Before we go into this let’s understand initial rendering and re-rendering

1. **Initial Rendering**This occurs whenever a component is shown for the first time.
2. **Re-rendering**This refers to subsequent renders of a component that is already visible on the screen. Re-renders are typically triggered when React needs to update the app components with new data.

**Initial rendering** is necessary for every component to display it on the screen, and we cannot do anything to remove it, so we will focus on re-rendering, its common causes, and how to avoid it in React

**Re-rendering** can happen due to user interactions (e.g., button clicks, changes in text input values), the arrival of external data through asynchronous requests (e.g., API calls, socket connections), or updates from a subscription model (e.g., listeners).

**How does rendering works in react**

React uses a Virtual DOM as a layer between the developer’s description of how the UI should appear and the actual rendering effort performed by the application. To render UI in the browser, the app must modify the Document Object Model (DOM) of the browser.

A diagram of a diagram of a virtual dom and a subtree

AI-generated content may be incorrect.

**How is re-rendering triggered?**

After the initial rendering has been completed, there are a few different ways to tell React to queue a re-render:

1. Re-render the component when the state and/or props change
2. Change in the value of the Context Store
3. Re-render when change in key prop

**How to prevent unnecessary Re-rendering**

There are two types of re-rendering

1. Necessary re-rendering

After a component’s state or props change, it should refresh itself and related components.

1. Unnecessary re-rendering

App components in React may refresh more frequently than necessary.

React provides two hooks based on memorization to improve performance on re-renders:

* useMemo()
* useCallback()