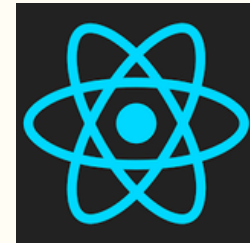




Controlled and Uncontrolled state

INTERVIEW QUESTIONS-80



Follow on 

@Duvvuru Kishore

Controlled and uncontrolled states in React refer to how the state of a component is managed.

Controlled State:

In a controlled state,

- **React manages and controls the state of the component.**
- **Any changes to the state are handled through React.**

Usage:

Controlled components are commonly used for form elements (like input fields, checkboxes, etc.), where React is responsible for maintaining and updating the state based on user input.

```
import React, {useState} from 'react'

export default function App() {
  const [value, setValue] = useState('');

  const handleChange = (e) => {
    setValue(e.target.value);
  };

  return <input type="text" value={value} onChange={handleChange}/>
}
```

Pros:

- **Predictable state management.**
- **Easier to implement dynamic UI updates based on state changes.**
- **Suitable for form validation and controlled data flow.**

Uncontrolled State 

Uncontrolled State:

In an uncontrolled state, **the state is not managed by React**. Instead, the component relies on the DOM to handle its state. Refs are often used to interact with the DOM directly.

Usage:

- Uncontrolled components might be used when integrating React with non-React code

or

- when you want to delegate the control of state to the DOM itself.

```
import React, { useRef } from "react";

export default function UnControlledState() {
  const inputRef = useRef(null);

  const handleClick = () => {
    alert(`Input Value: ${inputRef.current.value}`);
  };

  return (
    <>
      <input type="text" ref={inputRef} />
      <button onClick={handleClick}>Get Value</button>
    </>
  );
}
```

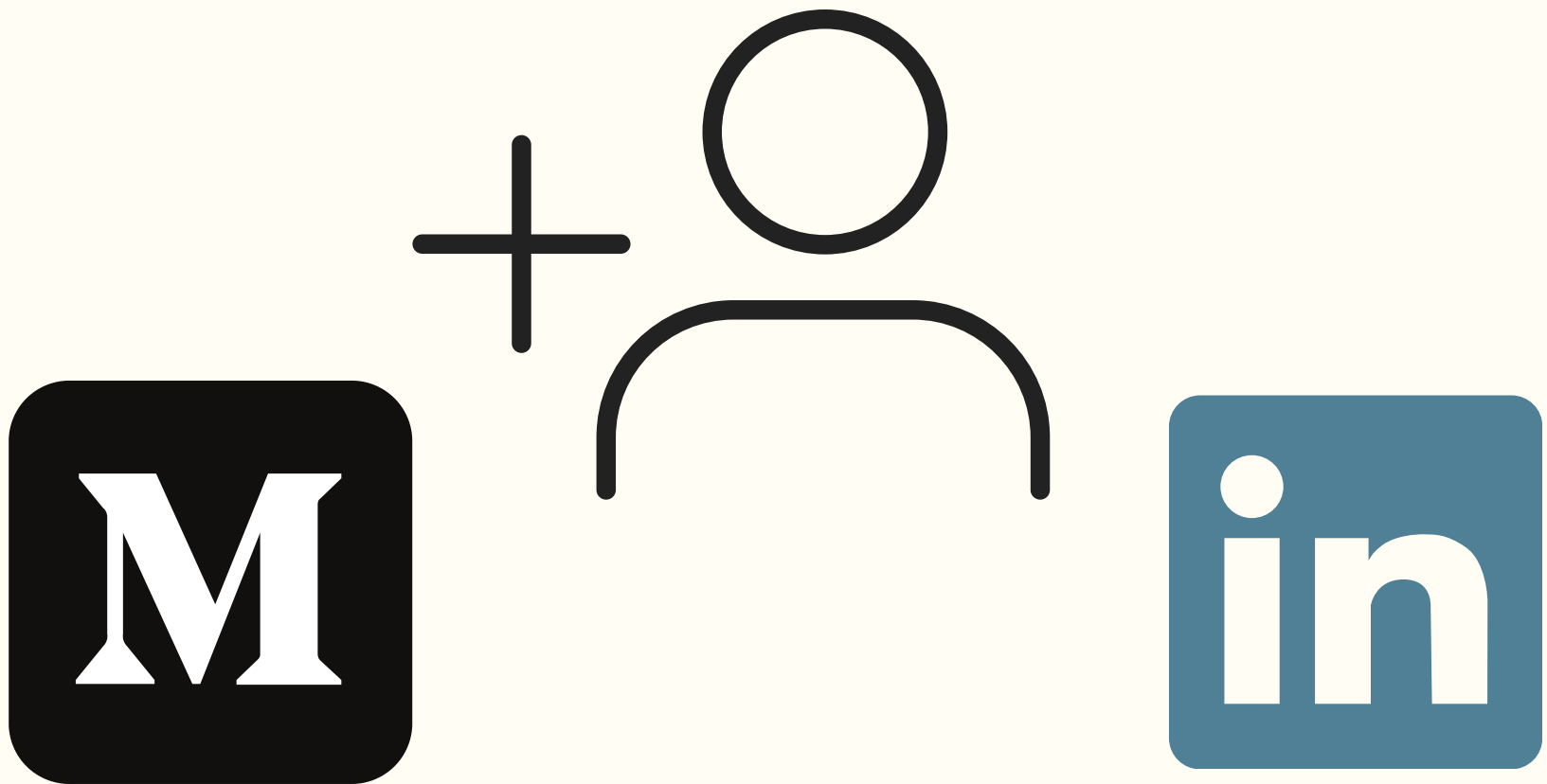
In this example:

- We have a simple form with an input field and a submit button.
- The input field is an uncontrolled component because its value is managed directly by the DOM using the `defaultValue` attribute.
- We use the `useRef` hook to create a reference (`inputRef`) to the input element.
- When the form is submitted, we access the current value of the input field directly from the DOM using `inputRef.current.value`.
- This demonstrates the use of uncontrolled components where React doesn't manage the state of the input field, and instead, we interact with the DOM directly.

Pros:

- **Can be simpler in certain scenarios, especially when integrating with non-React code.**
- **Might be suitable for integrating React with third-party libraries that manage their own state.**

In summary, controlled components have React managing the state, while uncontrolled components rely more on the DOM



Follow on  **@DUVVURU KISHORE**

