useContext() React Hooks

- React Context is a way to manage state globally.
- It can be used together with the useState Hook to share state between deeply nested components more easily than with useState alone.
- In React, the useContext hook is used to **consume data from a React context**. Context provides a way to share data between components **without having to pass props** manually at every level of the component tree.
- It allows you to create a global state that **can be accessed by any component that needs it**, regardless of where it is in the component hierarchy.

How to use “useContext”

1. Create the Context

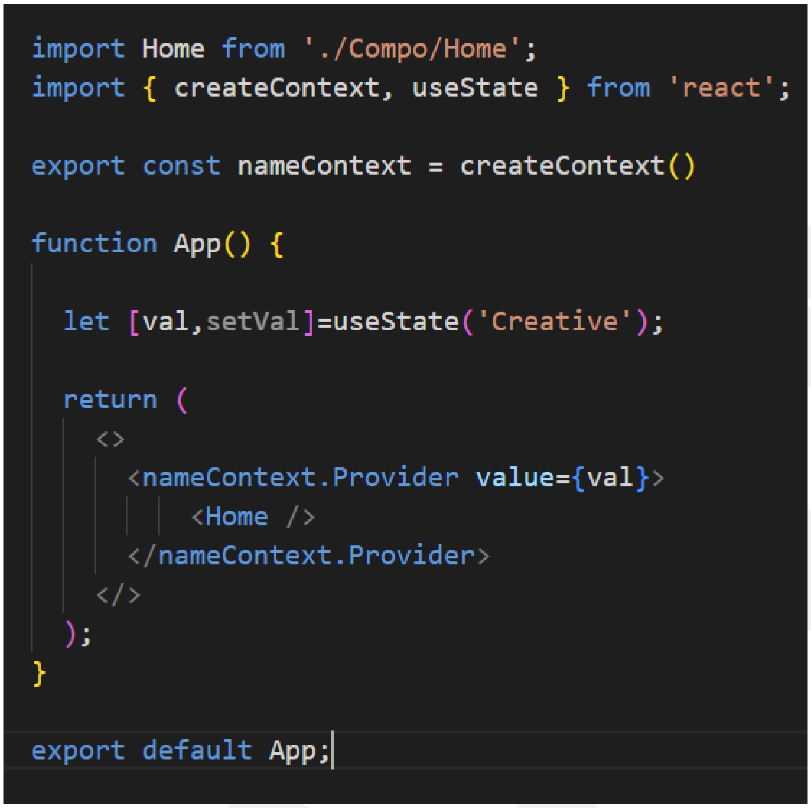
- First, you need to create a context using the **React.createContext()** method. This method returns an object with two properties: Provider and Consumer.

2. Provide the Data

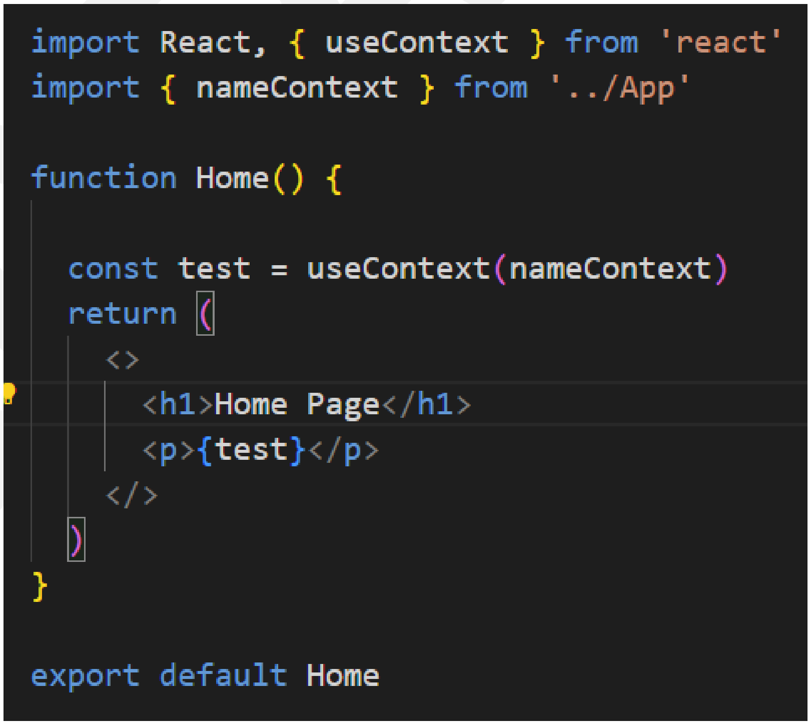
- Wrap the component(s) that should have access to the shared data with the Provider. This is typically done at the highest level in component tree.

3. Consume the data using useContext

- Now, in any component that needs to access the shared data, you can use the useContext hook to consume the context.



```
import Home from './Compo/Home';
import { createContext, useState } from 'react';

export const nameContext = createContext()

function App() {

  let [val,setVal]=useState('Creative');

  return (
    <>
      <nameContext.Provider value={val}>
          <Home />
      </nameContext.Provider>
    </>
  );
}

export default App;
```



```
import React, { useContext } from 'react'
import { nameContext } from '../App'

function Home() {

  const test = useContext(nameContext)
  return (
    <>
      <h1>Home Page</h1>
      <p>{test}</p>
    </>
  )
}

export default Home
```