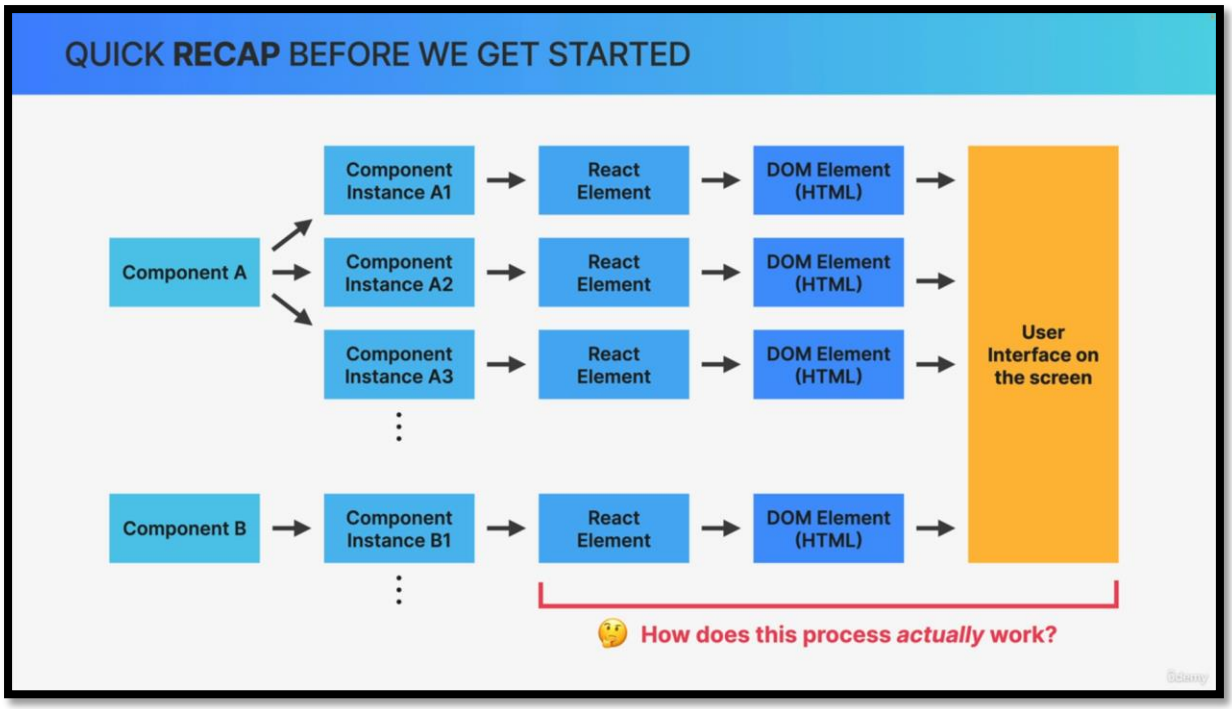
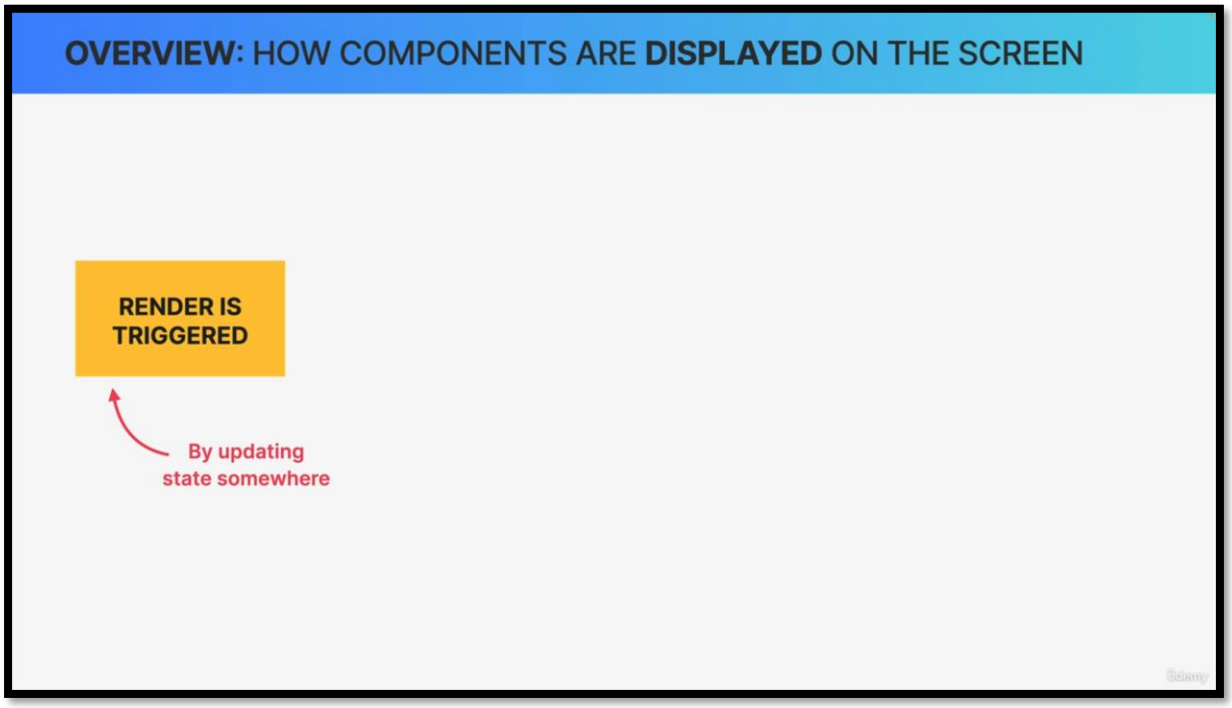
