ReactJS basic concepts

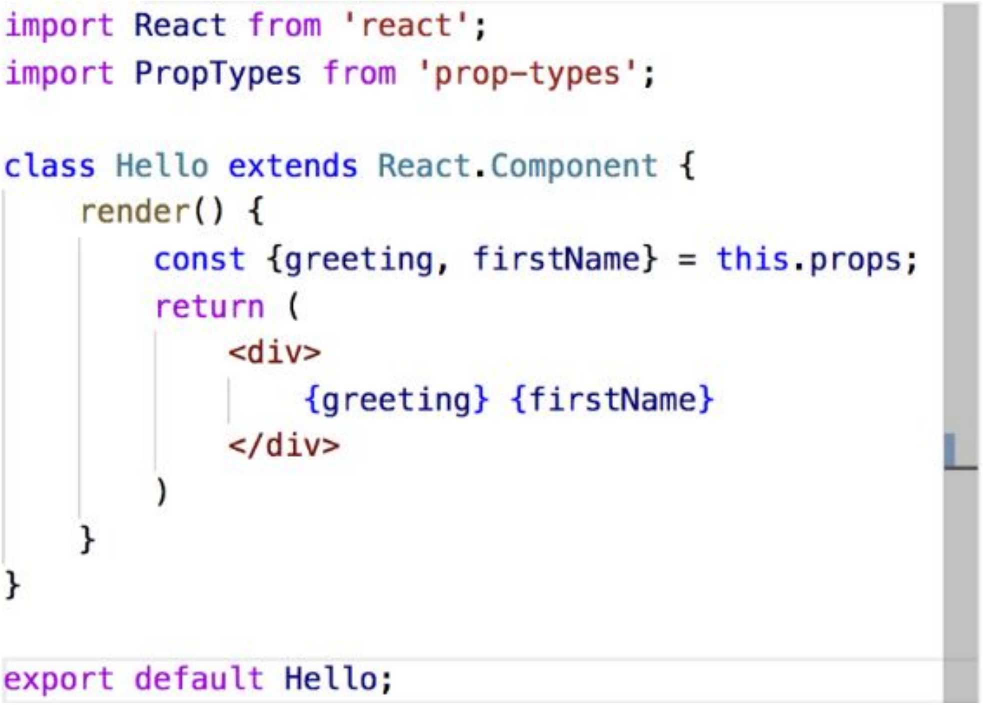
Presented by: Hau Nguyen

NORDIC
CODER

Agenda

- Functional component
- Class component
- Props vs State
- Handle events
- Lab: Simple Todo App
- Q & A

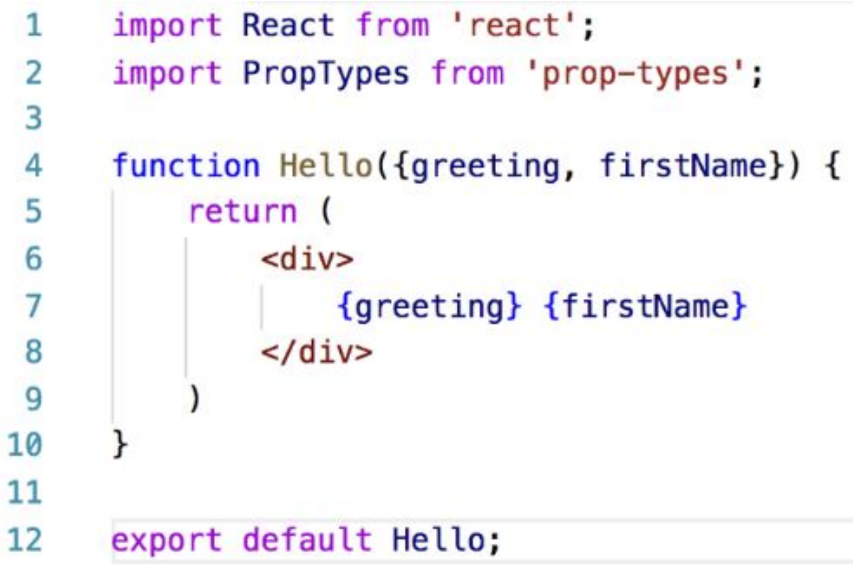
Functional vs Class component



```
import React from 'react';
import PropTypes from 'prop-types';

class Hello extends React.Component {
    render() {
        const {greeting, firstName} = this.props;
        return (
            <div>
                {greeting} {firstName}
            </div>
        )
    }
}

export default Hello;
```

