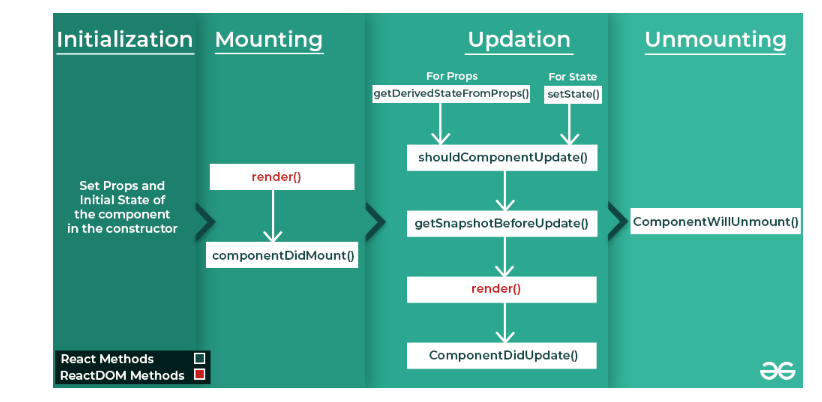
***React – Components***

***React Component /life cycle***

******

Every React Component has a lifecycle of its own, lifecycle of a component can be defined as the series of methods that are invoked in different stages of the component’s existence. The definition is pretty straightforward but what do we mean by different stages? A React Component can go through four stages of its life as follows.

* **Initialization:** This is the stage where the component is constructed with the given Props and default state. This is done in the constructor of a Component Class.
* **Mounting:** Mounting is the stage of rendering the JSX returned by the render method itself.
* **Updating:** Updating is the stage when the state of a component is updated and the application is repainted.
* **Unmounting:** As the name suggests Unmounting is the final step of the component lifecycle where the component is removed from the page.

import React, { Component } from 'react';

class Life extends Component {

    constructor(props) {

        super(props);

        this.state = { Count: 0 };

        console.log("Initialization");

    }

    componentDidMount() {

        console.log("Component Mounted");

    }

    componentDidUpdate(prevProps, prevState) {

        console.log("Component Updated");

    }

    componentWillUnmount() {

        console.log("Component Unmounted");

    }

    render() {

        return (

            <div>

                <h4>React Lifecycle</h4>

                <div>Count: {this.state.Count}</div>

                <button onClick={() => this.setState({ Count: this.state.Count + 1 })}>Increment</button>

                <button onClick={() => this.setState({ Count: this.state.Count - 1 })}>Decrement</button>

                <button onClick={() => this.setState({ Count: 0 })}>Reset</button>

            </div>

        );

    }

}

export default Life;

import React, { Component } from 'react';

import Car from './Car';

class Life extends Component {

  constructor(props) {

    super(props);

    this.state = { Count: 0, delete: false };

    console.log("Initialization");

  }

  // before

  componentWillMount() {

    console.log("Component will ");

  }

  // after

  componentDidMount() {

    console.log("Component Mounted");

  }

  // update

  componentDidUpdate(prevProps, prevState) {

    console.log("Component Updated");

}

  // remove

  componentWillUnmount() {

    console.log("Component is removed");

  }

  render() {

    return (

      <div>

        <h4>React Lifecycle</h4>

        <div>Count: {this.state.Count}</div>

        <button onClick={() => this.setState({ Count: this.state.Count + 1 })}>Increment</button>

        <button onClick={() => this.setState({ Count: this.state.Count - 1 })}>Decrement</button>

        <button onClick={() => this.setState({ Count: 0 })}>Reset</button>

        <button onClick={() => this.setState({ delete: true })}>Delete</button>

        <br />

        <div> {this.state.delete ? null : <Car />} </div>

      </div>

    );

  }

}

export default Life;