**Context API and useContext() Hook**

Here's a step-by-step example to clarify:

1. **Creating the Context**: First, you create a context:

import React from 'react';

const MyContext = React.createContext();};

1. **Using the Context Provider**: You'll usually want to provide a value to this context using the **MyContext.Provider** component:

function App() {

return (

<MyContext.Provider value={{ data: 'Hello from context!' }}>

<ChildComponent />

</MyContext.Provider>

);

}

1. **Consuming the Context**: Inside any child component of **App**, you can consume this context value using the **useContext** hook:

function ChildComponent() {

const contextValue = useContext(MyContext);

return <div>{contextValue.data}</div>;

}

In the **ChildComponent**, we pass **MyContext** (which we created using **React.createContext()**) as an argument to the **useContext** hook. This allows us to access the value provided by the nearest **MyContext.Provider** in the component tree.

Note: If **ChildComponent** (or the component where you're calling **useContext**) is not a descendant of any **MyContext.Provider**, **useContext** will return the default value from **React.createContext()**. If no default value was provided, it would return **undefined**.