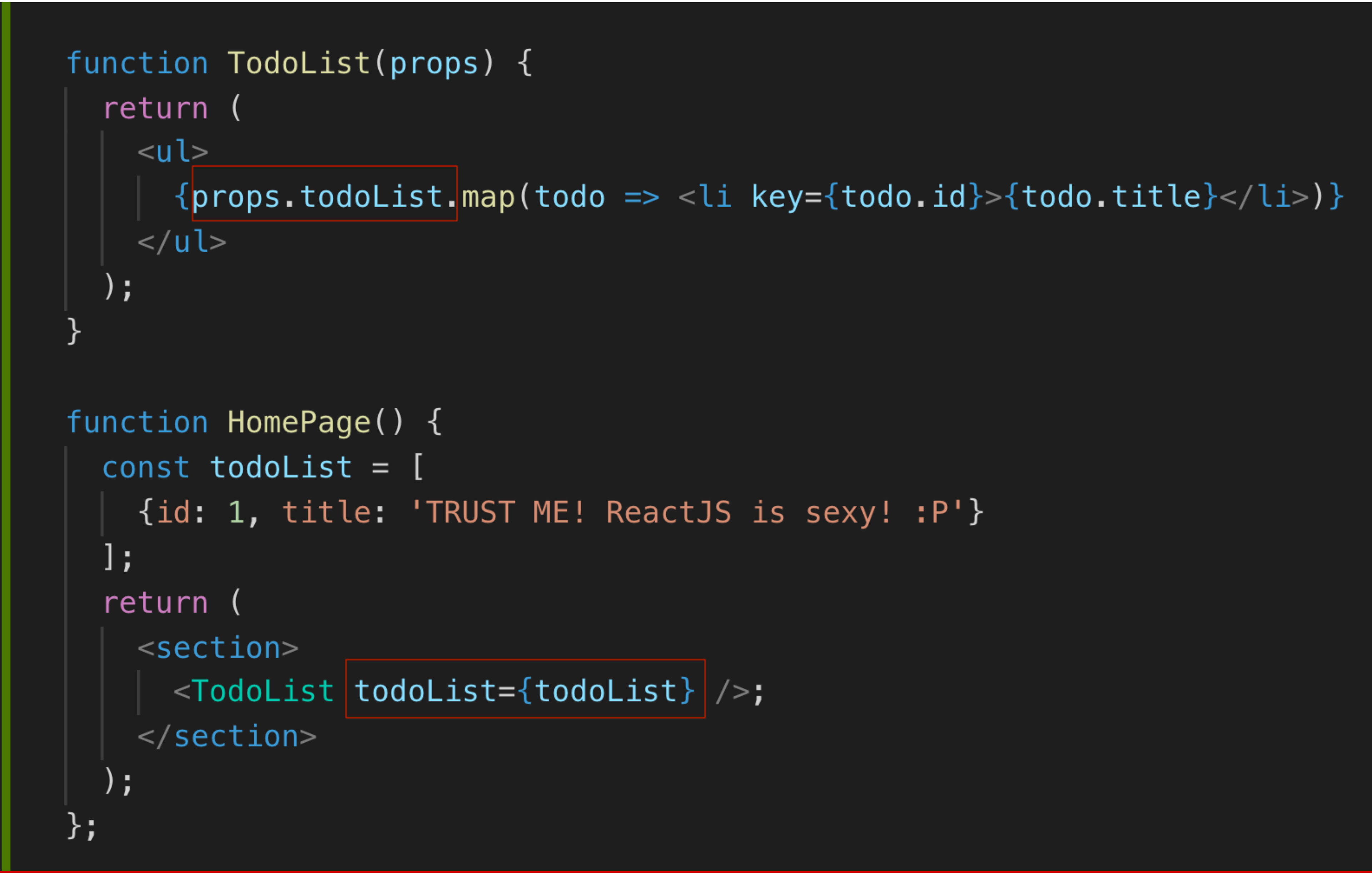


```
import React from 'react';
import PropTypes from 'prop-types';

function Hello({greeting, firstName}) {
    return (
        <div>
            {greeting} {firstName}
        </div>
    )
}

export default Hello;
```

