**React의 State란?**

컴포넌트 내부에서 관리되는 동적인 데이터ㅇ이다.

컴포넌트의 상태를 나타내며, 사용자 입력 등에 따라 변경될 수 있다.

state가 변경되면 리액트는 자동으로 해당 컴포넌트와 자식 컴포넌트를 다시 렌더링 한다.

useState() 훅를 통해 state 상태를 관리할 수 있다.

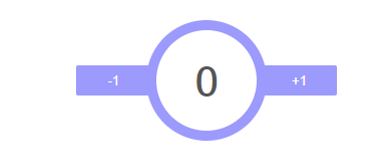
|  |
| --- |
| **리액트 훅(Hooks)**  함수형 컴포넌트에서 상태와 생명주기 기능을 사용할 수 있도록 해주는 기능으로 단순하고 이해하기 쉬우며, 클래스 인스턴스를 생성하지 않으므로 빠르다. 다양한 종류의 훅이 있으며, 대표적으로 useState, useEffect, useContext, useReducer, useCallback, useMemo, useRef 등이 있다. |

**State 선언**

state 변수와, state 값을 변경할 함수 이름을 함께 선언하며, 우변에는 초기값을 useState() 의 괄호안에 작성한다. - useState 사용을 위해 useState 모듈 추가

|  |
| --- |
| importReact, { useState } from'react';  **[ 사용 법 ]**  const [상태 값 저장 변수, 상태 값갱신함수] = useState(상태 초기 값);  **[ 예제 ]**  const [no, setNo] = useState(0); |

