**6. useState로 컴포넌트에서 바뀌는 값 관리하기**

* **Hooks?**

리액트에서 Hooks라는 기능이 도입 🡪 함수형 컴포넌트에서도 상태 관리 가능

* **useState?**

리액트의 Hooks중 하나인 함수

<button을 누르면 숫자가 바뀌는 Counter 컴포넌트 만들기>

* 버튼 클릭 이벤트 발생시 특정 함수가 호출되도록 설정
* **Counter.js**

import React from 'react';

function Counter() {

const onIncrease = () => {

console.log('+1')

}

const onDecrease = () => {

console.log('-1');

}

return (

<div>

<h1>0</h1>

<button onClick={onIncrease}>+1</button>

<button onClick={onDecrease}>-1</button>

</div>

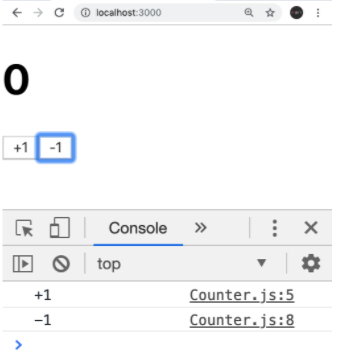
);

}

export default Counter;

* 리액트에서 엘리먼트에 이벤트 설정 :  on이벤트이름={실행하고싶은함수} 형태로 설정

\* onClick={onIncrease()} 로 함수 실행하면 안됨 \*

 <<실행 화면

* **동적인 값 끼얹기, useState**
* **Counter.js**

import React, { useState } from 'react'; //리액트 패키지에서 useState 함수 호출

function Counter() {

const [number, setNumber] = useState(0); //첫원소:현재상태, 두번째:Setter함수

//배열 비구조화 할당

const onIncrease = () => {

setNumber(number + 1);

}

const onDecrease = () => {

setNumber(number - 1);

}

return (

<div>

<h1>{number}</h1> //0이 아닌 {number}

<button onClick={onIncrease}>+1</button>

<button onClick={onDecrease}>-1</button>

</div>

);

}

export default Counter;

* **함수형 업데이트**

setNumber(prevNumber => prevNumber + 1);

* setNumber을 사용할 때 그 다음 상태를 파라미터가 아닌 값을 업데이트하는 함수를 파라미터로 넣어줌.
* 함수형 업데이트는 주로 나중에 컴포넌트를 최적화를 하게 될 때 사용하게 된다.