

리액트 심화 (3)_ref

: 직접 DOM을 참조할 수 있게 해준다.

: 상태를 사용하지 않고 DOM의 값들을 조작할 수 있다.

1. Referencing values with refs

when you want a component to "remember" some information, but you don't want that information to `trigger new renders`, you can use a ref.

you will learn

- how to add a ref to your component
- how to update a ref's value
- how refs are different from state
- how to use refs safely
- **adding a ref to your component**

you can add a ref to your component by importing the `useRef` hook from react:

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

2. Manipulating the DOM with Refs

react automatically updates the DOM to match your render output, so your components won't often need to manipulate it. However, sometimes you might need access to the DOM elements managed by React—for example, to focus a node, scroll to it, or measure its size and position. there is no built-in way to do those things in react, so you will need a ref to the DOM node.

you will learn

- how to access a DOM node managed by react with the ref attribute.
- how the ref JSX attribute relates to the useRef hook
- how to access another component's DOM node
- in which cases it's safe to modify the DOM managed by react
- Getting a ref to the node

to access a DOM node managed by React, first, import the `useRef` Hook:

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

then, use it to declare a ref inside your component:

```
const myRef = useRef(null);
```

The `useRef` Hook returns an object with a single property called `current`.

3. practice 4-3

```
const AddUser = (props) => {  
  const [enteredUsername, setEnteredUsername] = useState('');  
  const [enteredAge, setEnteredAge] = useState('');  
  const [error, setError] = useState();  
}
```

```
const AddUser = (props) => {  
  const [enteredUsername, setEnteredUsername] = useState('');  
  const [enteredAge, setEnteredAge] = useState('');  
  
  const nameInputRef = useRef();  
  const ageInputRef = useRef();  
  
  const [error, setError] = useState();  
  
  const addUserHandler = (event) => {  
    event.preventDefault();  
    const enteredName = nameInputRef.current.value;  
    const enteredUserAge = ageInputRef.current.value  
  
    if(  
      enteredName.trim().length === 0 ||\  
      enteredUserAge.trim().length === 0  
    ){  
      setError({  
        title: "유효하지 않은 입력값",  
        message: "유효한 나이와 이름을 입력해주세요 (빈값이 아닌)",  
      });  
      return;  
    }  
  }  
}
```

-
- ref를 사용하면 직접 dom을 참조할 수 있다.
 - ref를 사용하면 상태를 사용하지 않고 dom의 값들을 조작할 수 있다.