

# Bringing Everything Together: Buttons

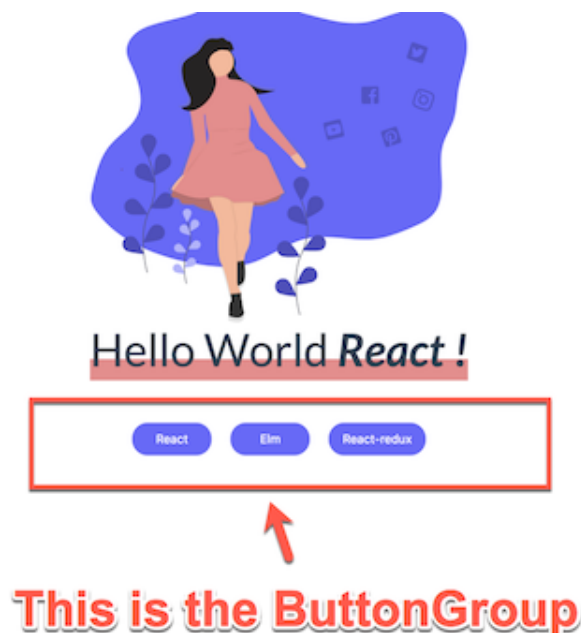
The creation of our app's buttons will be handled in this lesson using ButtonGroup.js. It will also discuss several aspects of our button template.

In the previous lesson, we looked at the implementation for **App.js**

```
import React, { Component } from "react";
import HelloWorld from "../HelloWorld";
import ButtonGroup from "../ButtonGroup";
import { store } from "../store";
class App extends Component {
  render() {
    return [
      <HelloWorld key={1} tech={store.getState().tech} />,
      <ButtonGroup key={2} technologies=["React", "Elm", "React-redux"] />
    ];
  }
}
export default App;
```

We shall now discuss the **ButtonGroup** component.

The ButtonGroup component takes in an array of technologies and spits out buttons. It handles the rendering of the three buttons below the “Hello World” text.



Also, you may notice that the **App** component returns an array. That's a React 16 goodie. With **React 16**, you don't have to wrap adjacent **JSX** elements in a `div`. You can use an array if you want - but pass in a `key` prop to each element in the array.

```
class App extends Component {
  render() {
    return [
      <HelloWorld key={1} tech={store.getState().tech} />,
      <ButtonGroup key={2} technologies={["React", "Elm", "React-redux"]} />
    ];
  }
}

export default App;
```



That is it for the `App.js` component.

The implementation of the `ButtonGroup` component is quite simple. Here it is:

**ButtonGroup.js:**

```
import React from "react";
const ButtonGroup = ({ technologies }) => (
  <div>
    {technologies.map((tech, i) => (
      <button
        data-tech={tech}
        key={`btn-${i}`}
        className="hello-btn"
      >
        {tech}
      </button>
    ))}
  </div>
);
export default ButtonGroup;
```

ButtonGroup.js

`ButtonGroup` is a stateless component that takes in an array of technologies, denoted by **technologies**.

It loops over this array using **map** and renders a `<button></button>` for each of the tech in the array. In this example, the buttons array passed in is, `["React", "Elm", "React- redux"]`

The buttons generated have a few attributes. There's the obvious `className` for styling purposes. There's key to prevent the pesky react warning about rendering multiple items without a key prop. Gosh, that error haunts me every time :(

Lastly, there is a **data-tech** attribute on each button. This shall be covered in the next lesson.