Hema Coding School

YouTube Link: https://www.youtube.com/@HemaCodingSchool

useContext

Purpose

- useContext is a React hook that allows you to consume context values in functional components. Context provides a way to pass data through the component tree without having to pass props down manually at every level.

1. Creating Context

- To use useContext, you first need to create a context using React.createContext.

  CreateContext()

```
import {createContext} from 'react';

let myContext = createContext()

export default myContext;
```

2. Import Context inside the Parent Component (A Component)

3. Wrap Child Component of A inside pre-defined component called Provider

< <context>.Provider>

<B/>

</ <context>.Provider >

4. Provider Component will take one props i.e value : {}

<myContext.Provider value={state}>

<B />

</myContext.Provider>

```
import React, { useState } from "react";
import "../../../Component/ClassBased.css";
import B from "./B.js";
import myContext from "./Context.js";

function A() {
  let [state, setState] = useState({
    name: "Hema",
    age: 20,
    email: "hema@gmail.com",
  });
  return (
    <div className="train-info">
      <h2>A Component</h2>
```

```
      <myContext.Provider value={state}>
        <B />
      </myContext.Provider>
    </div>
  );
}

export default A;
```

5. Use useContext() hook to access the data in child component

useContext() is hook in react, using which we can access the data on context inside the child Component

import { useContext } from "react";

import myContext from "./context";

const data = useContext(myContext);

```
import React,{useContext} from 'react';
import myContext from './Context.js'


function D() {
   let data= useContext(myContext)
   console.log(data)
  return (
    <div>
       <h2>D Component email : {data.email} </h2>
    </div>
  )
}

export default D
```

Interview Questions:

What is useContext in React?

- **Answer**: useContext is a hook that allows functional components to consume context values directly. It provides a way to share data, like themes or user information, across the component tree without passing props down manually at every level.

How do you create and use a context in React?

- **Answer**: You create a context using React.createContext and provide it using the Context.Provider component. To consume the context in a functional component, use the useContext hook.

Hema Coding School

YouTube Link: https://www.youtube.com/@HemaCodingSchool

What are the main use cases for useContext?

- **Answer**: Main use cases include sharing global data like themes, user authentication, language preferences, and other global settings that need to be accessed by multiple components without passing props through many levels of the component tree.

How does useContext help in avoiding prop drilling?

- **Answer**: useContext allows components to directly access the context values without passing them down through every intermediate component. This avoids the need to pass props at multiple levels, making the code cleaner and easier to maintain.

Can you explain the relationship between createContext and useContext?

- **Answer**: createContext is used to create a context object, which includes a Provider component and a Consumer component. useContext is a hook that allows functional components to access the value provided by the nearest Provider component without using a Consumer component.

What happens if a context value changes? How does it affect the components consuming the context?

- **Answer**: When a context value changes, all components that consume the context using useContext will re-render to reflect the new value. This can lead to performance issues if not managed carefully, as it may cause unnecessary re-renders.