React란

UI 컴포넌트를 만드는 일을 하며 캡슐화를 통한 재사용성이 매우 높다.

기존 코드를 모두 바꿀 필요 없이 필요한 부분을 조금씩 변경하여 사용 가능하다.

React 작업 환경 준비

**새 프로젝트 만들기(git bash 이용)**

$ npx create – react –app begin-react

begin-react 라는 디렉터리를 생성하고 그 안에 프로젝트가 생성된다.

이후

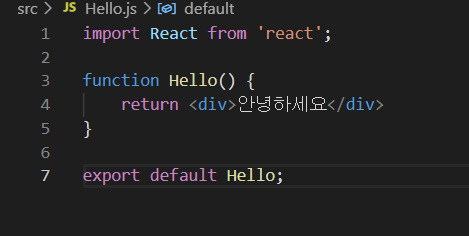
$cd begin-react

$yarn start

를 통해 자동으로 열리는 페이지에 코드 결과물을 볼 수 있다.

리액트 컴포넌트

src 디렉터리에 Hello.js라는 파일을 작성한다.



리액트 컴포넌트를 만들 때는

import React from ‘react’;

를 통하여 리액트를 불러와야 한다.

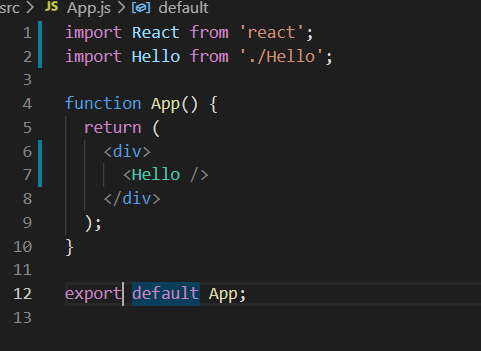
리액트 컴포넌트는 함수형태로 작성할 수도 있고, 클래스 형태로도 작성할 수 있다.

지금은 **함수로 작성하는 방법**에 대해서만 얘기한다.

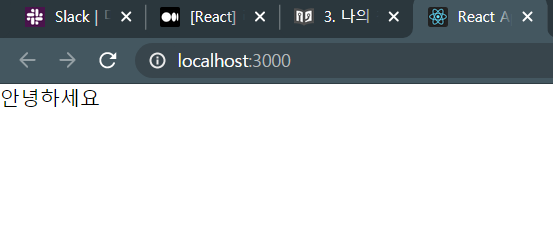
함수형 컴포넌트와 클래스형 컴포넌트의 주요 차이점은, state와 LifeCycle이 빠져있다는 점. 그래서 컴포넌트 초기 마운트가 아주 미세하게 빠르고, 메모리 자원을 덜 사용한다.

export default Hello;

이 코드를 통해 Hello 컴포넌트를 내보낼 수 있다. 따라서 다른 컴포넌트에서 import 하여 사용할 수 있는데, 한 번 App.js에서 불러와서 사용해보자



다음과 같이 작성하면 아래와 같은 결과를 얻을 수 있다.



JSX란

JSX는 리액트에서 생김새를 정의할 때 사용하는 XML과 비슷한 자바스크립트 확장 문법이다.

XML 형태로 코드를 작성한 JSX를 JAVASCRIPT로 변환해주고 있다.

**지켜야 할 문법**

1. 태그 꼭 닫기
2. 태그와 태그 사이에 내용이 들어가지 않을 때는, self closing 태그 사용하기

ex) <Hello />

1. 두 개 이상의 태그는 무조건 하나의 태그로 감싸기

리액트의 fragment <>를 사용하면 불필요한 상황을 피할 수 있음

1. JSX의 변수는 {}으로 감싸기
2. 주석은 {/\* \*/}

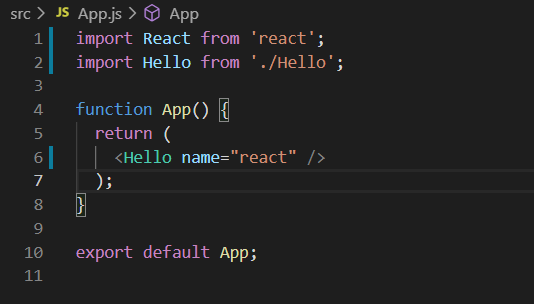
Props

Props는 properties의 줄임말로, 우리가 어떠한 값을 컴포넌트에게 전달해줘야 할 때 사용된다. props는 부모 컴포넌트가 자식 컴포넌트에게 주는 값으로 자식 컴포넌트에서는 props를 받아오기만 하고, 받아온 props를 직접 수정 할 수는 없다.

