1. Need and Benefits of Component Lifecycle

What is the Component Lifecycle?

The **component lifecycle** in React refers to the **series of phases** a component goes through during its existence:

- **Mounting** (when the component is being created and inserted into the DOM)
- **Updating** (when state or props change)
- **Unmounting** (when the component is removed from the DOM)

Why is it Needed?

- To **control what happens when** a component:
    - First appears on the screen
    - Gets updated with new data
    - Is removed from the page
- To **perform side effects** such as:
    - API calls
    - Setting timers
    - Subscribing to data streams
    - Cleaning up resources

Benefits:

Benefit	**Description**
Control	Gives control over what happens at each phase of the component's life
Optimization	Helps optimize performance by managing updates efficiently
External Interaction	Useful for interacting with APIs or the browser (e.g., event listeners, local storage)
Cleanup	Ensures proper cleanup (like removing intervals or event listeners)

2. Lifecycle Hook Methods

These are special methods you define in a **class component** to run code at specific points.

Mounting (Initial Render)

Method	Purpose
constructor()	Initialize state and bind methods
static getDerivedStateFromProps()	Sync state from props before render (rare)
render()	Returns JSX (required)
componentDidMount()	Runs once after the component is added to the DOM (good for API calls)

Updating (Props or State Changes)

Method	Purpose
static getDerivedStateFromProps()	Called before every render when props/state changes
shouldComponentUpdate()	Determines if the component should re-render
render()	Updates UI
getSnapshotBeforeUpdate()	Captures values (like scroll position) before DOM changes
componentDidUpdate()	Called after the DOM has been updated

Unmounting (Component Removal)

Method	Purpose
componentWillUnmount()	Cleanup tasks like clearing intervals, unsubscribing from APIs

3. Sequence of Steps in Rendering a Component

When a Component is Mounted:

1. constructor()
2. getDerivedStateFromProps()
3. render()
4. componentDidMount()

When a Component is Updated (due to props or state):

1. getDerivedStateFromProps()
2. shouldComponentUpdate()
3. render()
4. getSnapshotBeforeUpdate()
5. componentDidUpdate()