

Let's start this section by discovering what thinking in React is actually all about. So as you might have noticed by now, building React applications requires a completely new mindset because it's just very different from building applications with vanilla JavaScript. So to build React apps, you not only need to learn how to work with the React API in practice, like with all the different functions like use state, but you also need to be able to think in react. So you need to basically enter the React mindset. Now once you have both of these skills, you will have mastered react and you will be well on your way to building professional react applications. Now, what does thinking in React actually mean? Well, while you're building an application, thinking in React means that you have a very good mental model of how and when to use all the react tools like components, state props, general data flow effects, and many more. It's also about always thinking in terms of state transitions rather than in element mutations, as we have learned before. Now, you can also view thinking in React as a whole process, which can help us build apps in a more structured way. And the first step in this process is to break the desired UI into components and establish how these components are related to one another. So to establish the component tree. This also includes thinking about reusability and composability of components. After that, we can start by building a static version of the application. So without any state and interactivity. And this is great because by doing this we do most of the coding up front before having to worry about state and interactivity. That part comes next. So in step three, where we think about state. So here we decide when we need state, what types of state we need and where to place each piece of state. Then finally, we think about establishing how data flows through the application. This includes thinking about one way data flow, child to parent communication and the way global state should be accessed. So these points three and four is what we collectively call state management, which is the main focus of this section. Now, of course, this is not a rigid process that we always need to follow to the letter In practice. There's always a lot of back and forth between these different steps, and things are never this linear. But it's still good to have a process like this as an overall guideline. Okay, Now once you really know how to think in React, you'll be able to answer questions like how to break up my UI, design into components, how to make some of my components truly reusable, or how to assemble a user interface from reusable components. Now, we also think a lot about state, like what pieces of state do I need for the interactivity that I want and where to then place each of these states? Or in other words, what component should actually own each piece of state or what types of state can or should I use? And in more general terms, how to make my data flow through the application. Now, as you might have noticed, we already started to talk about some of these topics before, but I still wanted to have one section where I really focus on some of these skills so that you can start getting more and more into the React mindset. Now, of course, you will only really master these skills through practice and writing code, and lucky for you, we will do lots of that in this course, but we will also take a bit of a theoretical approach from time to time, just like in this video, because I believe that it's really, really important that I teach you these things besides just the React API itself. And by doing so, I'll make this course really as good as possible for you and in my opinion, way better than all the other React courses out there. But anyway, let's now move on to looking at the fundamentals of state management.