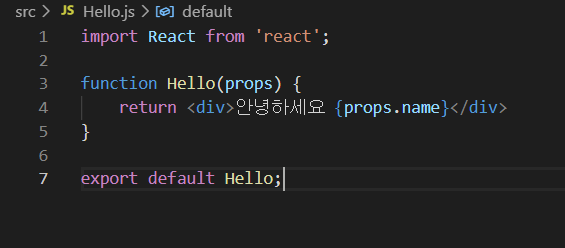
반면 나중에 배울 state는 컴포넌트 내부에서 선언하며 내부에서 값을 변경 할 수 있다.

예를 들어, App 컴포넌트에서 Hello 컴포넌트를 사용할 때 name이라는 값을 전달해주고자 하면

App.js 에서는 다음과 같이



이를 Hello 컴포넌트에서 사용하고자 할 때는 다음과 같이 작성한다.



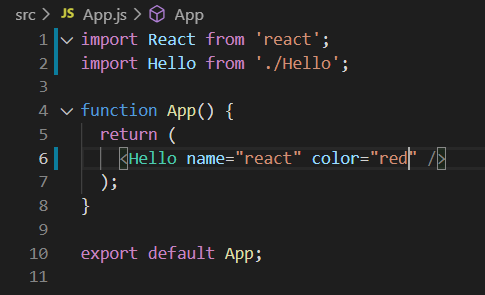
컴포넌트에게 전달되는 props는 파라미터를 통하여 조회할 수 있다. props는 객체 형태로 전달되며, name 값을 조회할 시에는 props.name 으로 조회할 수 있다.

props를 사용하기 위해서 Hello function안에 props 파라미터를 설정해주자 !!

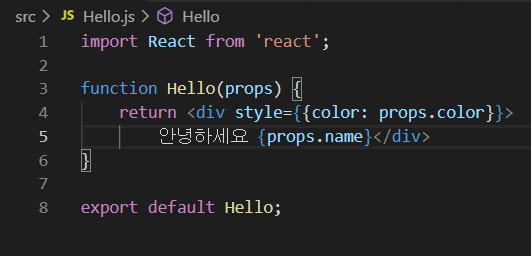
**여러 개의 props**

이번에는 color라는 props를 전달해보자

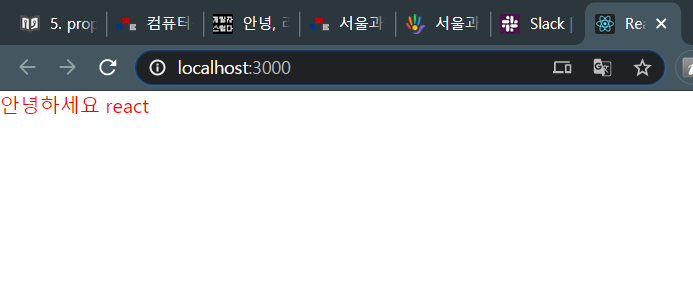
App.js는 다음과 같이 작성하고



Hello.js에서는 다음과 같이 작성하자



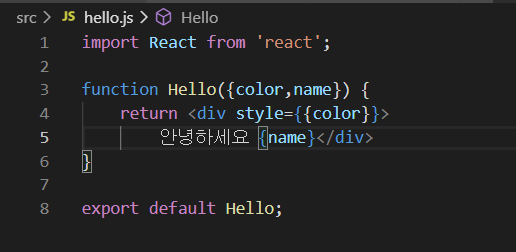
이와 같이 작성하면 다음과 같은 결과를 얻을 수 있다.



**비구조화 할당**

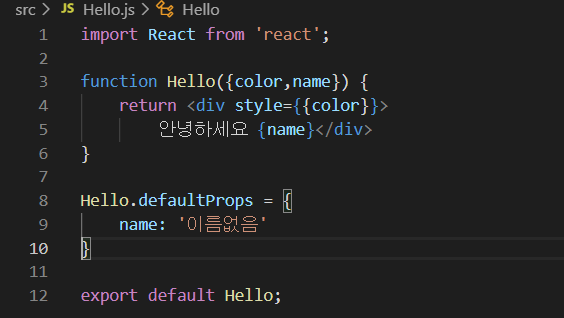
**클**

비구조화 할당 문법을 사용하면 다음과 같이 조금 더 코드를 간결하게 작성 할 수 있다.



**defaultProps로 기본값 설정**

props를 지정하지 않았을 때 기본적으로 사용할 값을 설정하고 싶으면 컴포넌트에 defaultProps 값을 설정하면 된다.



이와 같이 defaultProps를 설정하면 name이나 color의 props의 값을 따로 설정해두지 않아도 자동으로 default 값이 설정된다.

**props.children**

컴포넌트 태그 사이에 넣은 값을 조회하고 싶을 땐, props.children을 사용한다.

