

## 14. 배열에 항목 제거하기

이 섹션에서 사용된 코드는 다음 페이지에서 확인 할 수 있습니다.

이번에는 배열에 항목을 제거 할 때에는 어떻게 해야 하는지 알아보겠습니다.

우선, `UserList` 에서 각 `User` 컴포넌트를 보여줄 때, 삭제 버튼을 렌더링해주겠습니다.

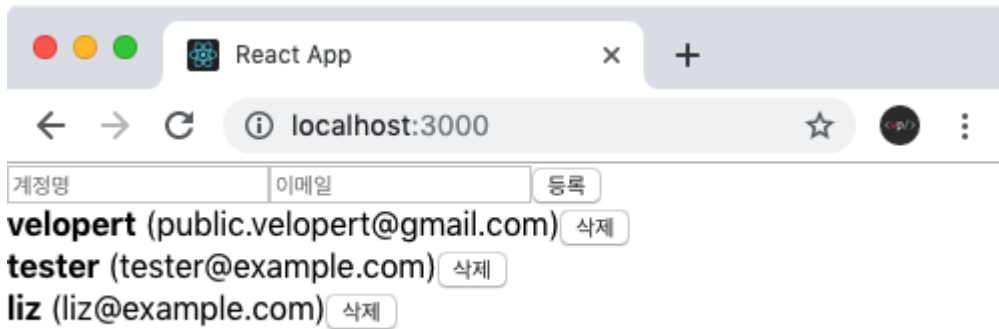
### UserList.js

```
import React from 'react';

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>
  );
}

export default UserList;
```



User 컴포넌트의 삭제 버튼이 클릭 될 때는 `user.id` 값을 앞으로 props 로 받아올 `onRemove` 함수의 파라미터로 넣어서 호출해주어야 합니다. 여기서 `onRemove` "id 가 \_\_인 객체를 삭제해라" 라는 역할을 가지고 있습니다.

이 `onRemove` 함수는 `UserList` 에서도 전달 받을것이며, 이를 그대로 User 컴포넌트에게 전달해줄것입니다.

이제, `onRemove` 함수를 구현해봅시다. 배열에 있는 항목을 제거할 때에는, 추가할때와 마찬가지로 불변성을 지켜가면서 업데이트를 해주어야 합니다.

불변성을 지키면서 특정 원소를 배열에서 제거하기 위해서는 `filter` 배열 내장 함수를 사용하는것이 가장 편합니다. 이 함수는 배열에서 특정 조건이 만족하는 원소들만 추출하여 새로운 배열을 만들어줍니다. ([참고](#))

App 컴포넌트에서 `onRemove` 를 다음과 같이 구현후, `UserList` 에게 전달해주세요.

## App.js

```
import React, { useRef, useState } from 'react';
import UserList from './UserList';
import CreateUser from './CreateUser';

function App() {
  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.concat(user));

  setInputs({
    username: '',
    email: ''
  });
  nextId.current += 1;
};

const onRemove = id => {
  // user.id 가 파라미터로 일치하지 않는 원소만 추출해서 새로운 배열을 만들
  // = user.id 가 id 인 것을 제거함
  setUsers(users.filter(user => user.id !== id));
};
return (
  <>
    <CreateUser
      username={username}
      email={email}
      onChange={onChange}
      onCreate={onCreate}
    />
    <UserList users={users} onRemove={onRemove} />
  </>
);
}

export default App;

```

이제 버튼들을 눌러서 삭제가 잘 이루어지는지 확인해보세요.

