TW-012 TEAM LEAD VERSION

TEAMWORK

CLARUSWAY
WAY TO REINVENT YOURSELF

Meeting Agenda

- Icebreaking
- Questions
- Interview Questions
- Coffee Break
- Coding Challenge
- Video of the week
- Retro meeting
- Case study / project

Teamwork Schedule

Ice-breaking 5m

- Personal Questions (Study Environment, Kids etc.)
- Any challenges (Classes, Coding, studying, etc.)
- Ask how they're studying, give personal advice.
- Remind that practice makes perfect.

Team work 5m

- Ask what exactly each student does for the team, if they know each other, if they care for each other, if they follow and talk with each other etc.

Ask Questions 15m

1. React Context makes it easier to do which of the following?

A. Pass data to different components
B. Update components
C. Reuse components
D. Create components

Answer: A

2. What is required to use React Context?

A. A ContextProvider component
B. A ContextConsumer component
C. A ContextProvider and a ContextConsumer component
D. A ContextProvider, a ContextConsumer, and a component that is connected to a ContextProvider

Answer: C

3. What does a ContextProvider component provide?

A. A way to bind data to context
B. A way to get data from context
C. A way to create context
D. A way to update context

Answer: A

4. What does a ContextConsumer component provide?

A. A way to get data from context
B. A way to create context
C. A way to update context
D. A way to bind data to context

Answer: A

5. How do you pass parameters in React Router?

A. By using the <Params> component
B. By passing them as props to the <Route> component
C. By using the <Parameters> component
D. None of the above

Answer: B

Explanation: You can pass parameters in React Router by passing them as props to the <Route> component.

6. Which method is used to create a new Context in React?

A. createContext()
B. newContext()
C. makeContext()
D. useContext()

Answer: A

7. How can you access data from the Context within a functional component?

A. Using the this.context object
B. Importing the Context directly into the component
C. Utilizing the useContext hook
D. None of the above

Answer: C

8. What are Styled Components in React?

A. Components that allow you to create custom styles for React components using CSS classes
B. Components that allow you to create custom styles for React components using inline styles
C. Components that allow you to create custom styles for React components using CSS-in-JS
D. Components that allow you to create custom styles for React components using SASS

Answer: C

Explanation: Styled Components are a popular library in React that allow you to create custom styles for React components using CSS-in-JS.

9. How do you create a styled component in React?

A. By defining a new CSS class and passing it to the component as a prop
B. By defining a new styled component using the styled() function provided by the Styled Components library
C. By using the <StyledComponent> component provided by React Styled Components
D. By defining a new React component and passing it to the styled() function as an argument

Answer: B

Explanation: You can create a styled component in React by defining a new styled component using the styled() function provided by the Styled Components library.

10. How do you pass props to a styled component in React?

A. By using the props object inside the CSS template literal
B. By passing them as props to the component rendered by the styled component
C. By using the ${props => ...} syntax inside the CSS template literal
D. All of the above

Answer: D

Explanation: You can pass props to a styled component in React by using the props object inside the CSS template literal, passing them as props to the component rendered by the styled component, or using the ${props => ...} syntax inside the CSS template literal.

11. What are the benefits of using Styled Components in React?

A. It allows you to write cleaner and more maintainable CSS code
B. It provides better performance and smaller bundle sizes compared to traditional CSS stylesheets
C. It allows you to create reusable and composable styles for React components
D. All of the above

Answer: D

Explanation: The benefits of using Styled Components in React include writing cleaner and more maintainable CSS code, better performance and smaller bundle sizes compared to traditional CSS stylesheets, and creating reusable and composable styles for React components.

12. What is the purpose of the "ThemeProvider" component in Styled Components?

A. It allows you to define a global theme object that can be used in your styled components
B. It allows you to define custom CSS properties that can be used in your styled components
C. It allows you to define global CSS rules that apply to all components in your application
D. It allows you to define custom CSS animations that can be used in your styled components

Answer: A

Explanation: The "ThemeProvider" component in Styled Components allows you to define a global theme object that can be used in your styled components. This can help you maintain a consistent visual style throughout your application.

13. What is the output?

```
const numbers = [33, 2, 8];
numbers.sort();
console.log(numbers[1])
```

A. 8
B. 2
C. 33
D. 1

Answer: C

Interview Questions 15m

1. What issues does the React Context API solve?
The React Context API solves the issue of prop drilling. Prop drilling is when a component needs data that is located several levels up in the component hierarchy. In order to get that data, the component has to pass the data down through all of the intermediate levels until it finally reaches the component that needs it. This can be a very tedious and error-prone process. The React Context API provides a way for components to access data that is located anywhere in the component hierarchy without having to pass the data down through all of the intermediate levels.

2. Why are components created in React, and what is the rationale behind their creation?
Components are created in React to encapsulate and organize the user interface into smaller, reusable, and manageable pieces. The rationale behind creating components is to promote modularity, reusability, and maintainability of the codebase. By breaking down the user interface into smaller components, developers can build complex UIs more efficiently and understand the application's structure better.

3. What is a Hook in React, and what is its purpose?
A Hook in React is a feature introduced with React 16.8 that allows functional components to use React features and state. It enables developers to add React-specific functionalities to functional components without the need for using class components.

The purpose of Hooks is to simplify the management of state, lifecycle, and other React features within functional components. Before Hooks, developers had to use class components to access and manage state and lifecycle methods. Hooks provide an alternative approach that promotes code reusability, separation of concerns, and a more straightforward and cleaner component structure.

4. If there are 5 nested components, how do you pass data from the parent component to the bottom-most child component where the data is held in the state?

To pass data from the parent component to the bottom-most child component where the data is held in the state, there are two common approaches: Props (Component Drilling) and Global State Manegement like React Context.

1. Prop Drilling: In this approach, data is passed down the component tree hierarchically using props. The data is initially held in the state of the parent component, and then it is passed as props to each child component in the tree until it reaches the bottom-most child component.
2. Global State Management like React Context, Redux etc. : In this approach, you can use global state management like React Context to share the data across the component tree without the need to pass props explicitly through each level.

Both approaches have their use cases, and the choice between them depends on the complexity of your application and your data sharing needs. Prop drilling is more suitable for simpler cases, while global state management with React Context or similar libraries becomes more advantageous when the data sharing complexity increases or when you have multiple levels of nested components.

Coding Challenge 15m

- RC-CC-04 Film Info

☕

Coffee Break 10m

☕

Video of the Week 10m

- Install Tailwind CSS for React

Case study/Project 15m

1. RP-08 Movie App

- - Movie App is a self study project and live solution will be provided on Monday and Tuesday(31.07-01.08.2023). Try to finish before live session.

Retro Meeting on a personal and team level 10m

Ask the questions below:

- What went well?
- What could be improved?
- What will we commit to do better in the next week?

Closing 5m

-Next week's plan

-QA Session
