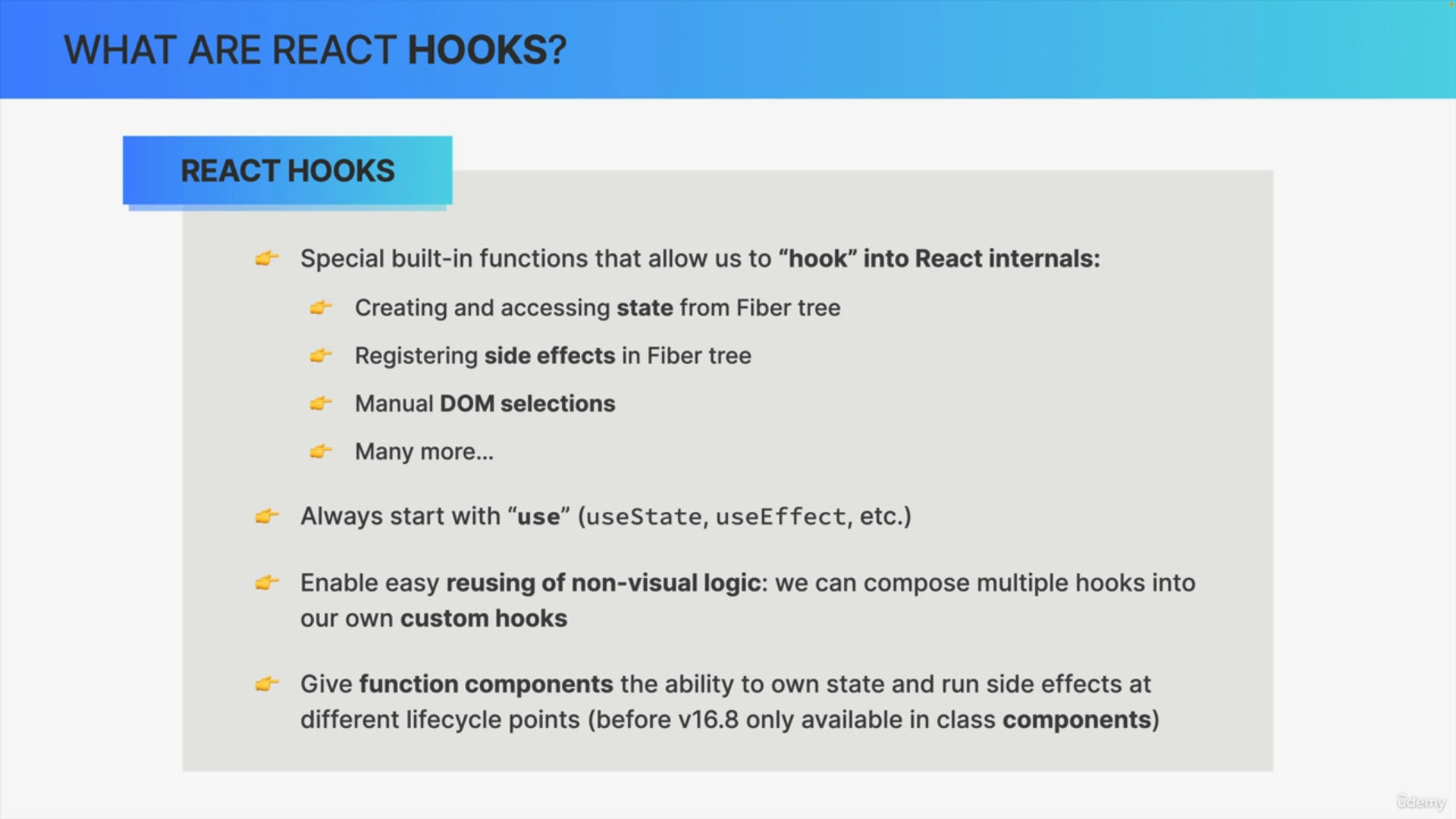
