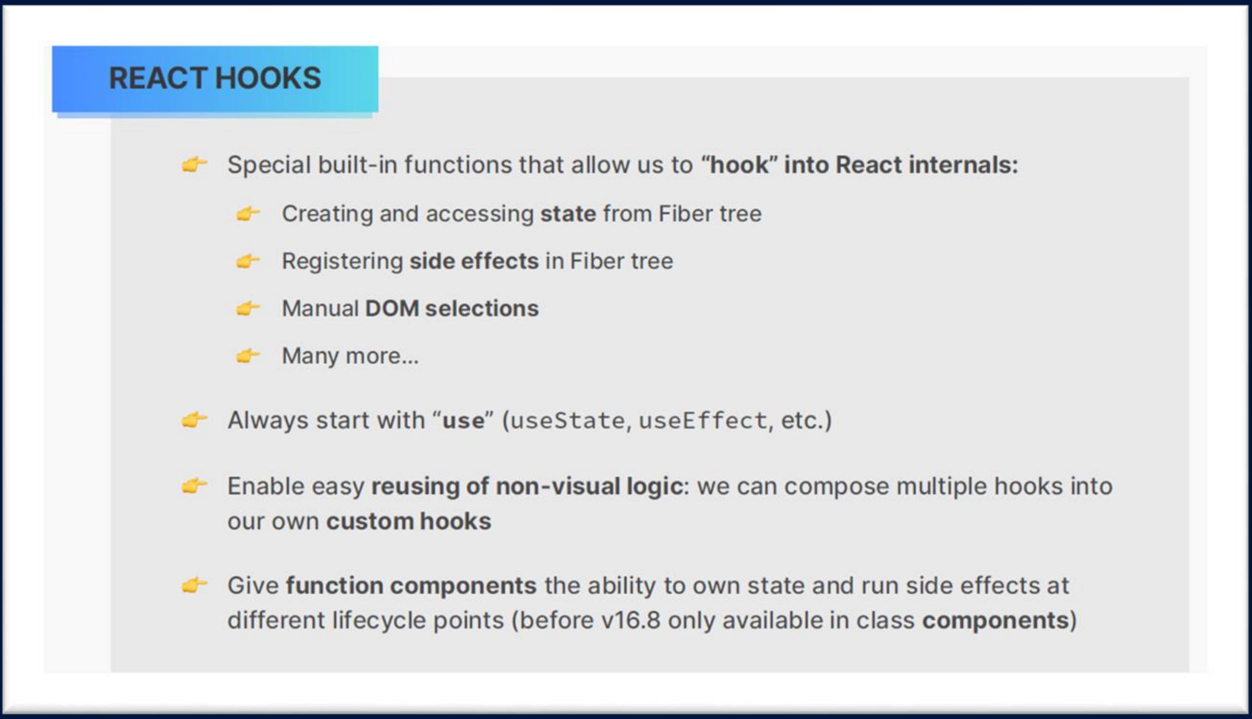
React

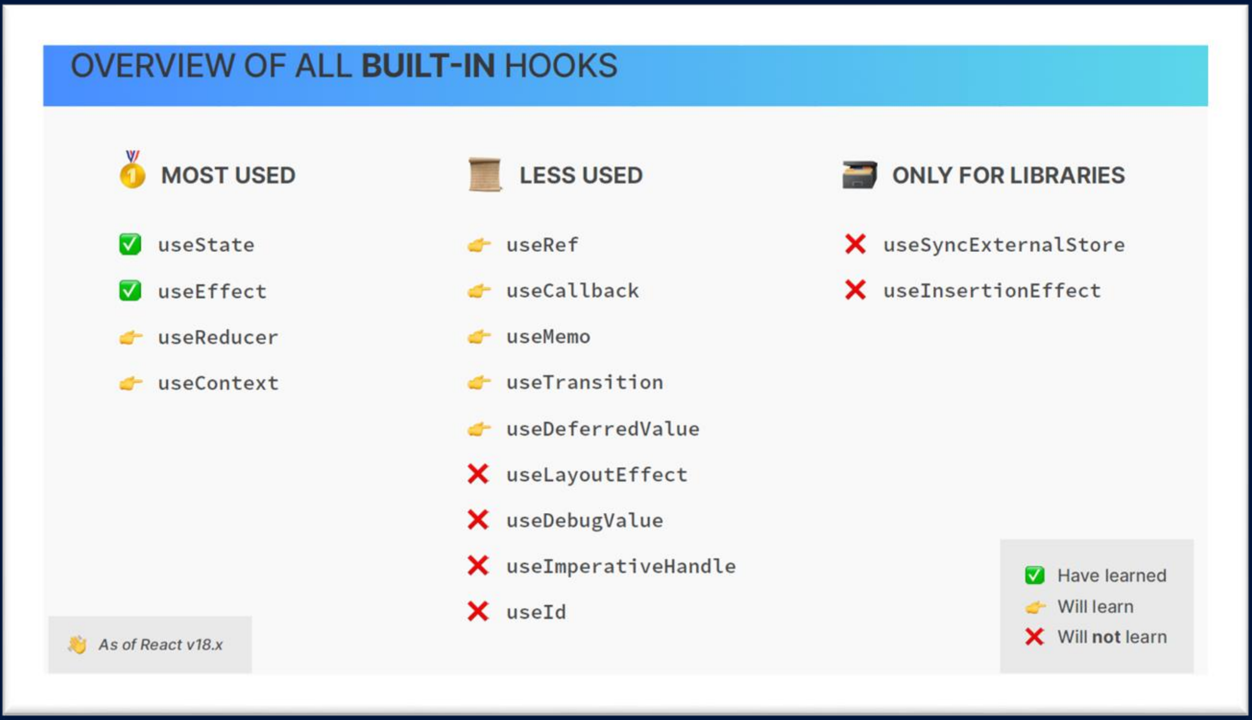
REACT HOOKS



REACT HOOKS

- Special built-in functions that allow us to **"hook" into React internals:**
  - Creating and accessing **state** from Fiber tree
  - Registering **side effects** in Fiber tree
  - Manual **DOM selections**
  - Many more...
- Always start with "**use**" (`useState`, `useEffect`, etc.)
- Enable easy **reusing of non-visual logic**: we can compose multiple hooks into our own **custom hooks**
