

12. useRef 로 컴포넌트 안의 변수 만들기

컴포넌트에서 특정 DOM 을 선택해야 할 때, `ref` 를 사용해야 한다고 배웠었습니다. 그리고, 함수형 컴포넌트에서 이를 설정 할 때 `useRef` 를 사용하여 설정한다고 배웠었습니다.

`useRef` Hook 은 DOM 을 선택하는 용도 외에도, 다른 용도가 한가지 더 있는데요, 바로, 컴포넌트 안에서 조회 및 수정 할 수 있는 변수를 관리하는 것 입니다.

`useRef` 로 관리하는 변수는 값이 바뀐다고 해서 컴포넌트가 리렌더링되지 않습니다. 리액트 컴포넌트에서의 상태는 상태를 바꾸는 함수를 호출하고 나서 그 다음 렌더링 이후로 업데이트 된 상태를 조회 할 수 있는 반면, `useRef` 로 관리하고 있는 변수는 설정 후 바로 조회 할 수 있습니다. 이 변수를 사용하여 다음과 같은 값을 관리 할 수 있습니다.

- `setTimeout`, `setInterval` 을 통해서 만들어진 `id`
- 외부 라이브러리를 사용하여 생성된 인스턴스
- `scroll` 위치

우리는, `App` 컴포넌트에서 `useRef` 를 사용하여 변수를 관리해볼건데요, 용도는 우리가 앞으로 배열에 새 항목을 추가할건데, 새 항목에서 사용 할 고유 `id` 를 관리하는 용도입니다.

`useRef` 를 사용하여 변수를 관리하기 전에 해야 할 작업이 있습니다.

지금은 우리가 `UserList` 컴포넌트 내부에서 배열을 직접 선언해서 사용을 하고 있는데요, 이렇게 `UserList` 에서 선언해서 사용하는 대신에, 이 배열을 `App` 에서 선언하고 `UserList` 에게 `props` 로 전달을 해주겠습니다.

App.js

```
import React from 'react';

import UserList from './UserList';

function App() {
  const users = [
    {
      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'
    }
  ]
}
```

```

];
return <UserList users={users} />;
}

export default App;

```

UserList.js

```

import React from 'react';

function User({ user }) {
  return (
    <div>
      <b>{user.username}</b> <span>({user.email})</span>
    </div>
  );
}

function UserList({ users }) {
  return (
    <div>
      {users.map(user => (
        <User user={user} key={user.id} />
      ))}
    </div>
  );
}

export default UserList;

```

이제 App 에서 `useRef()` 를 사용하여 `nextId` 라는 변수를 만들어보겠습니다.

App.js

```

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

function App() {
  const users = [
    {
      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 = () => {  
  // 나중에 구현 할 배열에 항목 추가하는 로직  
  // ...  
  
  nextId.current += 1;  
};  
return <UserList users={users} />;  
}
```

export default App;

useRef() 를 사용 할 때 파라미터를 넣어주면, 이 값이 `.current` 값의 기본값이 됩니다.

그리고 이 값을 수정 할 때에는 `.current` 값을 수정하면 되고 조회 할 때에는 `.current` 를 조회하면 됩니다.
간단하지요?

이제 다음 섹션에서 새 항목을 배열에 추가하는 방법을 알아보겠습니다.