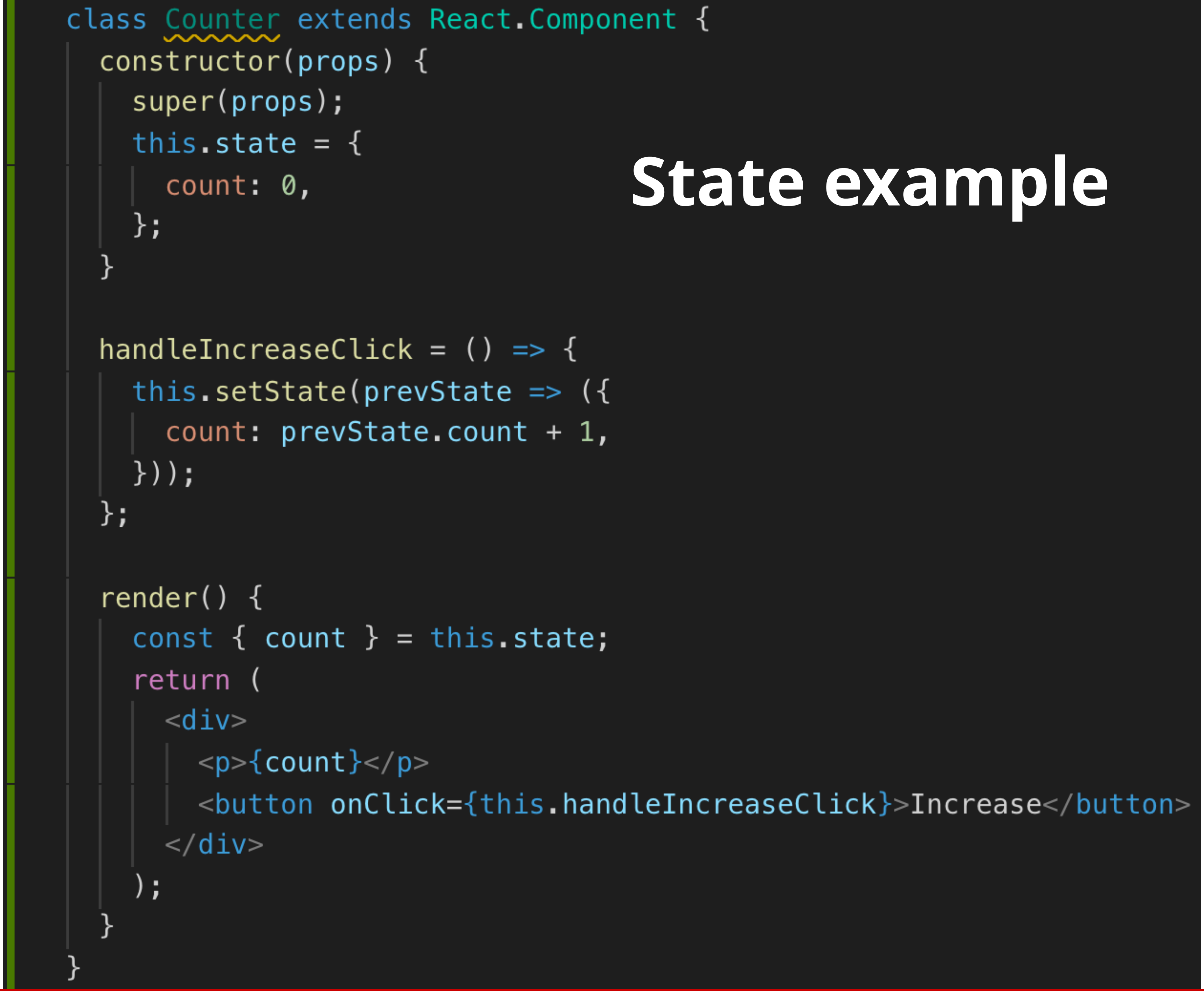
Props example



```
function TodoList(props) {
  return (
    <ul>
      {props.todoList.map(todo => <li key={todo.id}>{todo.title}</li>)}
    </ul>
  );
}

function HomePage() {
  const todoList = [
    {id: 1, title: 'TRUST ME! ReactJS is sexy! :P'}
  ];
  return (
    <section>
      <TodoList todoList={todoList} />;
    </section>
  );
};
```

