

What is JSX ?

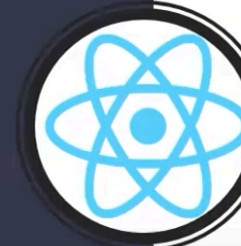
ANS: to render (display the data in the browser) in html and Javascript we written code like below



In Html	With JS
<pre>&lt;ul&gt;   &lt;li&gt;Home&lt;/li&gt;   &lt;li&gt;About&lt;/li&gt; &lt;/ul&gt;</pre>	<pre>let ul = document.createElement('ul'); let li1 = document.createElement('li'); li1.innerText = "Home"; let li2 = document.createElement('li'); li2.innerText = "About"; ul.appendChild(li1); ul.appendChild(li2); document.body.appendChild(ul);</pre>

The same unordered list in React.js, when rendered without using JSX, will look like this:

This is the syntax code to render elements in the browser



```
let li1 = React.createElement("li", {}, "Home")
let li2 = React.createElement("li", {}, "About")
let ul = React.createElement("ul", {}, [li1, li2]);
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(ul);
```

If we observe the above code, rendering elements in the browser using that syntax can be difficult and complex.

To simplify the process of writing code like the example above, the JSX syntax was introduced in React.

## Using JSX

# Using JSX

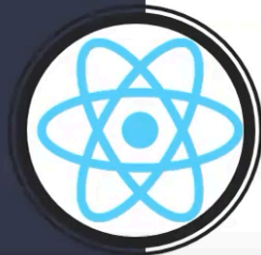
```
let menu = (<ul>
  <li>Home</li>
  <li>About</li>
</ul>)

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(menu);
```

The above code is written using JSX. While it looks like HTML, it's not actually HTML. JSX makes writing and managing code easier and simpler.

If you write code like the above in React, the browser won't understand it directly. A library called **Babel** is used to **internally** convert the **JSX** code into React methods. React then creates a virtual DOM to efficiently render the elements on the screen

Babel converts the code as shown below, then React to create the virtual DOM and render it in the browser.



```
let li1 = React.createElement("li", {}, "Home")

let li2 = React.createElement("li", {}, "About")

let ul = React.createElement("ul", {}, [li1, li2]);

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(ul);
```