**A Beginner’s Guide to JSX –**

[Sadman Sakib Mugdho](https://medium.com/@ss-mugdho?source=post_page-----d7576dad9fec--------------------------------) Nov 4, 2020

What is JSX and why should you learn it? Let’s talk about it.

JSX looks like a regular HTML in most cases. JSX is an HTML-like syntax used by React that extends ECMAScript so that HTML-like text can co-exist with JavaScript/React code. The syntax is intended to be used by preprocessors (i.e., transpilers like Babel) to transform HTML-like text found in JavaScript files into standard JavaScript objects that a JavaScript engine will parse.

By using JSX you can write HTML-like structures in the same file as you write JavaScript code, then Babel will transform these expressions into actual JavaScript code. Unlike the past, instead of putting JavaScript into HTML, JSX allows us to put HTML into JavaScript.

By using JSX one can write the following JSX code:

var list = (  
 <ul id="list">  
 <li>Apple</li>  
 <li>Orange</li>  
 <li>Banana</li>  
 <li>Mango</li>  
 <li>Jackfruit</li>  
 </ul>  
);

Bable will transform it into this:

var list = React.createElement(  
 "ul",  
 { id: "list" },  
 React.createElement(  
 "li",  
 null,  
 "Apple"  
 ),  
 React.createElement(  
 "li",  
 null,  
 "Orange"  
 ),  
 React.createElement(  
 "li",  
 null,  
 "Banana"  
 ),  
 React.createElement(  
 "li",  
 null,  
 "Mango"  
 ),  
 React.createElement(  
 "li",  
 null,  
 "Jackfruit"  
 )  
)

So, you can think of JSX as a shorthand for calling React.createElement() . JSX is easier to read and write over large pyramids of JavaScript function calls or object literals. Additionally, the React team clearly believes JSX is better suited for defining UI’s than a traditional templating solution.

*Markup and the code that generates it are intimately tied together. Additionally, display logic is often very complex and using template languages to express it becomes cumbersome. We’ve found that the best solution for this problem is to generate HTML and component trees directly from the JavaScript code such that you can use all of the expressive power of a real programming language to build UIs.*

1. JavaScript expressions can be used inside of JSX. We just need to wrap it with curly brackets **{}**.
2. HTML tags always use **lowercase** tag names, while React components start with **Uppercase**.
3. In the case of styling, react recommends using inline styles. When we want to set inline styles, we need to use **camelCase** syntax.
4. You cannot use **if-else** statements inside JSX, instead, you can use **conditional (ternary)** expressions.

You can leverage the full power of JavaScript in HTML and avoid learning or using a templating language. JSX is not a templating solution. Though it looks like a template. It is a declarative syntax used to express a tree structure of UI components.