REACTJS

1. **What is React?**

* An open-source JS library, made by Facebook for creating Single Page Applications.
* It is used to create component based mobile and desktop apps.

1. **What are Single Page Applications?**

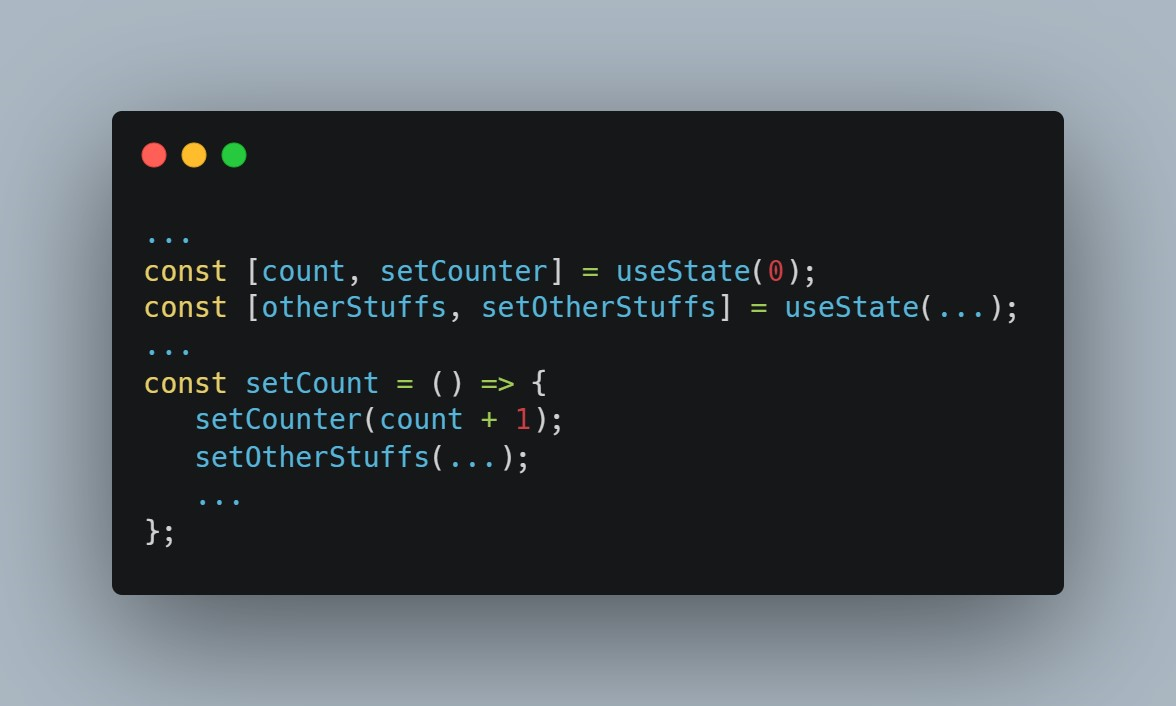
* Single Page Applications (SPAs) are a type of web application or website that operates on a single HTML page
* Instead of loading entire new pages from the server in response to user actions, SPAs dynamically update the content on the existing page by using JavaScript to manipulate the Document Object Model (DOM).
* This approach provides a smoother and more responsive user experience by avoiding full-page reloads.

1. **List some important features of React.**

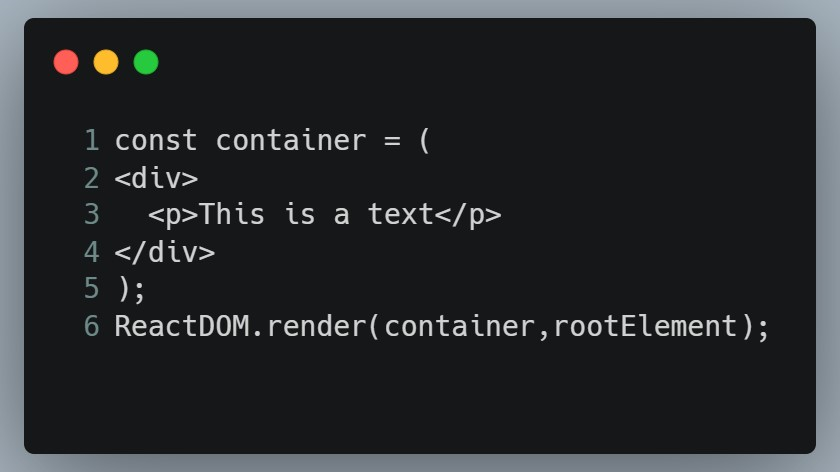
* It supports server-side rendering.
* It will make use of the virtual DOM rather than real DOM (Data Object Model) as RealDOM manipulations are expensive.
* It follows unidirectional data binding or data flow.
* It uses reusable or composable UI components for developing the view.

