ReactJs : Day4

Sunday, December 1, 2024 8:33 AM

Why use hook ?

- **"In class components, lifecycle methods can be easily used. However, in function-based components, we cannot directly use lifecycle methods."**
- **Simplify state and lifecycle management.**
- **Enable logic reuse with custom hooks.**
- **Avoid complexity tied to class components.**
- **Provide fine control over side effects.**
- **Optimize performance.**
- **Make functional components powerful and modern.**

What is HOOK ?
A **hook** in React is a special function that allows you to use React features like **state**, **lifecycle methods**, and **context** in **functional components**, making them more powerful and easier to use.

Syntax :- const [<state>, <setStateFunction>] = <HookName>(<initialValue>);

For.e.g :- const [count, setCount] = useState(0);

1 . **useState**:

The useState() hook in React is used to manage **state** in functional components. It allows you to add stateful logic to a function-based component, which was previously only possible in class components.

```
import React,{useState} from 'react';
function UseState()
{
    const [name,setName] = useState("Ram");
    return(
        <>
            <div style={{display:'grid',justifyContent:'center'}}>
            <h1>{name}</h1>
            <button onClick={()=>setName("Rohit")}>Update Name</button>
            </div>
        </>
    )
}
export default UseState;
```

2. useEffect :

useEffect replaces lifecycle methods like componentDidMount, componentDidUpdate, and componentWillUnmount.

- Replacing multiple lifecycle methods.
- Providing precise control with dependencies.
- Enabling automatic cleanup.
- Promoting reusable and cleaner code.

Syntax :- **useEffect(callback, [dependencies]);**

```
import React, { useEffect, useState } from 'react'
function Useeffect() {
    const [cnt,setCnt] = useState(0);
    const [cnt1,setCnt1] = useState(0);
     // Only called for cnt
    useEffect(()=>{
        console.log("UseEffect Called");
    },[cnt])
  return (
    <>
        <div style={{display:'grid',justifyContent:'center'}}>
        <h1>{cnt}</h1>
        <h2>{cnt1}</h2>
        <button onClick={()=>setCnt(cnt+1)}>Update Count</button>
        <button onClick={()=>setCnt1(cnt1+1)}>Update Count</button>
        </div>
    </>
  )
}
export default Useeffect;
```

```
import React, { useEffect, useState } from 'react'
function Useeffect() {
    const [cnt,setCnt] = useState(0);
    const [cnt1,setCnt1] = useState(0);
     // Do not call if any state is updated
    useEffect(()=>{
        console.log("UseEffect Called");
    },[])
  return (
    <>
        <div style={{display:'grid',justifyContent:'center'}}>
        <h1>{cnt}</h1>
        <h2>{cnt1}</h2>
        <button onClick={()=>setCnt(cnt+1)}>Update Count</button>
        <button onClick={()=>setCnt1(cnt1+1)}>Update Count</button>
        </div>
    </>
  )
}
export default Useeffect;
```

```
import React, { useEffect, useState } from 'react'
function Useeffect() {
    const [cnt,setCnt] = useState(0);
    const [cnt1,setCnt1] = useState(0);
     // Call for every state update
    useEffect(()=>{
        console.log("UseEffect Called");
    })
  return (
    <>
        <div style={{display:'grid',justifyContent:'center'}}>
        <h1>{cnt}</h1>
        <h2>{cnt1}</h2>
        <button onClick={()=>setCnt(cnt+1)}>Update Count</button>
        <button onClick={()=>setCnt1(cnt1+1)}>Update Count</button>
        </div>
    </>
  )
}
export default Useeffect;
```

In Props :

```
import React, { useEffect, useState } from 'react'
function Useeffect1(props) {
    const [cnt,setCnt] = useState(0);
    useEffect(()=>{
        console.log("UseEffect Called");
    },[props.cnt])
  return (
    <>
        <div style={{display:'grid',justifyContent:'center'}}>
        <h1>{props.cnt}</h1>
        </div>
    </>
  )
}
export default Useeffect1;
```

```
import React,{useState} from "react"
// import Usestate from "./component/Useeffect"
// import Useeffect from "./component/Useeffect"
import Useeffect1 from "./component/Useeffect1"
function App() {
  const [cnt,setCnt] = useState(0);
  return (
    <>
  {/* <Usestate></Usestate> */}
  {/* <Useeffect></Useeffect> */}
  <Useeffect1 cnt={cnt}></Useeffect1>
  <button onClick={()=>setCnt(cnt+1)}>Update Count</button>
    </>
  )
}
export default App
```

Different ways to add style:

