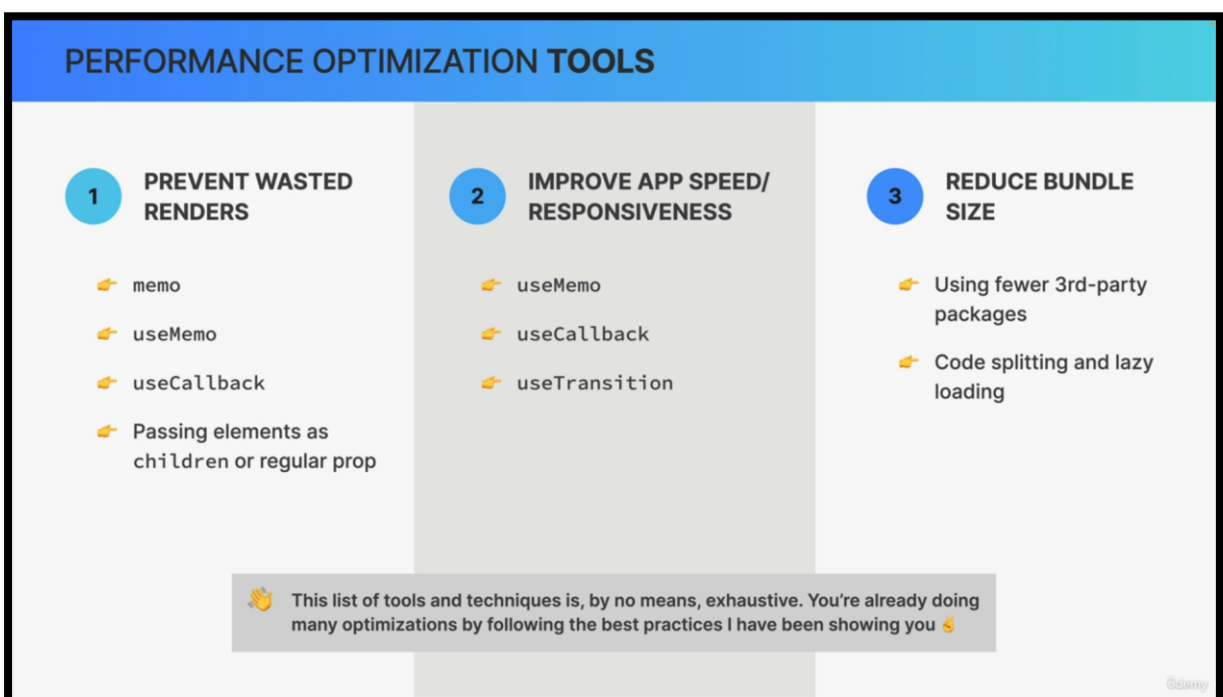


There are three main areas that we can focus on when we need to optimize performance of React apps.

First, we can try to prevent wasted renders.

Second, we can improve the overall app speed and responsiveness to make sure that the app is 100% fluid and without delays.

Third, we can reduce the bundle size.



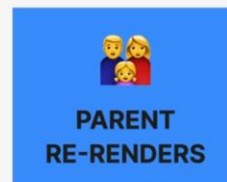
WHEN DOES A COMPONENTS INSTANCE RE-RENDER?

👉 A component instance only gets re-rendered in 3 different situations:

In React, a component instance only gets re-rendered in three different situations.

WHEN DOES A COMPONENTS INSTANCE RE-RENDER?

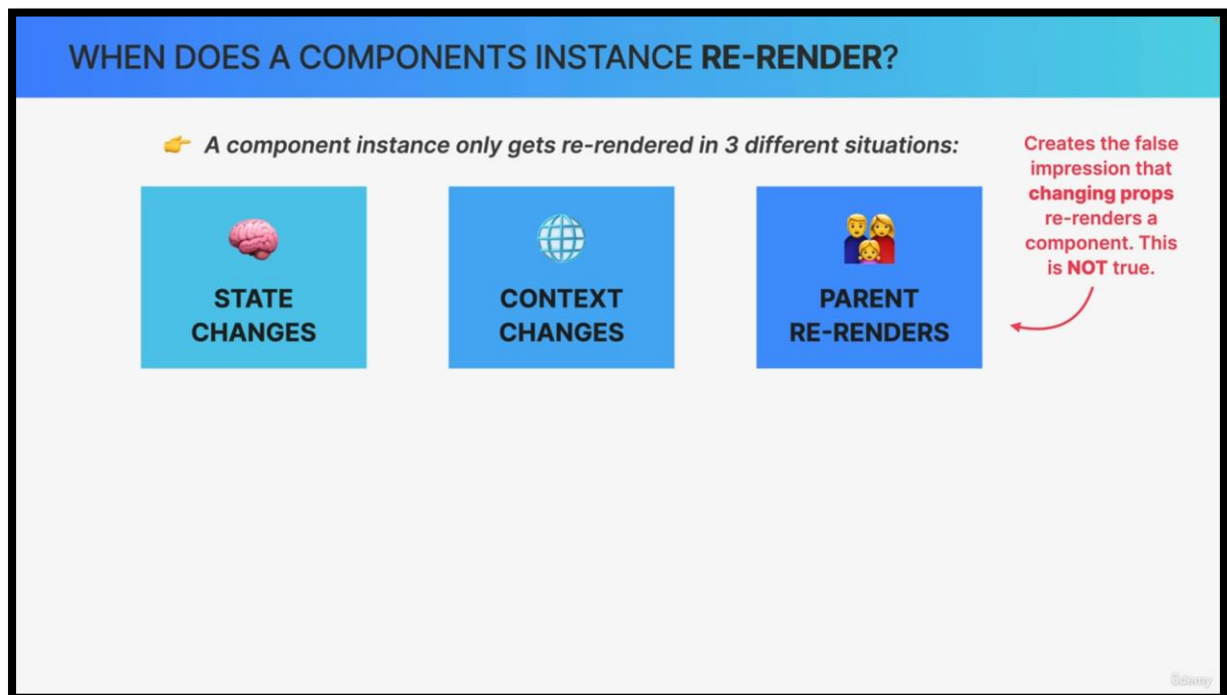
👉 A component instance only gets re-rendered in 3 different situations:



First, when the component state changes.

