QUÉ ES

Una librería para componer interfaces y renderizarlas

QUÉ NO ES

Un framework completo con router, cliente HTTP, etc

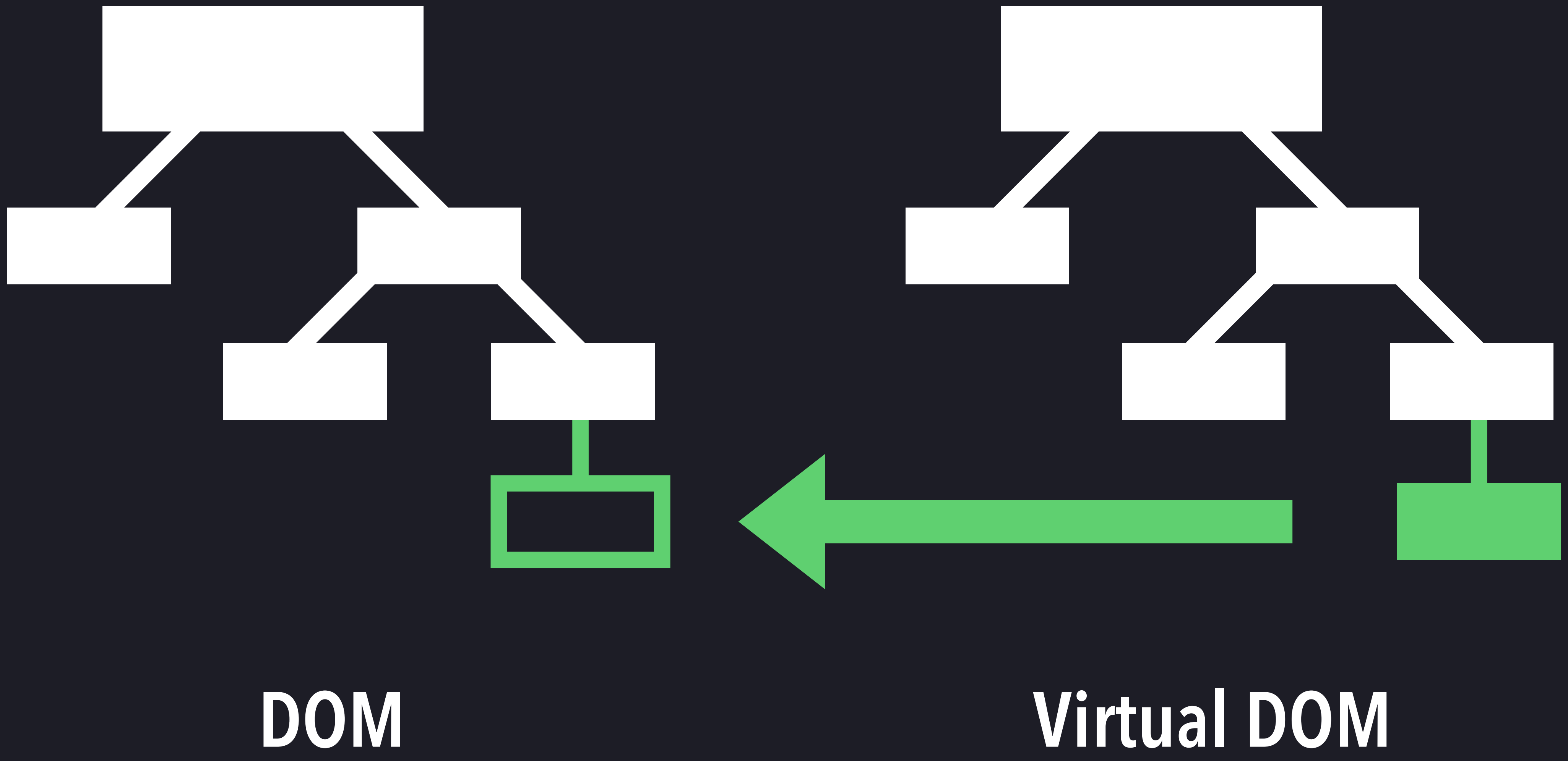
¿PARA QUÉ?

- Separar la UI en componentes reusables
- Actualizar la UI automáticamente cuando los datos cambian