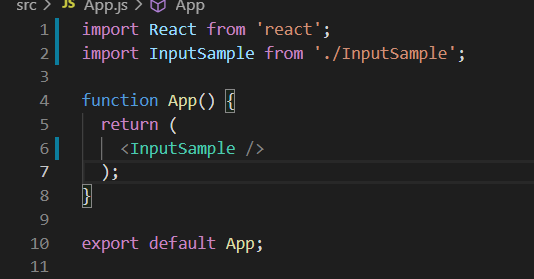
Input 상태 관리하기

**InputSample.js**

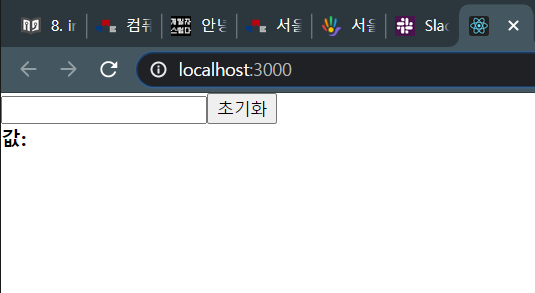
라는 파일을 생성한다.



이를 App에서 렌더링(data를 html로 변환해서 renderer에게 전달하는 일관의 작업)한다. import 변환 해주기



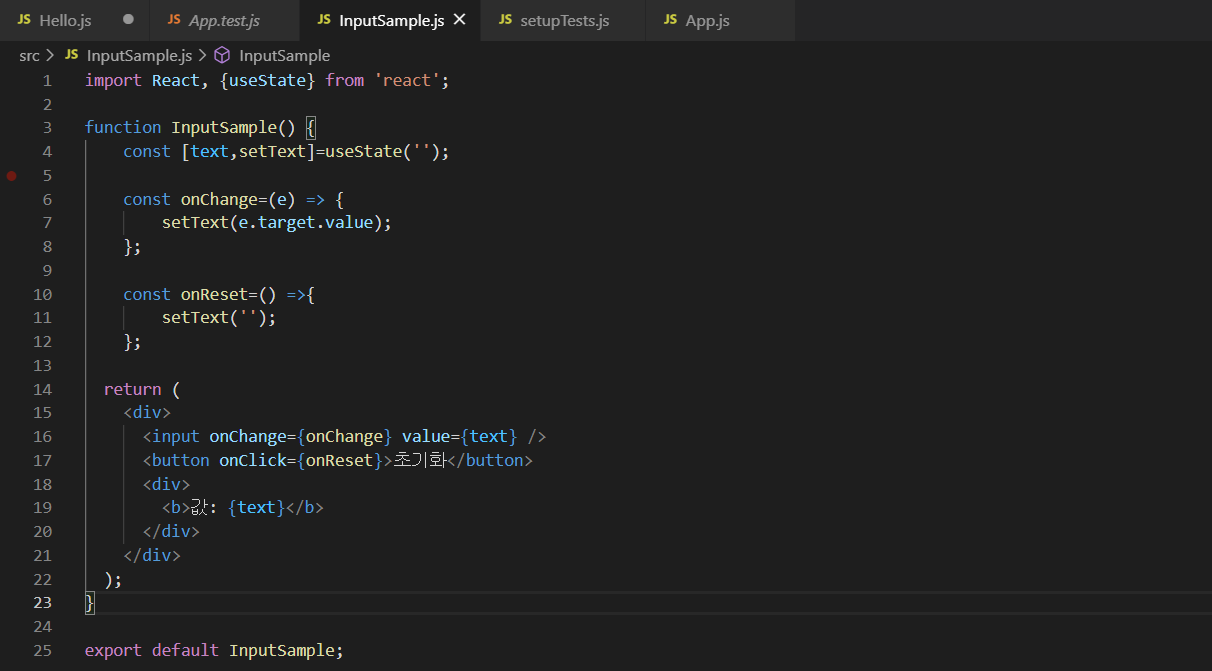
위와 같이 코드를 작성하게 되면 아래와 같은 결과를 얻을 수 있다.



input에 입력하는 값이 하단에 나타나고, 초기화 버튼을 누르면 input의 값이 비워지도록 구현해보자.

이벤트 함수는 이벤트 객체 e를 파라미터로 받아와서 사용할 수 있는데, 이 객체의 e.target는 이벤트가 발생한 DOM인 input DOM을 가르키게 된다. 이 DOM의 value 값 즉 e.target.value 를 조회하면 현재 input에 입력한 값이 무엇인지 알 수 있다. 또한 이 값을 useState를 통해서 관리한다.

다음과 같이 바꿔보자



input의 상태를 관리할 때에는 input 태그의 value 값도 설정해주는 것이 중요하다. 그렇게 해야 상태가 바뀌었을 때 input의 내용도 업데이트 된다.