**배운 내용 요약**

첫 글제목이 '여자코트 추천' 으로 바뀌는 <button>

var newArray = [...글제목];

newArray[0] = '여자 코트 추천';

글제목변경(newArray);

 spread 연산자라고하는 ES6 신문법

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

    let [좋아요, 좋아요변경] = useState(0);

    // state의 복사본을 만들어서 수정해야 함

    // 근데 그냥 복사하면 참조형이 가리키는 어쩌고 알지??

    // 그래서 [...이렇게 복사해야함] spread operator, deep copy

    function 제목변경(){

        // var newArray = 글제목에 있던 0번째 데이터를 여저코트추천으로 바꿈

        var newArray = [...글제목];

        newArray[0] = '여자 코트 추천';

            // state는 직접 건들지 말고, copy를 해서 건들어야함

        글제목변경(newArray);

        // 글제목변경(['여자 코트 추천', ' 강남 우동 맛집', '파이썬 독학']);

    }

    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>

            <div className = 'list' >

                <h3> {글제목[2]}</h3>

                <p>2 월 17 일 발행</p>

                <hr/>

            </div>

        </div>

    );

저번 시간 내드렸던 숙제가 잘 안되어서 그냥 다음 강의를 누르셨군요. 처음이니 봐드림

클릭하면 첫 글제목이 '여자코트 추천' 으로 바뀌는 <button>을 만들어야합니다.

state를 수정하라는건데 수정하려면 특별한 함수를 써야한댔죠?

근데 그 함수는 완전히 데이터를 바꿔치기, 대체해주는 함수라

state 데이터를 일부만 변경하고 싶을 땐 어떤 식으로 사용하는지 알아봅시다.

**일단 버튼을 만들고 버튼에 기능을 추가하려면**

function App(){

let [글제목, 글제목변경] = useState( ['남자코트 추천', '강남 우동맛집', '파이썬 독학'] );

return (

<HTML 많은 곳>

<button onClick={ 어쩌구~ }> 수정버튼 </button>

<h3> { 글제목[0] } <span>?</span> { 따봉 }</h3>

</HTML 많은 곳>

)

}

기존에 있던 코드에 <button>을 아무데나 하나 추가했고,

onClick 속성을 열어서 뭔가 적어주면 기능이 완성되겠죠?

근데 onClick안에 뭘 적을지가 문제입니다.

일단 onClick 안에는 함수만 들어갈 수 있댔죠? 함수를 만들어 집어넣어봅시다.

function App(){

let [글제목, 글제목변경] = useState( ['남자코트 추천', '강남 우동맛집', '파이썬 독학'] );

function 제목바꾸기() {

어쩌구~

}

return (

<HTML 많은 곳>

<button onClick={ 제목바꾸기 }> 수정버튼 </button>

<h3> { 글제목[0] } <span>?</span> { 따봉 }</h3>

</HTML 많은 곳>

)

}

함수는 저렇게 function App() 아무데나 만드실 수 있으며

만든 함수를 onClick안에 집어넣었습니다.

미리 정의된 함수를 넣으실 때는 소괄호를 넣지 않는게 포인트입니다.

그럼 즐겁게 **제목바꾸기** 함수 기능개발을 해보도록 합시다.

**제목바꾸기 함수는 state를 변경해야하는데..**

function App(){

let [글제목, 글제목변경] = useState( ['남자코트 추천', '강남 우동맛집', '파이썬 독학'] );

function 제목바꾸기() {

글제목[0] = '여자코트 추천'

}

return (

<HTML 많은 곳>

<button onClick={ 제목바꾸기 }> 수정버튼 </button>

<h3> { 글제목[0] } <span>?</span> { 따봉 }</h3>

</HTML 많은 곳>

)

}

제목바꾸기() {} 함수 안을 잘 보십시오.

글제목이라는 state의 첫째 데이터를 변경하고싶으니까 저렇게 쓰면 될까요?

안됩니다. state는 = 등호기호로 직접 조작할 수 없다니까염.

만들어놓은 state 변경함수를 꼭 사용하셔야합니다.

**글제목** state만들 때 같이 만든 **글제목변경()** 이라는 함수를 사용하시면 됩니다.

function 제목바꾸기() {

글제목변경( ['여자코트 추천', 강남 우동맛집', '파이썬 독학'] )

}

그래서 state 변경함수인 **글제목변경()**을 이용해서

**제목바꾸기()** 함수를 이렇게 디자인했습니다.

**Q. 글제목변경() 함수에 array를 왜 저렇게 넣죠?**

A. state변경함수들은 state를 아예 대체할 수만 있다고 저번에 배웠습니다.

그래서 기존 state인 ['남자코트 추천', 강남 우동맛집', '파이썬 독학'] 을 바꿔치기할 새로운 array를 넣어주어야합니다.

이렇게 똑같이 따라해주시면 여러분도 버튼을 눌렀을 때 정상적으로 글제목 state가 변경됩니다.

성공!

너무 주먹구구식이라고요?

**약간 프로그래머 스타일로 다시만들어보면**

array 내용이 3개밖에 없어서 위처럼 하드코딩해도 상관없겠지만

array가 조금이라도 길어지면 하드코딩하기 어렵겠죠?

