vivasoft

Learnathon 2.0 - React Class 2

...

Kazi Tajnur Islam | Software Engineer (L-2) | Vivasoft Limited

vivasoft

Functional Component VS Class Component

Functional Component

- Simplicity
- Hooks
- Readability
- Performance
- No ‘this’ keyword

Class Component

- Complex & Legacy code
- Lifecycle methods
- ‘this’ keyword
- More boilerplate
- State management

vivasoft

Component Lifecycle

Lifecycle Phases

- Mounting
- Updating
- Unmounting

Lifecycle Methods

- componentDidMount()
- componentDidUpdate()
- componentWillUnmount()

vivasoft

Props and State in a React Component

Props (Properties): Props are a way to pass data from a parent component to a child component. They are read-only and cannot be modified by the child component. Props are typically used for configuring or customizing child components. They allow you to provide data, such as values, functions, or objects, from a parent component to its child components.

State: State is used to manage component-specific data that can change over time and influence the component's rendering. Unlike props, state is mutable and can be modified using the '**setState**' function in class components or state hooks (e.g., **useState**) in functional components.

vivasoft

Conditional Rendering

```jsx
import React from 'react';

function MyComponent({ isLoggedIn }) {
  return (
    <div>
      {isLoggedIn ? (
        <p>Welcome, User!</p>
      ) : (
        <p>Please log in to access this content.</p>
      )}
    </div>
  );
}

export default MyComponent;
```

vivasoft

Lists and Keys in React

```
import React from 'react';

function MyListComponent() {
  const items = [
    { id: 1, text: 'Item 1' },
    { id: 2, text: 'Item 2' },
    { id: 3, text: 'Item 3' },
  ];

  return (
    <ul>
      {items.map((item) => (
                                        <li
key={item.id}>{item.text}</li>
      ))}
    </ul>
  );
}

export default MyListComponent;
```

vivasoft

Event handling

```
import React, { useState } from 'react';

function EventHandlingExample() {
  const [clickCount, setClickCount] = useState(0);

  const handleButtonClick = () => {
    setClickCount(clickCount + 1);
  };

  return (
    <div>
      <p>Click count: {clickCount}</p>
      <button onClick={handleButtonClick}>Click me</button>
    </div>
  );
}

export default EventHandlingExample;
```

vivasoft

Controlled and Uncontrolled Component

Controlled Component: The value of a form element (like an input field) is controlled by React's state. You store the value in the component's state and update it via state management functions.

Uncontrolled Component: The form element maintains its state internally, outside of React's state management. React does not control the value; it's managed directly by the DOM.

vivasoft

Routing in React with React Router

```
import Root from "./routes/root";

const router = createBrowserRouter([
  {
    path: "/",
    element: <Root />,
  },
]);

ReactDOM.createRoot(document.getElementById("root")).render(
  <React.StrictMode>
    <RouterProvider router={router} />
  </React.StrictMode>
);
```

vivasoft

Q&A time

Learnathon - 2.0

vivasoft

Thank You! Goodbye Everyone!

Learnathon - 2.0