

As we finish this first dive into state, I want to share a few more important thoughts or ideas about state as well as some practical guidelines. So first of all, there is one important technical detail that you should be aware of, and this might seem obvious, but it's still worth mentioning. So what I'm talking about is the fact that each component really has and manages its own state. So even if we render the same component multiple times on one page, each of these component instances will operate independently from all the other ones. So in this example, the three counter components all start with a piece of state called Score, which is set initially to zero. Then if one of the buttons is clicked, that increases the score by one for each click, but only in that component. The state in all the other components stays the same. So again, if we change the state in one of the components that won't affect the other components at all. And so the same thing, of course, is going to happen when we click one of the other buttons or even when one of the components is removed from the UI entirely. So state really is isolated inside of each component. Now, if we analyze everything that we just learned about state, we can come to the conclusion that we can basically think of the entire application view. So the entire user interface as a function of state. Or in other words, the entire UI is always a representation of all the current states in all components. And taking this idea even one step further, a React application is fundamentally all about changing state over time and of course also correctly displaying that state at all times. And this is really what the declarative approach to building user interfaces is all about. So instead of viewing a UI as explicit Dom manipulations with state, we now view a UI as a reflection of data changing over time. And as you know by now, we describe that reflection of data using state event handlers and JSX. So we describe the UI, react does the rest. Now, this might all sound a bit philosophical at this point in your journey, but trust me, as you become more and more experienced in building React apps and working with State, you will truly and deeply understand everything I just said here. Now. Okay. And now to finish, let me give you a few guidelines on how to use state in practice. So practical guidelines is always what students like the most. And here, this also kind of serves as a summary of state in general. And these guidelines are for you to keep as a reference. So there is a lot of text here that I will just quickly go over now. So first of all, you should create a new state variable for any data that a component should keep track of over time. And the easy way of figuring this out is to think of variables that need to change at some point in the future. So if you're used to building apps in vanilla JavaScript, those would be variables defined with Let or with Var or also an array or object that you mutate over the application's lifecycle. So in React, you use state for those. Another way of figuring out when you need state is this Whenever you want something in a component to be dynamic, create a piece of state related to that thing and then update the state when the thing should change. Or in other words, when you need it to be dynamic. Now, since this thing is a bit abstract, let's think of a modal window that can be either open or closed. So for a modal window we can create a state variable called is open that will keep track of whether the modal is currently open or not. Then when is open is true, we display the window on the screen and if it's false, we hide it. Simple, right? So whenever you want to change the way a component looks like or the data that it displays, just update its state, which you usually do inside an event handler function. Now, when you're actually building your components, it's going to be useful to always imagine the components view. So the component rendered on the screen as a reflection of state changing and evolving over time. Finally, there is one common mistake that many beginners make, which is to use state for every single variable that you need in a component. But that's really not necessary. So do not use state for variables that should not trigger a rerender. Okay. Because that will just cause unnecessary rerenders which can cause performance issues. So it's very common to need some variables that are not state. And so for those you can just use regular variables defined with const. But we will come back to this in the next section. All right. So this is my first set of guidelines about state, which should be more than enough for now. So if you truly internalize these, then building react applications in the future should be a lot easier for you. And I say this because I really believe that mastering state is the most difficult part of learning react. But once you overcome this hurdle and truly internalize when you need state and how it all works, it will unlock react development for you. And so that's why I spent so much time here showing you how state works.