Lists and Hooks

Ques: Explain Life cycle in Class Component and functional component with Hooks.

Answer:

In React, a **component** is like a building block of your application. Think of it as a small part of a webpage, like a button or a user profile.

*Class components;*

Class components have a defined lifecycle with several methods that can be overridden to run code at specific points in a component’s life.

Mounting: This is when the component is created and added to the webpage.

Constructor: This is where you set up the component, like creating a variable to keep track of information.

Render: This function returns what the component should look like (like the HTML).

ComponentDidMount: This runs right after the component appears on the page. You might use it to load data from an API (like getting information from a server).

Updating: This happens when the component's data changes.

ShouldComponentUpdate: This checks if the component needs to update or not. It can help make your app faster.

Render: Called again to update what the component looks like.

ComponentDidUpdate: This runs after the component has updated, useful for doing things like updating charts or graphs.

Unmounting: This is when the component is removed from the webpage.

ComponentWillUnmount: This is where you clean up things, like stopping timers or removing event listeners, so you don’t waste memory.

Functional Components with Hooks

Functional components are a simpler way to create components. With hooks, you can do similar things without needing to write a lot of extra code.

Mounting:

You use a hook called useEffect() to run code when the component is added to the page. It’s like componentDidMount.

If you want to clean up (like stopping a timer), you can do that inside useEffect() too.

Updating:

You can use useEffect() again, but this time you tell it to run whenever certain data changes. It’s like componentDidUpdate.

Unmounting:

Inside useEffect(), you can return a cleanup function. This runs when the component is removed from the page, just like componentWillUnmount.