**Episode 3: Laying the foundation**

**Topics:**

* JSX
* React.createElement vs JSX
* Benefits of JSX
* Behind the Scenes of JSX
* Babel and Parcel role in JSX
* Components
* Functional Components – Composing Components

**Assignment:**

1. What is JSX ?

JSX is a HTML like Syntax. It is syntax extension to Javascript. It comes with the full power of Javascript. JSX produces ‘React Elements’. It basically allows us to write HTML in React.

It is not HTML inside Javascript. It is a Javascript syntax to create React elements.

2. Superpowers of JSX

Using JSX, we can write markup inside Javascript, providing us with a superpower to write logic and markup of a component inside a single jsx file.

JSX converts HTML tags into react elements.

3. Role of type attribute in script tag ? What options can I use there ?

The type attribute gives the language of the script or the format of the data. If the attribute is present, its value must be a valid MIME type. The charset parameter must not be specified. The default value which is used, if the value is absent is “test/javascript”.

1. text/javascript – It is the basic standard of writing javascript code inside script tag.

2. text/ecmascript – This value indicates that the script is following the ecmascript standards.

3. module – This value tells the browser that the script is a module and it can import or export other files or modules inside it

4. text/babel – This value indicates that the script is a babel type and required babel to transpile it.

5. text/typescript – As the name suggests, the script is written in typescript.

4. {TitleComponent} vs <TitleComponent /> vs <TitleComponent></TitleComponent> in JSX

{TitleComponent} – This value describes the TitleComponent as a Javascript expression or a variable. The {} can embed a Javascript expression or a variable.

<TitleComponent /> - This value represents a Component that is returning some JSX value. In simple terms TitleComponent is a function that is returning JSX value. A component is written inside < /> expression.

<TitleComponent></TitleComponent> - <TitleComponent /> and <TitleComponent></TitleComponent> are equivalent only when the <TitleComponent /> has no child components. The opening and closing tags are created to include the child components.

**Coding Assignment:**

1. Create a Nested header Element using React.createElement(h1, h2, h3 inside a div with class “title”)

* Create the same element using JSX
* Create a functional component of the same with JSX
* Pass attributes into the tag in JSX
* Composition of Component (Add a component inside another)
* {TitleComponent} vs {<TitleComponent/>} vs {<TitleComponent>;</TitleComponent>} in JSX

2. Create a Header Component from scratch using Functional Components with JSX

* Add a Logo on left
* Add a search bar in middle
* Add User icon on right
* Add CSS to make it look nice

**References:**

1. Babel – <https://babeljs.io/>

2. Attribute Type – <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/script#attr-type>

3. JS Modules – <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Modules>

4. Babel Playground – <https://babeljs.io/repl>

5. React without JSX - <https://legacy.reactjs.org/docs/react-without-jsx.html>