**4. props을 통해 컴포넌트에 값 전달해보기**

* **props?**

properties의 줄임말, 어떠한 컴포넌트에게 값을 전달할 때 사용

* **기본사용법**

<App 컴포넌트에서 Hello 컴포넌트를 사용할 때 name이라는 값을 전달해주고 싶을 때>

* **App.js**

import React from 'react';

import Hello from './Hello';

function App() {

return (

<Hello name="react" />

);

}

export default App;

* 이때 Hello 컴포넌트에서 name값을 사용하고 싶을 때?
* **Hello.js**

import React from 'react';

function Hello(props) {

return <div>안녕하세요 {props.name}</div>

}

export default Hello;

* 컴포넌트에 전달되는 props는 객체 형태
* 만약 name 값을 조회하고 싶다면 props.name을 조회
* **여러 개의 props, 비구조화 할당**

<또 다른 props 전달, color 라는 값을 설정하기>

* **App.js**

import React from 'react';

import Hello from './Hello';

function App() {

return (

<Hello name="react" color="red"/>

);

}

export default App;

* **Hello.js**

import React from 'react';

function Hello({ color, name }) {

return <div style={{ color }}>안녕하세요 {name}</div>

}

export default Hello;

* **defaultProps로 기본값 설정**
* **Hello.js**

import React from 'react';

function Hello({ color, name }) {

return <div style={{ color }}>안녕하세요 {name}</div>

}

Hello.defaultProps = {

name: '이름없음'

}

export default Hello;

* 이제 App 에서 name이 없는 Hello 컴포넌트를 렌더링
* **App.js**

import React from 'react';

import Hello from './Hello';

function App() {

return (

<>

<Hello name="react" color="red"/>

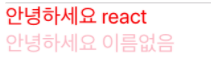
<Hello color="pink"/>

</>

);

}

export default App;

<<실행결과

* **props.children**

컴포넌트 태그 사이에 넣은 값을 조회하고 싶을대 props.children을 조회

<이번엔, props.children을 사용하는 새로운 컴포넌트 만들어 보기>

* Wrapper.js

import React from 'react';

function Wrapper({ children }) {

const style = {

border: '2px solid black',

padding: '16px',

};

return (

<div style={style}>

{children}

</div>

)

}

export default Wrapper;

* 여기서 {children}을 안해주면 검은색 네모박스안에 글자가 나타나지 않음.
* App.js

import React from 'react';

import Hello from './Hello';

import Wrapper from './Wrapper';

function App() {

return (

<Wrapper>

<Hello name="react" color="red"/>

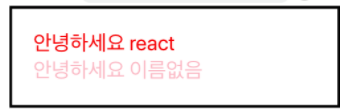
<Hello color="pink"/>

</Wrapper>

);

}

export default App;

<<실행화면