**삭제기능 만들기**

function User({ user, onRemove }) {

    return (

        <div>

                <b>{user.username}</b>  <span>({user.email})</span>

**<button onClick={() => onRemove(user.id)}>삭제</button>**

        </div>

    );

}

function UserList({ users, **onRemove**}) {

    return (

        <div>

            {

                users.map(user => (

**<User user={user} key={user.id} onRemove={onRemove} />**

                ))

            }

        </div>

    );

}

**onClick={() => onRemove(user.id)} 로 쓰는 이유**

={onRemove} : 그냥 함수이름 선언. 인자를 넘길 수 없음

={onRemove(user.id)} : 함수 자체를 선언. 인자 넘길 수 있지만, 읽자마자 바로 실행되서 무용지물

={() => onRemove(user.id)} : 함수 자체를 선언. 인자 넘길 수 있고, 이벤트 발생 때마다 호출됨.

**filter 내장함수 사용**

-불변성을 지키면서 배열에서 특정원소를 제거하는 방법

const onRemove = id => {

setUsers(users.filter(user => user.id !== id));

};

-user.id 가 파라미터로 일치하지 않는 원소만 추출해서 새로운 배열을 만듬.

-user.id가 id인 것을 제거한다는 것과 동의