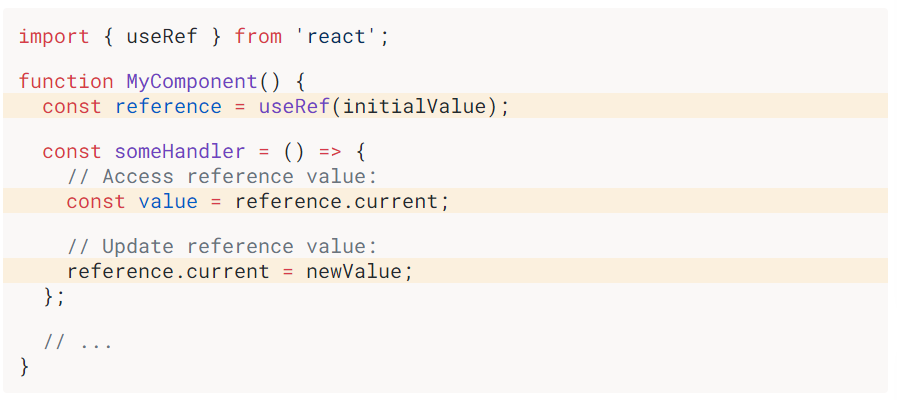
useRef



useRef –bu react hook hisoblanib,u o`zida boshlang`ich qiymat sifatida 1ta argumentni o`zida qabul qiladi va reference qaytaradi. Reference – bu maxsus current degand property ga ega object hisoblanadi

const value = reference.current; => orqli reference qiymatidan foydalanishimiz mumkun

reference.current = newValue; => orqali reference qiymatini yangilashimiz mumkun

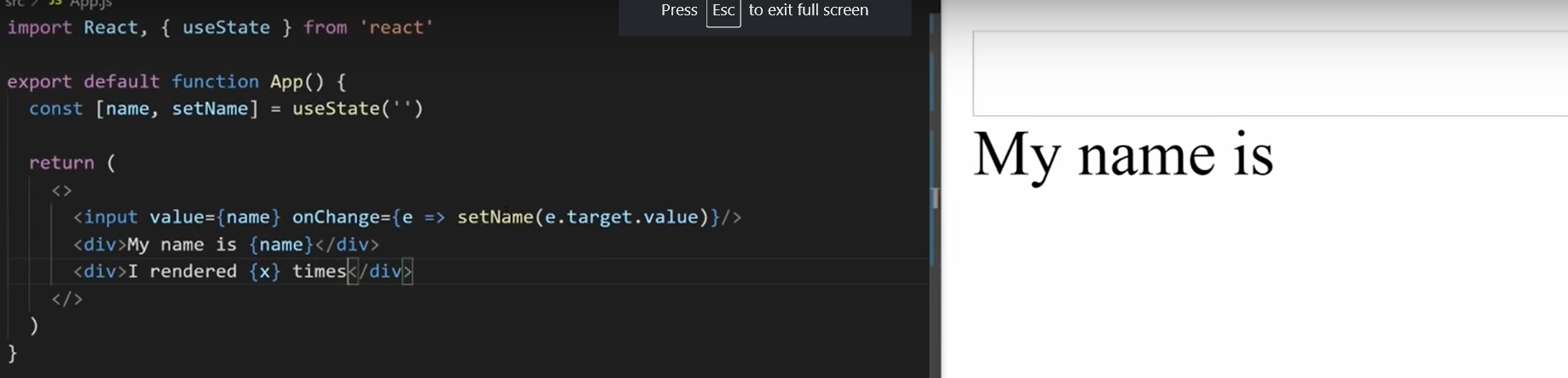
Reference ga tegishli 2ta qoida mavjud:

1. Component ta qayta rendender bo`lganda reference ning qiymati saqlanib qoladi
2. Reference qiymati yangilanganda componentani qayta render qilmaydi.