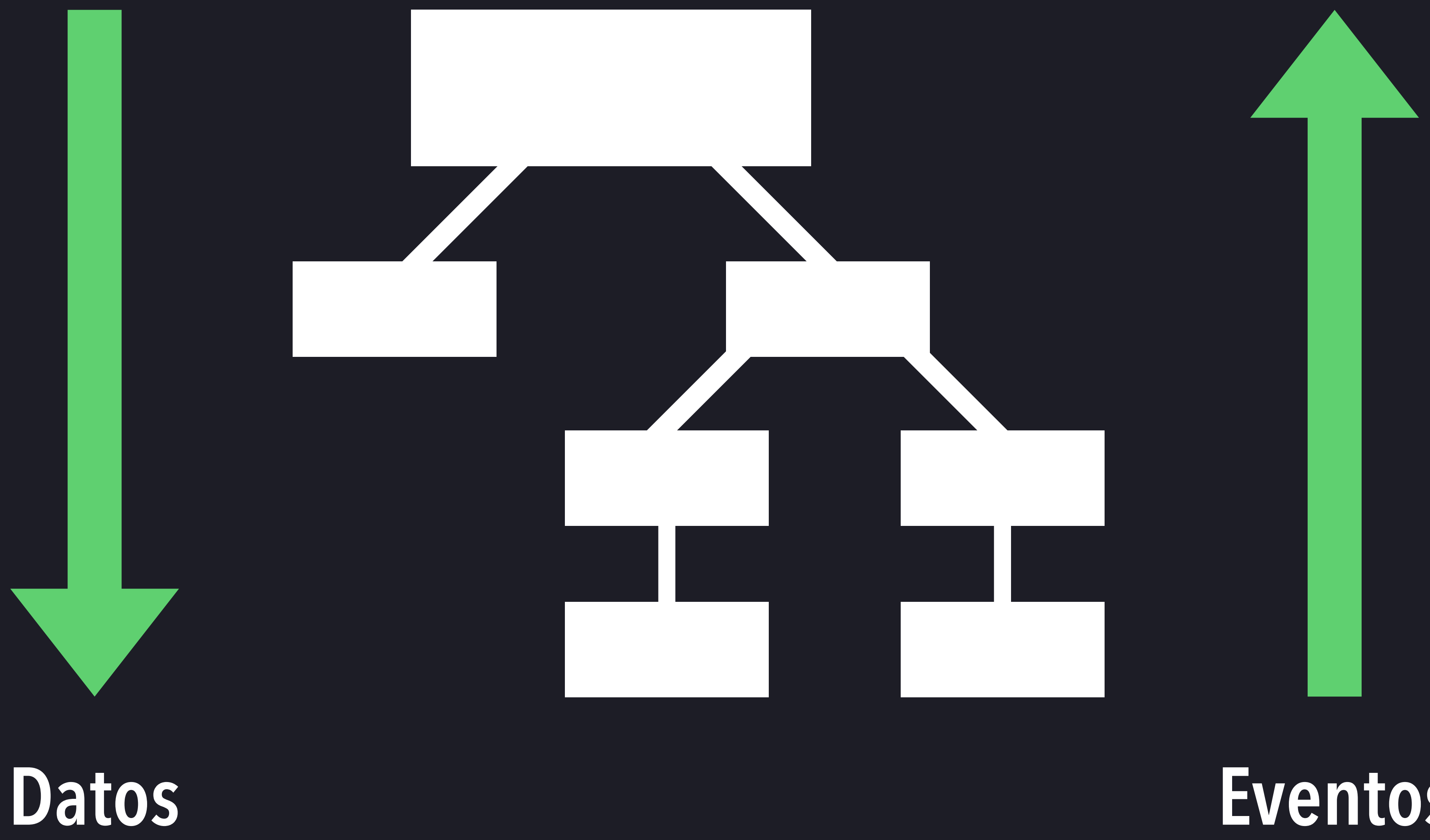
CARACTERÍSTICAS

1. Virtual DOM
2. Flujo unidireccional de datos
3. JSX
4. Componentes

VIRTUAL DOM


DOM
Virtual DOM

FLUJO UNIDIRECCIONAL DE DATOS

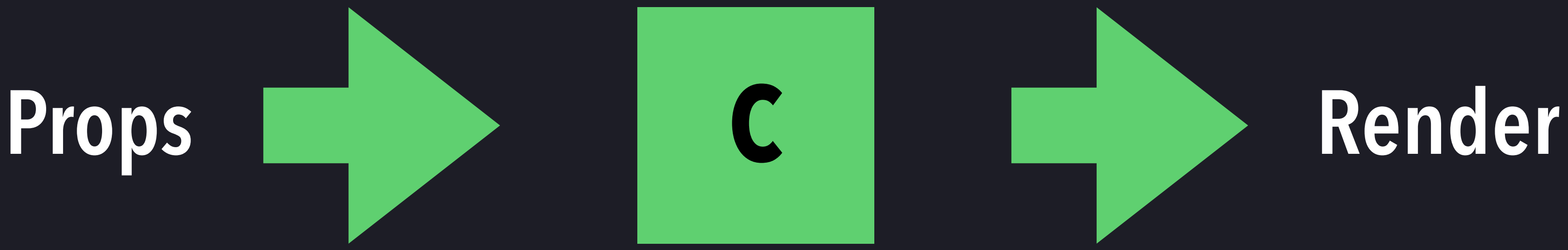

Datos
Eventos

JSX

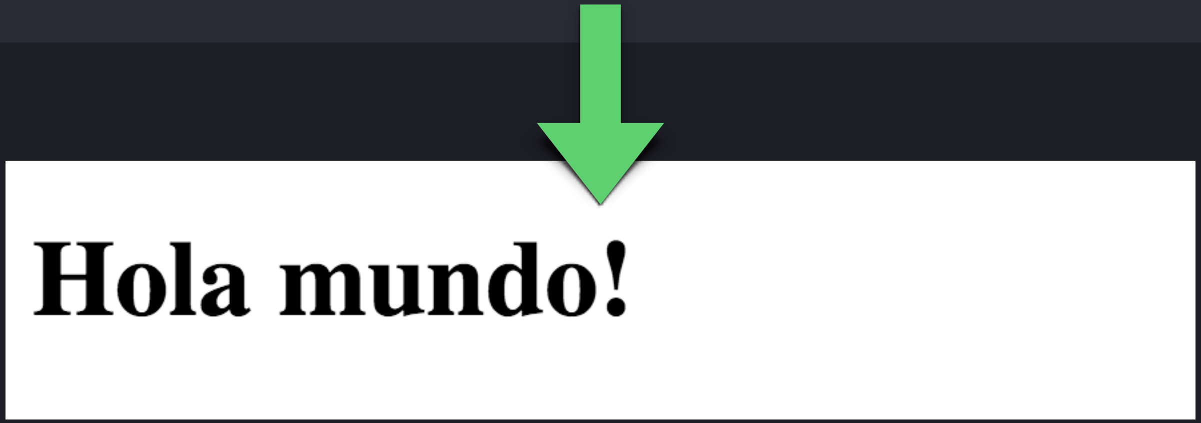
```
<a href="https://xkcd.com/221/" className="xkcd">
    <img src="https://imgs.xkcd.com/comics/random_number.png" />
</a>
```

```
React.createElement('a', {
        href: "https://xkcd.com/221/",
        className: "xkcd"
    },
    React.createElement('img', {
        src: "https://imgs.xkcd.com/comics/random_number.png"
    })
);
```

