**배열에 변화를 주는 두 가지 방법**

* 배열에 변화를 줄 때는 객체와 마찬가지로 불변성을 지켜줘야함
  + 따라서 배열의 push, splice, sort 등의 함수는 사용하면 안됨.
* 다음과 같이 하면 버튼을 누를때마다 배열 요소가 추가됨.

//App.js

const [inputs, setInputs] = useState({

username: ‘’,

email: ‘’

});

const { username, email } = inputs;

const onChange = e => {

const { name, value } = e.target;

setInputs({

…inputs,

[name]: value

});

};

const [users, setUsers] = useState([

{

id: 1,

username: 'velopert',

email: 'public.velopert@gmail.com'

},

{

id: 2,

username: 'tester',

email: 'tester@example.com'

},

{

id: 3,

username: 'liz',

email: 'liz@example.com'

}

]);

const nextId = useRef(4);

const onCreate = () => {

const user = {

id: nextId.current,

username,

email

};

setUsers([…users, user]); // 또는 setUsers(users.concat(user));

setInputs({

username: ‘’,

email: ‘’

});

nextId.current += 1;

};

…

<CreateUser

…

onChange={onChange}

onCreate={onCreate}

/>

1. spread 연산자 사용 : 기존 배열을 넣고, 새로운 원소를 기존 배열에 추가하는 느낌.

setUsers([…uesrs, user]); 처럼 […users, user] 로 씀

1. concat 함수 사용 : 기존의 배열을 수정하지 않고, 새로운 원소가 추가된 새로운 배열을 만들어줌.

setUsers(users.concat(user)); 로 씀.