

REACT PROPS



REACT PROP PATTERNS



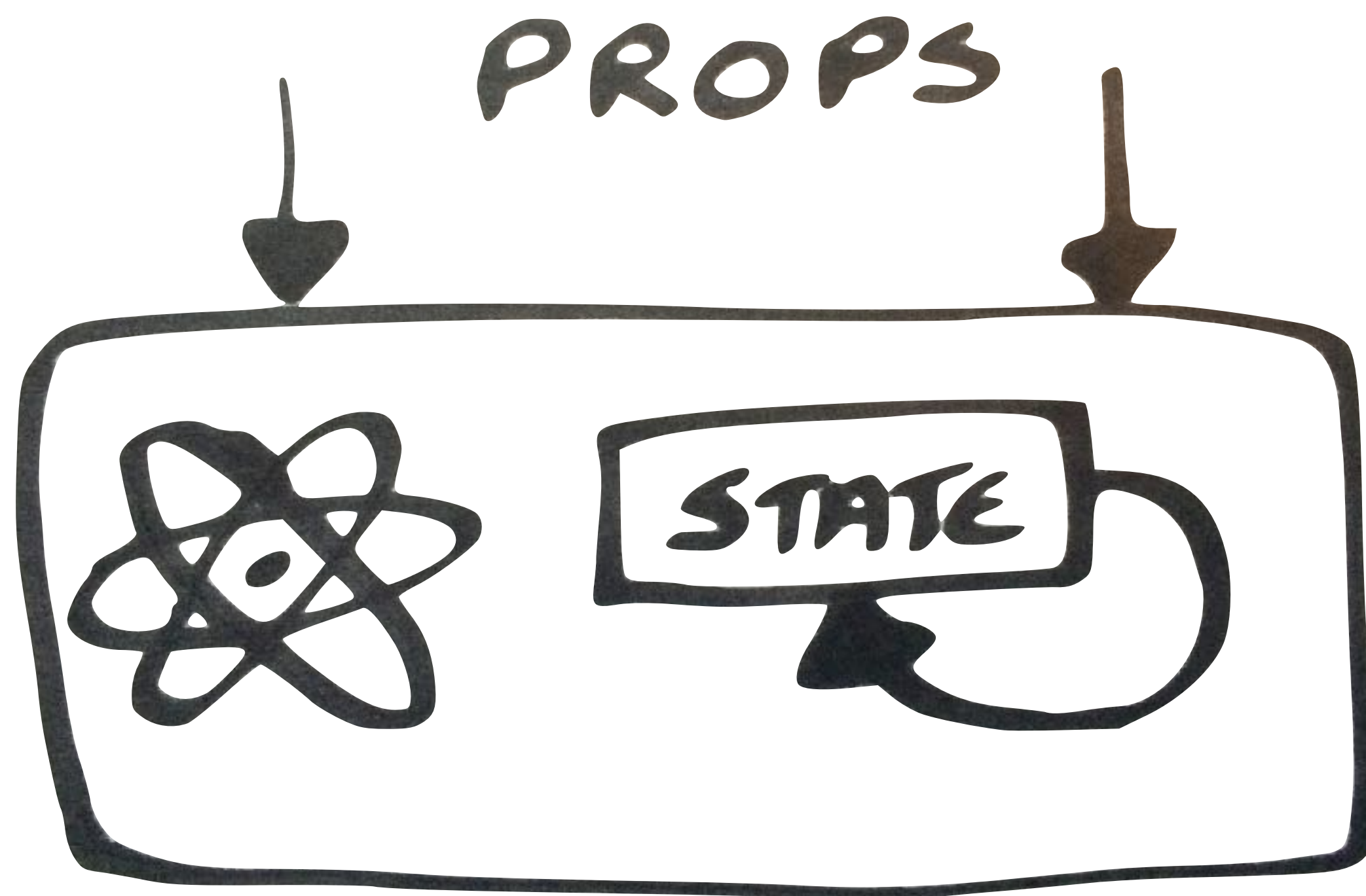
REACT PROP PATTERNS

For Component Composition

COMPONENTS

- **Self-contained, concern-specific building blocks**
- **Components:**
 - are reusable
 - are composable
 - lead to separation of concerns
- **Allows a “Divide and conquer” approach to development**

PROPS VS STATE



- Think of Props as components attributes, or “configuration”. They are received from parent components.
- A component manages its own state internally. State usually starts with a default value defined in the component’s constructor and then suffers from mutations in time (in. eg. generated from user events).

