REFS TO PERSIST DATA BETWEEN RENDERS

In React, the useRef hook is often used to persist data between renders without causing a re-render when the data changes. Unlike the useState hook, changes to the current property of a ref do not trigger a re-render. This makes it useful for holding mutable values that you don't want to cause a component to re-render when they change.

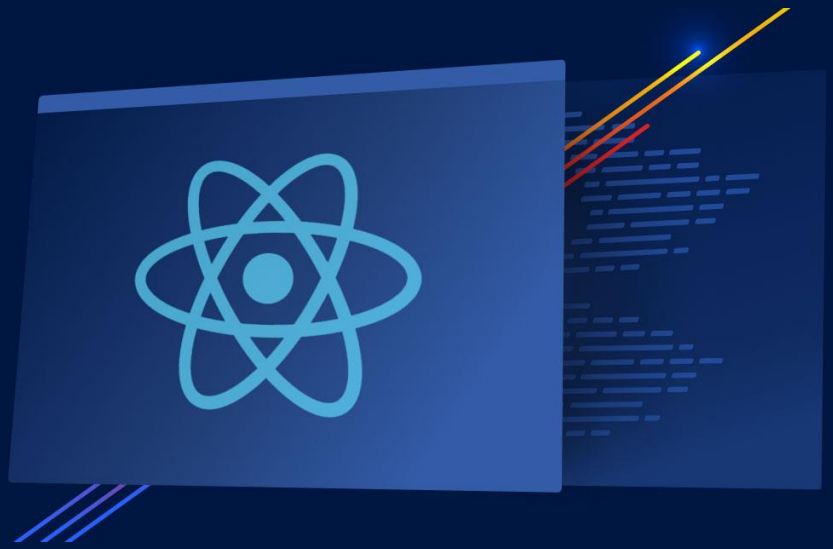
Remember that while useRef allows you to persist data between renders without causing re-renders, it doesn't trigger re-renders on its own. If you need to trigger a re-render when data changes, you might want to use useState instead.


```jsx
import React, { useRef, useState, useEffect } from 'react';

function MyComponent() {
  // useRef to persist data between renders
  const myDataRef = useRef('initialValue');

  // State to trigger a re-render for demonstration purposes
  const [renderCount, setRenderCount] = useState(0);

  useEffect(() => {
    // Log the current value of the ref on each render
    console.log('Current value of myDataRef:', myDataRef.current);

    // Increment render count for demonstration purposes
    setRenderCount(prevCount => prevCount + 1);
  }, [renderCount]); // Only re-run the effect when renderCount changes

  // Function to update the ref's current value
  const updateData = () => {
    myDataRef.current = 'updatedValue';
  };

  return (
    <div>
      <p>Render Count: {renderCount}</p>
      <p>Data from useRef: {myDataRef.current}</p>
      <button onClick={updateData}>Update Data</button>
    </div>
  );
}

export default MyComponent;
```

React

Do you find it helpful?

Let me know down in the comments.



Click To Follow For More On LinkedIN