

Advanced Hooks

We've gotten so far, but there are still more hooks to discover. Two down but eight more to come.

WE'LL COVER THE FOLLOWING

- Rest of the Hooks
- Interactive Examples of Advanced Hooks

We have only learned two of the 10 hooks React provides!

Knowledge of how `useState` and `useEffect` work will help you quickly learn the other hooks.

Rest of the Hooks

If you're curious to learn about those, I have created a [hooks cheatsheet](#) with live editable examples.

The image shows a screenshot of a 'hooks cheatsheet' website. The website has a sidebar with a search bar and a list of hooks: Home, useState, useEffect, useContext, useLayoutEffect, useReducer, useCallback, useMemo, useRef, and Examples. The 'Hooks' section is highlighted with a red arrow and the text '1. The hooks'. The 'Examples' section is highlighted with a red arrow and the text '5. Some advanced examples'. The main content area is titled 'Starter Example' and contains a code snippet for a component named 'App'. The code uses `useState` to manage age and `useEffect` to update the UI. The code is highlighted with a red arrow and the text '3. Edit Code'. The UI shows a button 'Get older!' and a text 'Today I am 99 Years of Age'. The text 'Follow the instructions above as closely as possible' is highlighted with a red arrow and the text '4. View Live Result of the code'. Below the code, there is a note: 'NB: Even though the Instructions child component uses React.memo to optimize performance, it is still re-rendered.' and a question: 'How can this be fixed? i.e prevent <Instructions /> from re-rendering needlessly?'. The text '2. Some explanation' is written next to the note.

1. The hooks

3. Edit Code

2. Some explanation

4. View Live Result of the code

5. Some advanced examples

Starter Example

The following example will form the basis of the explanations and code snippets that follow.

```
const App = () => {
  const [age, setAge] = useState(99)
  const handleClick = () => setAge(age + 1)
  const someValue = "someValue"
  const doSomething = () => {
    return someValue
  }

  return (
    <div>
      <Age age={age} handleClick={handleClick}>
      <Instructions doSomething={doSomething}>
    </div>
  )
}
```

Today I am 99 Years of Age

- click the button below 📌

Get older!

Follow the instructions above as closely as possible

NB: Even though the Instructions child component uses React.memo to optimize performance, it is still re-rendered.

How can this be fixed? i.e prevent <Instructions /> from re-rendering needlessly?

Interactive Examples of Advanced Hooks

This is important because you can immediately begin to tinker with real examples that'll reinforce your knowledge of how hooks work.

Remember that learning is reinforced when you actually solve problems and build stuff.

After you get through the live examples for each of the hooks, there's an extra section for other generic examples that don't exactly fit one hook or require a separate case study.

In the example section, you'll find [examples](#) such as fetching data from a remote server using hooks and more.

Go [check it out](#).

Let's move on to the conclusion of this course and see what lies ahead.