**useCallback을 사용하여, 이전에 정의한 onCreate, onRemove, onToggle 함수를 재활용하기**

-이전에 만든 저 세함수는 컴포넌트가 리렌더링 될 때마다 새로만들어짐.

-한번 선언하는 건 큰 부하가 안걸리지만, 매번 그러면 빡침

-따라서 컴포넌트에서 props가 바뀌지 않았으면 Virtual DOM에 새로 렌더링하는 것조차 하지 않고

컴포넌트의 결과물을 재사용할 건데, 그 전에 함수 재활용이 필수적이다.

//App.js

import React, { useRef, useState, useMemo, useCallback } from ‘react’;

…

function App() {

…

const onChange = useCallback(

e => {

…

},

[inputs]

);

…

const onCreacte = useCallback(() => {

…

}, [users, username, email]);

…

const onRemove = useCallback(

id => {

…

}, [users]

);

…

const onToggle = useCallback(

id => {

…

}, [users]

);

설명

1. const onRemove = useCallback( id => { … }, [users]);
   1. 첫번째 파라미터에 사용할 함수 정의
   2. 두번째 파라미터엔 함수 안에서 사용하는 상태 혹은 props가 있다면 꼭 넣어줘야한다.
2. 현재 챕터에선 눈에띄는 최적화 효과는 없음. 다음 챕터에서 컴포넌트 렌더링 최적화 작업을 해줘야 최적화가 됨.

**어떤 컴포넌트가 렌더링 되고 있는지 확인하기 위한 도구 React DevTools**

<https://chrome.google.com/webstore/detail/react-developer-tools/fmkadmapgofadopljbjfkapdkoienihi?hl=ko>

에서 크롬 확장프로그램으로 설치가능

이후 개발자 도구에서 Components -> 설정 -> highlight updates ,,,를 클릭하면 렌더링 될때마다 렌더링 되는 곳이 하이라이트 됨.