

REACT TERMS

component One discrete, re-usable portion of React code that can be

props JavaScript variant allowing HTML in JS

lifecycle methods Methods that have special names in React which are triggered at certain points in a React component's lifecycle

function / stateless component Many components might not need any state, only props, and can be re-written in a function syntax short-hand

unidirectional data-flow The idea that parents pass data to children via props, while children can never interact with siblings or with their parents directly

Virtual DOM Novel technique to speed up rendering while seemingly rerenders entire page (does "dry run" to render a "virtual DOM", compares what changed with the real DOM, and only makes minimum tweaks)

```
// Using components
import React, { Component } from "react";
import Button from './components/Button/Button.js';
class App extends Component {
  state = {
    count: 0,
  }

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

  render() {
    return (
      <Button onClick={this.increment}>
        Click me {this.state.count}
      </Button>
    )
  }
}
```

1 REACT EXAMPLE

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

COMPONENT EXAMPLES

src/components/Button/Button.js

```
import './Button.css';
class Button {
  render() {
    <button className="Button"
      onClick={this.props.onClick}>
      {props.children}
    </button>
  }
}

// If no state, can be re-written as
// "functional component" short-hand
const Button = (props) => (
  <button className="Button"
    onClick={props.onClick}>
    {props.children}
  </button>
);
export default Button;
```