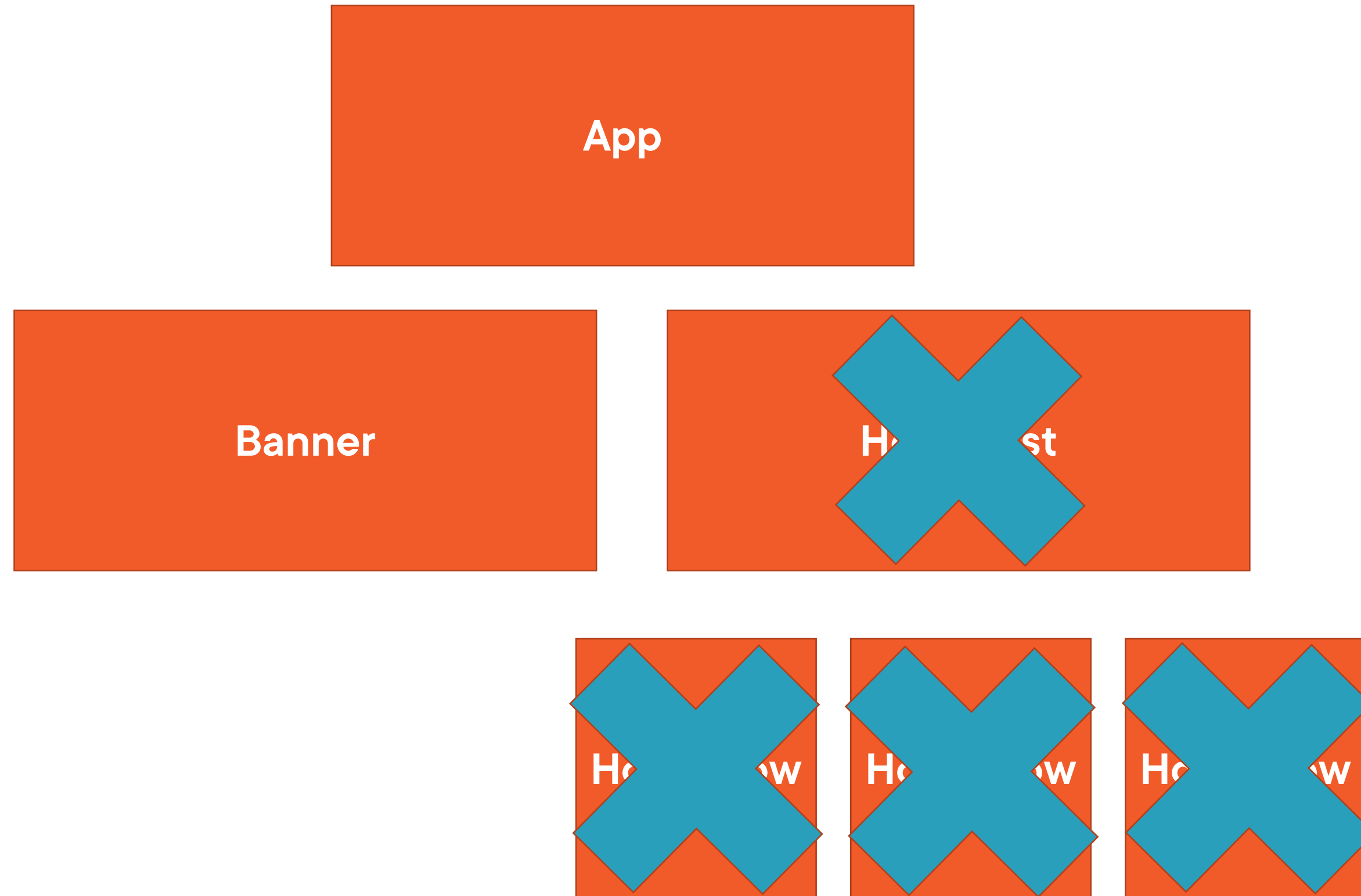
# State Positioning



# Mounting

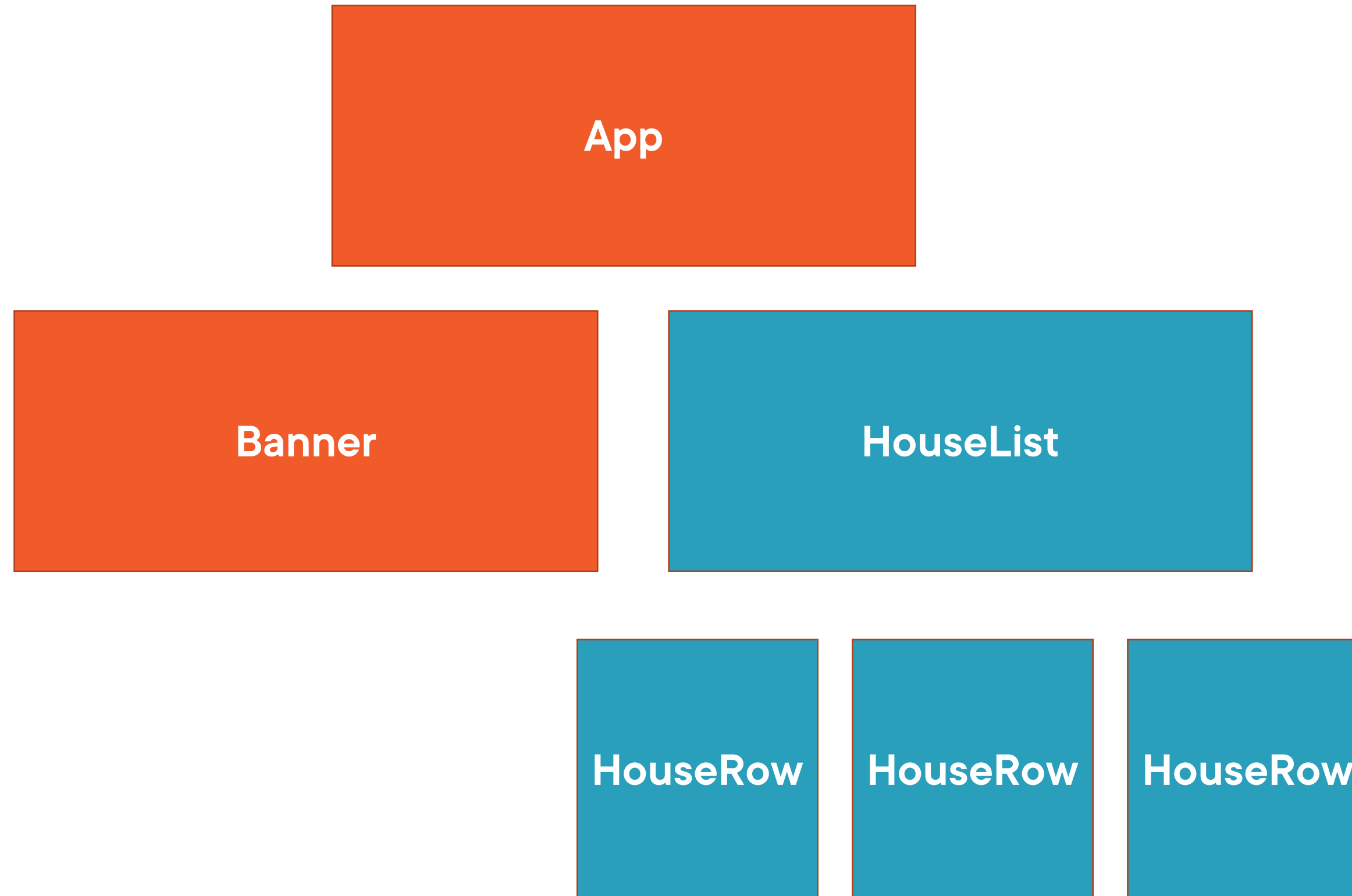


# Unmounting





# Mounting



# Recreating Functions

**Function object is recreated**

**New reference**

**Watch out with memoized components**

**Or when the function is in a dependency array**

**Wrap with useCallback if needed**

**Memoizes the contained function**

**Comes with overhead**



# The Callback Hook

```
const setHousesWrapper =  
  useCallback((house) => {  
    setSelectedHouse(house);  
  }, []);
```



# The Callback Hook

```
const setHousesWrapper =  
  useCallback((house) => {  
    setSelectedHouse(house);  
  }, []);
```



# Custom Hooks

**Function**

**When returned state changes, the component that uses it rerenders**

**For separation of concerns**

**And reusability**

**Isolated state**



# Conditional Rendering

**Every JavaScript conditional statement or expression can be used in JSX**

**if statement**

**? :**

**&&**

**As well as: early returns from a component**



# State Sharing

**Using a prop**

**Functions as a prop**

**From a custom hook**



Next up:  
Context and Navigation