- Give **function components** the ability to own state and run side effects at different lifecycle points (before v16.8 only available in class **components**)

React



OVERVIEW OF ALL **BUILT-IN** HOOKS

🥇 **MOST USED**

- ✅ `useState`
- ✅ `useEffect`
- 👉 `useReducer`
- 👉 `useContext`

📜 **LESS USED**

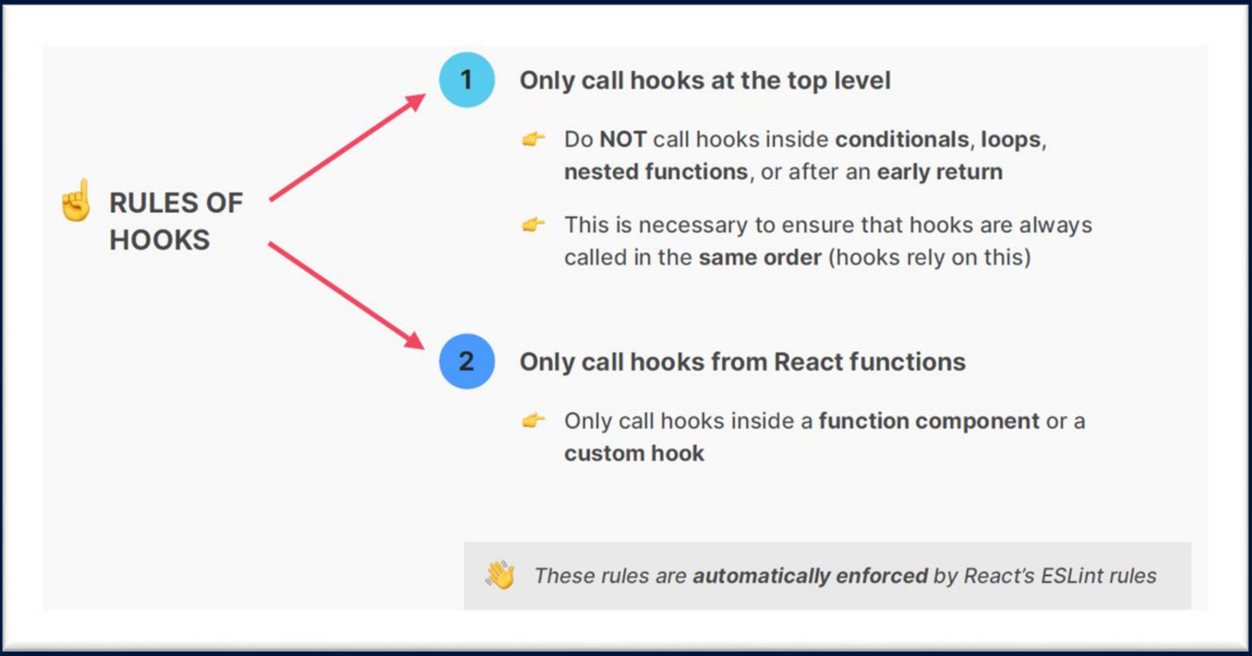
- 👉 `useRef`
- 👉 `useCallback`
- 👉 `useMemo`
- 👉 `useTransition`
- 👉 `useDeferredValue`
- ❌ `useLayoutEffect`
- ❌ `useDebugValue`
- ❌ `useImperativeHandle`
- ❌ `useId`

