Improvements in REACT 19

follow me for other new react features

REACT 19

coming soon...


Muhammad Mohsin

SWIPE LEFT

refs as a prop

then

Previously, refs were created using createRef( ) outside the component, making it challenging to pass them down to child components.

now

React 19 allows directly passing ref as a prop to functional components, streamlining ref usage and facilitating communication between parent and child components.

```
1   function ParentComponent() {
2     const myRef = React.createRef();
3
4     return (
5       <ChildComponent ref={myRef} />
6     );
7   }
8
9   function ChildComponent(props) {
10    return (
11      <div ref={props.ref}>Child component!</div>
12    );
13  }
14
```

pass the ref as a prop to the chlid

Access the ref from props

Muhammad Mohsin

SWIPE LEFT

Context as a provider

then

Without a clear way to define and consume context, sharing data across multiple levels of the component hierarchy required prop drilling, which can lead to code duplication and maintenance challenges.

now

React 19 reinforces the concept of Context as a provider, enabling developers to create a centralized location to store and share application-wide data. This promotes better code organization and simplifies data management.

```
1   const MyContext = React.createContext(defaultValue);
2
3   function MyProvider({ children, value }) {
4     return (
5       <MyContext.Provider value={value}>
6         {children}
7       </MyContext.Provider>
8     );
9   }
10
11  function ChildComponent() {
12    const value = React.useContext(MyContext);
13    return <div>Context value: {value}</div>;
14  }
15
```

provide context value

consume context valure

Muhammad Mohsin

SWIPE LEFT

Cleanup funnctions for refs

then

Memory leaks could occur if refs weren't cleaned up correctly, especially in situations like uncontrolled components or side effects within useEffect.

now

React 19 introduces a cleanup function as an optional second argument to React.createRef(). This function allows you to perform necessary cleanup tasks when the component unmounts or the ref value changes.

```
1   const myRef = React.createRef(() => {
2     // Cleanup logic here, e.g., removing event
          listeners
3   });
4
5   useEffect(() => {
6     // ...
7   }, [myRef.current]); // Dependency on ref value
8
```

miro

Muhammad Mohsin

SWIPE LEFT

useDeferredValue initial value

then

Previously, useDeferredValue didn't allow specifying an initial value, potentially leading to unexpected behavior if accessed before the initial value was resolved.

now

React 19 now accepts an initial value as an argument to useDeferredValue, preventing potential issues when reading the value before it's settled.

```
1    const deferredValue = useDeferredValue(initialValue);
2
```

miro

Muhammad Mohsin

SWIPE LEFT

Diffs for hydration errors

then

Debugging hydration mismatches, which occur when the server-rendered HTML doesn't match the client-side rendered output, could be difficult due to a lack of clear error messages.

now

React 19 provides more detailed diffs for hydration errors, making it easier to identify and fix discrepancies between server and client rendering. This helps in ensuring a seamless user experience.

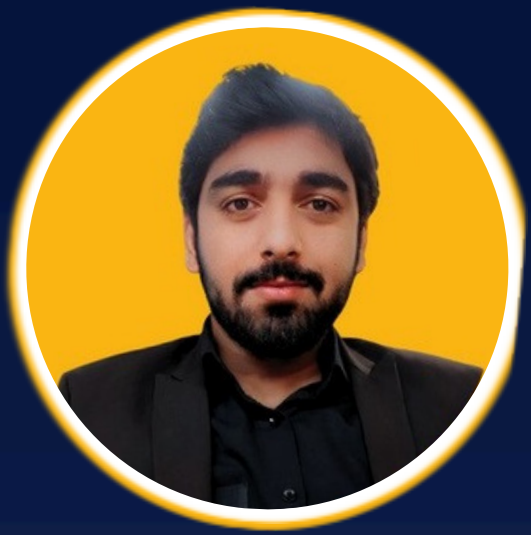
Document Metadata, stylesheets, Asynce Scripts, Preloading Resources

then

While React doesn't directly provide APIs for these features, the improvements in React 19 enable better integration with browser capabilities for managing document metadata, stylesheets, async scripts, and preloading resources.

now

Modern browsers offer APIs like document.head.append() and link and script elements with the rel and as attributes for these purposes.



Follow me for more content.

Follow me

Muhammad Mohsin Senior Fronted Engineer | React js | Next js | Node js