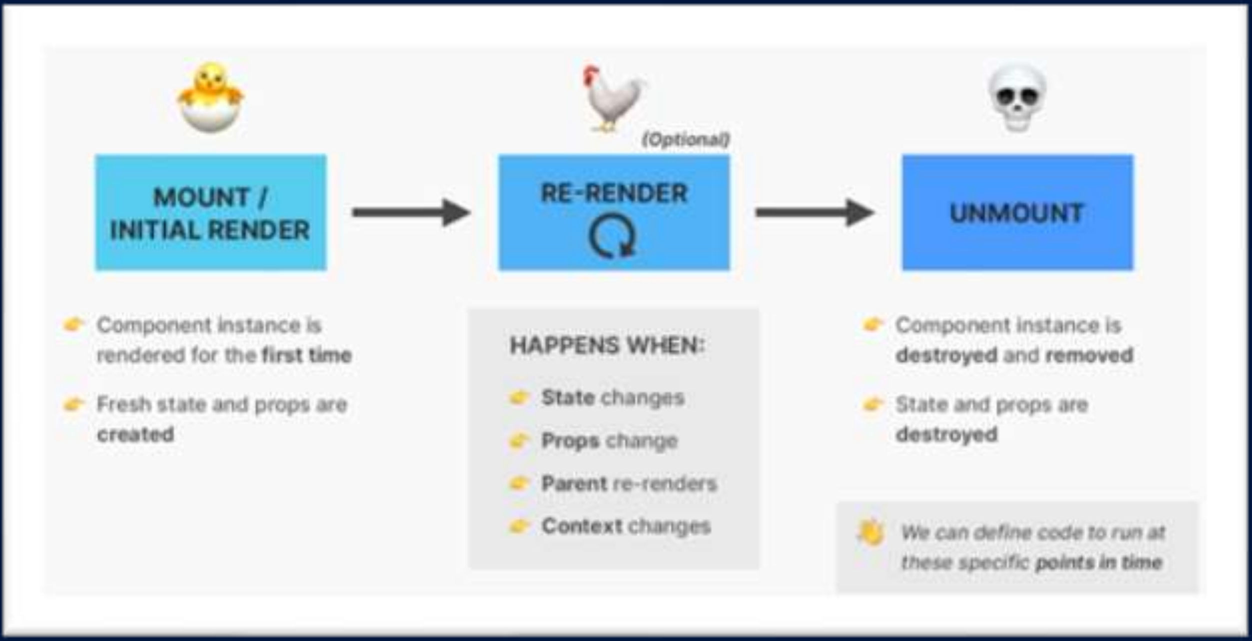
React

LIFE CYCLE


MOUNT / INITIAL RENDER
Component instance is rendered for the first time
Fresh state and props are created
(Optional)
RE-RENDER
HAPPENS WHEN:
State changes
Props change
Parent re-renders
Context changes
UNMOUNT
Component instance is destroyed and removed
State and props are destroyed
We can define code to run at these specific points in time

In React, the component lifecycle refers to the series of events that occur during the lifespan of a component, from its creation to its removal from the DOM. In React class components, these lifecycle events are methods that you can override to perform specific tasks at different points in the component's life.

Here's an overview of the different phases in the component lifecycle:

React

Mounting Phase

1. These methods are called when an instance of a component is being created and inserted into the DOM.

2. constructor(props): This is the constructor for the component. It is called before the component is mounted. You can initialize state and bind event handlers in the constructor.

3. static getDerivedStateFromProps(props, state): This is a static method that is called when the component is instantiated as well as when it receives new props. It returns an object to update the state or null to indicate that the state doesn't need to change.

4. render(): This is a required method that determines what gets rendered to the DOM. It should be a pure function and not modify the component state.

5. componentDidMount(): This method is called after the component has been rendered to the DOM. It's often used for tasks like fetching data from a server.

React

Updating Phase

1. These methods are called when a component is being re-rendered as a result of changes to either its props or state.

2. static getDerivedStateFromProps(nextProps, nextState): Similar to the mounting phase, this is called when the component is being re-rendered due to changes in props or state.

3. shouldComponentUpdate(nextProps, nextState): This method is called before rendering when new props or state are received. It can be used to optimize performance by preventing unnecessary re-renders.

4. render(): Again, this is the required method to determine what gets rendered.

5. getSnapshotBeforeUpdate(prevProps, prevState): This method is called right before the changes from the virtual DOM are to be reflected in the DOM. It receives the previous props and state as parameters and returns a value that is passed to the next method.

6. componentDidUpdate(prevProps, prevState, snapshot): This method is called after the component is updated in the DOM. It's often used for tasks like interacting with the DOM or performing additional data fetching.

Unmounting Phase

1. This method is called when a component is being removed from the DOM.

2. componentWillUnmount(): This method is called just before the component is removed from the DOM. It can be used to perform cleanup tasks such as cancelling network requests or clearing up subscriptions.

Error Handling

1. These methods are related to error handling.

2. static getDerivedStateFromError(error): This method is called when there is an error during rendering. It allows the component to capture an error in state and display a fallback UI.

React

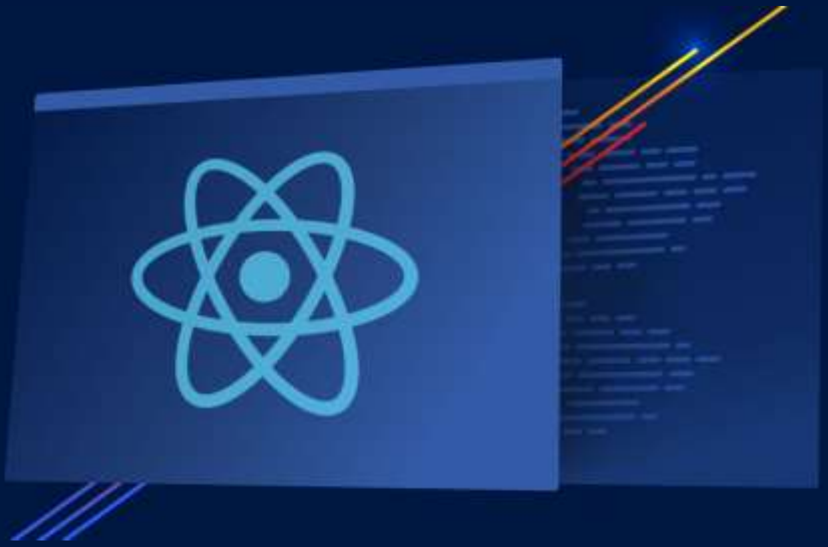
3. **componentDidCatch(error, info): This method is called after an error has been thrown during rendering. It's used for logging errors or reporting them to a service.**

It's important to note that with the introduction of React Hooks, functional components have their own equivalent lifecycle methods using hooks like useEffect, useLayoutEffect, useMemo, and useCallback. These hooks allow functional components to manage side effects and state in a way similar to class components.

React

Do you find it helpful?

Let me know down in the comments



Click To Follow For More On LinkedIN