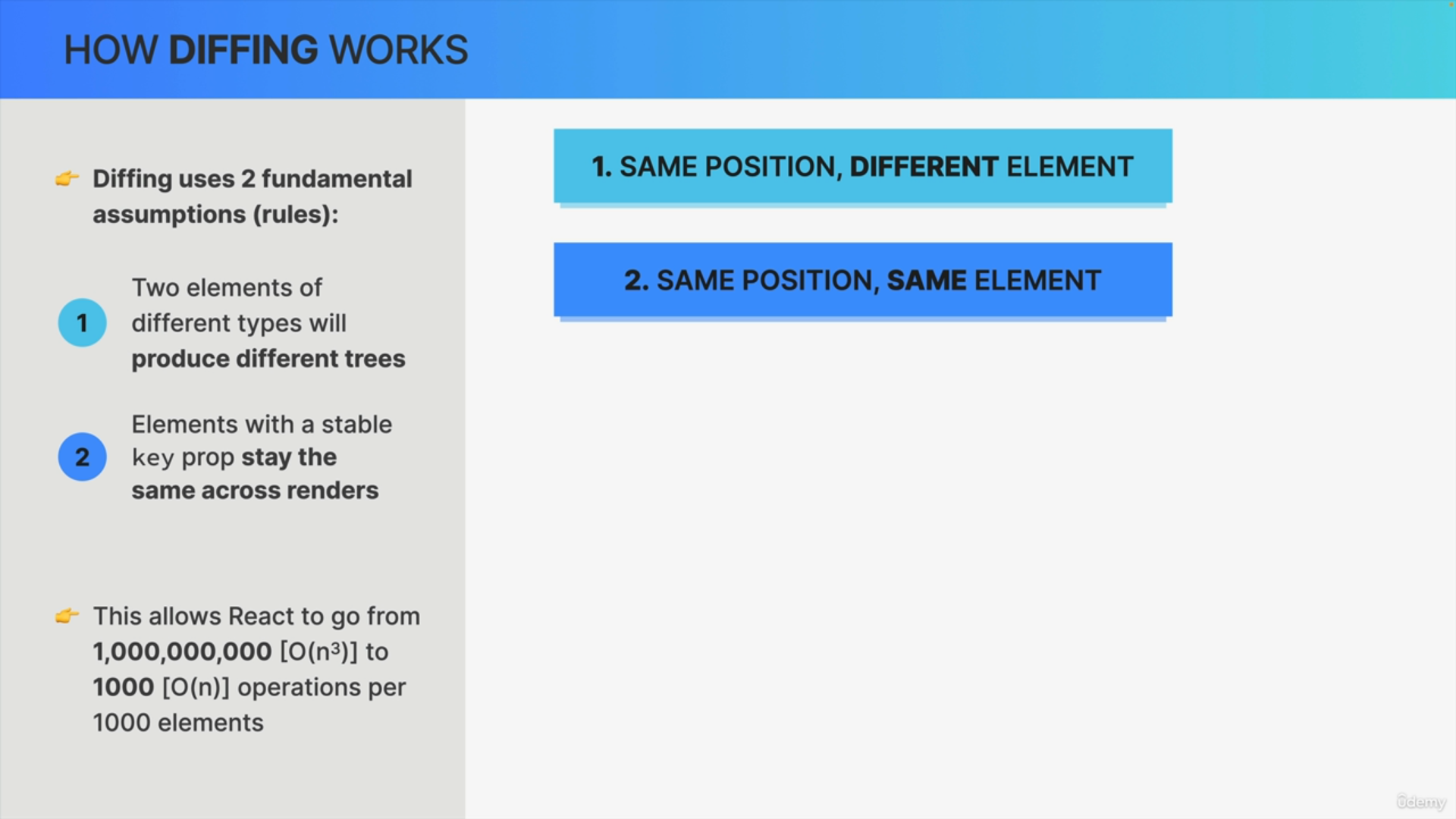
**Diffing**



**Purpose**

* These assumptions enable the algorithm to perform significantly faster, reducing operations drastically.