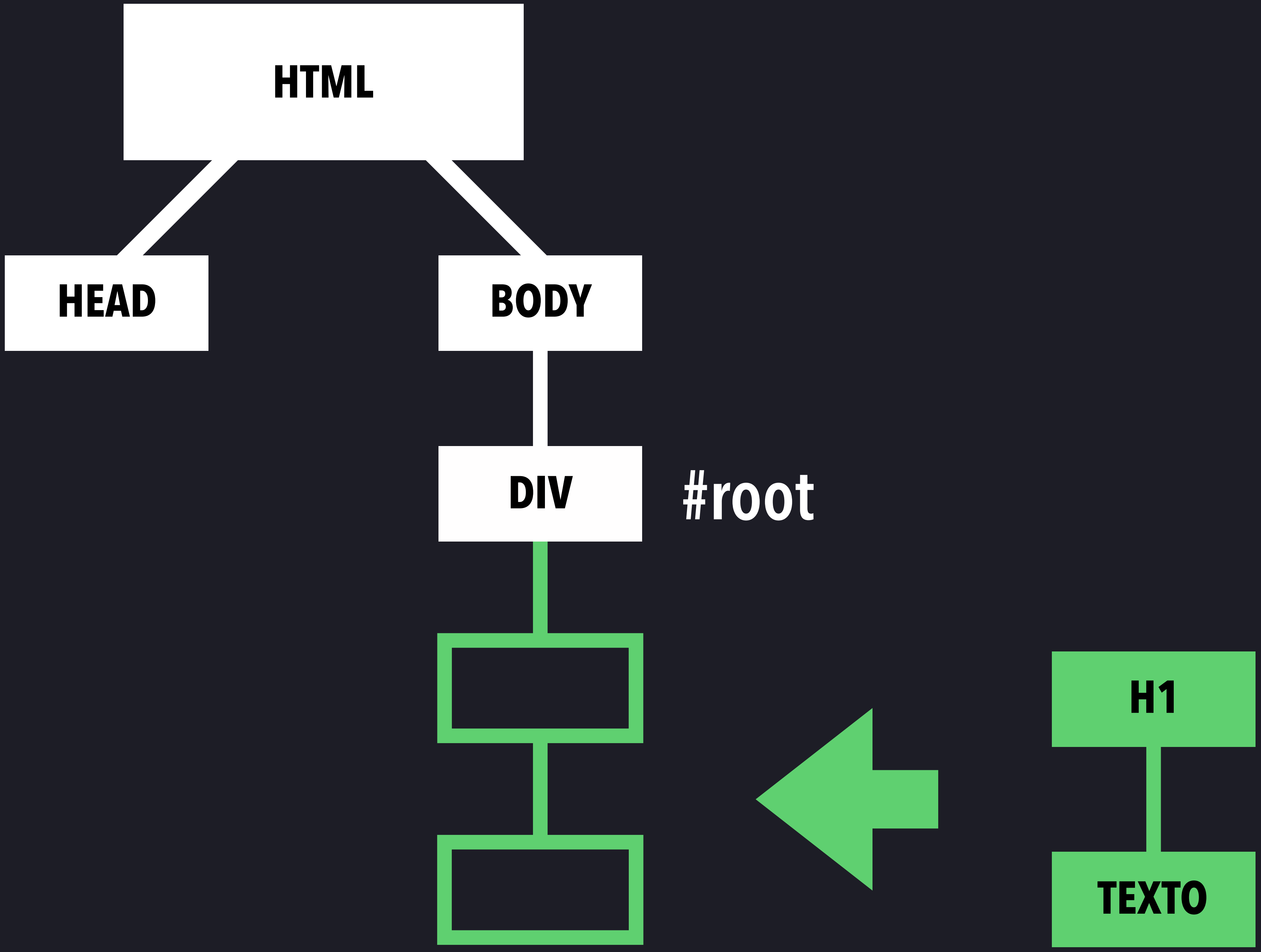
COMPONENTES

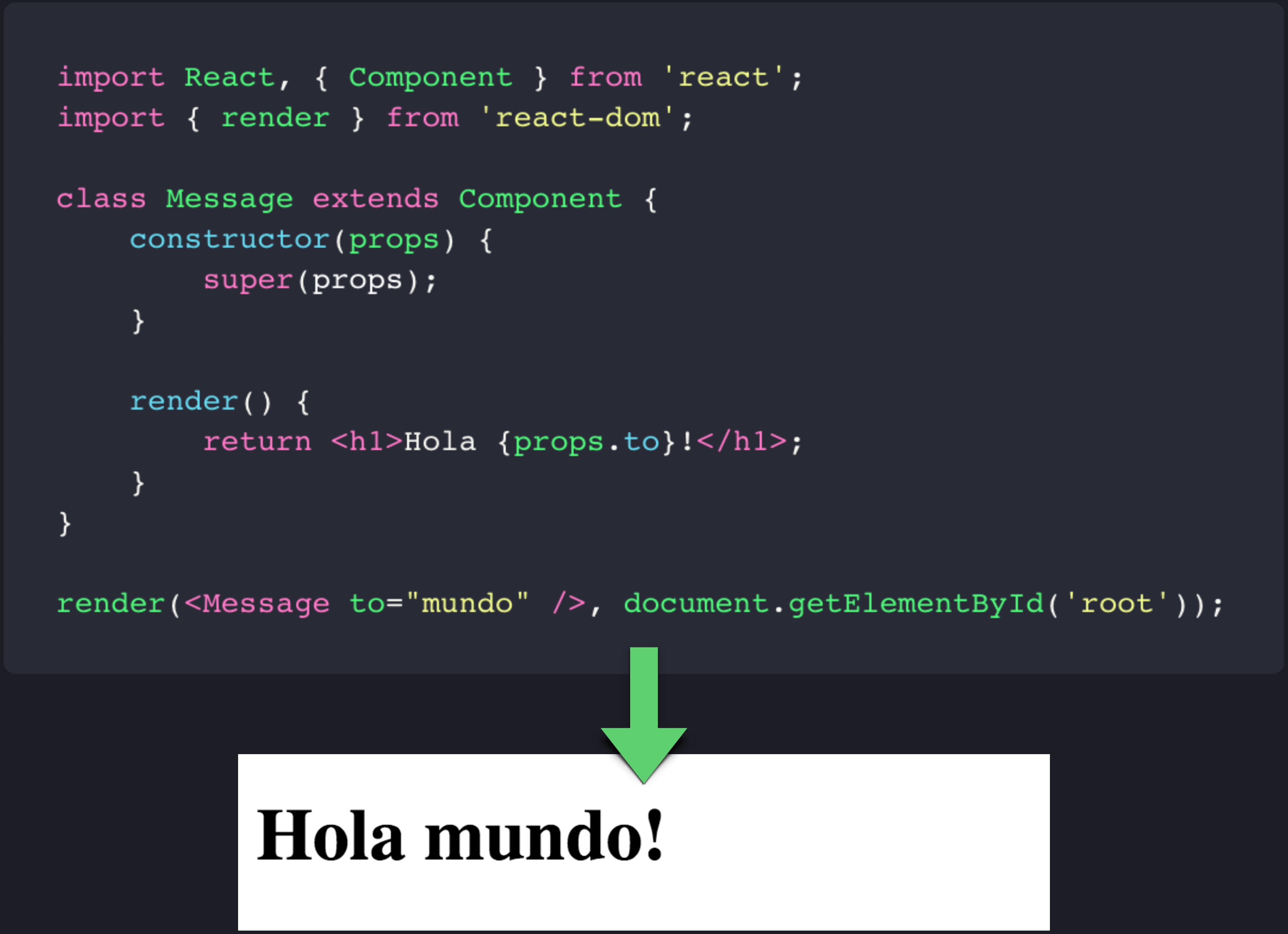

Props
C
Render

```
import React from 'react';
import { render } from 'react-dom';

const Message = props => <h1>Hola {props.to}!</h1>;

render(<Message to="mundo" />, document.getElementById('root'));
```



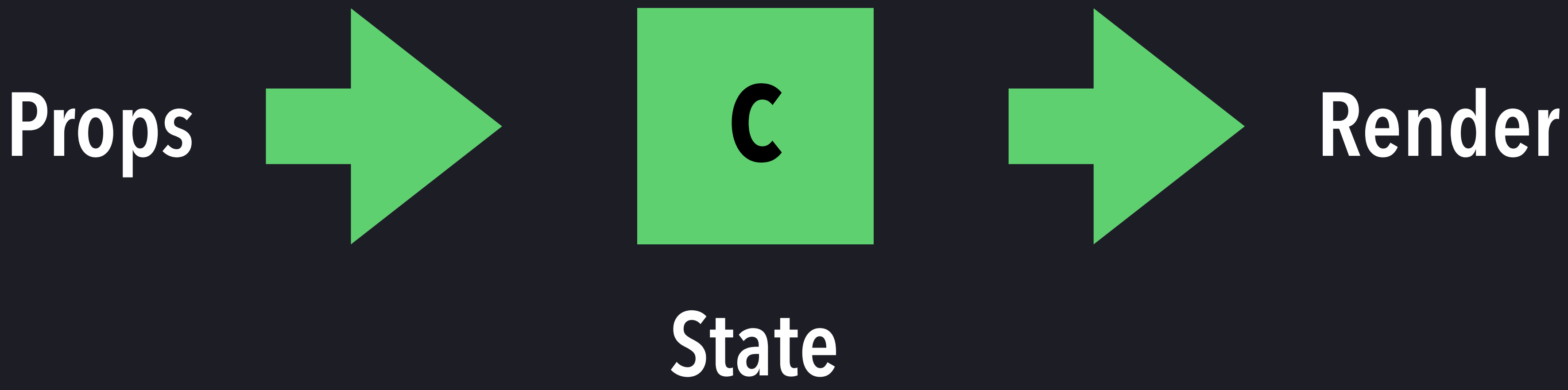
Hola mundo!

