

day50-rn-useRef

☰ 태그	
📅 날짜	@2022년 12월 9일

우리는 웹 프로그래밍 시 특정 DOM을 선택해야 하는 상황에서 `getElementById` 혹은 `querySelector` 와 같은 DOM Selector 함수를 사용한다. 리액트 네이티브에서도 웹 프로그래밍처럼 특정 컴포넌트를 선택해야 하는 경우가 있다. 예를 들어, 컴포넌트로 포커스를 설정하고 싶은 경우 해당 컴포넌트를 선택할 수 있어야 한다.

- 기본형태

```
const ref = useRef(initialValue);
```

`useRef`를 사용할 때 주의해야 할 점은 두 가지이다.

1. 컴포넌트의 `ref`로 지정하면 생성된 변수에 값이 저장되는 것이 아니라 변수의 `.current` 프로퍼티(property)에 해당값을 담는다는 것이다.
2. `useState`를 이용하여 생성된 상태와 달리 `useRef`의 내용이 변경돼도 컴포넌트는 다시 렌더링되지 않는다는 점이다.

`useRef`를 사용할 때는 이 부분에 유의해야 한다. 이제 `useRef`를 이용해 Form 컴포넌트를 수정해보자.

```

import React, {useState, useEffect, useRef} from "react";
import styled from "styled-components/native";

const StyledTextInput = styled.TextInput.attrs({
  autoCapitalize : 'none',
  autoComplete : false,
})`
  border : 1px solid #757575;
  padding : 10px;
  margin : 10px 0;
  width : 200px;
  font-size : 20px;
`;

const StyledText = styled.Text`
  font-size : 24px;
  margin : 10px;
`;

const Form = () => {
  const [name, setName] = useState('');
  const [email, setEmail] = useState('');
  const refName = useRef(null);
  const refEmail = useRef(null);

  useEffect(() => {
    console.log( '\n=== Form Component Mount ===\n');
    refName.current.focus();
    return () => console.log( '\n=== Form Component Unmount ===\n');
  }, []);
}

```

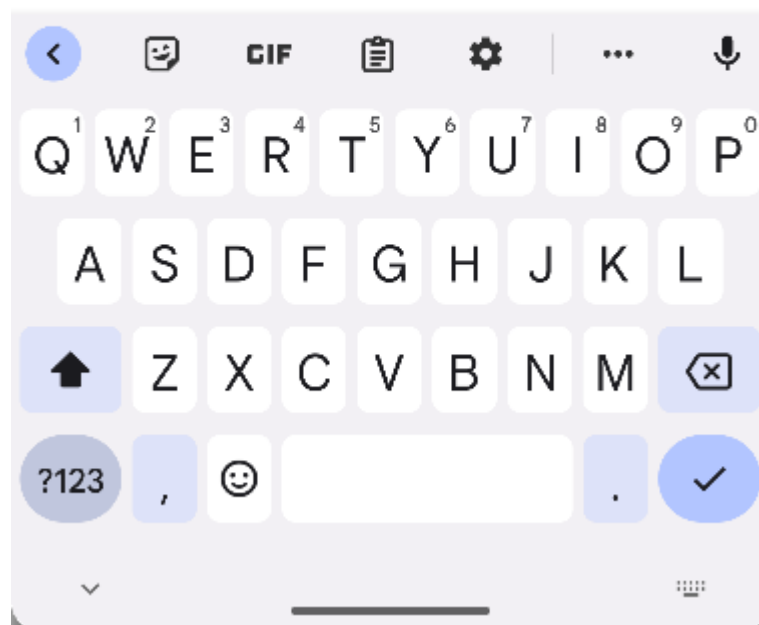
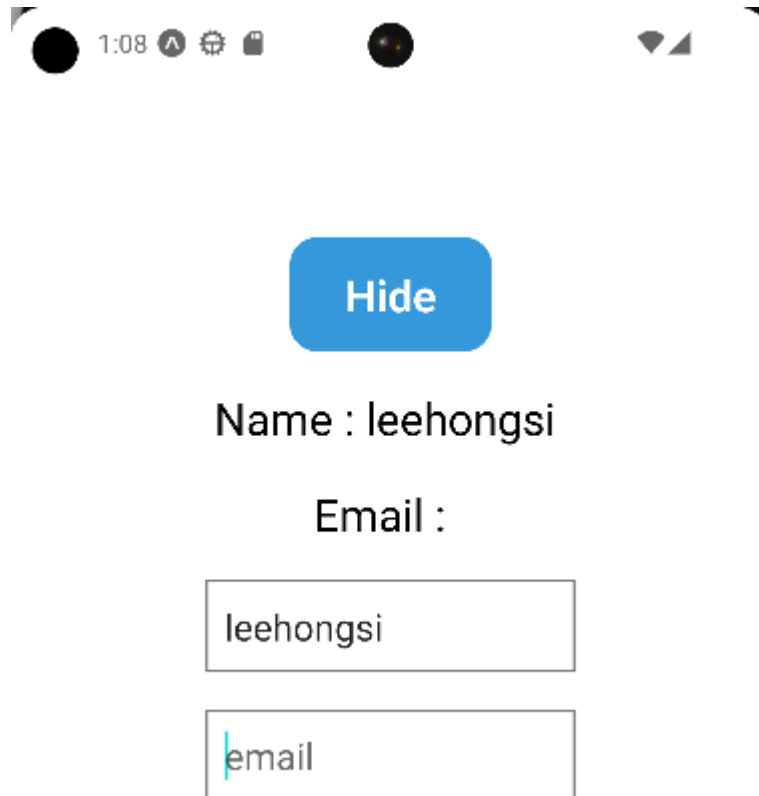
```

return (
  <>
    <StyledText>Name : {name} </StyledText>
    <StyledText>Email : {email} </StyledText>
    <StyledTextInput
      value={name}
      onChangeText={text => setName(text)}
      placeholder="name"
      ref={refName}
      returnKeyType="next"
      onSubmitEditing={() => refEmail.current.focus()}
    />
    <StyledTextInput
      value={email}
      onChangeText={text=> setEmail(text)}
      placeholder="email"
      ref={refEmail}
      returnKeyType="done"
    />
  </>
)
}

export default Form;

```

useRef함수를 이용하여 refName과 refEmail을 생성해 각각 이름과 이메일을 입력받는 TextInput컴포넌트의 ref로 설정했고, 키보드의 완료 버튼을 각각 next와 done으로 변경했다. 또한 이름을 입력받는 컴포넌트의 확인(next) 버튼을 클릭하면 이메일을 입력받는 컴포넌트로 포커스가 이동하도록 작성했으며, Form컴포넌트가 마운트될 때 포커스가 이름을 입력받는 컴포넌트에 있도록 수정했다.



이름을 적고 확인을 누르니 아래로 이동한다. 포커스가 잘 이동한것 같다.