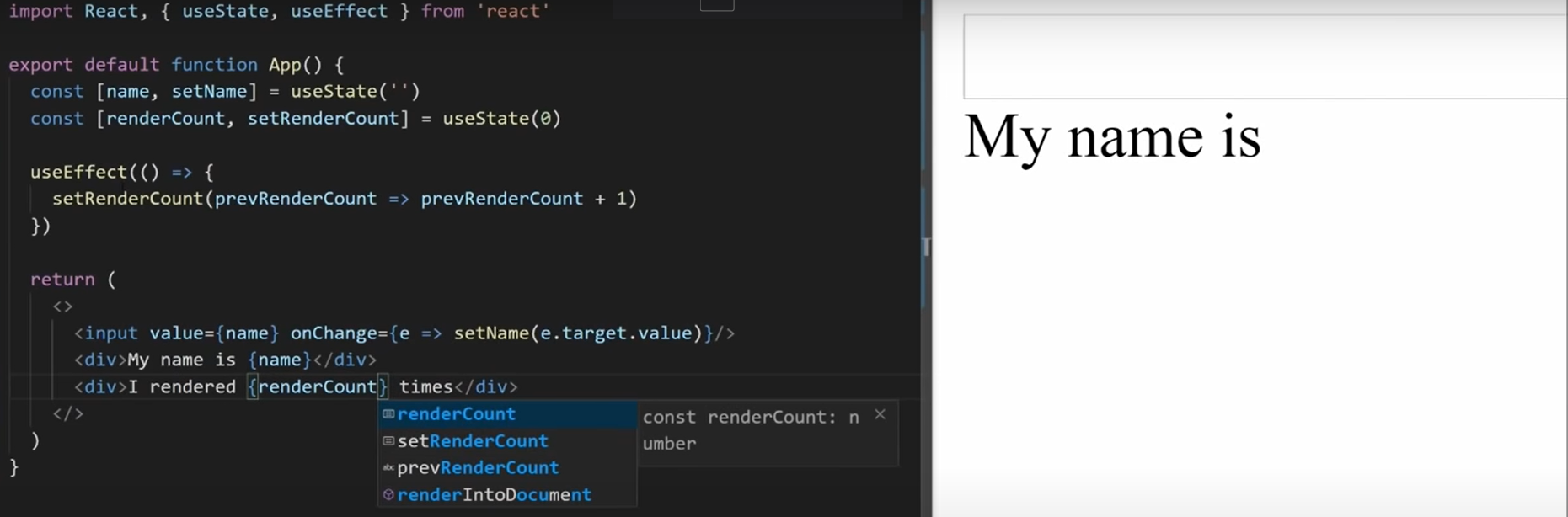
useRef

in order to start talking about what useRef is instead of getting into the nitty gritty details of the implementation of it I wanna talk about what you want to do if you wanted to show the number of times a component rendered on the screen.

So if we wanted to have another div that says I rendered and then some variable and here {x} times.



How you go about implementing this. Well most people would think, ok I`m gonna to create some state variable called renderCount and setRenderCount than set that to useState and we`re going to default it to zero and then every time we render our component we can do that instead of a useEffect so now what we want to do is every single time that our component rerenders which is just a basic useEffect we want to set our renderCount to prevRenderCount and we want to do prevRenderCount + 1. So this would correctly update our render count and we could print that out down here. So this would correctly update our render count every time we render our component.



But huge huge red flag with this is that when you update your state you cause the component to render so first time I component renders is gonna set this state which causes it to re-render



And then just gonna set the state again and re-render and re-render and re-render essentially you`re setting yourself up inside of an infinite loop



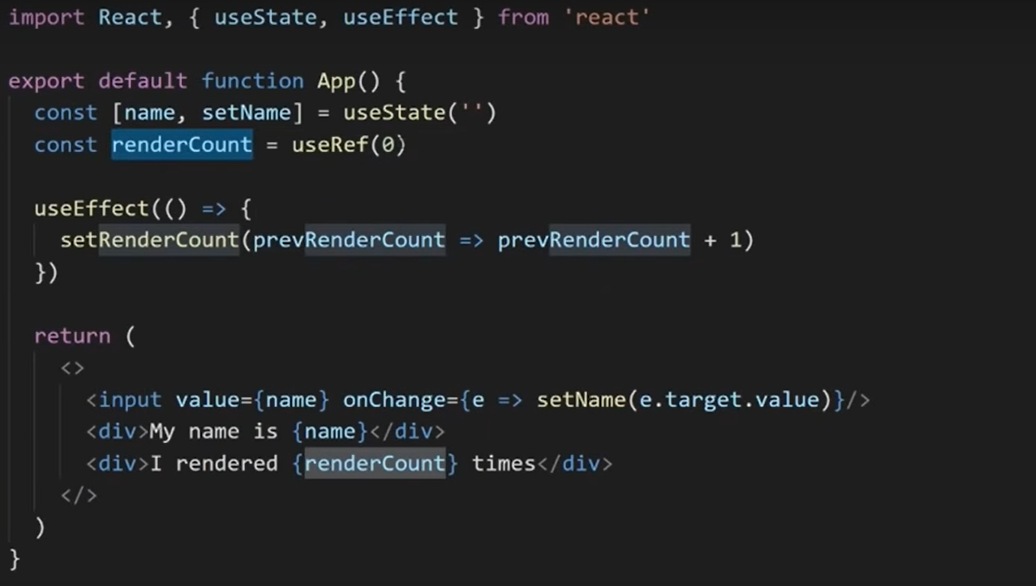
I`m not actually gonna save this or run it because it will throw an infinite loop and I don`t wanna deal with that but just know that state is not the way to do this because you can`t do it with state there is no way to do that because state will just cause you to get into that infinite loop.