🗃 **ONLY FOR LIBRARIES**

- ❌ `useSyncExternalStore`
- ❌ `useInsertionEffect`

👋 *As of React v18.x*

✅ Have learned
👉 Will learn
❌ Will **not** learn



☝ **RULES OF HOOKS**

1 Only call hooks at the top level

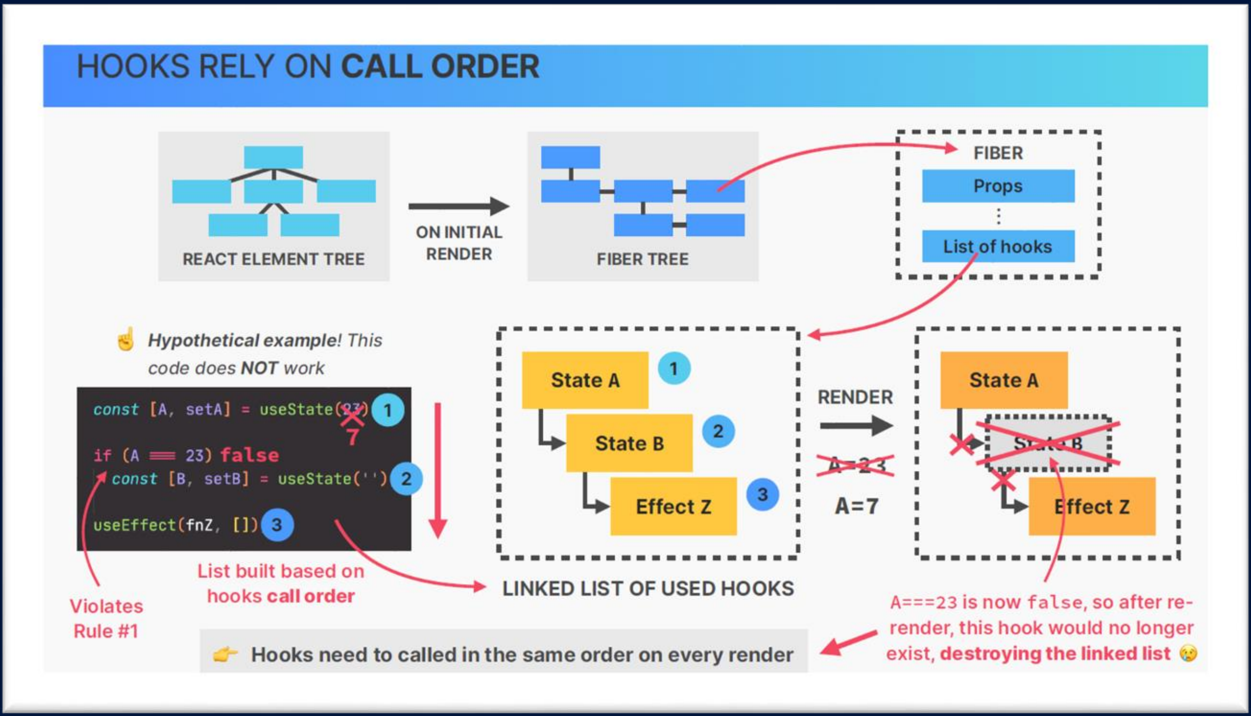
- 👉 Do **NOT** call hooks inside **conditionals**, **loops**, **nested functions**, or after an **early return**
- 👉 This is necessary to ensure that hooks are always called in the **same order** (hooks rely on this)