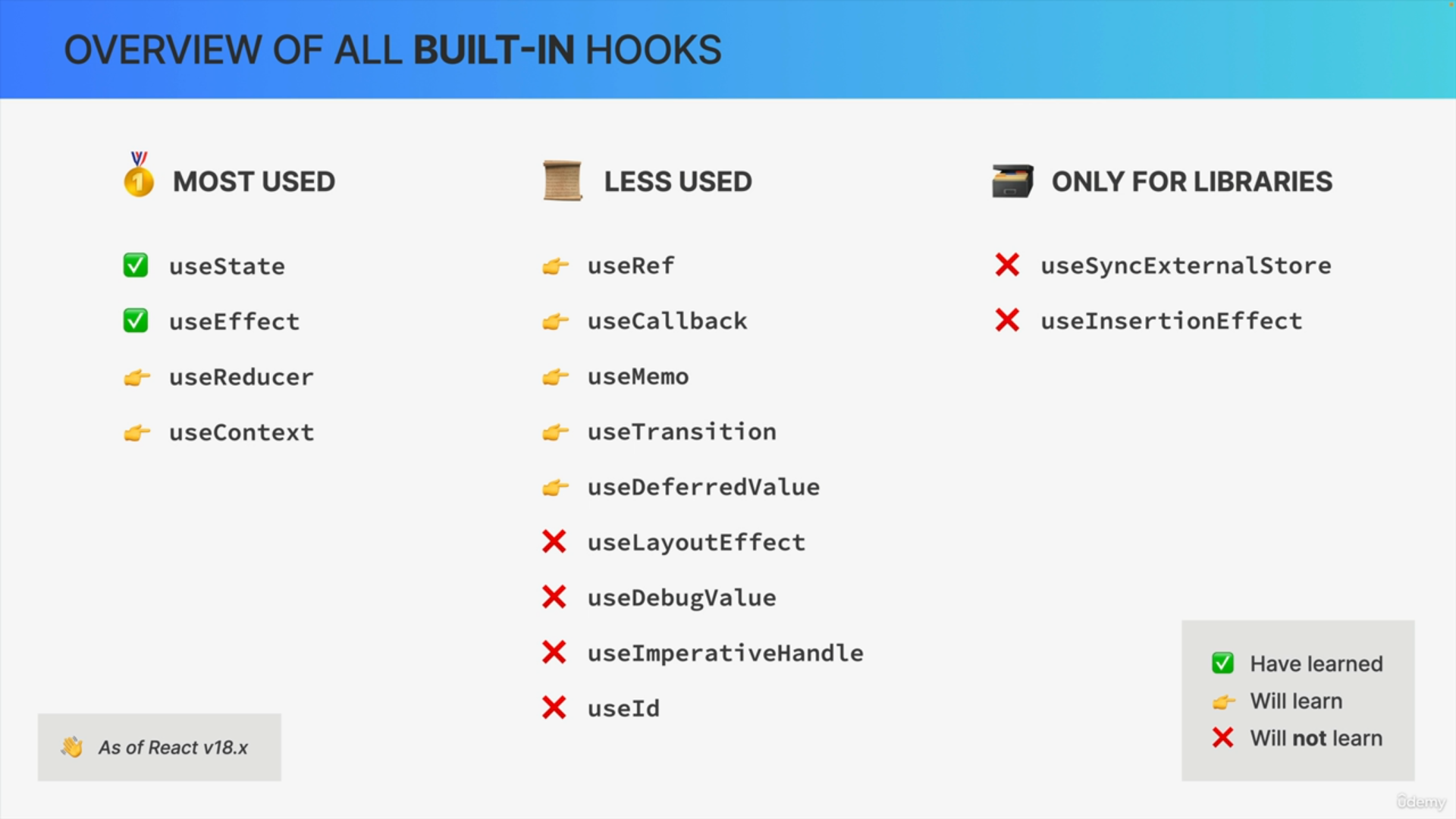
**Introduction to React Hooks**

