**What is a Ref?**

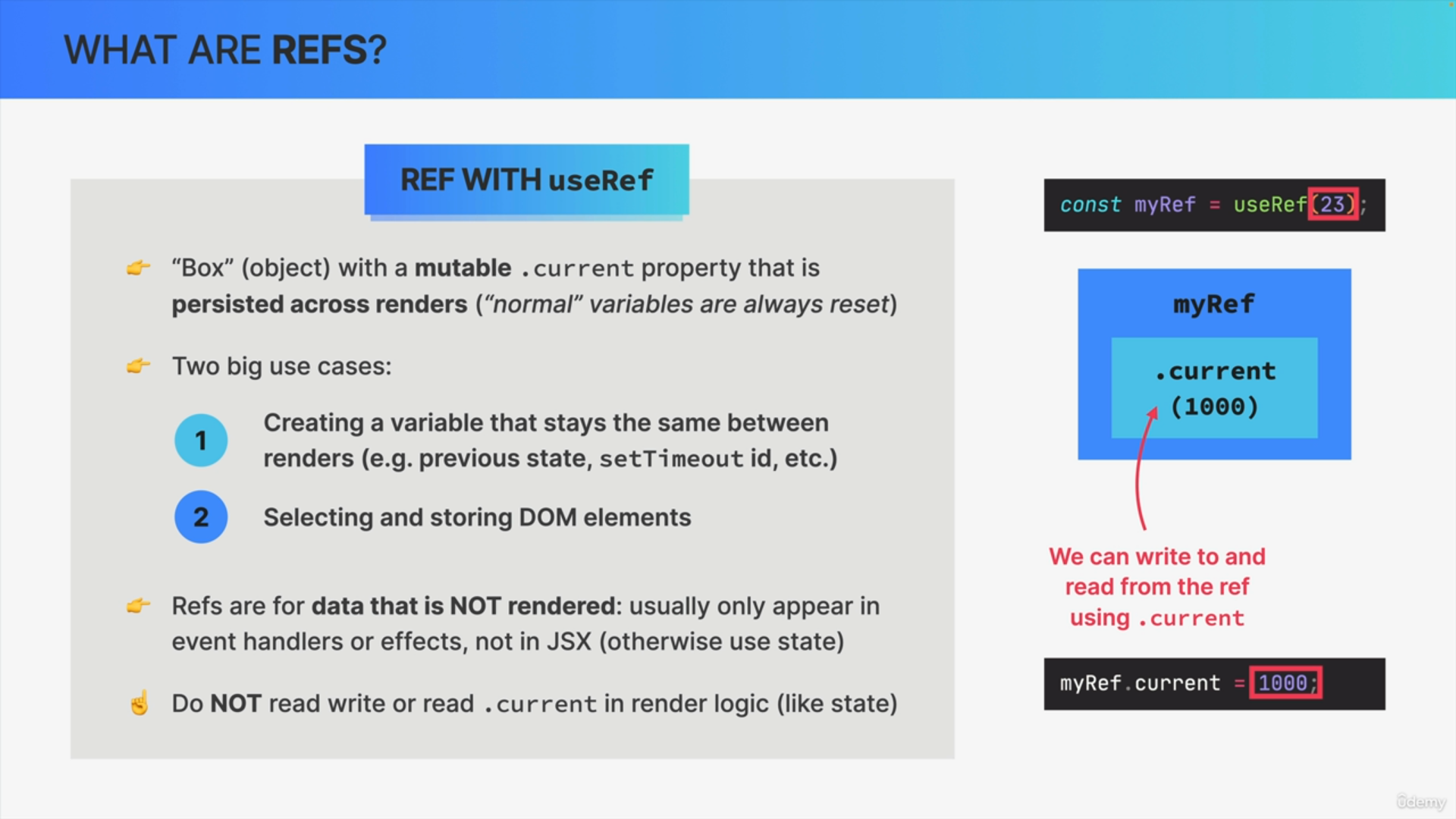
* Ref stands for reference and acts like a box to store data that needs to be preserved between renders.
* useRef creates an object with a mutable current property to store and read data.

**Key Characteristics of Refs**

* Persistence: current property value remains the same across renders.
* Mutability: Unlike other parts of React, the current property is mutable.
* Non-Rendering: Updating a ref does not cause the component to re-render, unlike state.

**Use Cases for Refs**

* Persistent Variables: For data that should stay the same between renders (e.g., previous state, setTimeout IDs).
* DOM Elements: To select and store DOM elements, as they need to be preserved across renders.



**Refs vs. State**

* Both refs and state are persisted across renders.
* State: Causes component re-rendering when updated; immutable and updated asynchronously.
* Refs: Do not cause re-rendering; mutable and updated synchronously.

**Best Practices**

* Use refs for data that doesn’t participate in the component’s visual output.
* Perform mutations inside useEffect to avoid side effects in render logic.
* Use state for data that should trigger re-rendering when changed.

**Summary of Differences Between Refs and State**

**Ref**

* Does not re-render the component on update.
* Mutable and updated synchronously.
* Suitable for data not used in JSX rendering.

**State**

* Re-renders the component on update.
* Immutable and updated asynchronously.
* Suitable for data that impacts the visual output of the component.

