**LIFECYCLE METHODS IN REACT JS**

**1.What Are Components ?**

**2.Functional Components ?**

**3.What Are Props And State ?**

**4.Component Life Cycle Phases And Methods ?**

Components:

What you see in a web page ui it is built over components.

Components means nothing but a small piece of code which specifies the some data in a web page. For example, if there is a web page which it contains card,buttons, paragraph text elements here button may be a component,card is a also a component like that it works.

And we can use components in 2 ways:

1.Class Components

2.React.components

Note:

1.React removed the class components and converted all into functional components.

2.In near future if there is any update react will do it in functional components only.

3.But knowing about class components is very important because if you know this you will understand the basic life cycle of the components. It will be more helpful.

Component = state + UI(html + css) + props

1.Class component:

A class component in React is a type of component defined using ES6 class syntax, extending React.Component. It is a traditional way of creating components in React, offering capabilities for managing internal state and utilizing lifecycle methods.

**Key characteristics of React class components:**

* **Definition:** They are defined as a JavaScript class that extends React.Component.
* render() method: A mandatory method within the class that returns JSX, describing the UI to be rendered.
* **State management:** Class components can manage their own internal data using this.state and update it using this.setState().
* **Lifecycle methods:** They provide specific methods (e.g., componentDidMount, componentDidUpdate, componentWillUnmount) that allow developers to execute code at different stages of a component's lifecycle.
* **Props:** They receive data from parent components through this.props.
* **Constructor:** Often used to initialize state and bind event handlers. The super(props) call is essential within the constructor to correctly pass props to the parent React.Component constructor.

**Example of a basic class component:**



2.Functional Component:

React functional components are JavaScript functions that accept an object of "props" (short for properties) as an argument and return a React element, typically written in JSX, to describe what should be rendered on the screen.

**Here are the key characteristics of React functional components:**

* **Simplicity:**

They are essentially plain JavaScript functions, making them easier to read, write, and understand compared to class components.

* **Props as Arguments:**

They receive data from parent components through the props object passed as an argument to the function. Props are read-only, meaning a component should not modify its own props.

* **Return JSX:**

They must return a React element, commonly expressed using JSX, which is a syntax extension for JavaScript that looks similar to HTML.

* **State and Lifecycle with Hooks:**

While initially stateless, the introduction of React Hooks (e.g., useState, useEffect, useContext) in React 16.8 enabled functional components to manage state and access lifecycle features previously exclusive to class components.

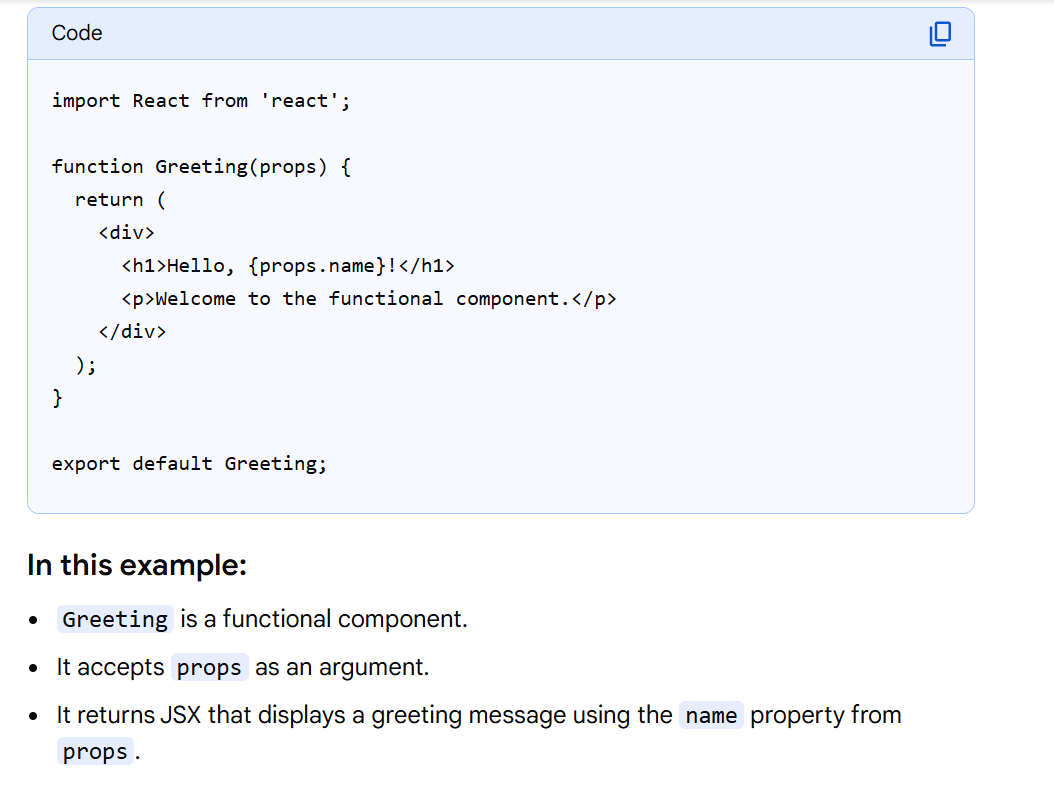
* **Reusability:**

They are designed to be reusable building blocks of your UI, allowing you to create modular and maintainable applications.

