9. useRef로 컴포넌트 변수 생성하기

다중 선택	react 입문

useRef

- 함수형 컴포넌트에서 DOM 요소나 컴포넌트의 값(또는 어떤 값)을 저장하고 참조할 수 있게 해주는 훅(Hook)

1. DOM 요소 접근

`useRef`는 특정 DOM 요소에 직접 접근해야 할 때 사용됩니다. 예를 들어, 입력 필드에 포커스를 설정하거나, 스크롤 위치를 조작해야 할 때 유용합니다.

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

function InputFocus() {
  const inputRef = useRef(null);

  const handleFocus = () => {
    inputRef.current.focus(); // input 요소에 포커스를 설정
  };

  return (
    <div>
      <input ref={inputRef} type="text" />
      <button onClick={handleFocus}>Focus Input</button>
    </div>
  );
}

export default InputFocus;
```

- `inputRef`는 `input` DOM 요소를 참조합니다.

- `inputRef.current`를 통해 DOM 요소에 접근할 수 있습니다.

---

2. 값 유지

`useRef`는 렌더링 간에 변하지 않는 값을 저장하는 데도 사용됩니다. 이는 컴포넌트가 다시 렌더링되더라도 `useRef`에 저장된 값이 초기화되지 않음을 의미합니다.

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

function Counter() {
  const renderCount = useRef(0); // 렌더링 횟수를 저장

  const [count, setCount] = useState(0);

  renderCount.current += 1;

  return (
    <div>
      <p>Count: {count}</p>
      <p>Rendered: {renderCount.current} times</p>
      <button onClick={() => setCount(count + 1)}>Increment
</button>
    </div>
  );
}

export default Counter;
```

- `renderCount.current`는 렌더링 횟수를 추적하며, 상태 업데이트로 인해 컴포넌트가 **다시 렌더링되어도 값이 유지**됩니다.

---

3. 비제어 컴포넌트

폼 요소에서 `useRef`를 사용하여 비제어 컴포넌트를 구현할 수 있습니다. 이 접근 방식은 상태를 사용하지 않고 입력 값을 다룰 때 사용됩니다.

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

```
function UncontrolledForm() {
  const inputRef = useRef(null);

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

  return (
    <form onSubmit={handleSubmit}>
      <input ref={inputRef} type="text" />
      <button type="submit">Submit</button>
    </form>
  );
}

export default UncontrolledForm;
```

- 상태를 관리하지 않고도 입력 값을 가져옵니다.

---

4. useRef와 상태 비교

`useRef`와 `useState`의 주요 차이점:

- `useRef`의 값은 업데이트해도 컴포넌트가 **재렌더링되지 않습니다**.
- `useState`는 값이 업데이트되면 컴포넌트를 **다시 렌더링**합니다.

→ `useRef`는 간단한 데이터 참조, DOM 조작, 성능 최적화 등에 유용합니다. 하지만 상태 관리가 필요한 경우에는 `useState`를 사용하는 것이 더 적합합니다.