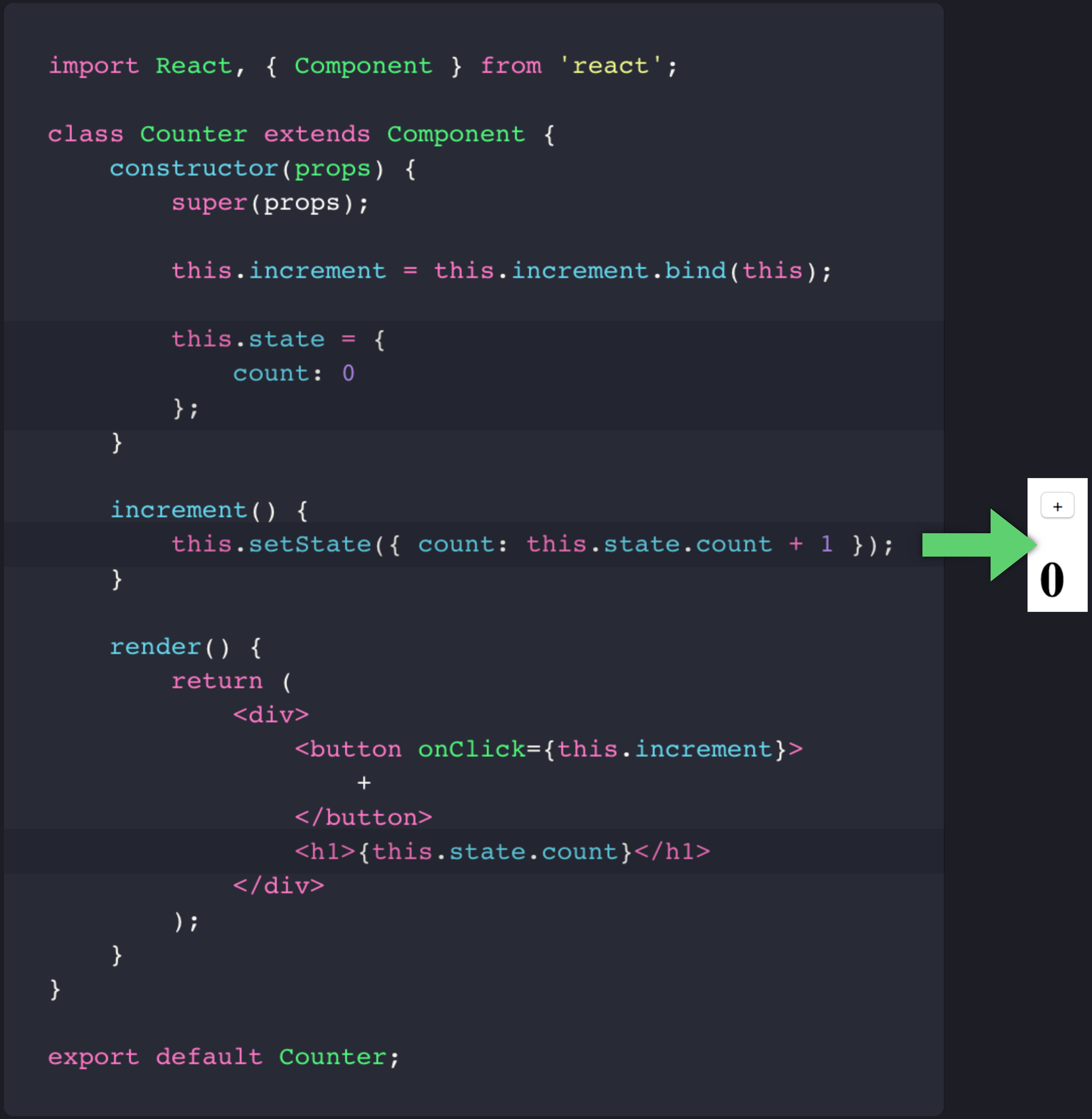

HTML
HEAD
BODY
DIV
#root
H1
TEXTO



```
import React, { Component } from 'react';
import { render } from 'react-dom';

class Message extends Component {
    constructor(props) {
        super(props);
    }

    render() {
        return <h1>Hola {props.to}!</h1>;
    }
}

render(<Message to="mundo" />, document.getElementById('root'));
```

Hola mundo!

