**Learn React: Quick Introduction to JSX –**

[Aphinya Dechalert](https://medium.com/@PurpleGreenLemon?source=post_page-----c60708dcbca9--------------------------------) Nov 3, 2021

Part 1 of Learn React from Scratch Series

JSX stands for JavaScript Extension. It is not JavaScript, per se, but an extension of it. This means that if you try and run JSX code in the browser, it won’t work. You’d need a compiler to compile the JSX code into semantic JavaScript before anything can work.

JSX was created by Facebook to solve the problem they had — how do you elegantly combine HTML, CSS, and JavaScript together without it exploding into an unmanageable number of files, modules, and component parts?

And from this issue, JSX was born.

Here is what JSX looks like in its most basic form:

const h1 = <h1>Some text here</h1>;

A line of JSX can include HTML as your variable.

For multiple HTML as your variable, use () . Here is a syntax example of a multi-line JSX:

const footer = (  
 <footer>  
 <p>Copyright dottedsquirrel.com</p>  
 </footer>  
);

A JSX HTML variable can contain id and class attributes as per how you'd use it normally when building interfaces for apps. Here is an example:

const footer = (  
 <footer>  
 <p className="copyright">Copyright dottedsquirrel.com</p>  
 </footer>  
);

When it comes to JSX HTML, a block expression can only have a single parent render. This means that you cannot have multiple blocks of HTML sitting on their own. They need to be wrapped by a single parent.

For example:

const iWontWork = (  
 <div>I won't work</div>  
 <div>Because I am two sibling divs with no parent</div>  
);const wrapMe = (  
 <div className="sampleParent">  
 <div className="children">I will work</div>  
 <div className="children">Because I have a parent wrapper</div>  
 </div>  
);

When you want to render JSX with React, you need to import two modules: React and ReactDOM. Here is how you'd do it:

import React from 'react';  
import ReactDOM from 'react';

ReactDOM is the module that displays your JSX on the screen. To make it work, use .render() method.

render() takes two arguments. The first argument is what you are going to render and the second argument is where you want to render it. Here is an example:

ReactDOM.render(<h1>hi</h1>, document.getElementById('app'));

For document.getElementById(), app can be anything. For example, your HTML may look like this:

<div id='app'></div>  
<div id='secondApp'></div>

Your React JSX render can also do something like this:

ReactDOM.render(<h1>hi</h1>, document.getElementById('app'));  
ReactDOM.render(<h1>hi there</h1>, document.getElementById('secondApp'));

The first render will place hi where app is. The second render will place hi there where secondApp is.

You can also pass in constants and variables into render(). For example:

import React from 'react';  
import ReactDOM from 'react';const hi = <h1>hi</h1>ReactDOM.render(hi, document.getElementById('app'));

And that’s basically JSX in a nutshell.

**React FAQ:**

**Can browsers read/render JSX?**

No. JSX needs to be compiled and no browsers currently support JSX. However, if you want your React project to work, you need to compile it before deployment. To do this, use the following command:

npm run build

npm will compile your JSX code and produce a build file where your deployment bundle is located.

**Is JSX HTML the same as normal HTML?**

A JSX element can contain HTML syntax. It is not HTML itself but you can think of it as a placeholder that can later be used for rendering in your React app.

JSX is a description of what to do and what we see.

**Do I have to use valid HTML tags when creating JSX elements?**

In theory, yes. In practice, it is not always the case. JSX renders whatever tags you put in. How the browser interprets it is based on what else you’ve done with your app. For example, custom tags might be used to render views, which is a collection of other tags.

**Can I create my own JSX element attributes?**

Yes. Custom attributes can be used to pass data between React components.

**Why do we need parentheses around multi-line JSX expressions?**

() lets the compiler know when a block of code starts and ends. It also lets the compiler know that the contents inside is a block to be rendered and not something else like an array (which uses {} instead).

**Can I wrap multi-line JSX expressions in an element other than <div>?**

Yes. As long as there is one parent, it doesn’t matter what element you end up using.

**What’s the difference between React and ReactDOM?**

React is the library for building interfaces. ReactDOM is the library that lets React interact with the DOM. React is a collection of libraries that work together to produce an outcome based on specific structures.

**Is ReactDOM library built into the React library?**

ReactDOM and React are two separate libraries but are often shipped together because of what they do and how they are used together. You can use ReactDOM as a stand-alone library if you wanted to.

**Can I have multiple calls to ReactDOM.render() in a single JavaScript file?**

Yes. But we usually just have one per app but it is possible to have more than one ReactDOM.render(). For it to show, you will need to call it against different element id or class so it knows where to render your view.

**Why use a variable as the first argument in ReactDOM.render() instead of JSX?**

You don’t have to but it is recommended. It’s just easier to read than putting large blocks of JSX into render().