### ****Why Lifecycle Methods Are Needed****

React components go through a **lifecycle** from creation to deletion. Lifecycle methods let you **run code at specific points** — like fetching data after the component appears or handling errors.

#### ****Benefits:****

Manage data loading (e.g., API calls)

Handle cleanup (e.g., event listeners)

Control rendering behavior

Catch and handle errors

**Types of Lifecycle Phases**

React class components go through three main phases:

**Mounting** – Component is being created and inserted into the DOM

**Updating** – Component is being re-rendered due to changes in state or props

**Unmounting** – Component is being removed from the DOM

**Error Handling** – Catching and handling errors in child components

**Common Lifecycle Hook Methods**

| **Phase** | **Lifecycle Methods** |
| --- | --- |
| Mounting | constructor(), componentDidMount() |
| Updating | shouldComponentUpdate(), componentDidUpdate() |
| Unmounting | componentWillUnmount() |
| Error | componentDidCatch() |

**Sequence of Lifecycle Execution (Mounting)**

constructor() – Initializes state

render() – Renders the UI

componentDidMount() – Called after the component is rendered

Practical Task Summary (Blog App Project)

**App Name:**

**blogapp**

**Component Overview:** Post.js

Defines a class-based component representing a single blog post with title and body.

#### Posts.js

Another class-based component that:

Initializes a **list of posts** using state

Implements loadPosts() method using the **Fetch API** to get data from:  
https://jsonplaceholder.typicode.com/posts

**Lifecycle Methods Used:**

componentDidMount()

Called after the component is mounted

Invokes loadPosts() to fetch and set post data

render()

Displays the list of posts (title as <h2> and body as <p>) in the UI

componentDidCatch()

Catches and handles any runtime errors during rendering or inside lifecycle methods

Displays errors using an alert

**Final Output in Browser:**

When the app is opened at http://localhost:3000, it displays a list of blog posts fetched from the API.  
Each post includes:

A title as a heading

The body as a paragraph  
If any error occurs, it is caught by componentDidCatch() and shown as an alert.