**Why we can not write onClick={update()}**

In JavaScript and React, when you use `onClick={update()}`, it means that the `update` function is immediately invoked when the component renders. This is not what you typically want for event handlers in React.

When you use `onClick={update}`, you are passing a reference to the `update` function, and it will be called only when the actual click event occurs. This is the correct way to set up event handlers in React.

If you use `onClick={update()}`, it results in the function being called immediately when the component renders, which is not the intended behavior for handling a click event. This is why it's important to omit the parentheses when passing a function as a handler to events in React.

**Note** : It is similar to the reason why we cannot write onClick(setState());

**Certainly! Let's break down event.target.value:**

1. event: This represents the event that occurred, in this case, the onChange event. The event object contains information about the event, such as the type of event, the target element (where the event originated), and other relevant details.

2. event.target: This refers to the element that triggered the event. For example, if the onChange event occurred on an input field, event.target would point to that input element.

3. event.target.value: On an input element, this property represents the current value entered by the user. In the context of an input field, it provides the text that the user has typed or the value they have selected (in the case of a dropdown, for instance).

Putting it all together, event.target.value in the onChange handler of an input field gives you access to the text that the user is typing or has typed in that specific input field. This is useful for capturing and updating the state with the current input, allowing you to respond to changes in real-time as the user interacts with the input element.