Exercise 3 – Creating Components using JSX

Objective

To be able to modify a standard ReactJS project to allow the use of JSX files

To be able to create components in JSX and render them to the browser

Overview

This exercise takes the same project set up as defined in Exercise 2 and modifies it to be able to use JSX files for creating components. Once this has been done, a component is then created in JSX and rendered to the browser.

Part 1 – Modifying the Project

1.1. In a suitable text editor, navigate to the **EG03_CreatingComponentsWithJSX** project folder and then open the **starter** folder.

1.2. On the command line/terminal, also in the **EG03_CreatingComponenetsWithJSX/starter** folder, run the **npm install** command to install the dependencies in a node_modules folder (automatically created)

Part 2 – Creating the JSX Component

2.1. In the **starter** folder, create a file called **App.jsx** and open it for editing in a suitable text editor. This will be the file in which we create the component to render to the screen.

2.2. This file needs access to the **react** library, so use an `import` statement for the `React` class.

2.3. Declare two exported classes called `Header` and `Content` that extend the `React.Component` class. The classes should have `render` methods that produce appropriate content.

Remember, if you are intending to use more than one element in a component, then this needs to be wrapped in an outer element such as a `<div>`.

Part 3 – Modifying the main.js and index.html files to use App.jsx

3.1. Navigate to the **main.js** file and open it in a suitable text editor.

3.2. Add an `import` statement so that access is granted to the `Header` and `Content` classes from **App.jsx**.

3.3. Rewrite the `render` method so that the `Header` component is deposited into a `<header>` element.

3.4. Add a second `render` method to place the `Content` component in an element with an `id` of `content`.

3.5. Modify the **index.html** file so that it has an empty `<header>` element and an empty `<div>` with the `id` of `content`.

Part 4 – Viewing the Application

4.1. To view the application, execute the following on the command line or terminal after navigating to the correct folder:

```
npm start
```

4.2. Open a browser and navigate to:

```
localhost:8080
```

The Header and the Content components that you have created should be visible on the screen.

Part 5 – Checking the Application Updates

5.1. Return to the **App.jsx** file and change the **Header** and **Content** components, saving it when you are done.

5.2. The browser should refresh automatically. If you have problems, call your instructor.

If you have time…

Try to modify your application so that it only uses a single ReactDOM.render method. This will require you either:

- Create a single component as the default export in a file that will use 2 other components that are also created in the file (Nested classes)
- Create multiple JSX files for each component, importing and exporting them as appropriate