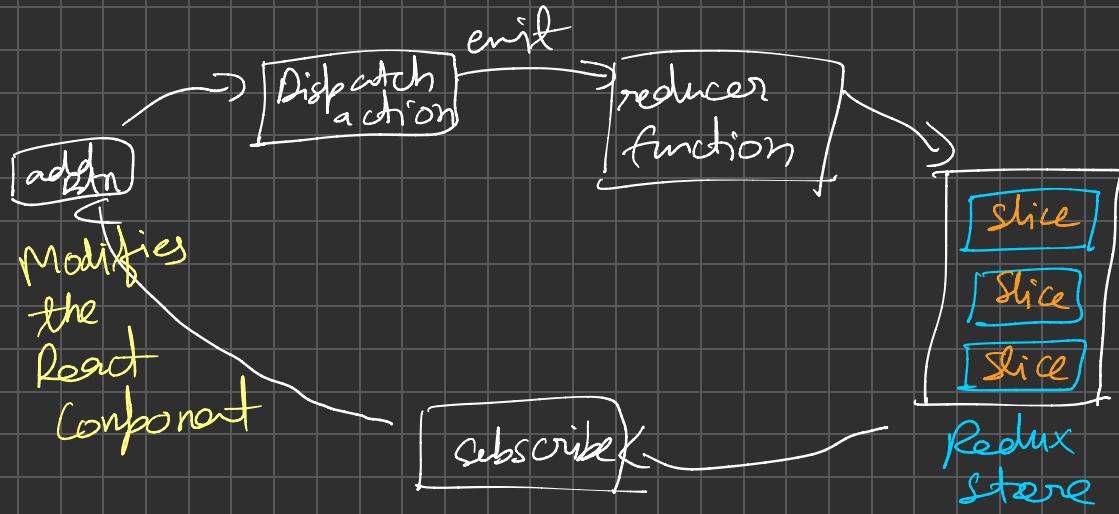




# Redux



## Some other hooks

Use memo "increase the performance of app".

It is a react hook that lets you cache the result of a calculation between renders.

When this hook should be used?

Suppose there is a heavy calculation, and the component rerenders when the state change because of that it calculates again & again. So to get rid of this we can memoize the calculation using this hook.

Suppose we are finding the prime number on particular input and as well as using another state for a timer. So if there change or input changes whole component will rerender.

That's the problem, we need to use useMemo.

```
const prime = useMemo(() => findprime([text]), [text])
```

front attribute function or  
value on which useMemo  
need to apply

dependencies

[when  
text]

change then only  
it recalculate  
otherwise uses the  
cache)

## useCallback

It is a react hook that lets you cache a function  
definition between re-renders.

## useRef

It is a react hook that lets you reference a value  
that's not needed for rendering.

```
const a = useRef(0); // { current: 0 }
console.log(a);
```

## React Interview Questions.

### 1. Hooks

useState

useEffect

useContext

useReducer (Need to study)

useMemo

useCallback (Need to study)

useRef

### 2. Higher order components.

### 3. Life cycle methods of class based components.

### 4. State management

(All about data)

Props, Proper drilling,  
context

5. Know how to pass data from one sibling to next sibling.
6. Redux / Zustand
7. Custom hooks. (clean, maintainable, readable, reusable)
8. Lazy loading | Code splitting | chunking | suspense/bundles.  
(If mentioned about performance, include it as well)
9. Virtual DOM. (All things diff algorithm, reconciliation, react fibers)
10. SSR vs CSR.  
Server side rendering vs Client side Rendering.
11. Routing (RBAC) (read about this)  
Role based access controller
  - react-router . Dynamic routing
  - . Protected routes .
12. Async Tasks ] → use effect in depth
  - API calls [fetch api]
  - events → click, scroll
  - promises
13. Reusability / Modularity / Testability [ use these words when interview happens ]
14. Performance (It is the important topic)  
(Shimmer UI) (lazy loading)  
(Asset optimization) (read about this) (server optimization)  
How to optimize images, jsx  
(add you should use setTimeout ).
15. Styling  
Tailwind CSS  
Style - By Meta ( competitor of Tailwind CSS )  
Bootstrap  
Material UI  
CSS, SCSS  
(3rd party which is better and why)