그러니 여러분이 좋아하는 프로그래밍 스킬을 잔뜩 넣어보겠습니다.

▼ 제목바꾸기 함수를 이렇게 프로처럼 바뀌봅시다.

function 제목바꾸기() {

var newArray = 글제목;

newArray[0] = '여자코트 추천';

글제목변경( newArray );

}

지금 뭘한거냐면..

0. 글제목이라는 state는 직접 수정할 수 없습니다. 그래서.. 복사본을 만들기로 합니다.

1. 글제목이라는 state의 복사본을 만들어**newArray**라는 변수에 저장합니다.

2. newArray의 0번째 데이터를 '여자코트 추천'으로 변경합니다.

3. 그리고 그걸 글제목변경() 함수 안에 넣어서 글제목 state를 변경합니다.

이론적으로 완벽하죠?

근데 동작하지 않습니다.

왜냐면 코드는 잘 짠건 맞는데, state를 복사하실 때 문제가 하나 있었기 때문입니다.

원래 자바스크립트 내에서 array나 object 자료형은 = 등호로 복사하시면 각각 별개의 자료형이 생성되는게 아니라

값을 공유합니다. (충격)

예를 들면

var data1 = [1,2,3];

var data2 = data1;

이런 식으로 사용하면 data1에 있던 자료를 data2에 복사한다는 뜻이죠?

근데 data1과 data2는 각각 [1,2,3]을 별개로 저장하는게 아니라

**data1과 data2는 똑같은 값을 공유합니다.**

data1을 변경하면 data2도 자동으로 변경되고 그렇습니다. (충격)

(자세한건 javascript reference data type이라고 검색해봅시다)

그래서 결론은 state도 등호 = 를 이용해서 복사하면 문제가 일어나기 때문에

완전히 개별 복사본을 만들어주는 카피를 하셔야합니다.

shallow/deep copy라고 부르는데 가장 쉬운 방법은

var 새로운어레이 = [...원본어레이]

이렇게 사용하는 것입니다.

그래서 이렇게 하시면 안되고

function 제목바꾸기() {

var newArray = 글제목;

newArray[0] = '여자코트 추천';

글제목변경( newArray );

}

이렇게 하시면 성공입니다.

function 제목바꾸기() {

var newArray = [...글제목];

newArray[0] = '여자코트 추천';

글제목변경( newArray );

}

성공적으로 글제목 state의 사본을 생성했으며,

그 사본을 입맛에 맞게 조작한 다음에

state 변경함수에 넣어주시면 state가 성공적으로 수정됩니다.

**저기요 쩜쩜쩜... 기호는 뭔데요**

 spread 연산자라고하는 ES6 신문법입니다.

array나 object 자료형 왼쪽에 붙일 수 있으며

뜻은 별거없고 **중괄호나 대괄호를 벗겨주세요~** 라는 뜻입니다.

**...[1,2,3]** 이렇게 쓰시면

그 자리에 **1,2,3** 이 남습니다. 걍 괄호 벗기기용 연산자입니다.

근데 두번째 용도도 있는데 array나 object 자료형을 shallow/deep copy할 때 많이 사용합니다.

var data1 = [1,2,3];

var data2 = [...data1]

그냥 data1에 있던 자료들을 괄호 벗긴담에 다시 array로 만들어주세요~ 라는 뜻인데

array를 이런 식으로 사용하면

완전**독립적인 array 복사본을 생성해주는 copy**가 가능합니다.

object 자료형도 똑같이 가능합니다.

아무튼 오늘의 결론입니다.

리액트에서 state를 수정하고 싶으면 보통 이런 패턴으로 코드를 짭니다.

**1. 수정하고 싶은 state의 deep/shallow 카피본을 하나 생성합니다.**

**2. 카피본을 입맛에 맞게 수정합니다.**

**3. 카피본을 state변경함수()에 집어넣습니다.**

이 패턴을 외워두시면 state 변경하는 법 마스터하신겁니다.

**응용문제 : 정렬 기능을 만들고 싶으면?**

Q. 이번엔 버튼을 누르면 글들이 가나다 순으로 정렬되어야합니다.

어떻게 버튼에 기능개발해야할까요?

이번 문제는 답은 따로 없습니다. 여러분이 잘 알아서해보길 바랍니다.

힌트를 드리자면

지금 3개의 글들이 state에 저장되어있고 이걸 HTML에 데이터바인딩했습니다.

그럼 여러분이 신경쓸 일은 **HTML을 어떻게 정렬할까**가 아니라

**state를 어떻게 정렬할까** 입니다.

리액트는 **state를 변경하면 HTML은 알아서 재렌더링되니까요.**

기존 전통 개발방식처럼 "잉 div 위치를 어떻게 바꿔줘야하나?" 는 전혀 신경쓸 일이 없습니다.

state 순서만 정렬하면 div 순서는 알아서 정렬 + 재렌더링 됩니다.

그러니 버튼을 눌렀을 때 글목록 array가 담긴 state를 한번 정렬해보시길 바랍니다.

당연히 array 내의 아이템을 가나다 순으로 정렬하는 법을 모를 땐 구글 검색 ㄱㄱ