ARYATECHNO

React - JSX

Topics : React JS
Written on January 02, 2024

JSX (JavaScript XML) is a syntax extension for JavaScript that looks similar to XML or HTML. It is commonly used with React to describe what the UI should look like. JSX makes the syntax more concise and readable, making it easier for developers to work with React components.

1. **Syntax:** JSX syntax allows you to write HTML-like code directly within your JavaScript. It looks similar to XML or HTML, but it's not a one-to-one mapping.

   const element = <h1>Hello, JSX!</h1>;

2. **Embedding Expressions:** You can embed JavaScript expressions inside JSX by wrapping them with curly braces {}.

   const name = "John";
   const element = <p>Hello, {name}!</p>;

3. **Attributes:** JSX supports HTML-like attributes.

   const element = Visit Example;

4. **Elements and Components:** JSX can represent both HTML elements and React components.

   const element = <div>This is a JSX element</div>;

   const MyComponent = () => {
   return <p>This is a JSX component</p>;
   };

5. **Self-Closing Tags:** Similar to HTML, you can use self-closing tags for elements that don't have content.

   const image = ;

6. **JSX is Not HTML:** JSX may look like HTML, but there are some differences. For example, you

use `className` instead of `class` to define CSS classes, and inline styles are expressed as objects.

const element = <div className="my-class" style={{ color: 'blue' }}>Styled Div</div>;

7. **Using JSX with React:** JSX is often used with React to define the structure of components. React then transforms the JSX into JavaScript that the browser can understand.

```
const MyComponent = () => {
return (
<div>
<h1>Hello, React with JSX!</h1>
</div>
);
};
```

8. **Babel Transpilation:** JSX code needs to be transpiled by tools like Babel to be converted into JavaScript that browsers can interpret. Babel takes JSX and converts it into `React.createElement` calls.

const element = React.createElement('h1', null, 'Hello, JSX!');