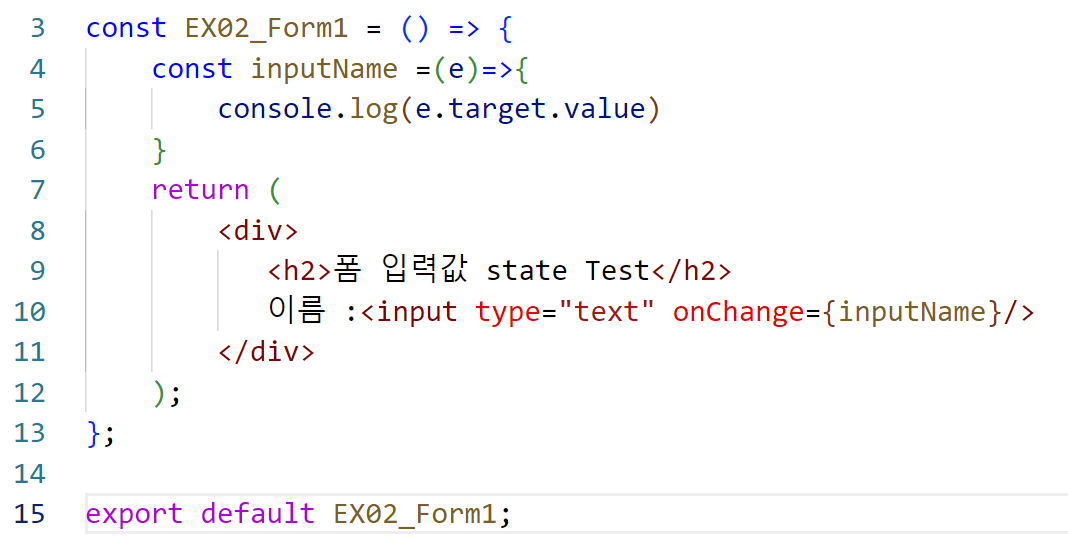
**useState 실습**

****

****

**State 사용하기 – Form01**

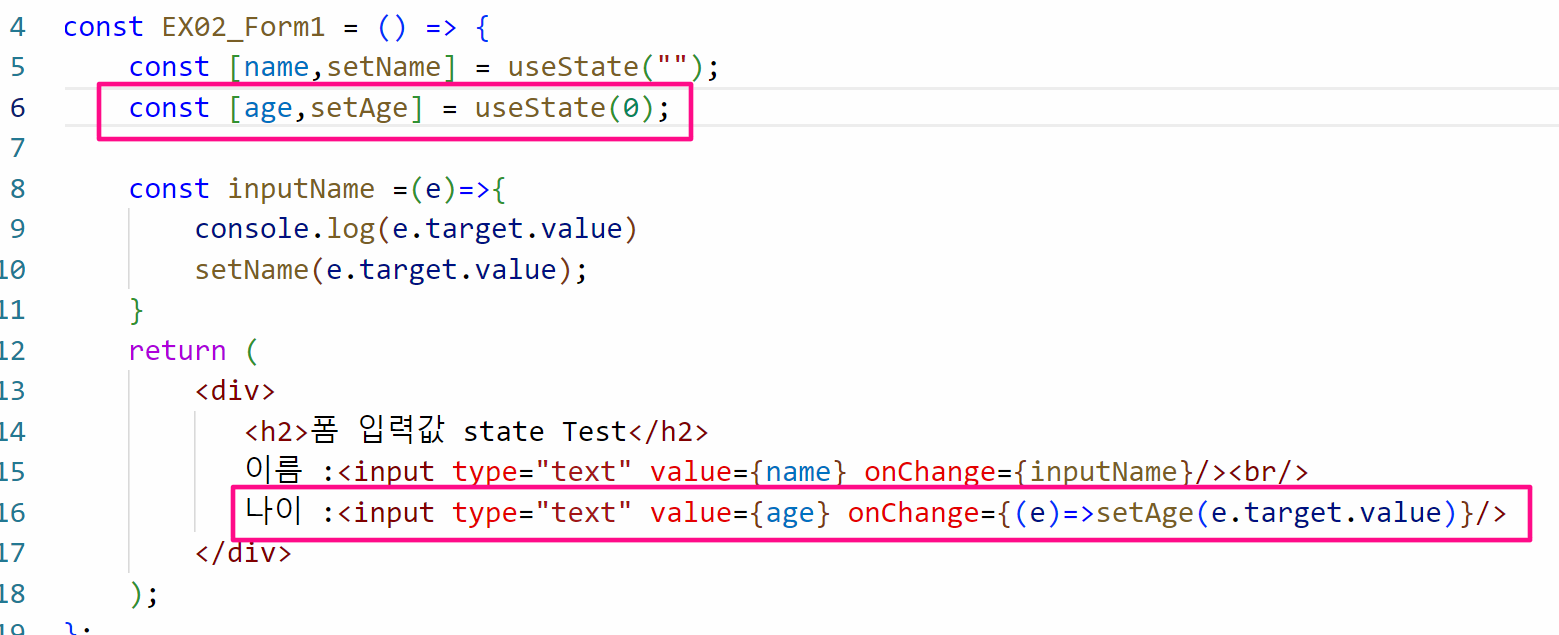
Form1.jsx 생성 및 input 태그 작성

****

**name이라는 state를 정의하여 입력값을 담기**

****

**나이 입력 폼 추가해보자.**

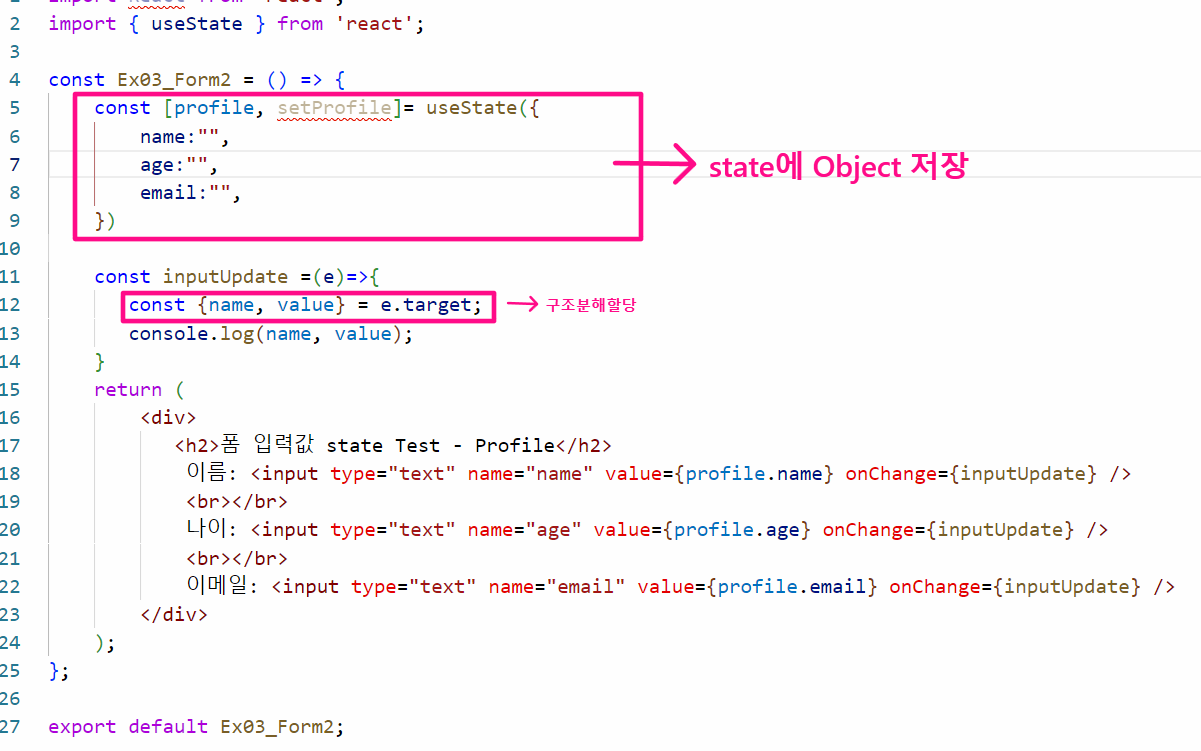
****

**State 사용하기 – Form02**

회원정보를 입력 받는 폼을 정의하고 입력된 정보를 profile이라는 하나의 object로 state 지정

Form2.jsx 생성 및 input 태그 작성 이름, 나이, 이메일에 각각 입력을 하면 inputUpdate 함수 호출 inputUpdate 함수는 입력한 값의 name, value를 콘솔에 출력한다.