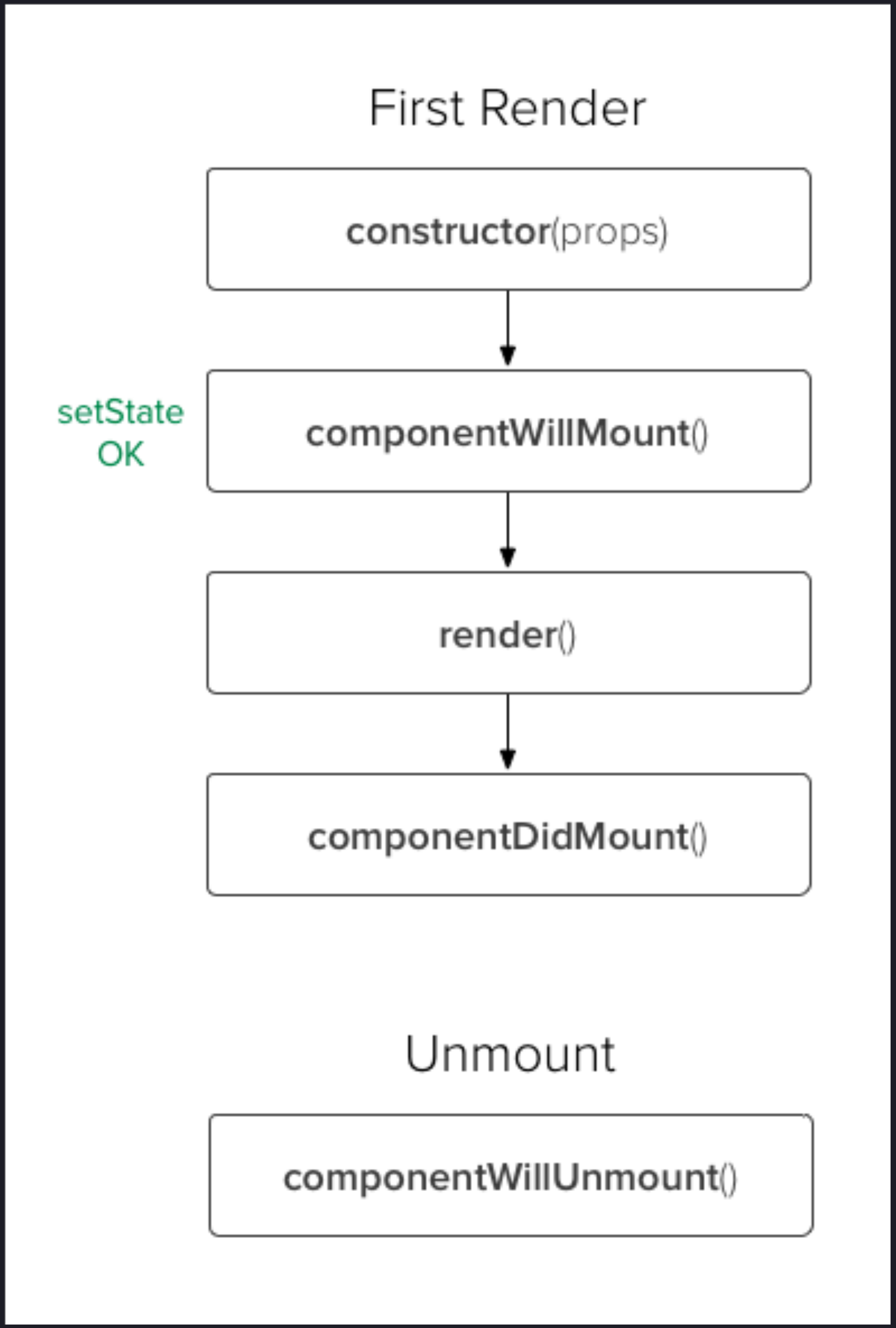

Props
C
State
Render



```
import React, { Component } from 'react';

class Counter extends Component {
    constructor(props) {
        super(props);

        this.increment = this.increment.bind(this);

        this.state = {
            count: 0
        };
    }

    increment() {
        this.setState({ count: this.state.count + 1 });
    }

    render() {
        return (
            <div>
                <button onClick={this.increment}>
                    +
                </button>
                <h1>{this.state.count}</h1>
            </div>
        );
    }
}

export default Counter;
```

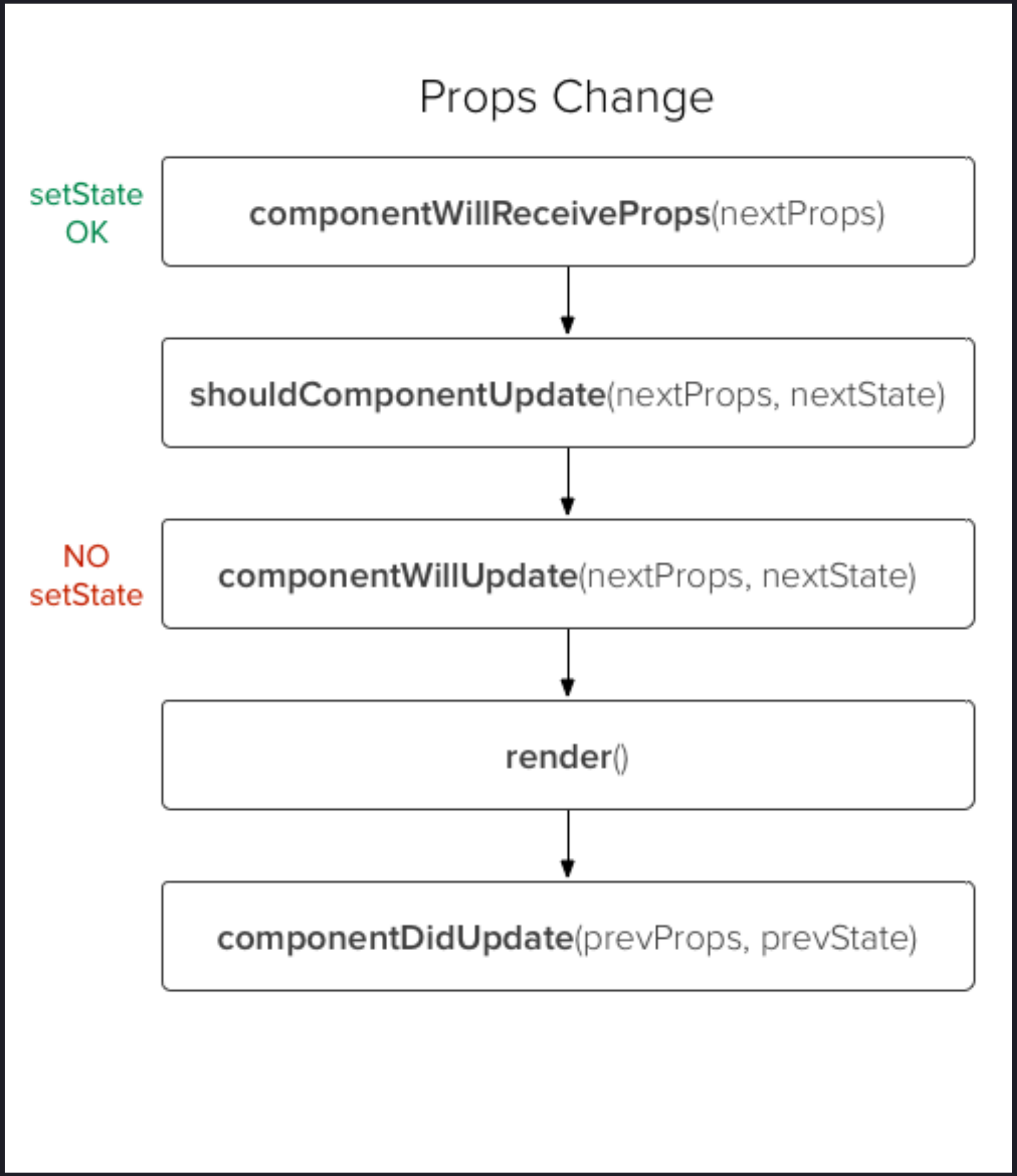
CICLO DE VIDA DE LOS COMPONENTES

```
import React, { Component } from 'react';

class Message extends Component {
    constructor(props) {
        super(props);
    }

    componentWillMount() {

    }

    render() {
        return <h1>Hola {props.to}!</h1>;
    }

    componentDidMount() {

    }

    componentWillUnmount() {

    }
}

export default Message;
```



