

2024-04-03 5.7.6 Conditional Rendering - 2

! 현행 문서: 5.7.6 Conditional Rendering - 2

1 Conditional Rendering 실습

1 Conditional Rendering 실습

조건부 렌더링 만들어 보세요!

Component

```
import React, {FC, useState} from "react";

const data = [
  {id: 1, name: 'Kim', age: 23},
  {id: 2, name: 'Lee', age: 45},
  {id: 3, name: 'Park', age: 30},
  {id: 4, name: 'Kim', age: 25},
];

const ArrayRendering = () => {
  const [users, setUsers] = useState(data);

  const handleDelete = (id: number) => {
    setUsers(prev => {
      // TODO: To be implemented

      return [...prev];
    });
  };

  return (
    <>
      {
        users.map(({id, name, age}) =>
          <User id={id} name={name} age={age} onDelete={handleDelete}/>)
      }
    </>
  )
}

export default ArrayRendering;

interface UserProp {
  id: number;
  name: string;
  age: number;
  onDelete: Function;
}

const User:FC<UserProp> = ({id, name, age, onDelete}) => {
  return (
    <div className="user">
      <div>{name}</div>
      <div>{age}</div>
      <button onClick={() => onDelete(id)}>Delete</button>
    </div>
  )
}
```

Component Test

```

import ArrayRendering from "../ArrayRendering";
import {render, screen, waitFor} from "@testing-library/react";
import userEvent from "@testing-library/user-event";

/*
  key
*/
describe('Array rendering test', () => {
  const error = jest.spyOn(global.console, "error").mockImplementation();
  test('      . (Each child in a list should have a unique "key" prop.)', () => {
    render(<ArrayRendering/>);

    expect(error).not.toBeCalled();
  });

  test('      Lee .', async () => {
    render(<ArrayRendering/>);

    expect(screen.getByText('Lee')).toBeDefined();

    userEvent.click(screen.getAllByText('Delete')[1]);

    await waitFor(() => {
      expect(screen.queryByText('Lee')).not.toBeInTheDocument()
    })
  });
});

```