Second, a component instance gets re-rendered whenever there's a change in a context that the component is subscribed to.

Finally, whenever a component re-renders, all its child components will automatically be re-rendered as well. So, the third reason for a component to re-render is a parent component re-rendering.



You might be wondering what about prop changes? Doesn't updating props also re-render a component?

Well, actually, that's technically not the case. So, that's a common misconception.


Actually, even we told you this at the beginning of the course, but it was only because you didn't know yet how rendering works.

Because it is true that it does look as if components re-render when their props change but what actually happens is that props only change when the parent re-renders but when the parent re-renders, the children who receives the prop will re-render anyway.


So, the real reason why a component re-renders when props change is that the parent has re-rendered.

WHEN DOES A COMPONENTS INSTANCE RE-RENDER?


👉 A component instance only gets re-rendered in 3 different situations:



**STATE
CHANGES**



**CONTEXT
CHANGES**



**PARENT
RE-RENDERS**

Creates the false impression that **changing props** re-renders a component. This is **NOT** true.

👉 Remember: a render does *not* mean that the DOM actually gets updated, it just means the component function gets called. But this can be an expensive operation.


Scrimby

It's important to also remember that rendering a component does not mean that the DOM actually gets updated. So, all that rendering means is that the component function gets called and that React will create a new virtual DOM and to all the diffing and reconciliation.


This can be an expensive and wasteful operation which brings us to an important topic i.e. wasted renders.

WHEN DOES A COMPONENTS INSTANCE RE-RENDER?


👉 A component instance only gets re-rendered in 3 different situations:



**STATE
CHANGES**



**CONTEXT
CHANGES**



**PARENT
RE-RENDERS**

Creates the false impression that **changing props** re-renders a component. This is **NOT** true.

👉 Remember: a render does *not* mean that the DOM actually gets updated, it just means the component function gets called. But this can be an expensive operation.

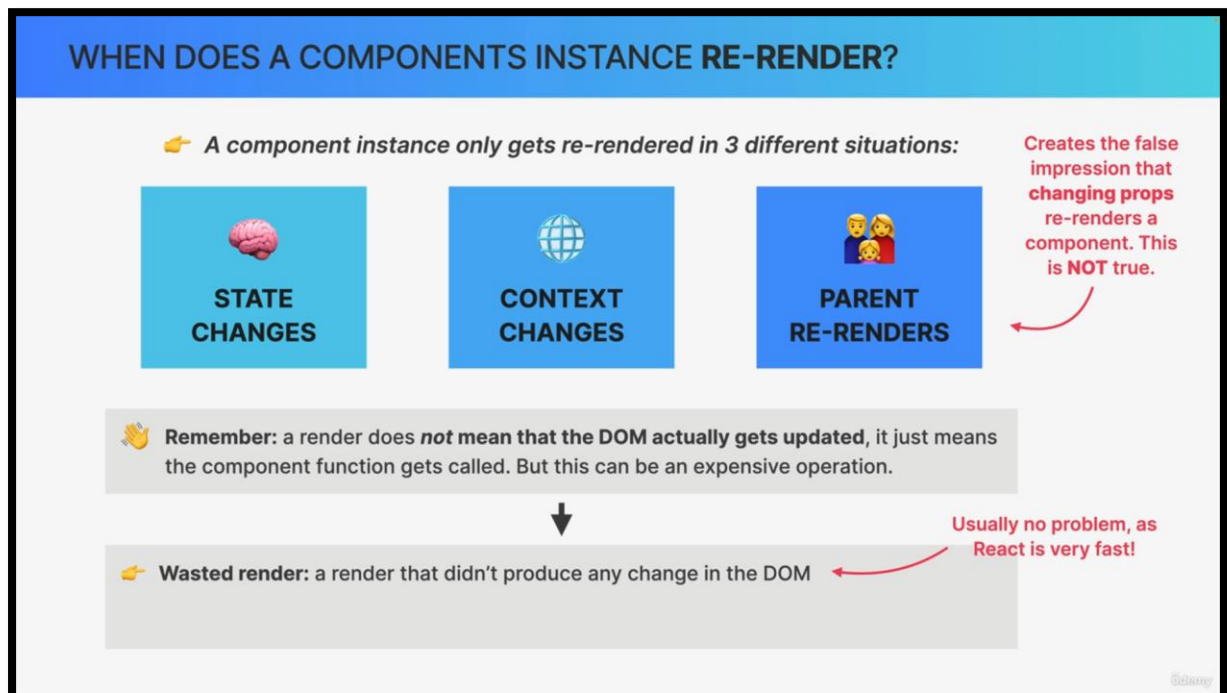
↓

👉 **Wasted render:** a render that didn't produce any change in the DOM

Scrimby

So, a wasted render is basically a render that didn't produce any change in the DOM. So, it's a waste because all the diffing calculations still

had to be done but it didn't result in any new DOM and so therefore, all the calculations were for nothing.



Most of the time, this is actually no problem at all because React is very fast.

However, it can become a problem when re-renders happen way too frequently or when the component is very slow in rendering.

So, this can then make the application feel leggy and unresponsive. For example, not updating the UI fast enough after the user performs a certain action.

So, we want to avoid situations like that at all costs.