

At this point. We have used to describe the appearance of some components and we have also used some JavaScript inside of them. And so now that we have a tiny bit of experience in writing components, I want to take a minute and go back to the fact that combines HTML, CSS and JavaScript into one single block of code, because you might be wondering why did React come up with this idea in the first place? So why not just keep HTML, CSS and JavaScript in separate places like we have always done before? And this might sound like a trivial question, something you think is not really relevant at all, but it's actually deeply relevant to understand why React was completely designed around components. And let's understand this topic from the very beginning. So from the rise of interactive single page applications, so before single page apps, we always had one file for HTML, one for JavaScript and one for CSS. So basically one technology per file. That was our traditional separation of concerns. And I'm pretty sure that just like me, this is how you first learned web development. However, as pages got more and more interactive, they became single page applications where the JavaScript started to determine the user interface and the content in general, or in other words, JavaScript became more and more in charge of the HTML. And we can see that here in this really small code example where the content and the presentation of these HTML elements are really completely determined by the JavaScript code. They are in fact tightly coupled together. So the HTML doesn't even make sense without the JavaScript here. Now the details of this code are really not important. So if you can't read this code, that's no problem at all. My point here is that if the JavaScript is in charge of the HTML anyway, so if the logic and the UI are so tightly coupled together, then why should we keep them separated in these different files and in different code blocks? Well, the answer to that question is what gave us React components and JS. So the fact that logic and UI are so coupled in modern web applications is really the reason why a react component contains the data, the logic and the appearance of one piece of the UI. In fact, it's the fundamental reason why React is all about components. And the same is actually also true for most other modern front end frameworks. Now returning to some code. In this React example, we can see how JavaScript and HTML markup live very happily together in this one single component. So this component has some JavaScript logic, it has JSX. And then inside that JSX there is yet another block of JavaScript which in turn has even more JSX inside of it. So everything is mixed together, but it all still works really well. So content and logic are tightly coupled together and so it makes sense that they are co-located here. And co-located simply means that things that change together should be located as close as possible together. And in the case of React apps, that means that instead of one technology per file, we have one component per file. So one component that contains data, logic and appearance all mixed together. Now when React and JS first came out a long time ago, many, many people just hated the way that JS looks like and they hate it that we are throwing separations of concerns out of the window. But actually, are we really? Is there really no separation of concerns in React? Well, I think that the people who say that React has no separation of concerns got it all wrong because React does actually have separation of concerns. It's just not one concern per file as we had traditionally, but one concern per component. So each component is in fact only concerned with one piece of the UI. Then within each of these components, of course, we still have the three concerns of HTML, CSS and JavaScript all mixed up as we have been discussing. So compared to the traditional separation of concerns, this is a completely new paradigm that many people were really not used to in the beginning. But now, many years later, we all got used to this and it works just great. So having all the information about a certain component in one single place really does work in an amazing way. All right. So this was a long lecture, longer than I had expected just to arrive at this conclusion that React does actually have separation of concerns, just a different separation of concerns. But I still hope that you liked the way how we arrived at that conclusion over the course of this video, because hopefully this gave you a lot of additional fundamental react knowledge.