**배운 내용 요약**

{// if대신 삼항연산자

1 < 3

? console.log('맞아요')

: console.log('틀려요')

}

            {/\* h3을 클릭하면 Modal이 등장하게 \*/}

            <div className = 'list' >

            {/\* modal이 false면 true로 바꾸고, true면은 false로 바꿔줘 \*/}

                <h3 onClick={()=>{modal===false ? modal변경(true) : modal변경(false)} }> {글제목[2]}</h3>

                <p>2 월 17 일 발행</p>

                <hr/>

            </div>

            {// if대신 삼항연산자

                modal === true

                ? <Modal/>

                : null

            }

<button onClick={ ()=>{modal변경(!modal)} }>버튼</button>

            {/\* 버튼 눌렀을 때 열리고 한번 더 누르면 닫히는 모달창  \*/}

            {

                modal === true

                ? <Modal/>

                : null

            }

function App() {

    let [글제목, 글제목변경] = useState(['남자 코트 추천', ' 강남 우동 맛집', '파이썬 독학']);

    let [좋아요, 좋아요변경] = useState(0);

    let [modal,modal변경] = useState(false); // 이거는 모달창을 켜고 닫는 스위치이다.

    function 제목변경(){

        var newArray = [...글제목];

        newArray[0] = '여자 코트 추천';

        글제목변경(newArray);

    }

    return (

        <div className="App">

            <div className="black-nav">

                <div>개발 Blog

                </div>

            </div>

            <button onClick={ 제목변경 }>버튼</button>

            <div className='list'>

                <h3> {글제목[0]} <span onClick={()=>{좋아요변경(좋아요++)}}> 👍 </span> {좋아요} </h3>

                    <p>2월 17일 발행</p>

                    <hr/>

            </div>

            <div className = 'list'>

                <h3> {글제목[1]} <span onClick={ ()=>{console.log(1)}}> 👍 </span> 0  </h3>

                    <p>2월 17일 발행 </p>

                    <hr/>

            </div>

            {/\* h3을 클릭하면 Modal이 등장하게 \*/}

            <div className = 'list' >

                <h3 onClick={ () => {modal변경(true)}}> {글제목[2]}</h3>

                <p>2 월 17 일 발행</p>

                <hr/>

            </div>

            {// if대신 삼항연산자

                modal === true

                ? <Modal/>

                : null

            }

        </div>

        /\*\*리액트에서는 ui를 만들 때 state데이터를 이용함

         \* 모달창을 켜고 닫는 스위치같이 실행할 수 있게

         \*/

    );

}

function Modal(){

    return(

        <div className="modal">

            <h2>제목</h2>

            <p>날씨</p>

            <p>상세내용 </p>

        </div>

    )

}

클릭하면 등장하는 간단한 UI는 어떻게 만드는지 알아봅시다.

쌩 자바스크립트에선 display : block 이런 스타일을 조정해서 보여주고 안보여주고 했었는데

리액트에선 모달창이 현재 보이는지 안보이는지 state를 저장해둔 뒤에

if문을 사용해서 state에 따라 모달창을 보여줍니다.

글만 읽으면 뭔소리인지 이해가 안되니 밑에서 자세히 알아봅시다.

**리액트에서 HTML UI를 조건에 따라 보여주고 싶으면**

특정상황에 맞춰서 HTML을 보여주고 안보여주고 이런건**if문**을 이용하시면 됩니다.

그럼 만약에 특정 조건이 참일 때만 <Modal />을 보여주고 싶으면 어떻게 할까요?

function App (){

return (

<div>

HTML 잔뜩있는 곳

{

1 < 3

? <Modal />

: null

}

</div>

)

}

▲ 실은 리액트 중괄호 내에서 if문을 사용할 수 없어서 if문 대용 역할을 할 수 있는 삼항연산자를 사용합니다.

삼항연산자는 if문이랑 똑같은 용도로 사용하실 수 있는 쌩자바스크립트 문법인데

조건식 ? 조건식 참일 때 실행할 코드 : 조건식 거짓일 때 실행할 코드

이렇게 if문 대신 쓸 수 있는 문법입니다.

그래서 이 문법을 이용해서 1 < 3일 때만 <Modal>을 보여주세요~라고 작성해본 것입니다.

(참고) 리액트에서 텅빈 HTML을 의미하는 코드는 null이라는 자료형입니다.

**(다시 돌아가서) 리액트에서 클릭시 보이는 UI 만드는 법**

**1. 일단 UI가 보이는/보이지않는 상태정보를 state로 만들어둠** (보통 true/false 자료형으로)

**2. state가 true일 때만 UI를 보여준다고 if문을 사용함**

**3. 심심하면 <열기버튼>을 누르면 state가 true로 바뀌도록 버튼에 기능개발함**

그럼 이제 열기버튼을 클릭시 UI가 보입니다.

이게 끝입니다. 모달창 뿐만 아니라 탭, 햄버거메뉴 같은 UI들은 다 이런식으로 만듭니다.

같이 모달창을 한번 만들어봅시다.

**1. 모달창이 보이는/보이지않는 상태정보를 저장할 state를 하나 만들어둡니다.**

function App (){

let [modal, modal변경] = useState(false);

return (

<div>

HTML 잔뜩있는 곳

{

1 < 3

? <Modal />

: null

}

</div>

)

}

modal 이라는 state를 만들었고 기본값은 false로 주었습니다.

일종의 온오프 스위치라고 생각하셔도 좋습니다.

**2. if문을 이용해서 스위치가 켜져있을 때만 모달창이 보인다고 명시해줍니다.**

function App (){

let [modal, modal변경] = useState(false);

return (

<div>

HTML 잔뜩있는 곳

{

modal === true

? <Modal />

: null

}

</div>

)

}

modal 이라는 state가 true일 때만 <Modal />을 보여준다고 작성했습니다.

(그러면 그 state의 기본값은 false기 때문에 사이트 첫 로드시 <Modal />은 보이지 않겠네요? ㅇㅇ)

**3. 열기버튼을 눌렀을 때 모달창이 보이게 만들어줍니다.**

function App (){

let [modal, modal변경] = useState(false);

return (

<div>

HTML 잔뜩있는 곳

<button onClick={ ()=>{ modal변경(true) } }> 열기 </button>

{

modal === true

? <Modal />

: null

}

</div>

)

}

그럼 이제 열기버튼을 눌렀을 때 modal이라는 state가 true로 변경되겠죠?

그럼 if문에 의해 자동으로 <Modal />이라는 요소가 보입니다.

**Q. 오잉 왜 state 변경할 때 복사본을 만들어서 수정하라매요?**

A. 사본만드는건 reference 자료형들만 하시면 됩니다. array, object 이런거요.

문자, 숫자, true/false 이런건 필요없이 직접수정하셈

그럼 모달창 띄우기 성공!

이제 모달창 닫기 버튼도 알아서 만들어보실 수 있겠죠?