**Fundamental Assumptions**

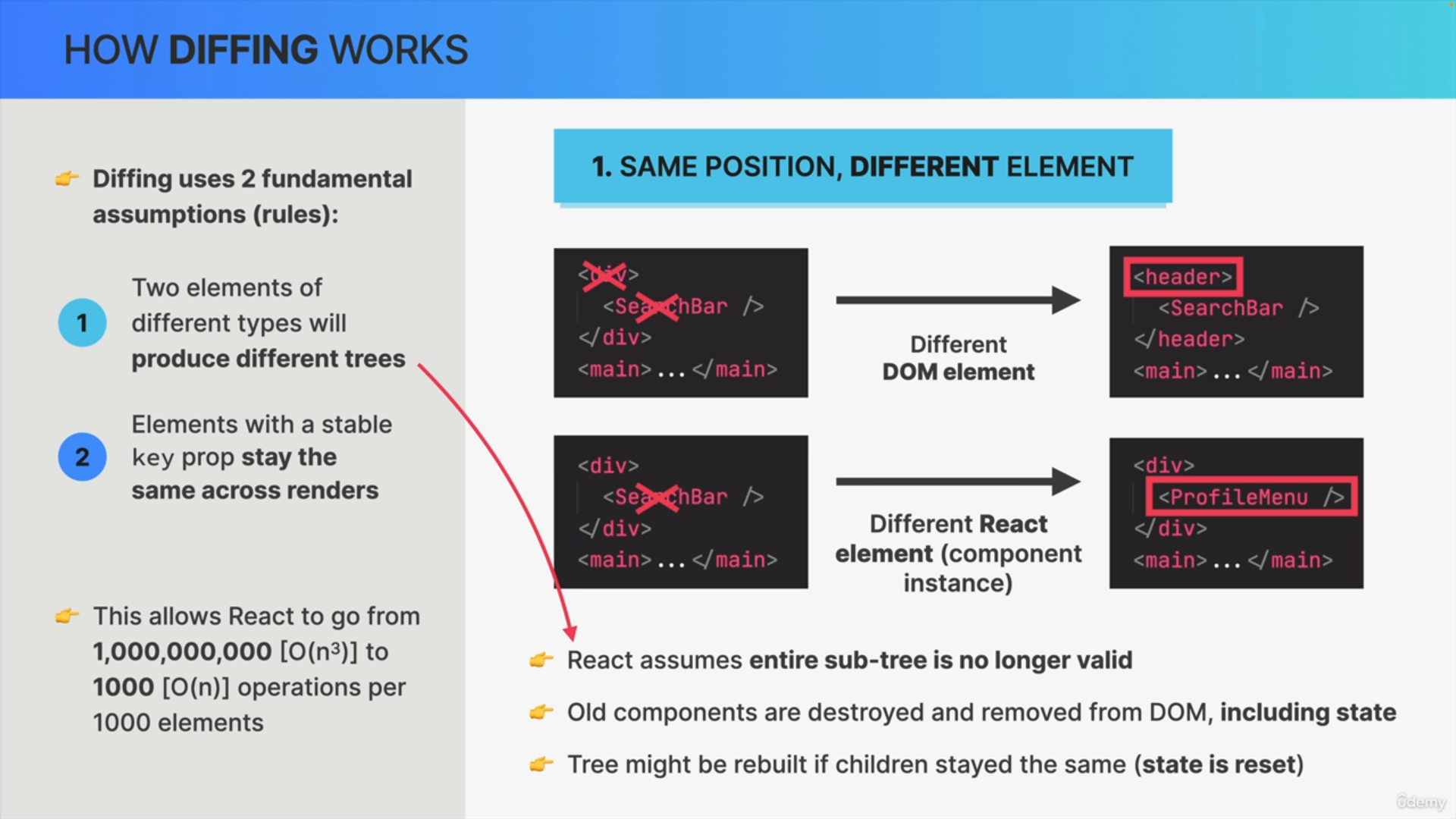
* Elements of different types produce different trees.
* Elements with a stable key remain the same across renders.

**Different Elements at the Same Position**

When an element type changes at a position in the tree (e.g., from <div> to <header>), React assumes the element and all its children are no longer valid. This leads to:

* Destruction and removal of the element and its children from the DOM.
* Loss of component state.

Example: Changing a <div> to a <header> results in the removal of both the div and its child components, including their state.

