

7 React Hooks

you
must
know

useId()

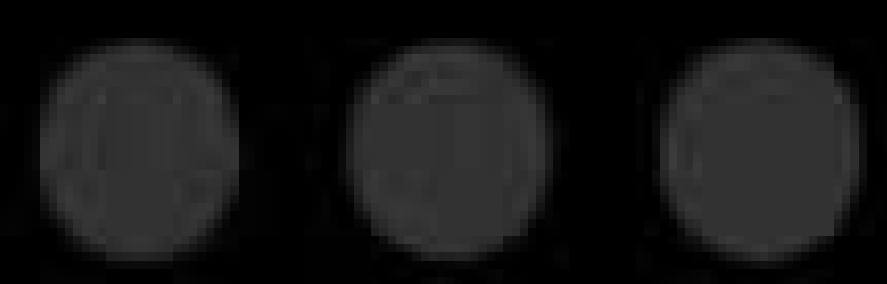
```
import { useId } from 'react';

function PasswordField() {
  const passwordHintId = useId();
  // ...
  return (
    <>
      <input type="password" aria-describedby={passwordHintId} />
      <p id={passwordHintId}>
    </>
  )
}
```

useId is a React Hook for generating unique IDs that can be passed to accessibility attributes

accessibility attributes let you specify that two tags are related to each other where you can use useId() generated id instead of hardcoding them

useState()



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

function MyComponent() {
  const [age, setAge] = useState(28);
  const [name, setName] = useState('Taylor');
  //
```

useState is a React Hook that lets you add a state variable to your component which returns an array with exactly two values

1. Current State
2. Set Function

You can pass an initial value as well like in the example:- 28 and 'Taylor'

useRef()

```
import { useRef } from 'react';

function MyComponent() {
  const intervalRef = useRef(0);
  const inputRef = useRef(null);
  // ...
}
```

useRef is a React Hook that lets you reference a value that's not needed for rendering.

Basically its like useState but the only difference is useRef doesn't cause a re-render when the value changes

useMemo()

```
import { useMemo } from 'react';

function TodoList({ todos, tab }) {
  const visibleTodos = useMemo(
    () => filterTodos(todos, tab),
    [todos, tab]
  );
  // ...
}
```

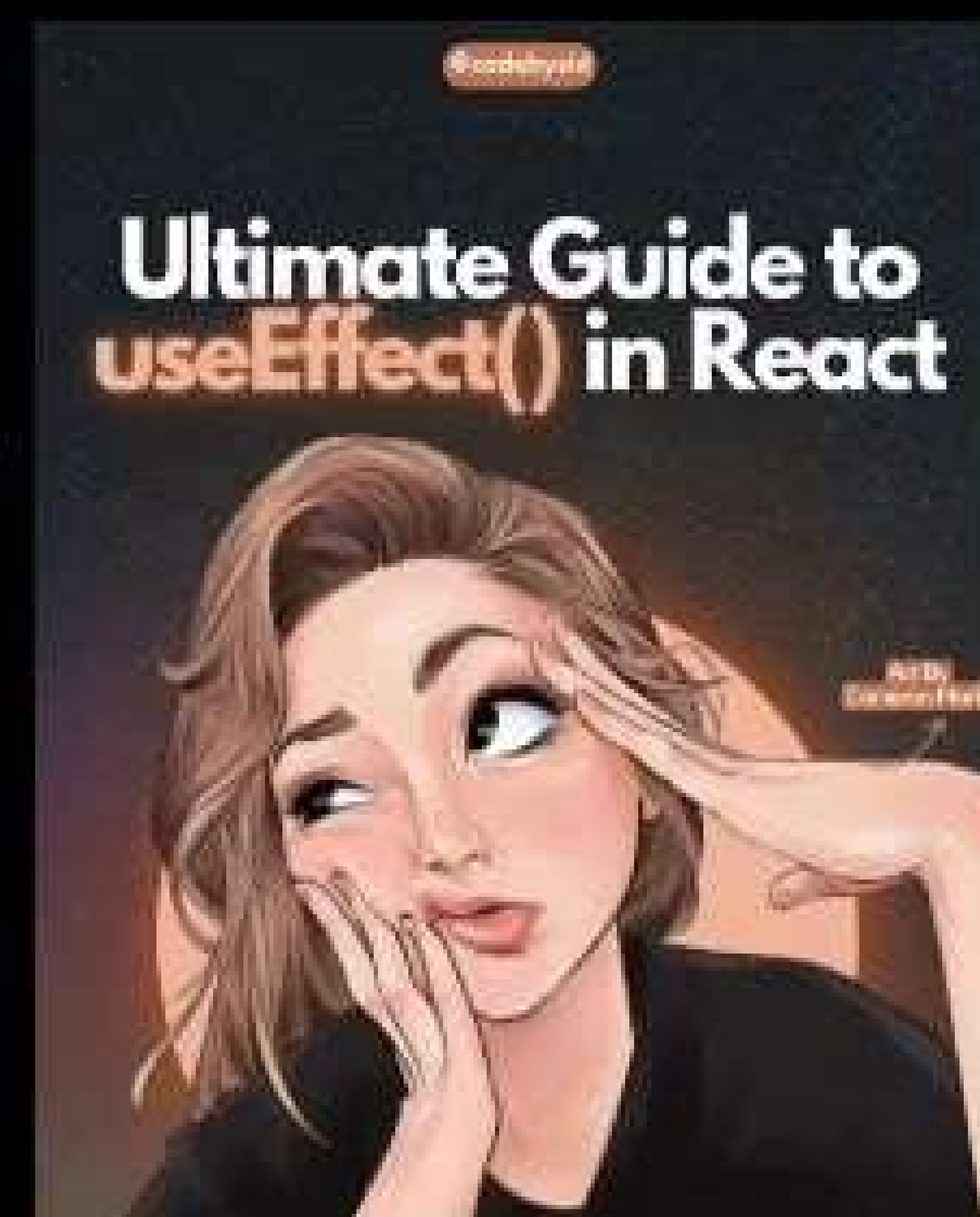
useMemo is a React Hook that lets you cache the result of a calculation between re-renders which prevents the unnecessary renders in your React Application

useEffect()

```
useEffect(() => {  
    const connection = createConnection(serverUrl, roomId);  
    connection.connect();  
    return () => {  
        connection.disconnect();  
    };  
}, [serverUrl, roomId]);  
// ...  
}
```

useEffect is a React Hook that lets you perform side-effects in the component. Side effects basically is an action which connects the component to the outside world

Check my broadcast
channel for detail
explanation post →



useCallback()

```
import { useCallback } from 'react';

export default function ProductPage({ productId, referrer, theme }) {
  const handleSubmit = useCallback((orderDetails) => {
    post('/product/' + productId + '/buy', {
      referrer,
      orderDetails,
    });
  }, [productId, referrer]);
}
```

useCallback is a React Hook that lets you cache a function definition between re-renders.

useCallback cache a function and useMemo cache a value/result of a calculation

useContext()

```
import { useContext } from 'react';

function MyComponent() {
  const theme = useContext(ThemeContext);
  // ...
}
```

useContext is a React Hook that lets you read and subscribe to context from your component just like a data store (Redux)

useContext hook lets you to read the data stored in a context which is a data store

This example is just to demonstrate useContext hook not creating a Context

we have more React hooks (check caption) but these 7 are good to learn when you are starting with React as a beginner and to build a project