2 Only call hooks from React functions

- 👉 Only call hooks inside a **function component** or a **custom hook**

👋 *These rules are **automatically enforced** by React's ESLint rules*

React



HOOKS RELY ON CALL ORDER

☝ ***Hypothetical example!*** *This code does* ***NOT*** *work*

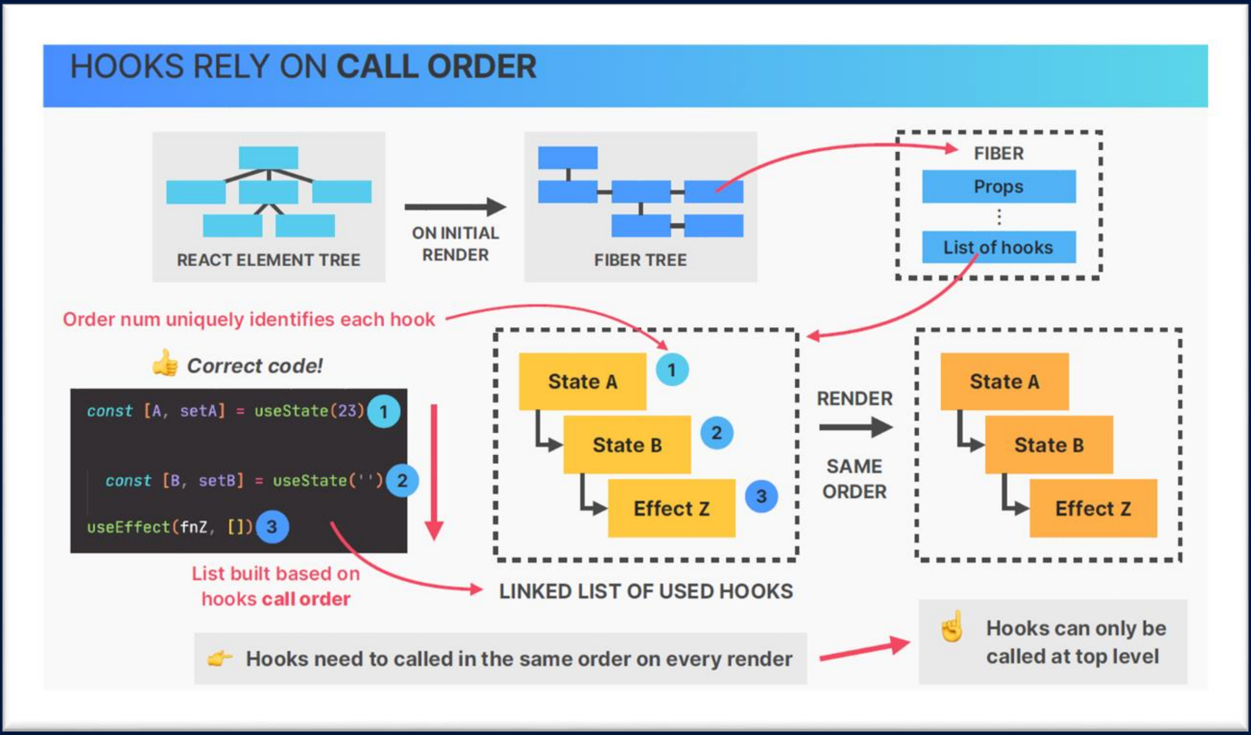
```
const [A, setA] = useState(23)

if (A === 23) false
  const [B, setB] = useState('')

useEffect(fnZ, [])
```

Violates Rule #1

List built based on hooks **call order**

A===23 is now false, so after re-render, this hook would no longer exist, **destroying the linked list** 😢

👉 Hooks need to called in the same order on every render



HOOKS RELY ON CALL ORDER

Order num uniquely identifies each hook

👍 *Correct code!*

```
const [A, setA] = useState(23)


  const [B, setB] = useState('')

useEffect(fnZ, [])
```

List built based on hooks **call order**

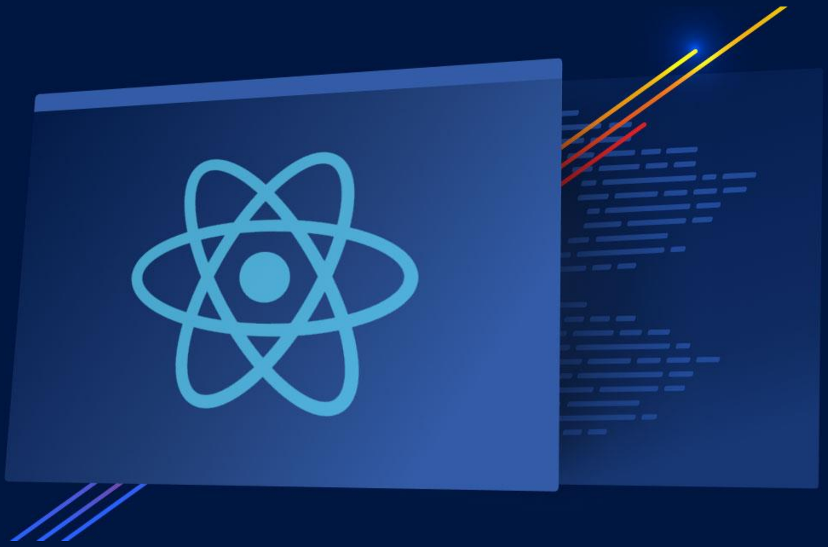
👉 Hooks need to called in the same order on every render

☝ Hooks can only be called at top level

React

Do you find it helpful?

Let me know down in the comments.



Click To Follow For More On LinkedIN