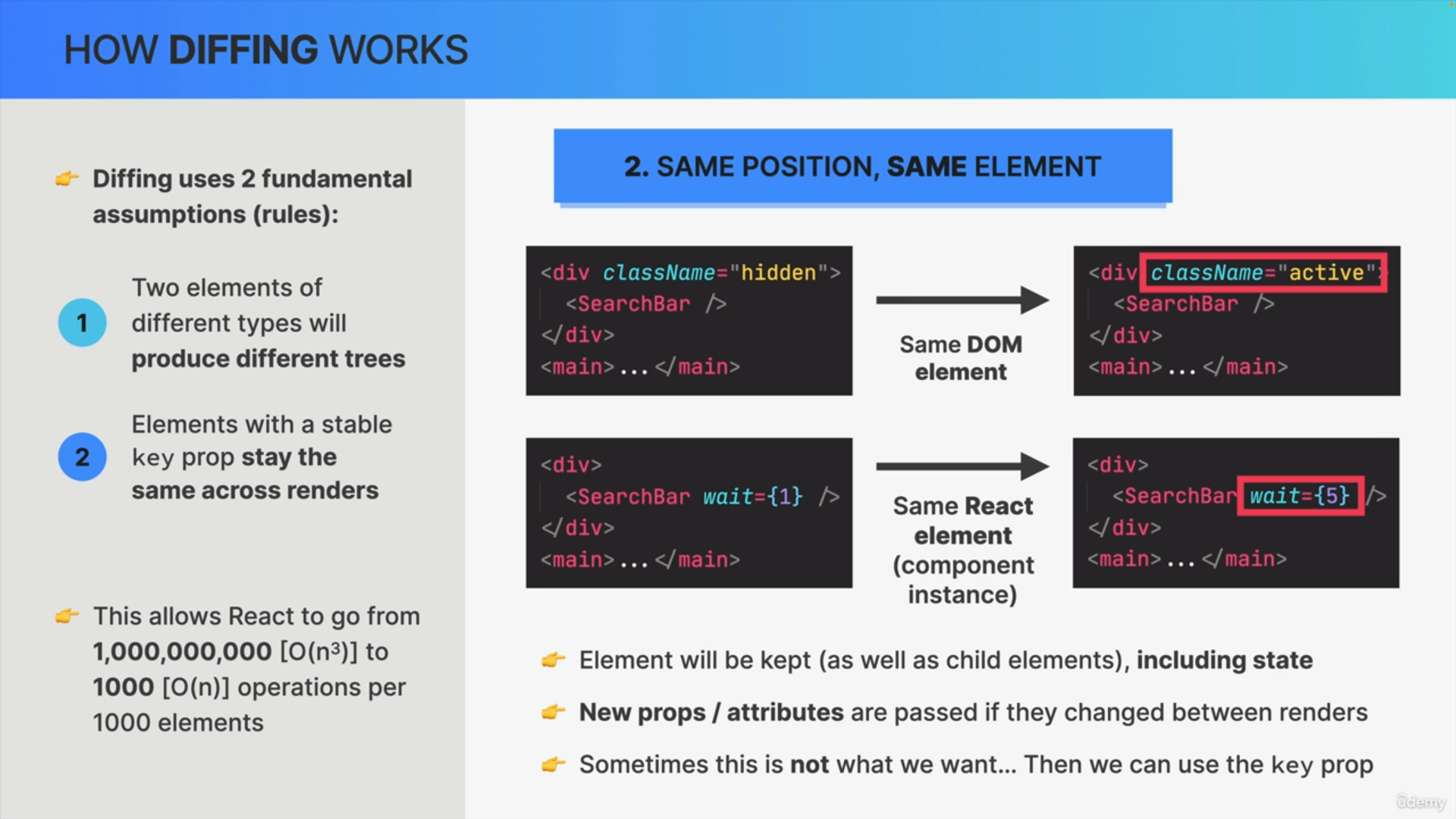


**Same Element at the Same Position**

If an element remains the same at a position in the tree between renders, it is kept in the DOM, preserving its state. React will:

* Mutate DOM element attributes or pass new props to React elements.
* Maintain component state.

Example: Changing the class name or prop value of an element will update the attributes without removing the element or its state.



**Practical Implications**

**Preservation and Loss of State:**

* Changing element types results in state loss, which is crucial to remember in application design.
* Same elements maintain state, leading to efficient updates.