Recap: React basics

Overview

Having worked through the two React-focused courses in this Front-End developer program, you are likely to know that React is all about components and that components rely heavily on the JSX syntax. In this reading, you'll be guided through a quick recap of the basic building blocks of React.

What is React?

React is a declarative and component-based JavaScript library used to build user interfaces. It allows you to build components and compose them together.

This feature, namely, the composability of React components, is very important because it allows you to construct React apps similar to how you'd create a structure with toy blocks. Simply put, it's a process that allows the construction of complex structures from simple building blocks.

React components

In modern React, every React component is a JavaScript function. To differentiate between regular HTML elements and custom React components, you need to capitalize the component function's name. Then, using JSX syntax, you can render a component from another component. JSX syntax relies on expressions and ternary operators.

Rendering components

For a component to be rendered (displayed) on a screen, it needs to be imported into another component. A component that imports another component is known as the parent component. The component that gets imported and rendered is referred to as a child component.

You can render multiple components from the parent component and pass data from a parent component to a child component using props.

For example, suppose you have a parent component named **`App`**, which imports a child component named **`Header`**:

```
1   import React from "react";
2   import Header from "./Header";
3
4   export default function App() {
5       return (
6           <Header />
7       )
8   }
9
```

The Header component returns some HTML with the heading element containing some text.

```
1   export default function Header() {
2       return (
3           <div className="main-grid">
4               <h1>This is the Header</h1>
```

```
            </div>
        )
    }
```

Props, state and event handlers

There are two ways that components work with data: `props` and `state`. If the data comes from a parent component, that data comes in the form of props. Props are immutable and external to a component, meaning they come from outside a component. State is mutable and internal to a component, meaning it exists inside a component.

Event handlers are a way to update state. Event handlers in React work similarly to the event attributes in plain HTML. There are several differences between the two syntaxes, but the two main ones are:

- HTML event attributes use lowercase; React event attributes use camelCase.
- HTML event attributes use parentheses; React event attributes don't use parentheses.

Conclusion

In this reading, you revisited the topic of React basics. Now that you've re-acquainted yourself with some of the basic React concepts, you are better prepared to take on your next exercise, which will involve setting up components for your project app. If you would like a more in-depth recap of the React basics, please revisit the following lesson items from the **React basics** course:

- Creating React components
- Transpiling JSX
- Types of events
- Parent-child data flow
- What is state?