* **Naming Convention:**

Functional components typically start with a capital letter (e.g., MyComponent) to distinguish them from regular JavaScript functions.

**Example of a simple functional component:**



* Now we can pass componets as <MyComponent /> like this with starting letter should be capital letter.
* JSX or React decided this because to distinguish between the HTML elements and React functions.
* h 1,p,input these all HTML elements are start with small letters and when you have a function starting with small letter react think it as a HTML element which makes messy so react came up with this approach the react components should start with capital letter.

Props:

Props is nothing but passing data to children componets from parent components.

State:

State is a data that can be string,number,Boolean,object or it can be anything.

State is a data that can be changed or it could be changed or updated later.

State,setState => here state will have current state data where the setState() will have the updated sate or data what we have done and it re renders the UI with this updated state data. Its like we rendering the UI everytime we make changes with this setState() react thinks we should re render the data or UI now.It will do it in-memory only or automatically that is the use of setState().

Components life cycle phases and methods:

Phase => the phase means nothing but from the creation of react component in ui and until its removed from the ui in between how many phases it went through is called phase.You can call it like as a component phase also.

Every component in react goes through 3 phases:

1.Mounting Phase

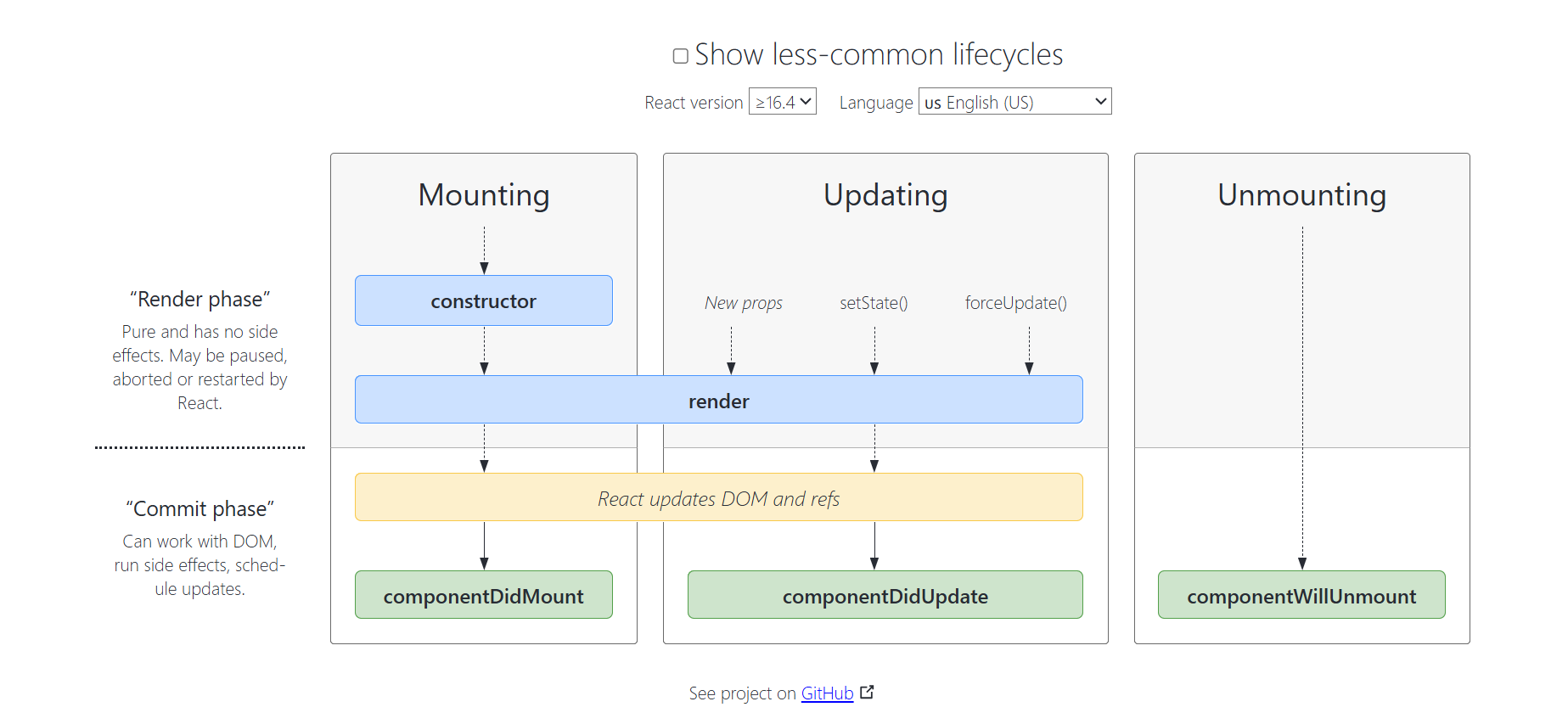
2.Updating phase

3.Unmounting phase

1.Mounting Phase – it is the creation phase

2.Updating Phase – when we update the data in the UI in that case everytime it re renders the UI that phase is called updating phase.A component can go through any number of updating phases.

3.Unmounting Phase – Unmounting means when the component is removed or deleted from the UI that phase we called the Unmounting phase.



For life cycle methods refer this link:

<https://legacy.reactjs.org/docs/react-component.html>