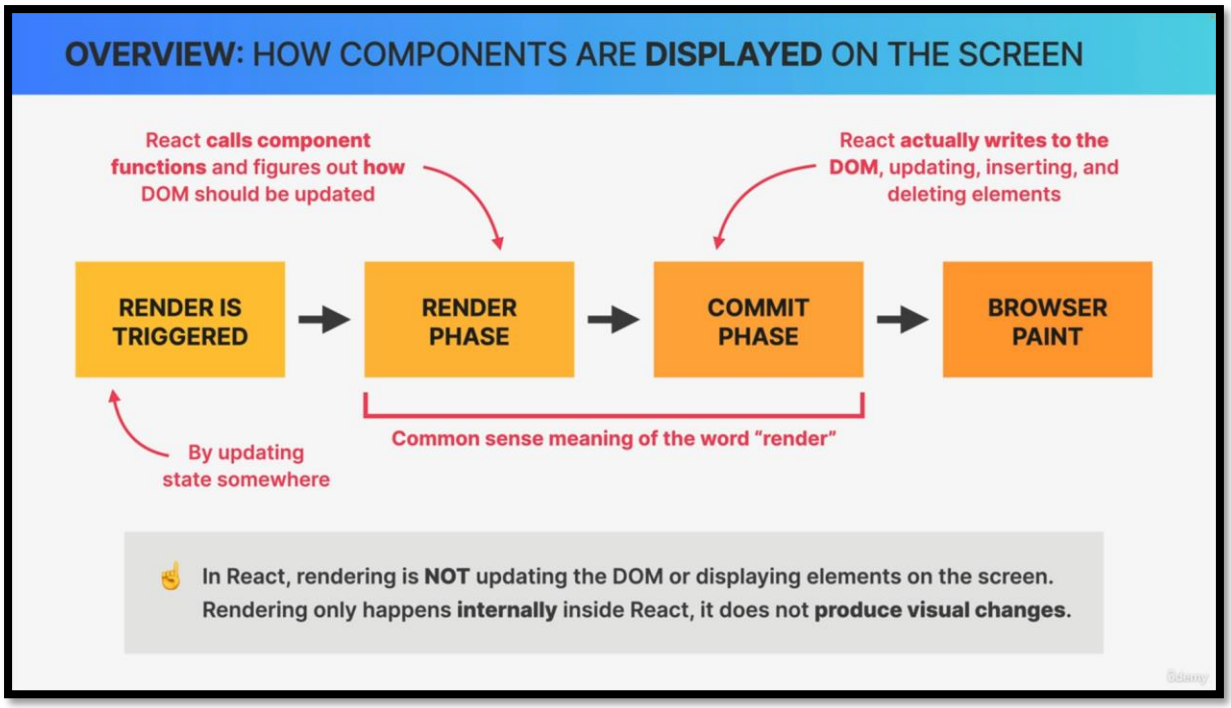