* React hooks are special functions built into React that allow developers to access and utilize some of React's internal mechanisms.
* Hooks essentially act as APIs, exposing internal React functionality, such as state management and side effects, which are part of the fiber tree.
* The fiber tree is an internal React structure, and hooks like useState and useEffect provide access to it.





**Key Features and Benefits of Hooks**

* Hooks enable developers to manage state and lifecycle methods in functional components, eliminating the need for class components.
* Custom hooks allow developers to create reusable non-visual logic, starting with the prefix "use."
* React provides nearly 20 built-in hooks, with useState and useEffect being the most commonly used.

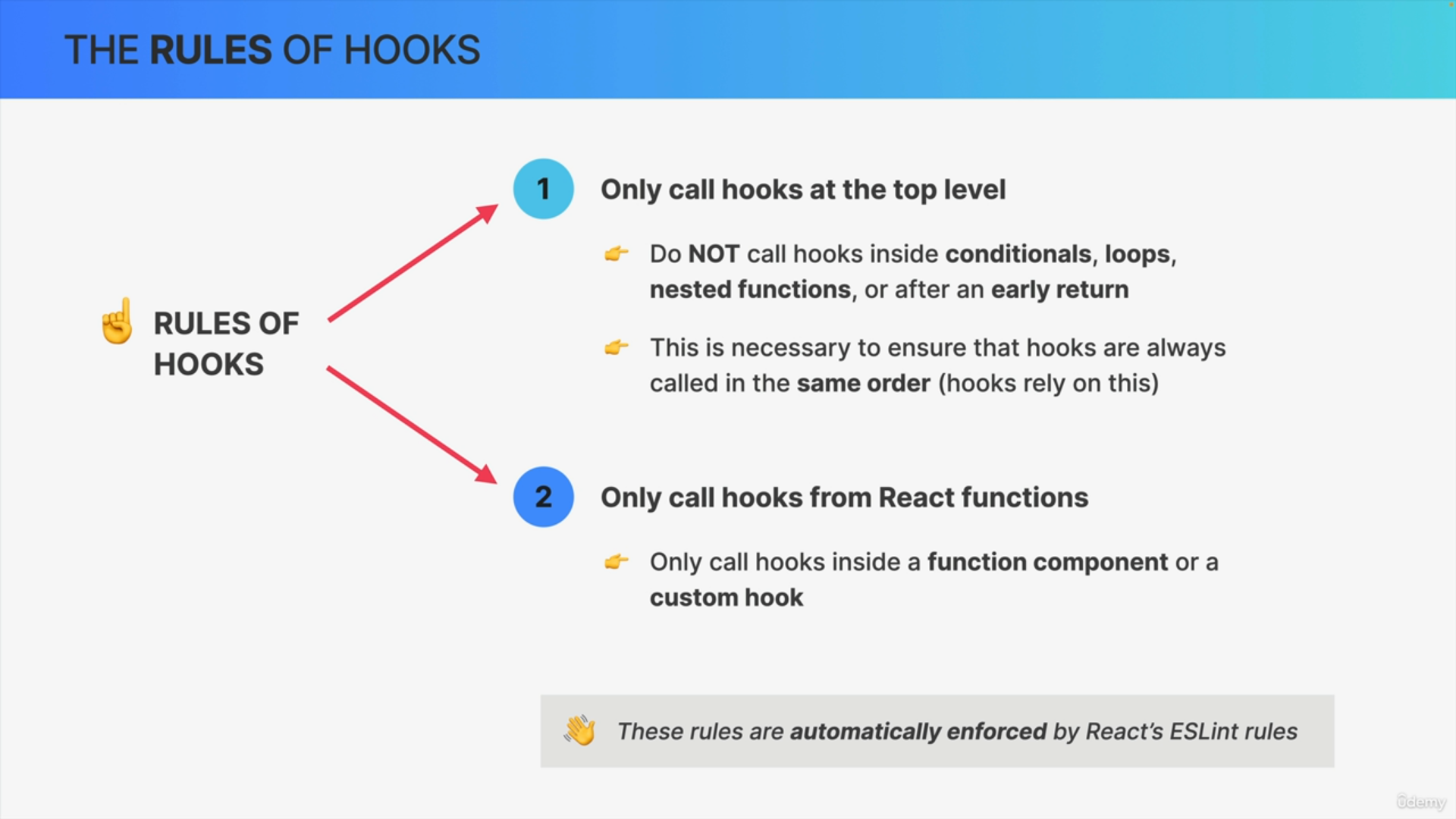
**Rules for Using Hooks**

**Top-Level Calls Only**

* Hooks must be called at the top level of a component, not inside loops, conditionals, or nested functions.
* This ensures hooks are called in the same order on every render, which is crucial for React to correctly maintain the hooks' state.

**React Functions Only**

* Hooks can only be used within React functional components or custom hooks, not in regular functions or class components.



**Why Hooks Must Follow These Rules**

* React creates a linked list of hooks in the order they are called during the initial render.
* If hooks are conditionally called, the order can change between renders, breaking the linked list and causing React to lose track of the hooks' state.
* By maintaining a consistent call order, React can correctly associate each hook with its value, simplifying state management without requiring developers to manually assign names to each hook.

**Conclusion**

* Following these rules ensures hooks work correctly and efficiently, enabling functional components to have state and lifecycle capabilities.
* Understanding these internals helps developers appreciate the design and constraints of hooks, leading to better coding practices.

