React-Essentials (Summary)

2026-01-19

0.1 Key Takeaways

- A component can be defined like any other JS function. Its return body must be some sort of mark-up. Furthermore, components must be capitalized
- Components (`.jsx`) wrap javascript, html, and css together into one file.
- Since .jsx code is built with the React compiler, the code that is written is not the same code that is delivered to the end-user.
- Essentially, React compiles the Component Tree into the DOM Tree
- To insert a component into the main program body i.e: App() => {}.

1 React Essentials

1.0.1 Components

Components (`.jsx`) wrap javascript, html, and css together into one file. Helping keep front-end codebases relatively small.

- A component can be defined like any other JS function. Its return body must be some sort of mark-up. Furthermore, components must be capitalized

To insert a component into the main program body i.e: App() => {}. We can either insert the component like any other HTML tag, or you can have the component be self-closing. For example:

```
function Body(){
    return (
        <p> Hi! </p>
    );
}

function App(){
    return (
        <title> Website </title>
        <Body/> //or <Body> </Body>
    );
}

export default App
```

1.0.2 Component Tree

Since .jsx code is built with the React compiler, the code that is written is not the same code that is delivered to the end-user. Using `ReactDOM`, we are able to serve a .jsx file as an entry-point into a specific element. In the starting-project example, we have a index.html file that contains a component (id = "root") which we use as the entry-point for the `App.jsx` file with following ReactDOM methods in our index.jsx file:

```
import App from "./App.jsx";
import ReactDOM from "react-dom/client";

const entryPoint = document.getElementById("root");
ReactDOM.createRoot(entryPoint).render(<App />);
```

Essentially, React compiles the Component Tree into the DOM Tree