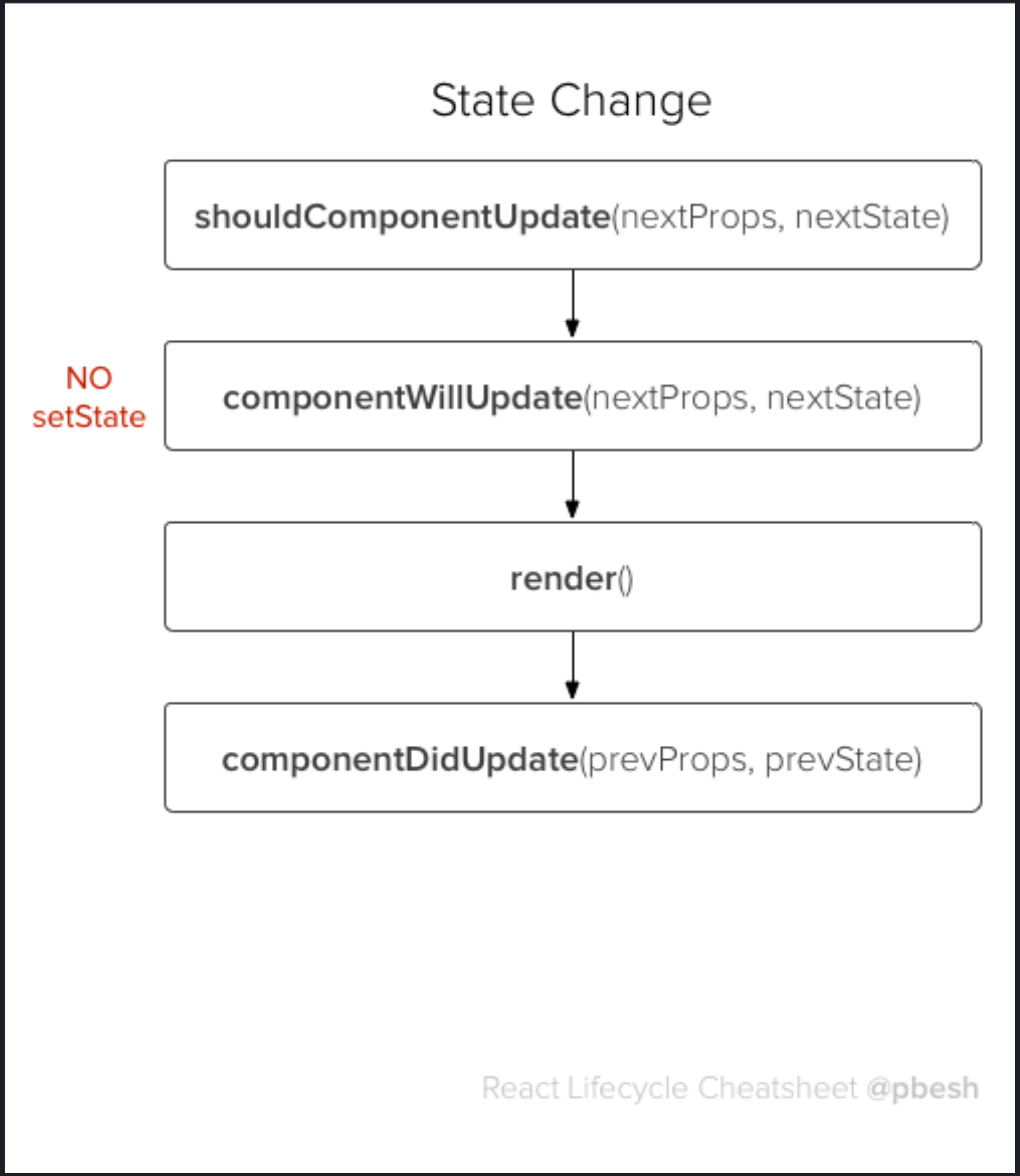
Cheatsheet by https://twitter.com/pbesh

```
import React, { Component } from 'react';

class Message extends Component {
    componentWillReceiveProps(nextProps) {

    }

    shouldComponentUpdate(nextProps, nextState) {
        return true;
    }

    componentWillUpdate(nextProps, nextState) {

    }

    render() {
        return <h1>Hola {props.to}!</h1>;
    }

    componentDidUpdate(prevProps, prevState) {

    }
}

export default Message;
```


Props Change
setState OK
componentWillReceiveProps(nextProps)
shouldComponentUpdate(nextProps, nextState)
NO setState
componentWillUpdate(nextProps, nextState)
render()
componentDidUpdate(prevProps, prevState)

Cheatsheet by https://twitter.com/pbesh

```
import React, { Component } from 'react';

class Message extends Component {
    shouldComponentUpdate(nextProps, nextState) {
        return true;
    }

    componentWillUpdate(nextProps, nextState) {

    }

    render() {
        return <h1>Hola {props.to}!</h1>;
    }

    componentDidUpdate(prevProps, prevState) {

    }
}

export default Message;
```


State Change
shouldComponentUpdate(nextProps, nextState)
NO setState
componentWillUpdate(nextProps, nextState)
render()
componentDidUpdate(prevProps, prevState)
React Lifecycle Cheatsheet @pbesh