State example

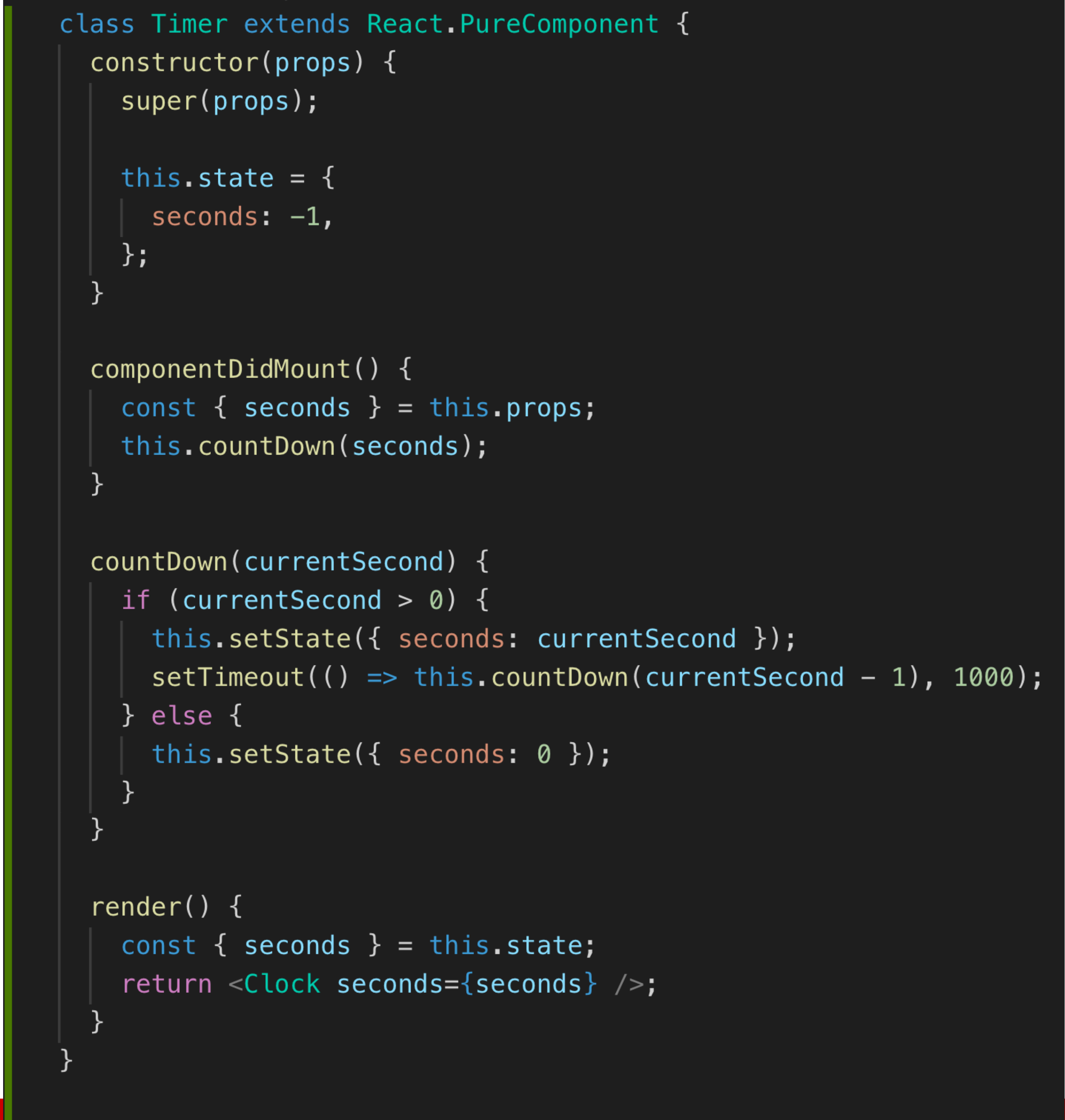


```
class Counter extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      count: 0,
    };
  }

  handleIncreaseClick = () => {
    this.setState(prevState => ({
      count: prevState.count + 1,
    }));
  };

  render() {
    const { count } = this.state;
    return (
      <div>
        <p>{count}</p>
        <button onClick={this.handleIncreaseClick}>Increase</button>
      </div>
    );
  }
}
```

