4 Overview

Now that you’re set up, let’s get an overview of React!

Inspecting the starter code

In CodeSandbox you’ll see three main sections:

1. The *Files* section with a list of files like `App.js`, `index.js`, `styles.css` and a folder called `public`
2. The *code editor* where you’ll see the source code of your selected file
3. The *browser* section where you’ll see how the code you’ve written will be displayed

The `App.js` file should be selected in the *Files* section. The contents of that file in the *code editor* should be:

```
export default function Square() {
  return <button className = "square"> X </button>;
}
```

The *browser* section should be displaying a square with a X in it like this:

Now let’s have a look at the files in the starter code.

`App.js`

The code in `App.js` creates a *component*. In React, a component is a piece of reusable code that represents a part of a user interface. Components are used to render, manage, and update the UI elements in your application. Let’s look at the component line by line to see what’s going on:

```
export default function Square() {
    return <button className = "square"> X </button>;
}
```

The first line defines a function called `Square`. The `export` JavaScript keyword makes this function accessible outside of this file. The `default` keyword tells other files using your code that it’s the main function in your file.

```
export default function Square() {
    return <button className = "square"> X </button>;
}
```

The second line returns a button. The `return` JavaScript keyword means whatever comes after is returned as a value to the caller of the function. `<button>` is a *JSX element*. A JSX element is a combination of JavaScript code and HTML tags that describes what you’d like to display. `className="square"` is a button property or *prop* that tells CSS how to style the button. `X` is the text displayed inside of the button and `</button>` closes the JSX element to indicate that any following content shouldn’t be placed inside the button.

`styles.css`

Click on the file labeled `styles.css` in the *Files* section of CodeSandbox. This file defines the styles for your React app. The first two *CSS selectors* (`*` and `body`) define the style of large parts of your app while the `.square` selector defines the style of any component where the `className` property is set to `square`. In your code, that would match the button from your Square component in the `App.js` file.

`index.js`

Click on the file labeled `index.js` in the *Files* section of CodeSandbox. You won’t be editing this file during the tutorial but it is the bridge between the component you created in the `App.js` file and the web browser.

```
import { StrictMode } from 'react';
import { createRoot } from 'react-dom/client';
import './styles.css';

import App from './App';
```

Lines 1-5 bring all the necessary pieces together:

- React
- React’s library to talk to web browsers (React DOM)
- the styles for your components
- the component you created in `App.js`.

The remainder of the file brings all the pieces together and injects the final product into `index.html` in the `public` folder.