React Exercises

Topics : React JS
Written on January 03, 2024

Exercise 1: Creating a Simple Component

Create a functional React component called `Greeting` that renders a greeting message. The component should take a prop called `name` and display a personalized greeting.

Exercise 2: Handling State

Create a class component called `Counter` that maintains a count in its state. Add buttons for incrementing and decrementing the count. Display the current count on the component.

Exercise 3: List Rendering

Create a functional component called `UserList` that takes an array of user objects as a prop and renders a list of users. Each user should be displayed with their name and age.

Exercise 4: Event Handling

Create a component called `Toggle` with a button. When the button is clicked, toggle the visibility of a message saying "Toggle is active" or "Toggle is inactive".

Exercise 5: Forms and Controlled Components

Create a form component called `LoginForm` with fields for username and password. Handle form submission and log the entered values to the console.

Exercise 6: Lifecycle Methods

Create a class component called `LifecycleDemo` that logs messages to the console in each of the lifecycle methods (e.g., `componentDidMount`, `componentDidUpdate`, `componentWillUnmount`).

Exercise 7: Conditional Rendering

Create a component called `Authentication` that takes a prop `isLoggedIn`. If `isLoggedIn` is true, display a "Welcome" message; otherwise, display a "Login" button.

Exercise 8: Routing with React Router

Create a simple React application with React Router. Define routes for a home page, an about page, and a contact page. Display different components for each route.

Exercise 9: State Management with Redux

Set up a React application with Redux. Create actions, reducers, and a store to manage a counter. Display the current count and buttons for incrementing and decrementing the count.

Exercise 10: Using React Context

Create a context for managing user authentication. Create a provider component that manages the authentication state and a consumer component that displays content based on whether the user is authenticated.