

Many beginners get quite confused when they start using JS in their own code and in fact JS can be a bit tricky to understand and to master. But that's why I'm here by your side, helping you along the way. And so let's quickly check out the rules of how JS works. Now, there are some general rules and there are some rules about how JSX is different from HTML. And starting with the general rules, you should know that JSX works essentially just like HTML. So it has a very similar syntax. However, we can enter a JavaScript mode by using curly braces anywhere in the markup where a value like text or an attribute is expected. Now into this JavaScript mode, we can place any JavaScript expression, so anything that produces a value. So we can reference variables, create arrays or objects, we can loop over arrays using the map method, or we can use operators like the ternary operator. What's not allowed are statements. So in JSX, you cannot use things like an if else statement for loops a switch or any other statement. Now what's super important to understand is that a piece of JSX produces a JavaScript expression, or in other words, a piece of JSX is just like any other JavaScript expression. And this makes sense because we already learned that JSX is simply converted to a create element function call, which is in fact also an expression. Now, this fact has two important implications. First, it means that we can place other pieces of JSX inside the curly braces. So inside the JavaScript node. And again, this is only possible because we can put any JavaScript expression inside those curly braces and that includes the expressions produced by JSX. The second implication of the fact that JSX produces an expression is that we can write JSX anywhere inside a component. For example, we can assign a piece of JSX to a variable like in this code snippet. We can also use it inside an if else statement, pass it into functions and many other things. Finally, a piece of JS can only have one root element. So basically one parent element. If you need more than that, for example, when you need to return two elements from a component, you can use something called a react fragment, which we will talk about later. Okay. And now let's see the differences between JS and regular HTML. Now, I will actually not go through this entire list here at this point because these are just some very simple, straightforward rules which are best explored by using code. But I still wanted to include this list in this slide here so that you can keep it as a reference when you download these slides. So again, we will encounter these rules as we keep writing code, which is exactly what we're going to do now.