



## JSX in React JS

- **JSX (Javascript XML)** is an extension to Javascript syntax that allows you to write **HTML-like code within your Javascript** code when working with React.
- It provides a **concise and expressive** way to describe the structure and appearance of your user interface.

```
import React from "react";

const App = () => {
  return (
    <div>
      <h1>Hello, JSX!</h1>
      <p>This is a React component.</p>
    </div>
  );
};

export default App;
```

- In this example, The code inside the return statement looks similar to HTML, but it's actually JSX.
- JSX elements can also include JavaScript **expressions** inside curly braces **{}**.
- This allows you to embed **dynamic values** or execute JavaScript code within JSX.

```
const name = "Creative Design"
const greeting = <h1>Hello, {name}!</h1>;
```

- **Features of JSX :**

- **Import React:** Begin by importing the React library.
- **One root element:** JSX expressions must have a **single root element**.
- **HTML-like syntax:** JSX uses **HTML**-like syntax to define elements and attributes.
- **Self-closing tags:** Use **self-closing** tags for elements without children.
- **Embed JavaScript expressions within curly braces { }.**
- **Class vs className:** Use **className** instead of class for CSS classes.
- **Inline styles:** Apply inline styles using the style attribute with a **JavaScript object**.
- **Capitalize component** names to differentiate them from HTML elements.
- **Javascript expressions, not statements:** Use JavaScript expressions, **not statements**, inside curly braces.
- **Commenting:** Place comments **within curly braces**.
- **Fragments:** Use fragments to **group elements** without adding extra nodes to the DOM.
- **Event handling:** Define event handlers using **camelCase** naming conventions.
- **Conditional rendering:** Use Javascript **expressions** or **logical operators** for conditional rendering.
- **JSX is not HTML:** Remember that JSX is **transpiled into JavaScript**, not HTML.
- **Keys in lists:** Assign **unique "key"** props to items in lists for efficient rendering.