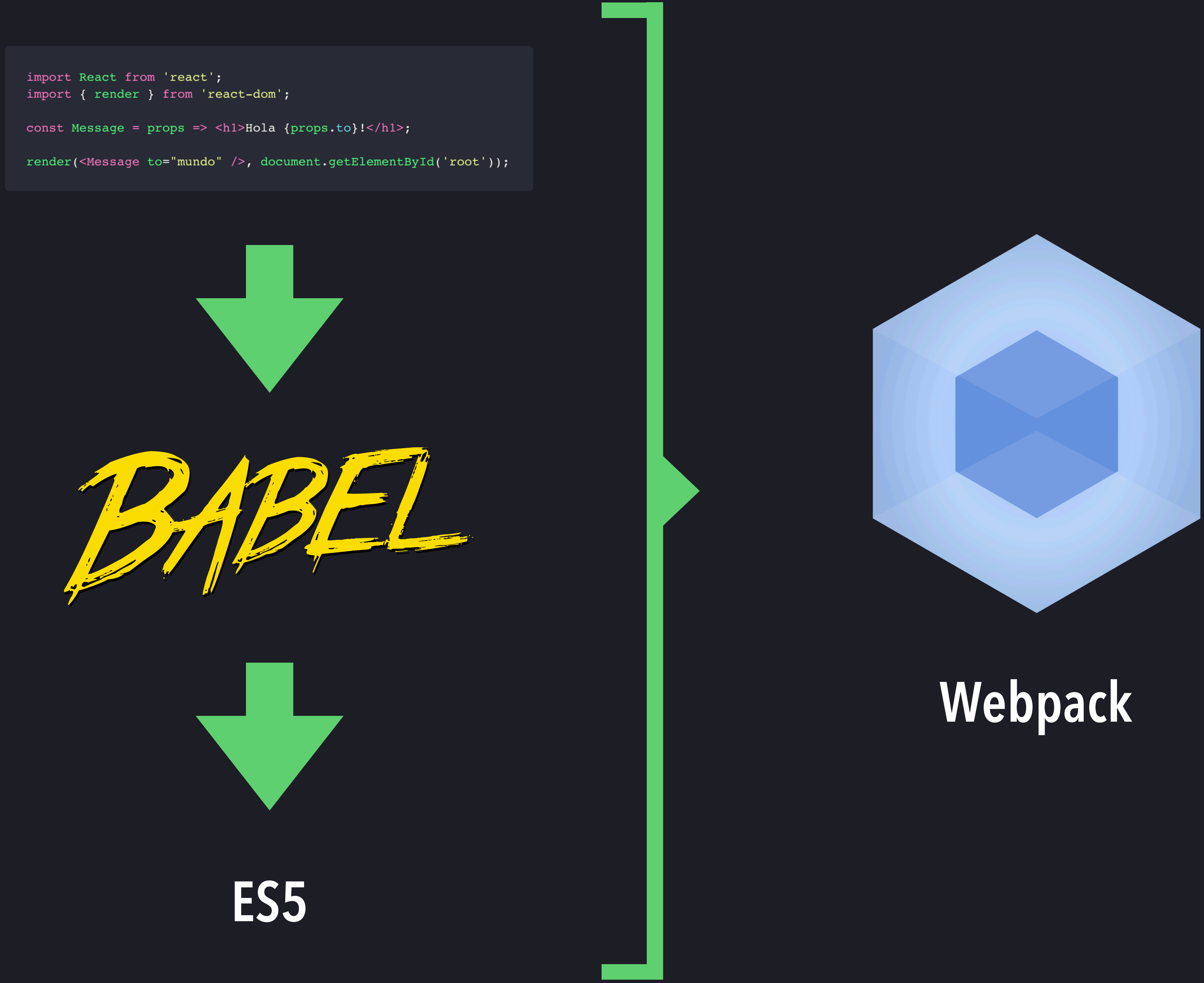
Cheatsheet by https://twitter.com/pbesh

EVENTOS

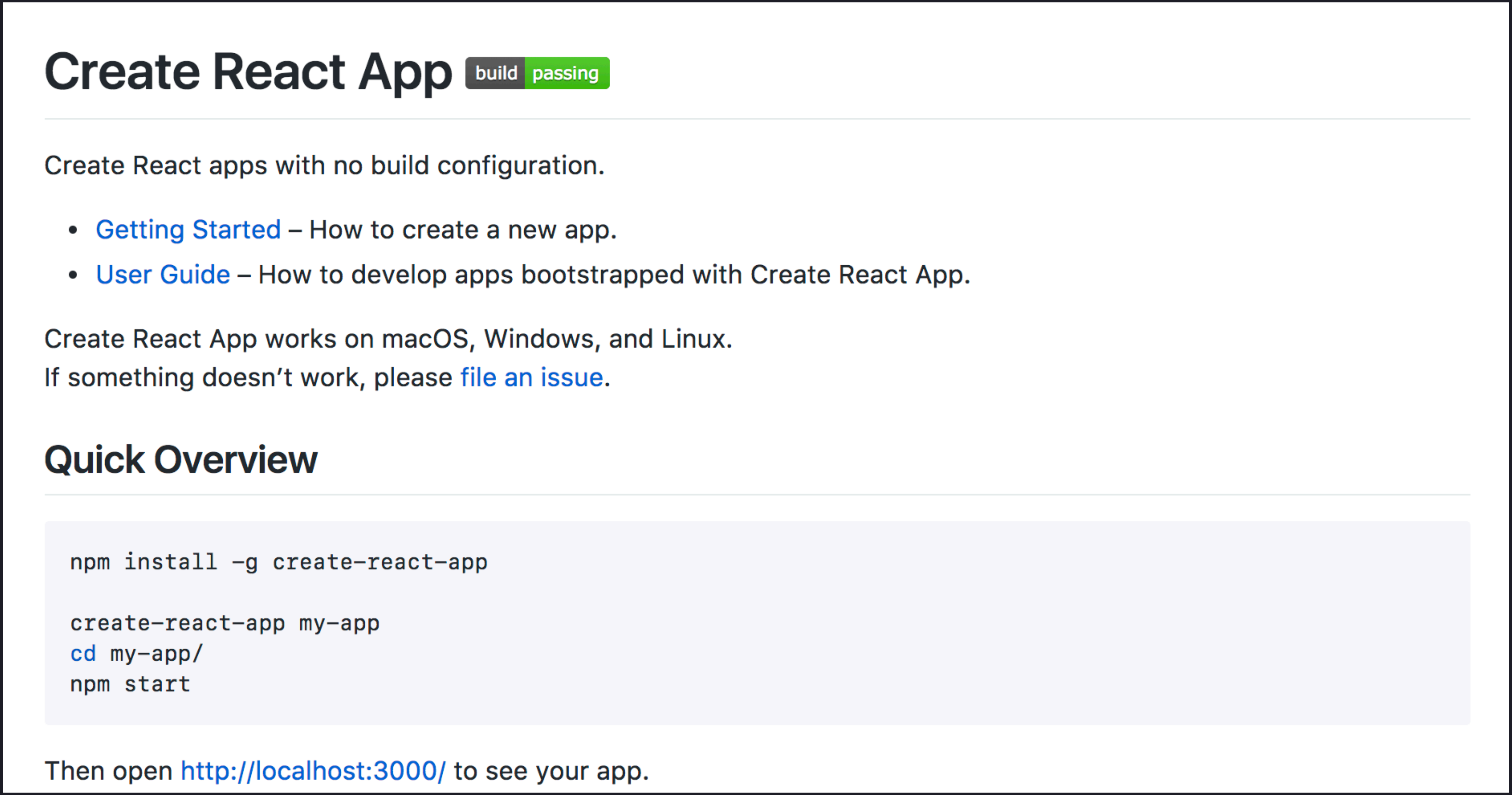
```
import React, { Component } from 'react';
import { render } from 'react-dom';
import Counter from './Counter';

class App extends Component {
    onCounterChange(value) {
        console.info(value);
    }

    render() {
        return <Counter onChange={this.onCounterChange} />;
    }
}

render(<App />, document.getElementById('root'));
```

```
import React, { Component } from 'react';

class Counter extends Component {
    constructor(props) {
        super(props);

        this.increment = this.increment.bind(this);

        this.state = {
            count: 0
        };
    }

    increment() {
        this.setState(
            { count: this.state.count + 1 },
            () => this.props.onChange(this.state.count)
        );
    }

    render() {
        return (
            <div>
                <button onClick={this.increment}>
                    +
                </button>
                <h1>{this.state.count}</h1>
            </div>
        );
    }
}

export default Counter;
```

TOOLING



```
import React from 'react';
import { render } from 'react-dom';

const Message = props => <h1>Hola {props.to}!</h1>;

render(<Message to="mundo" />, document.getElementById('root'));
```



Create React App

Create React apps with no build configuration.

- Getting Started – How to create a new app.
- User Guide – How to develop apps bootstrapped with Create React App.