State example



```
class Timer extends React.PureComponent {
  constructor(props) {
    super(props);

    this.state = {
      seconds: -1,
    };
  }

  componentDidMount() {
    const { seconds } = this.props;
    this.countDown(seconds);
  }

  countDown(currentSecond) {
    if (currentSecond > 0) {
      this.setState({ seconds: currentSecond });
      setTimeout(() => this.countDown(currentSecond - 1), 1000);
    } else {
      this.setState({ seconds: 0 });
    }
  }

  render() {
    const { seconds } = this.state;
    return <Clock seconds={seconds} />;
  }
}
```

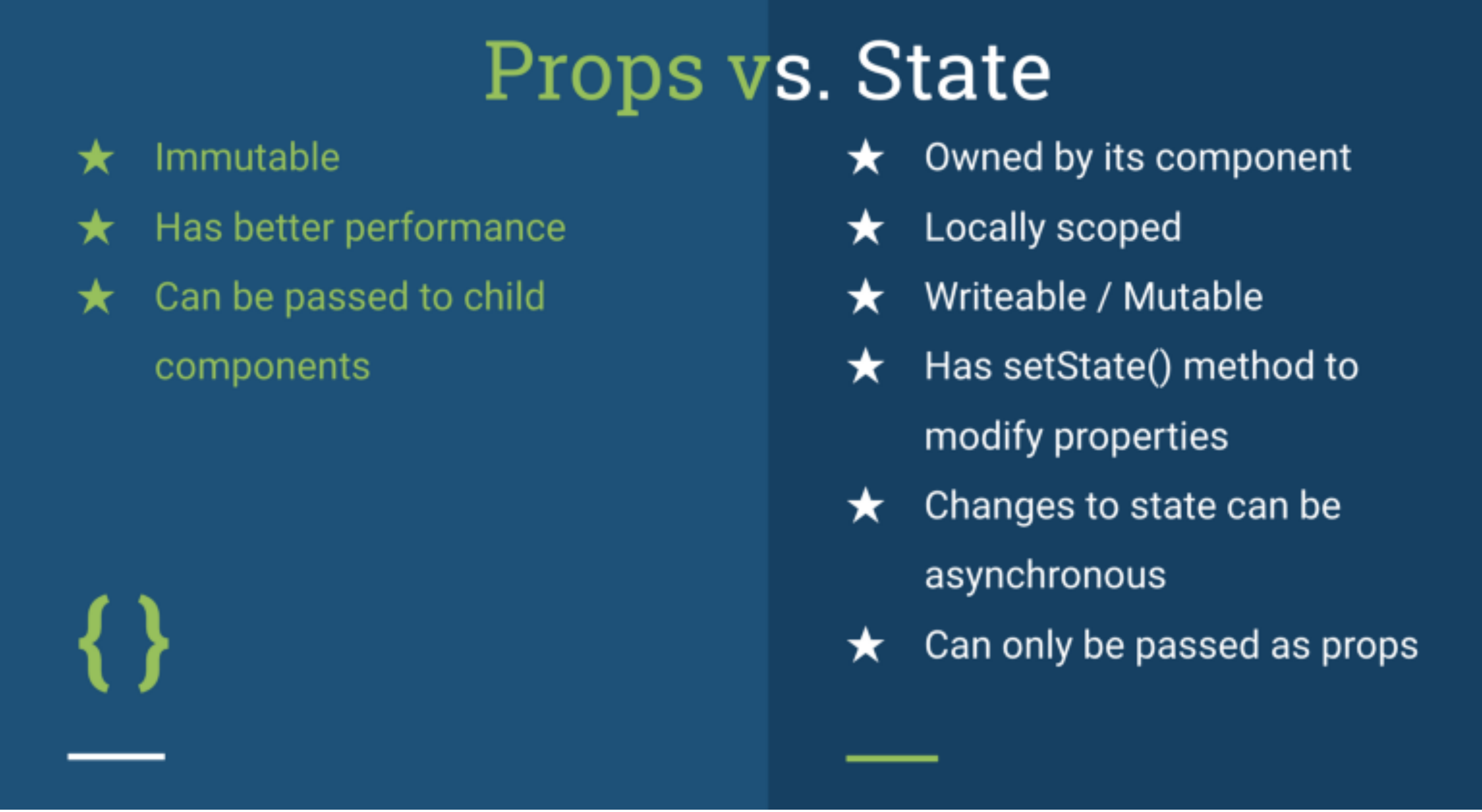
React

Functional Component vs Class Component

Functional vs Class

- Receive parameter (Props) - Optional
- **Stateless** or **dumb** component.
- Just Plain old JavaScript functions.
- Shorter to write
- For UI Components