|  |
| --- |
| **객체 비구조화 할당 (Object Destructuring Assignment)**  const { name, value } = e.target;  e.target에 저장된 name, value 속성에 담긴 값을 비구조화 할당 방식을 통해 추출 |

****

state 값을 쓰지 않았기 때문에 input에 값을 입력해도 화면에 입력 값이 보이지는 않는다.

**State에 값 담기**

inputUpdate 함수에 profile state 값을 담기 위한 코드 추가한다.

전개연산자(spread operator)를 사용하여 값을 담는다.

|  |
| --- |
| **전개 연산자(spread operator)**  전개 연산자는 배열이나 객체의 요소를 개별요소로 분리할 때 사용하는 연산자이다.  객체 값을 복사할 때 활용한다.    **…profile** : 기존 profile에 담긴값을 그대로 가져온다.  **[name] : value, :** 새롭게 추가된 값은 바꿔준다. |

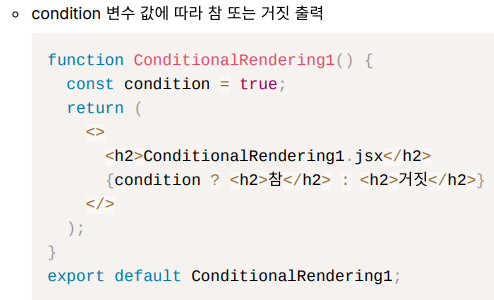
****

**조건부 랜더링**

state 값에 따라 화면에 보여줄 요소를 다르게 하는 것

삼항 연산자를 활용하는 것이 일반적이다.

|  |
| --- |
| **삼항연산자**  삼항연산자는 조건에 따라 실행할 내용을 구분할 수 있도록 하는 연산자로서 if, else 와 비슷한 기능을 한다. 표현이 간결하기 때문에 조건부 렌더링에 많이 활용된다. |

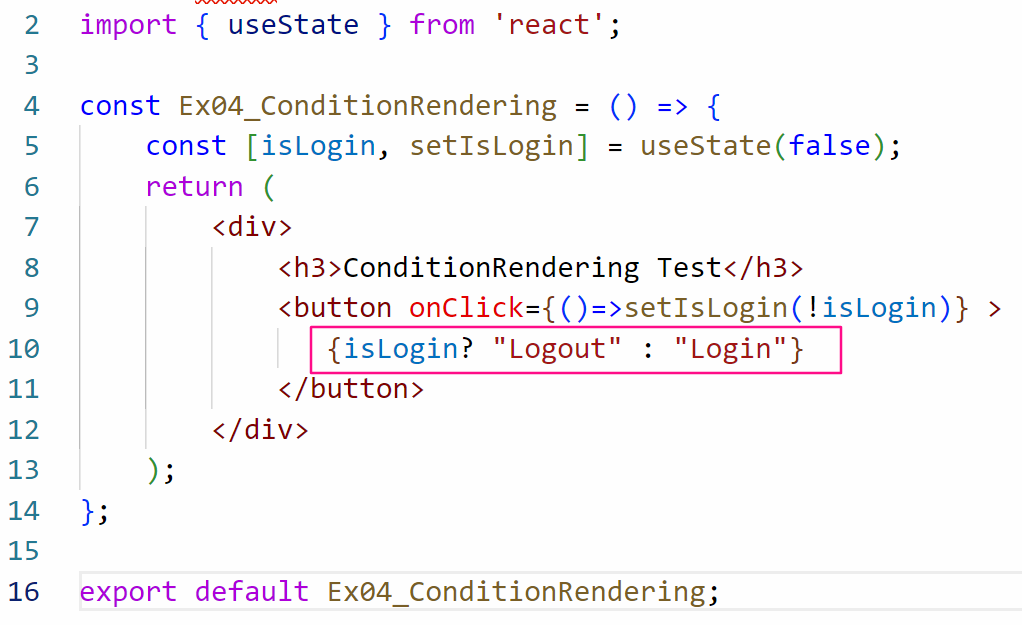
****

**조건부 렌더링 예제**

조건을 따질 변수를 state로 지정하여 값이 바뀜에 따라 화면에 보여지는 내용을 바꾼다.

버튼을 클릭할 때마다 isLogin의 반대 값을 isLogin 값으로 저장한다.

로그인상태(isLogin=true ⇒ Logout), 로그아웃상태(isLogin=false ⇒ Login) 출력한다.

****