

Namaste React Course by Akshay Saini

Chapter 03 - Laying the Foundation

Q: What is JSX?

A: JSX stands for JavaScript XML.

JSX allows us to write HTML elements in JavaScript and place them in the DOM without any createElement() and/or appendChild() methods.

JSX makes it easier to write and add HTML in React.

JSX converts HTML tags into react elements.

Example 1 using JSX:

```
const myElement = <h1>I Love JSX!</h1>;
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(myElement);
```

Example 2 Without JSX:

```
const myElement = React.createElement('h1', {}, 'I do not use JSX!');
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(myElement);
```

Q: Superpowers of JSX.

A: Using JSX, you can write markup inside Javascript, providing you with a superpower to write logic and markup of a component inside a single .jsx file. JSX is easy to maintain and debug.

Example

```
function greeting(user) {
//JSX
  return <h1>{user}, How are you!!!</h1>;
}
```

Q: Role of type attribute in script tag? What options can I use there?

A: The type attribute specifies the type of the script. The type attribute identifies the content between the <script> and </script> tags. It has a Default value which is "text/javascript".

type attribute can be of the following types:

- text/javascript : It is the basic standard of writing javascript code inside the <script> tag.

Syntax

```
<script type="text/javascript"></script>
```

- text/ecmascript : this value indicates that the script is following the EcmaScript standards.
- module: This value tells the browser that the script is a module that can import or export other files or modules inside it.
- text/babel : This value indicates that the script is a babel type and required babel to transpile it.
- text/typescript: As the name suggest the script is written in TypeScript.

Q: {TitleComponent} vs {<TitleComponent/>} vs {<TitleComponent></TitleComponent>} in JSX.

A: The Difference is stated below:

- {TitleComponent}: This value describes the TitleComponent as a javascript expression or a variable.
The {} can embed a javascript expression or a variable inside it.
- <TitleComponent/> : This value represents a Component that is basically returning Some JSX value. In simple terms TitleComponent a function that is returning a JSX value.
A component is written inside the {< />} expression.
- <TitleComponent></TitleComponent> : <TitleComponent /> and <TitleComponent></TitleComponent> are equivalent only when < TitleComponent /> has no child components. The opening and closing tags are created to include the child components.

Example

```
<TitleComponent>
  <FirstChildComponent />
  <SecondChildComponent />
  <ThirdChildComponent />
</TitleComponent>
```