- Has local State
- Receive parameter (Props) - Optional
- **Statefull** or **smart** component
- Has Lifecycle hooks.
- Can Handles fetching data via ajax calls



Props vs. State

- ★ Immutable
- ★ Has better performance
- ★ Can be passed to child components

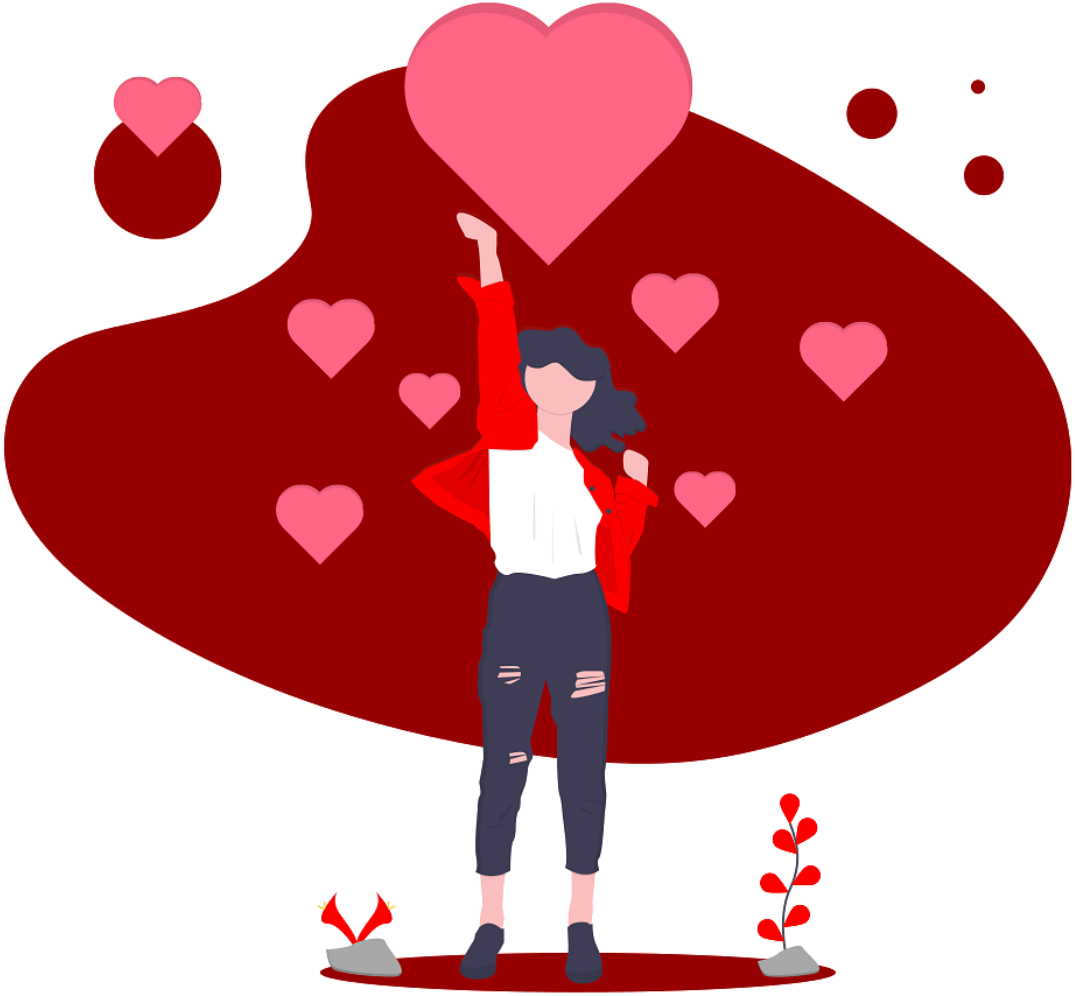
- ★ Owned by its component
- ★ Locally scoped
- ★ Writeable / Mutable
- ★ Has setState() method to modify properties
- ★ Changes to state can be asynchronous
- ★ Can only be passed as props

LIVE DEMO

NORDIC

< CODER >

Thank You
And see you ;)



NORDIC
< CODER >