The solution through is to use something called refs.

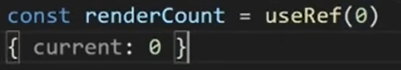
The ref – is very similar to state and that it persist between renders of your components but

The important thing to note - about refs is that a ref does not cause your component to re update when it gets changed.

So, instead of using useState here let`s do something called a useRef and we again pass it our default just like with this zero but instead of returning two values, it`s going to return just one single value. We gonna call renderCount

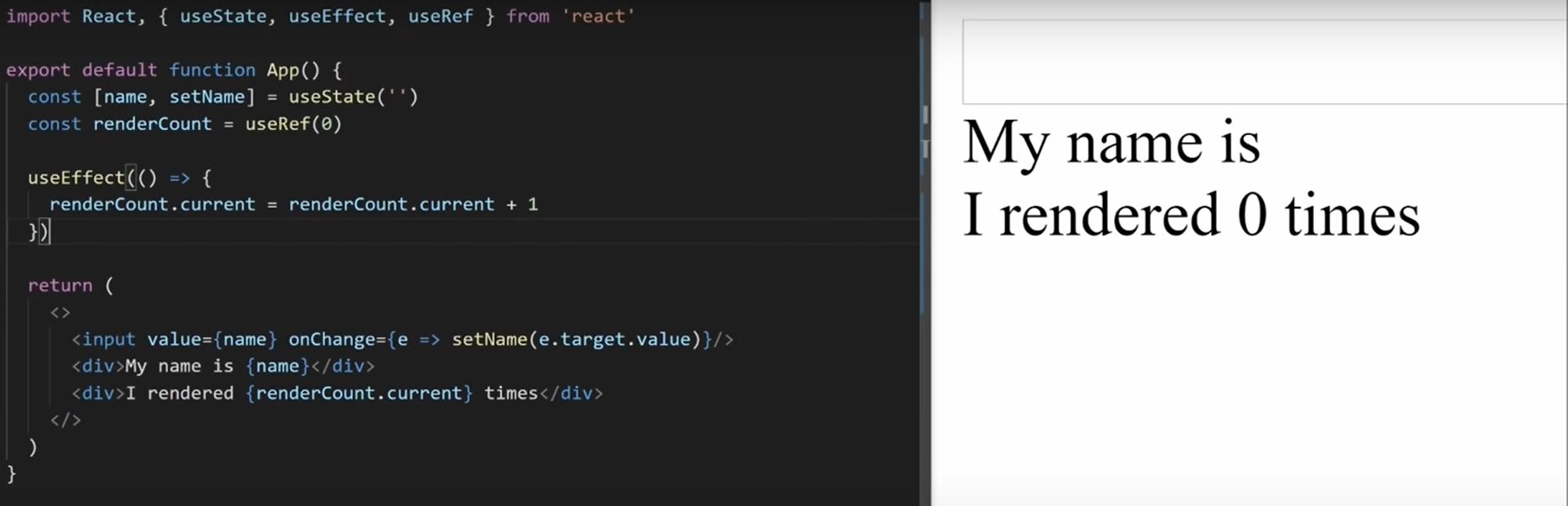


And the important thing about this is actually returns an object and this object looks like this



it just has a single property called current and it`s going to by default to set that current value to zero. That`s what we passed into useRef here

So rendreCount here is just object with a current property and when we update that current property that is what gets persisted between our different renders.



So instead of here we can take our render count and we can take the current property on it and just set that equal to renderCount.current + 1

The important thing to know is that we can change this renderCount.current as much as we want as many times as we want and we`ll never ever cause our component to re-render because it`s completely separate from our component render cycle.

So the important thing about ref is very very similar to state and that you can store a previous value in it and it persists between different renders but it does not cause you re-render like state does so that`s the basic use cases for using refs but probably the biggest use case that people are going to use refs for is reference elements inside of your HTML and this is actually so popular. That each element inside of your document has a ref attribute and you can set that to any ref that you want.