1. Inline css:

```
import React from 'react'
function Style1() {
  return (
    <div style={{color:'red'}}>Style1</div>
  )
}
export default Style1
```

2. External CSS :
   Step1 : create filename.css

   ```
   div{
       color: purple;
   }
   ```

   Step2 : import into component

   ```
   import React from 'react'
   import './style.css'
   function Style1() {
     return (
       <div>Style1</div>
     )
   }
   export default Style1
   ```

3. Using Module
   Step1 : create filename.module.css

   ```
   div{
       background-color: aqua;
   }
   ```

   Step2:

   ```
   import React from 'react'
   // import './style.css'
   import style from './custom.module.css';
   function Style1() {
     return (
       <div className={style.div}>Style1</div>
     )
   }
   export default Style1
   ```

4. Bootstarp :
   Step1 : install npm install react-bootstrap bootstrap
   Step2 : Import it into index.js/main.js

   ```
   import { StrictMode } from 'react'
   import { createRoot } from 'react-dom/client'
   import 'bootstrap/dist/css/bootstrap.min.css';
   import './index.css'
   import App from './App.jsx'
   createRoot(document.getElementById('root')).render(
     <StrictMode>
       <App />
     </StrictMode>,
   )
   ```

   // Step 3: Import where the Bootstrap component is used

   ```
   import React from 'react'
   import { Button } from 'react-bootstrap'
   function Bootstrap() {
     return (
       <Button varient="primary">First</Button>
     )
   }
   export default Bootstrap
   ```

Why for loop doesn't work directly in JSX:

JSX expects **expressions** and not statements like a for loop. Since for is a statement and not an expression, it cannot be directly used inside the JSX code.

Use map,filter

```
import React from 'react'
function ArrayandList() {
    const arr = ["Ram","Sham","Arjun","Krishna","Luvekesh"]
  return (
    <>
    <h1>Student List</h1>
    <ul>
        {
            arr.map((ele,i)=>{
                return <li key={i}>{ele}</li>
            })
        }
    </ul>
    </>
  )
}
export default ArrayandList
```

Render object :

```
import React from "react";
function Object() {
  const p = [
    {
      name: "John Doe",
      age: 28,
      profession: "Software Developer",
    },
    {
      name: "Jane Smith",
      age: 34,
      profession: "Graphic Designer",
    },
    {
      name: "Emily Johnson",
      age: 45,
      profession: "Teacher",
    },
    {
      name: "Michael Brown",
      age: 22,
      profession: "Marketing Specialist",
    },
    {
      name: "Sarah Davis",
      age: 30,
      profession: "Data Scientist",
    },
  ];
  return (
    <>
      <table style={{ border: "1px solid black", borderCollapse: "collapse", margin:'5px'}}>
        <thead>
          <tr>
            <th style={{ border: "1px solid black", padding: "8px" }}>Name</th>
            <th style={{ border: "1px solid black", padding: "8px" }}>Age</th>
            <th style={{ border: "1px solid black", padding: "8px" }}>Profession</th>
          </tr>
        </thead>
        <tbody>
          {p.map((ele, i) => (
            <tr key={i}>
              <td style={{ border: "1px solid black", padding: "8px" }}>{ele.name}</td>
              <td style={{ border: "1px solid black", padding: "8px" }}>{ele.age}</td>
              <td style={{ border: "1px solid black", padding: "8px" }}>{ele.profession}</td>
            </tr>
          ))}
        </tbody>
      </table>
    </>
  );
}
export default Object;
```

What is React Fragment?
In React, a **Fragment** is a lightweight component used to group multiple child elements without adding extra nodes to the DOM.

```
import React from 'react';

function Example() {
```

```
  return (
    <React.Fragment>
      <h1>Title</h1>
      <p>This is a paragraph.</p>
    </React.Fragment>
  );
}

export default Example;
```

OR

```
function Example() {
  return (
    <>
      <h1>Title</h1>
      <p>This is a paragraph.</p>
    </>
  );
}
```

What is Lifting Up State :
Send Data from child to parent component.

Example :

```
import React from 'react'
function LiftingState(props) {
    const data = "Virat Kohli"
  return (
    <>
        <div>
            <h2>LiftingState Component</h2>
            <button onClick={()=>props.alert(data)}>Click Me</button>
        </div>
    </>
  )
}
export default LiftingState
```

```
import LiftingState from "./component/LiftingState"
function App() {
  function pAlert(data)
  {
    alert(data);
  }
  return (
    <>
  <LiftingState alert={pAlert}></LiftingState>
    </>
  )
}
export default App
```