JavaScript XML (JSX)

JSX stands for JavaScript XML. It is a syntax extension to JavaScript.

JSX is a preprocessor step that adds XML syntax to JavaScript.

JSX produces React “elements”. It is possible to create element without JSX but JSX makes React a lot more elegant.

It is recommended to use JSX with React to describe what the UI should look like.

JSX is easier to read and write. Babel transform these expressions into a actual JavaScript Code.

It also allows React to show more useful error and warning messages.

Examples

Ex: -

const el = <h1> Hello Rahul </h1>	React.createElement("h1", null, "Hello Rahul");
const el = <h1 className="bg">Hello Rahul</h1>	React.createElement("h1", {className: "bg"}, "Hello Rahul");
const el = <h1>Hello {name}</h1>;	React.createElement("h1", null, "Hello ", name);
const el = <Student />	React.createElement(Student, null);
const el = <Student name=“Rahul” />	React.createElement(Student, {name: "Rahul“});

JavaScript Expressions in JSX

We can put any valid JavaScript expression inside the curly braces in JSX. You can pass any JavaScript expression as children, by enclosing it within {}.

Syntax:- {expression}

Ex:-

```
const el = <h1>{10+20}</h1>
const el = <h1> Value: {10+20}</h1>

const name = “Rahul”;
const el = <h1>Hello {name}</h1>

const el = <h1>Hello {show()}</h1>
const el = <h1>Hello {user.firstname}</h1>
```

Specifying Attributes with JSX

You may use quotes to specify string literals as attributes.

Syntax:-

```
const el = <h1 attribute=“value”></h1>
```

Ex:-

```
const el = <h1 className=“bg”>Hello</h1>
const el = <label htmlFor=“name”>Name</label>
```

You may also use curly braces to embed a JavaScript expression in an attribute.

```
const el = <h1 className={ac.tab}>Hello</h1>
```

```
ReactDOM.render(<App name="Rahul" />, document.getElementById("root"));
```

Note –

- Since JSX is closer to JavaScript than to HTML, React DOM uses camelCase property naming convention instead of HTML attribute names.
- Don’t put quotes around curly braces when embedding a JavaScript expression in an attribute. You should either use quotes (for string values) or curly braces (for expressions), but not both in the same attribute.

JSX Represents Objects

Babel compiles JSX down to React.createElement() calls.

```
const el = <h1 className=“bg”>Hello</h1>
```

```
const el = React.createElement("h1", {
  className: "bg"
}, "Hello");
```

```
const el = {
  type: 'h1',
  props: {
    className: 'bg',
    children: 'Hello'
  }
};
```

Nested Elements

You can provide more JSX elements as the children. This is useful for displaying nested components.

Ex:-

```
<MyContainer>
  <MyFirstComponent />
  <MySecondComponent />
</MyContainer>
```