



Let's break React's rules

Introduction

The fundamental rules of hooks:

Only call Hooks at the top level

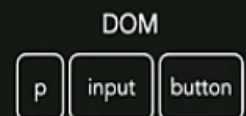
Only use Hooks from React functions

Only call Hooks at the top level

```
root.render(<Recipe />)

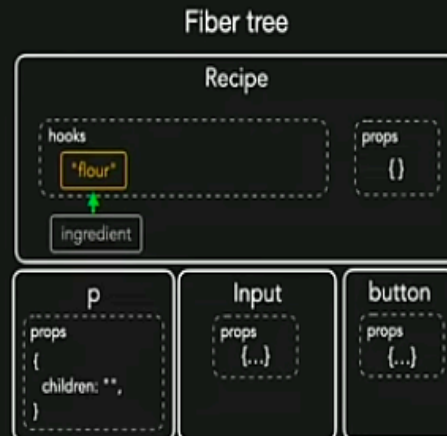
const Recipe = () => {
  const [ingredient, setIngredient] = useState(null)
  if (ingredient) {
    const [quantity, setQuantity] = useState(0)
  }
  const [needsPurchase, setNeedsPurchase] = useState(false)

  return (
    <p>{ingredient ? `${ingredient}` : ""}</p>
    <input type="checkbox" name="needsPurchase" checked={needsPurchase}/>
    <button onClick={() => setIngredient('flour')}>
      Add ingredient
    </button>
  )
}
```



Components and Hooks must be pure - Do not mutate hooks values

```
const Recipe = () => {  
  const [ingredients, setIngredients] = useState([]);  
  
  return (  
    <p>{ingredient ? `${ingredient}` : ""}</p>  
    <button  
      onClick={() => {  
        ingredients.push("flour");  
      }}  
    >  
      Add ingredient  
    </button>  
  );  
};
```



Do not use setState unconditionally in render

```
const Recipe = () => {  
  const [ingredient, setIngredient] = useState(null);  
  
  if(someCondition){  
    setIngredient("flour")  
  }  
  
  return (  
    <p>{ingredient ? `${ingredient}` : ""}</p>  
    <button onClick={() => setIngredient("flour")}>Add ingredient</button>  
  );  
};
```



Only use Hooks from React functions

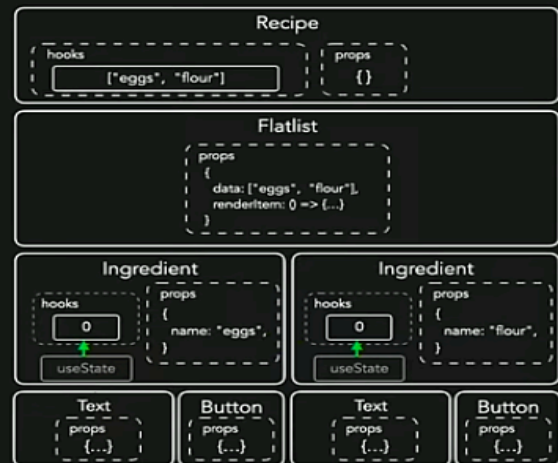
```
const Recipe = () => {
  const [ingredients, setIngredients] = useState(['eggs', 'flour']);

  return <FlatList data={ingredients} renderItem={renderIngredient} />;
};

const renderIngredient = ({ item }) => <Ingredient name={item} />;

const Ingredient = ({ name }) => {
  const [quantity, setQuantity] = useState(0);
  const increment = () => setQuantity(quantity + 1);

  return (
    <Text>`${quantity} ${name}</Text>
    <Button onPress={increment} title="Increment" />
  );
};
```



Only use Hooks from React functions

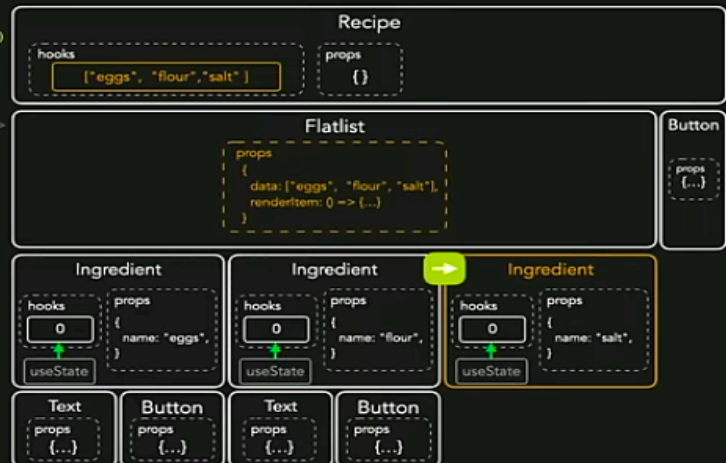
```
const Recipe = () => {
  const [ingredients, setIngredients] = useState(['eggs', 'flour']);
  const addIngredient = () => setIngredients(i => [...i, 'salt']);

  return (
    <FlatList data={ingredients} renderItem={renderIngredient} />
    <Button onPress={addIngredient} title="Add Ingredient" />
  );
};

const renderIngredient = ({ item }) => <Ingredient value={item} />;

const Ingredient = ({ name }) => {
  const [quantity, setQuantity] = useState(0);
  const increment = () => setQuantity(quantity + 1);

  return (
    <Text>{quantity} {name}</Text>
    <Button onPress={increment} title="Increment" />
  );
};
```



Conclusion

Only call Hooks at the top level

Only use Hooks from React functions