CONTENTS

- **Props and Component Composition**
- **Default Props**
- **Children prop**
- **React.Children API**

PROPS AND COMPONENT COMPOSITION

PROPS & COMPONENT COMPOSITION

- **Composing components is an essential pattern in React for building complex UIs and applications.**
- **Props are of key importance in component composition. They are the mechanism used in React for passing data from parent to child components.**
- **Props are Read-Only: They can't be changed from inside. they are passed and “owned” by the parent component.**

PROPS & COMPONENT COMPOSITION

```
function TodoApp() {  
  return (  
    <div>  
      <div className="todo-item"><input type="checkbox" checked={true}/>  
        <p>Buy Milk</p>  
      </div>  
      <div className="todo-item"><input type="checkbox" checked={false}/>  
        <p>Buy Cookies</p>  
      </div>  
      <div className="todo-item"><input type="checkbox" checked={false}/>  
        <p>Learn React</p>  
      </div>  
    </div>  
  );  
}
```

PROPS & COMPONENT COMPOSITION

```
function TodoItem({checked, children}) {  
  return <div className="todo-item">  
    <input type="checkbox" checked={checked}/><p>{children}</p></div>;  
}  
  
function TodoApp() {  
  return (  
    <div>  
      <TodoItem checked={true} children="Buy Milk" />  
      <TodoItem checked={false} children="Buy Cookies" />  
      <TodoItem checked={false} children="Learn React" />  
    </div>  
  );  
}
```

DEFAULT PROPS

DEFAULT PROPS

- **You can define default values for your props by assigning to the special `defaultProps` property:**
- **Used to ensure that a given prop will have a value if it was not specified by the parent component.**

DEFAULT PROPS

```
function TodoItem({checked, children}) {  
  return <div className="todo-item">  
    <input type="checkbox" checked={checked}/><p>{children}</p></div>;  
}  
  
function TodoApp() {  
  return (  
    <div>  
      <TodoItem checked={true} children="Buy Milk" />  
      <TodoItem checked={false} children="Buy Cookies" />  
      <TodoItem checked={false} children="Learn React" />  
    </div>  
  );  
}
```

DEFAULT PROPS

```
function TodoItem({checked, children}) {  
  return <div className="todo-item">  
    <input type="checkbox" checked={checked}/><p>{children}</p></div>;  
}  
TodoItem.defaultProps = {  
  checked: false  
};  
  
function TodoApp() {  
  return (  
    <div>  
      <TodoItem checked={true} children="Buy Milk" />  
      <TodoItem checked={false} children="Buy Cookies" />  
      <TodoItem checked={false} children="Learn React" />  
    </div>  
  );  
}
```

DEFAULT PROPS

```
function TodoItem({checked, children}) {  
  return <div className="todo-item">  
    <input type="checkbox" checked={checked}/><p>{children}</p></div>;  
}  
TodoItem.defaultProps = {  
  checked: false  
};  
  
function TodoApp() {  
  return (  
    <div>  
      <TodoItem checked={true} children="Buy Milk" />  
      <TodoItem children="Buy Cookies" />  
      <TodoItem children="Learn React" />  
    </div>  
  );  
}
```

CHILDREN PROP

PROPS.CHILDREN

- **props.children is a special property:**
 - Automatically assigned to the content between the opening and closing tags

PROPS.CHILDREN

```
function TodoItem({checked, children}) {  
  return <div className="todo-item">  
    <input type="checkbox" checked={checked}/><p>{children}</p></div>;  
}  
TodoItem.defaultProps = {  
  checked: false  
};  
  
function TodoApp() {  
  return (  
    <div>  
      <TodoItem checked={true} children="Buy Milk" />  
      <TodoItem children="Buy Cookies" />  
      <TodoItem children="Learn React" />  
    </div>  
  );  
}
```

PROPS.CHILDREN

```
function TodoItem({checked, children}) {  
  return <div className="todo-item">  
    <input type="checkbox" checked={checked}/><p>{children}</p></div>;  
}  
TodoItem.defaultProps = {  
  checked: false  
};  
  
function TodoApp() {  
  return (  
    <div>  
      <TodoItem checked={true}>Buy Milk</TodoItem>  
      <TodoItem>Buy Cookies</TodoItem>  
      <TodoItem>Learn React</TodoItem>  
    </div>  
  );  
}
```

PROPS.CHILDREN

PROPS.CHILDREN

- **props.children is a special property:**
 - Automatically assigned to the content between the opening and closing tags

PROPS.CHILDREN

- **props.children is a special property:**
 - Automatically assigned to the content between the opening and closing tags
 - Usually used with components that don't know what JSX they will render ahead of time. (especially common for components like Sidebar or Dialog that represent generic “boxes”)

PROPS.CHILDREN

- **props.children is a special property:**
 - Automatically assigned to the content between the opening and closing tags
 - Usually used with components that don't know what JSX they will render ahead of time. (especially common for components like Sidebar or Dialog that represent generic “boxes”)
 - Can have one element, multiple elements, or none at all

PROPS.CHILDREN

```
function FancyBorder(props) {  
  return (  
    <div className={'FancyBorder FancyBorder-' + props.color}>  
      {props.children}  
    </div>  
  );  
}
```


PROPS.CHILDREN

```
function FancyBorder(props) {  
  return (  
    <div className={'FancyBorder FancyBorder-' + props.color}>  
      {props.children}  
    </div>  
  );  
}  
  
function WelcomeDialog() {  
  return (  
    <FancyBorder color="blue">  
      <h1>Welcome</h1>  
      <p>  
        Thank you for visiting our spacecraft!  
      </p>  
    </FancyBorder>  
  );  
}
```

REACT'S CHILDREN API

REACT.CHILDREN

- **Top-Level React API**
- **Provides utilities for dealing with the `props.children`**
 - map, count etc...