

- `const [names, setNames] = useState([ ...])` 부분은 함수형 컴포넌트에서 `names` 라는 state 를 정의하고, setter 인 `setNames` 를 정의한 것입니다. 이 때 `{useState}`를 import 하는 것을 잊지 않아야 합니다.
  - `useState` 내부에는 객체 형식의 상태도 정의할 수 있으며, 위 코드는 객체 형태의 배열을 정의한 모습입니다.
- 이 외에도 `inputText`, `nextId` 등의 상태를 정의하였습니다.
- `onChange` 함수는 input 요소에서 값을 입력받을 때 `onChange` 이벤트 핸들링을 위해 반드시 필요합니다. 사용자로부터 받은 값을 입력받겠다고 할 때에는 `onChange` 가 필수라고 알고 있으면 될 것 같습니다.
- `onClick` 함수는 버튼 클릭 이벤트 핸들링을 위해 정의하였습니다.
  - 내부에서 `names.concat` 부분은 `names` 라는 객체 형태 배열의 상태에 새로운 항목을 추가하는 것입니다.
  - 그러나 단순히 `concat` 함수로만 새 항목이 추가되는 것은 아니며, 상태를 변경하려면 setter 함수를 활용해야 합니다. 즉, 밑에 부분에서 `setNames(nextNames)` 라는 부분이 상태를 새롭게 변경해 주는 것이며, 이로 인해 새로운 값이 배열에 들어갈 수 있게 됩니다.

---

이 포스팅은 리엑트를 다루는 기술(저자 김민준)을 학습하는 과정에서 개인적으로 정리한 내용입니다.