**Need and Benefits of Component Lifecycle**

* **Need:** The component lifecycle helps React manage what happens before, during, and after a component appears on the page. It gives you control over actions like fetching data, setting up timers, or cleaning up when a component is removed.
* **Benefits:**
  + Lets you run code at the perfect moment (e.g., after the component shows up, before it changes, or before it disappears).
  + Helps manage resources (like starting/stopping timers or network requests).
  + Makes building interactive, dynamic apps much easier and organized.

**Various Lifecycle Hook Methods**

In **class components**, the most common lifecycle methods are:

1. **constructor()**
   * Runs first, sets up initial state.
2. **render()**
   * Decides what appears on the page.
3. **componentDidMount()**
   * Runs once after the component shows up. Good for data fetching or starting things.
4. **componentDidUpdate(prevProps, prevState)**
   * Runs every time something changes (props or state), after rendering.
5. **componentWillUnmount()**
   * Runs before the component is removed. Use it for cleanup.

In **function components**, you use **React Hooks** like useEffect() for similar actions.

**Sequence of Steps in Rendering a Component**

When a React component is used, the order is:

1. **constructor()** (only class component – sets up)
2. **render()** (decides what to display)
3. **componentDidMount()** (after it first appears)

When the component updates (due to new props or state):

1. **render()** (again, to show the new output)
2. **componentDidUpdate()** (after the change is visible)

When the component is about to be removed:

1. **componentWillUnmount()**

You can remember it like:  
**Mount (appear)** → **Update (change)** → **Unmount (disappear)**