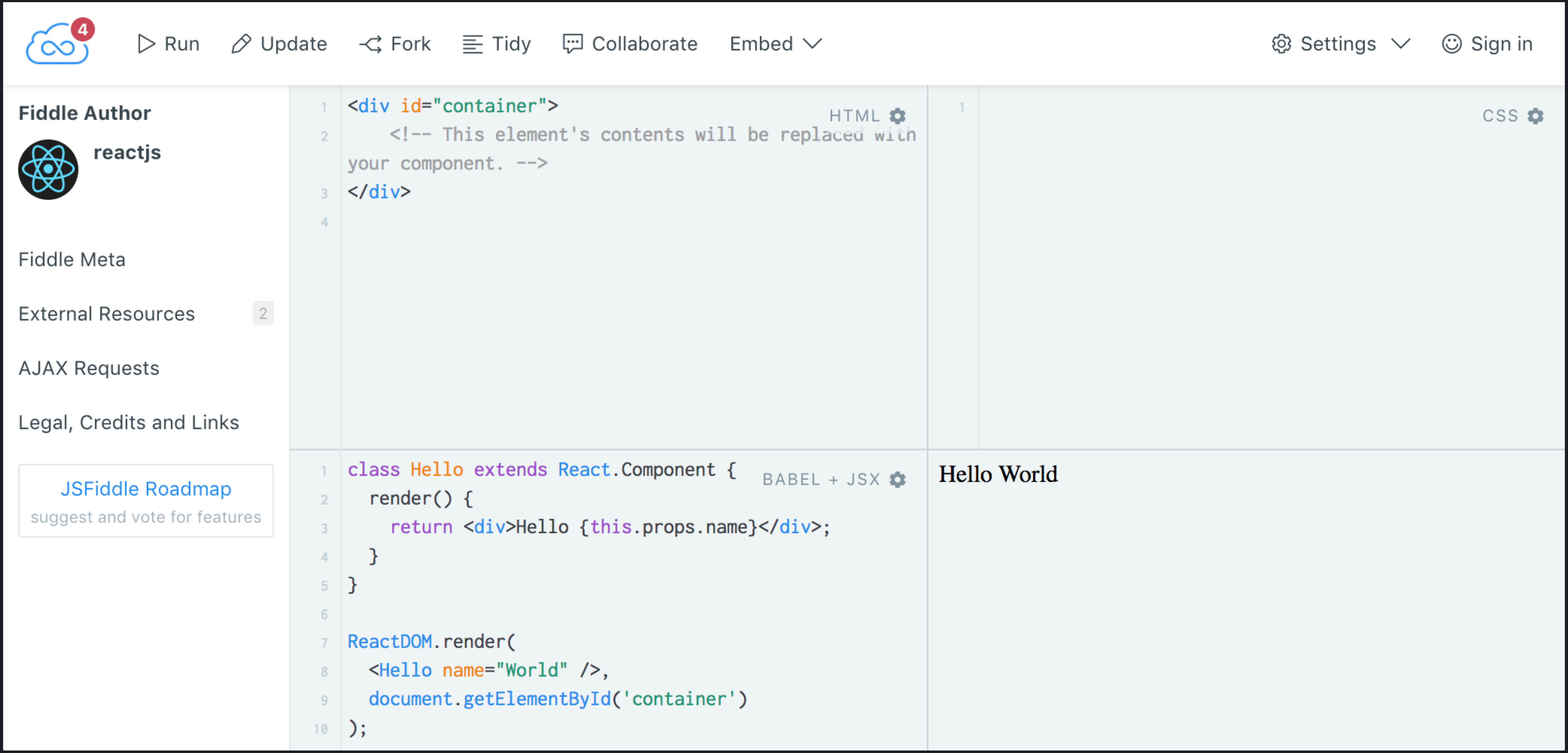
Create React App works on macOS, Windows, and Linux.
If something doesn't work, please file an issue.

Quick Overview

```
npm install -g create-react-app

create-react-app my-app
cd my-app/
npm start
```

Then open http://localhost:3000/ to see your app.

https://github.com/facebookincubator/create-react-app

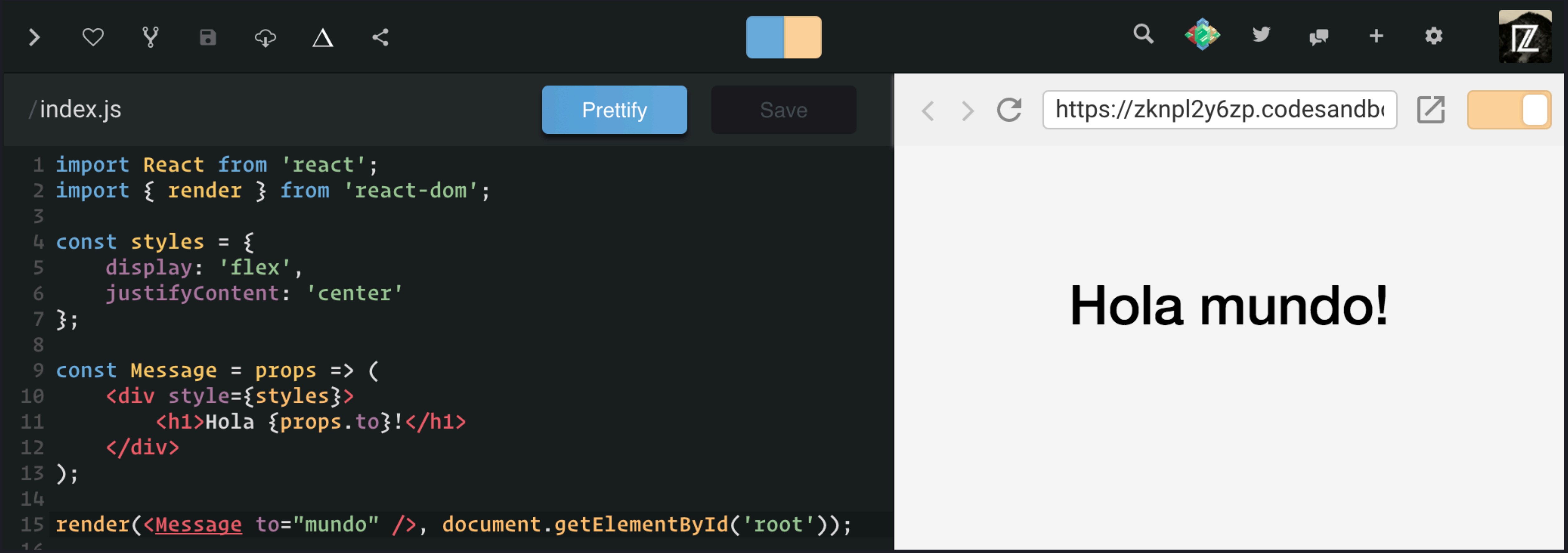


```html
<div id="container">
    <!-- This element's contents will be replaced with your component. -->
</div>
```

```jsx
class Hello extends React.Component {
  render() {
    return <div>Hello {this.props.name}</div>;
  }
}

ReactDOM.render(
  <Hello name="World" />,
  document.getElementById('container')
);
```

Hello World

https://jsfiddle.net/reactjs/69z2wepo



```
import React from 'react';
import { render } from 'react-dom';

const styles = {
    display: 'flex',
    justifyContent: 'center'
};

const Message = props => (
    <div style={styles}>
        <h1>Hola {props.to}!</h1>
    </div>
);

render(<Message to="mundo" />, document.getElementById('root'));
```

https://codesandbox.io

¿PREGUNTAS?