State changes trigger renders and so it makes sense that the next phase is the render phase.



Render Phase

In the render phase, React calls our component functions and figures out how it should update the DOM in order to reflect the latest state changes.

However, it does actually not update the DOM in this phase. React's definition of render is very different from what we usually think of as a render which can be quite confusing. Rendering is not about updating the DOM or displaying elements on the screen.

Rendering only happens internally inside of React and so it does not produce any visual changes.

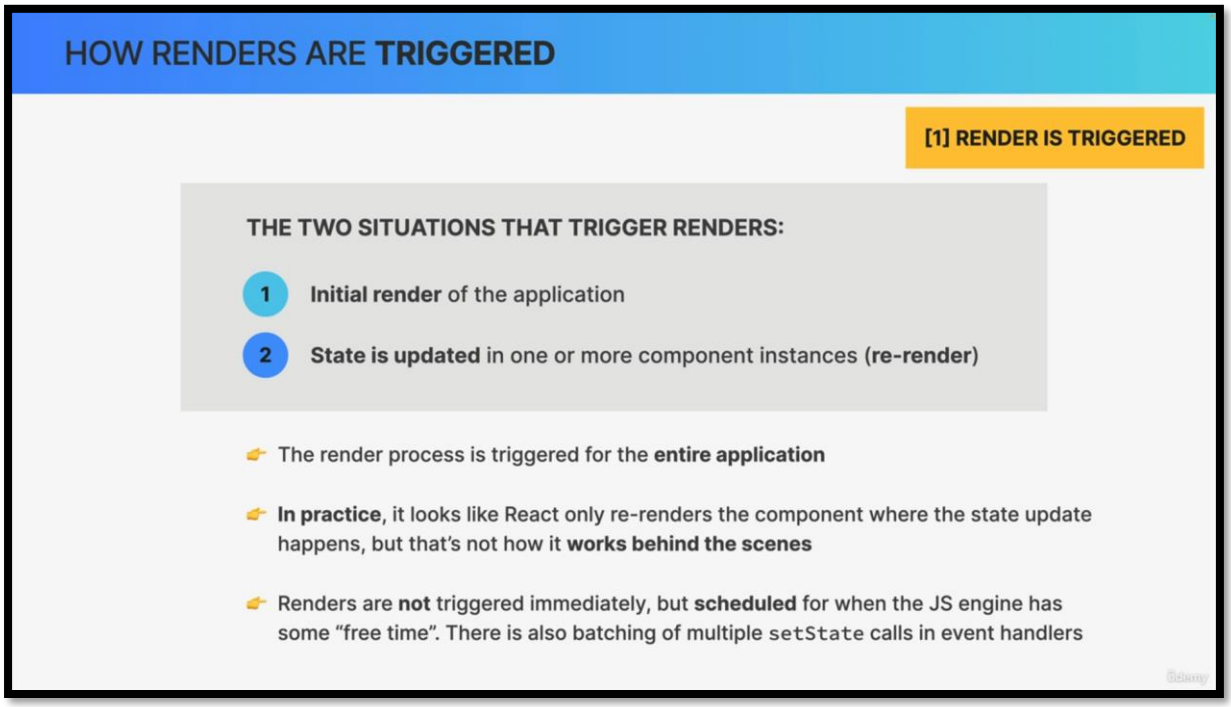
In all the previous sections, we have always used the term rendering with the meaning of displaying elements on the screen because that was just easy to understand.

Commit Phase

Once React knows how to update a DOM it does so in the commit phase. In this phase, new elements might be placed in the DOM and already existing elements might get updated or deleted in order to correctly reflect the current state of the application.

Finally, the browser will notice that the DOM has been updated and it repaints the screen. This final step that actually produces the visual change that users see on their screens.

How renders are triggered?



There are only two ways in which a render can be triggered.

1. The first one is the very first time the application runs which is what we call the initial render.

2. The second one is a state update happening in one or more component instances somewhere in the application which is what we call a re-render.

It's important to note that the render process really is triggered for the entire application, not just for one single component. Now that doesn't mean that the entire DOM is updated. Because in React, rendering is only about calling the component functions and figuring out what needs to change in the DOM later.

React looks at the entire tree whenever a render happens.

A render is actually not triggered immediately after a state update happens. Instead, it's scheduled for when the JavaScript engine basically has some free time on its hands. But this difference is usually just a few milliseconds that we won't notice.