1. **What is useState() in React?**

* The useState() is a built-in React Hook that allows you for having state variables in functional components
* It should be used when the DOM has something that is dynamically manipulating/controlling.



1. **What is JSX?**

* JSX stands for JavaScript XML.
* It allows us to write HTML inside JavaScript and place them in the DOM without using functions like appendChild( ) or createElement( ).
* An example of JSX: 

1. **What are the differences between functional and class components?**

* Before the introduction of Hooks in React, functional components were called stateless components and were behind class components on a feature basis.
* After the introduction of Hooks, functional components are equivalent to class components.
* Although functional components are the new trend, the react team insists on keeping class components in React. Therefore, it is important to know how these components differ.

|  |  |
| --- | --- |
| Functional Component Syntax | Class Based Component Syntax |
|  |  |

1. **What are props in react?**

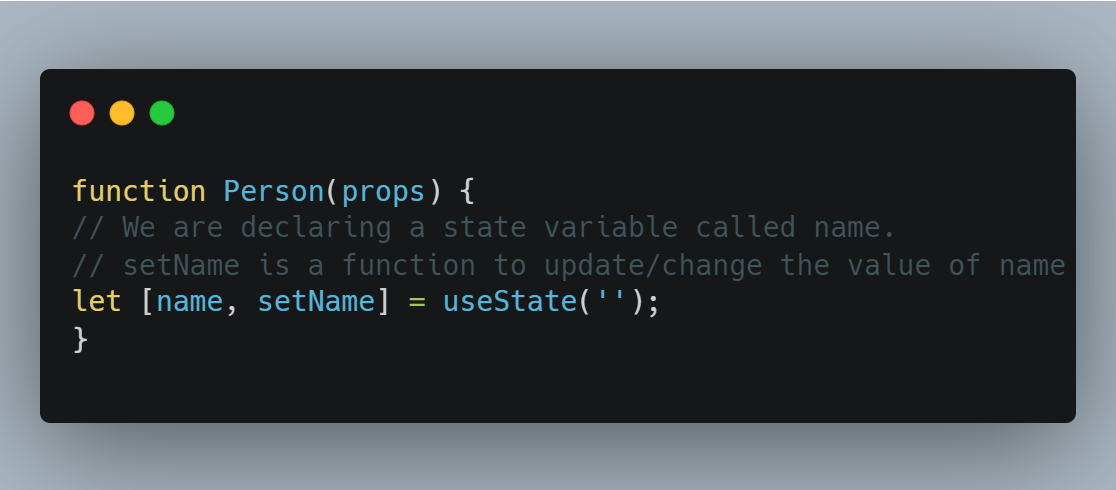
* In React, "props" (short for properties) are a way to pass data from a parent component to a child component.
* Props are a mechanism for communication between components, allowing you to customize and configure child components based on data from their parent components.

1. **Explain props drilling in react.**

* Props drilling, also known as prop threading or prop passing, refers to the process where props are passed through multiple layers of React components to reach a deeply nested child component.
* This can create a maintenance challenge and reduce the code's clarity because components that do not use the prop are still required to pass it down to their children.
* To address the issue of prop drilling, you can use solutions like React Context or state management libraries like Redux to manage global state and avoid passing props through multiple layers of components.

1. **What are react hooks?**

* Hooks are functions that let us “hook into” React state and lifecycle features from a functional component.
* React Hooks cannot be used in class components. They let us write components without class.



1. **What are the rules that must be followed while using React Hooks?**

* React Hooks must be called only at the top level. It is not allowed to call them inside the nested functions, loops, or conditions.
* It is allowed to call the Hooks only from the React Function Components.

1. **What is the use of useEffect React Hooks?**

* In React, the useEffect hook is used for handling side effects in functional components.
* Side effects can include data fetching, subscriptions, manually changing the DOM, or any other operation that needs to be performed after the initial render or in response to changes in state or props.



**What are controlled components in React?**

* Controlled components are components where form data is controlled by React state. This means that the React co

**What is the virtual DOM in React?**

* The virtual DOM is an in-memory representation of the real DOM elements. React uses the virtual DOM to efficiently update the browser's DOM by calculating the difference between the current virtual DOM and the desired virtual DOM state.
* mponent that renders a form also controls what happens in that form on subsequent user input.