

DEFINING COMPONENTS

src/components/Button/Button.js

```
import React from "react";
import "./Button.css";
function Button(props) {
  return (
    <button className="Button"
      onClick={props.onClick}>
      {props.children}
    </button>
  );
}
// Alternatively, you can use
// "arrow function" short-hand
const Button = (props) => (
  <button className="Button"
    onClick={props.onClick}>
    {props.children}
  </button>
);
export default Button;
```

REACT TERMS

component One discrete, re-usable, self-contained portion of React code that can be used multiple times in a project for repeatable graphical components

props Short for “properties”, props are *immutable* and represent the data passed down to components from the parent of a component as attributes

hooks Hooks are functions that let you “hook into” React state and lifecycle features from inside components. React provides a few built-in Hooks like `useState` and `useEffect`.

function / stateless component Many components might not need any state, only props, and can be rewritten 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)

DESTRUCTURING

```
const info = {name: "jane", age: 35};
const name = info.name;
const age = info.age;
// Equivalent to
const info = {name: "jane", age: 35};
const {name, age} = info;
```

USING COMPONENTS

```
import React, { useState } from "react";
import Button from "../components/Button/Button.js";
function App() {
  const [count, setCount] = useState(0);

  function increment() {
    setCount(count + 1);
  }

  return (
    <Button onClick={increment}>
      Click me {count}
    </Button>
  );
}
```

USEFUL SNIPPETS

Conditional rendering

```
if (!props.text) {
  return (
    <p><em>No text found...</em></p>
  );
}

return (
  /*... full normal render JSX here ... */
);
```

Using map to loop through data

```
<div>{
  props.data.map((item, index) => (
    <p onClick={doAction(index)}>
      {index}: {item}
    </p>
  ))
}</div>
```

Using ? : (ternary operator) for an “if-statement”

```
<div>{
  props.image ? (
    <img src={props.image} />
  ) : <em>No image provided.</em>
}</div>
```