배운 내용 요약

state를 이용해 데이터를 저장

**"변수가 변경될 때 자동으로 관련된 HTML을 재렌더링되게 만들고 싶으면"**

**useState( 저장할 데이터 )** 이렇게 사용하시면 state에 데이터를 저장

let [a,b] = useState([‘data’, ‘data’]);

a[0] a[1] 로 사용

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

리액트에서도 그냥 변수 자유롭게 만들고 변수에 데이터 아무렇게나 저장해서 쓰실 수 있는데

데이터바인딩과 관련된 중요한 데이터를 저장할 땐 변수 대신 state를 만들어씁니다.

state는 왜 쓰는지, state는 어떻게 만들어서 데이터를 저장하는지 알아봅시다.

**일단 블로그 글 리스트 UI를 하나 만들고 시작합니다.**

function App(){

let posts = '강남 고기 맛집';

return (

<div className="App">

<div className="black-nav">

<div>개발 blog</div>

</div>

<div className="list">

<h3>{ posts }</h3>

<p>2월 17일 발행</p>

<hr/>

</div>

</div>

)

}

(App.css)

.list {

margin-top : 30px;

text-align : left;

padding-left : 20px;

padding-right : 20px;

}

그냥 똑같이 치셔도 될 듯 합니다. 그럼 메인페이지에 멋진 글목록이 하나 생성됩니다.

이제 좀 코드짤 맛이 납니다.

**state를 만들어서 데이터를 저장해봅시다**

리액트에선 변수 말고 state를 만들어서 데이터를 저장해서 쓸 수도 있습니다.

이전 강의에서는 그냥 **let posts = '강남 어쩌구'** 이렇게 일반 변수에 데이터를 저장했었는데

이번엔 state를 이용해 데이터를 저장해보도록 합시다.

변수쓰면 되는거지 왜 굳이 그러냐고요? 이유는 좀 있다가 알게 됩니다.

여러분의 App.js 파일에 다음과 같은 코드를 추가합니다.

import React, { useState } from 'react';

import './App.css'

function App(){

let [a,b] = useState('남자 코트 추천');

let posts = '강남 고기 맛집';

return (

<div className="App">

<div className="black-nav">

<div>개발 blog</div>

</div>

<div className="list">

<h3>{ posts }</h3>

<p>2월 17일 발행</p>

<hr/>

</div>

</div>

)

}

두곳에 코드가 추가되었습니다.

맨 윗줄 import React 어쩌구에 {useState}라는게 추가가 되었고,

function App() 안에 useState() 어쩌구가 추가되었습니다.

**useState()**라는 함수는 state를 하나 만들어주는 함수입니다.

(이걸 이용하기 위해 위에 import {useState} 어쩌구를 추가했습니다.)

**useState( 저장할 데이터 )** 이렇게 사용하시면 state에 데이터를 저장하실 수 있습니다.

근데 useState()를 쓰면 데이터가 두개 남습니다. [데이터1, 데이터2] 이렇게 생긴 이상한 array가 남습니다.

이걸 각각 a와 b라는 변수명으로 ES6 destructuring 문법을 이용해 저장해서 쓰시면 됩니다.

a라는 변수엔 실제 저장할 데이터가 들어있고

b라는 변수엔 저장할 데이터를 변경시킬 함수가 들어있습니다.

**Q. destructuring 문법이 뭔데요?**

이론 설명보다 예시 하나를 보는게 이해가 더 쉽습니다.

let [a,b] = useState('남자 코트 추천');

▲ 위처럼 사용하면 a와 b라는 변수가 생성되는데,

a라는 변수는 **'남자 코트 추천'** 이라는 중요 데이터가 들어있는 **state**이며

b라는 변수는 '남자 코트 추천'이라는 **데이터 변경을 도와주는 함수**가 들어있습니다. (이건 아직 신경쓰지맙시다)

그래서 변수명을 약간 더 예쁘게 바꿔보면

let [글제목, 글제목변경] = useState('남자 코트 추천');

이렇게 하면 조금 더 직관적으로 이해할 수 있겠군요.

그럼 **글제목**이라는 변수에 **'남자 코트 추천'** 이라는 데이터가 들어갑니다.

실제 코드도 이렇게 바꿔줍시다.

**state 데이터도 똑같이 변수처럼 데이터바인딩 가능합니다**

그래서 아까 '남자 코트 추천'을 담은 글제목 state를 만들었는데

state도 변수와 똑같이 활용하실 수 있습니다.

{}를 이용하시면 원하는 곳에 데이터바인딩이 가능합니다.

import React, { useState } from 'react';

import './App.css'

function App(){

let [글제목, 글제목변경] = useState('남자 코트 추천');

let posts = '강남 고기 맛집';

return (

<div className="App">

<div className="black-nav">

<div>개발 blog</div>

</div>

<div className="list">

<h3>{ 글제목 }</h3>

<p>2월 17일 발행</p>

<hr/>

</div>

</div>

)

}

위 예제에서 **<h3> { 글제목 } </h3>** 이 부분은 미리보기화면에서 **<h3> 남자 코트 추천 </h3>** 으로 렌더링됩니다.

state도 일반 변수처럼 쉽게 데이터바인딩이 가능합니다.

**state에는 Array, Object 아무거나 다 넣을 수 있습니다**

많은 자료를 한번에 저장하고 싶으면 Array, Object 자료형으로 넣으셔도 됩니다.

import React, { useState } from 'react';

import './App.css'

function App(){

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

return (

<div className="App">

<div className="black-nav">

<div>개발 blog</div>

</div>

<div className="list">

<h3>{ 글제목[0] }</h3>

<p>2월 17일 발행</p>

<hr/>

</div>

<div className="list">

<h3>{ 글제목[1] }</h3>

<p>2월 17일 발행</p>

<hr/>

</div>

</div>

)

}

위 예제는 state를 만들 때 array 자료형을 집어넣었고,

{} 중괄호를 이용해 원하는 곳에 state 데이터들을 바인딩했습니다.

state도 만드는 과정이 귀찮아서 그렇지 용도는 그냥 변수랑 똑같습니다.

그럼 변수 만들어 쓰면 되는거지 왜 굳이 state 만들어쓰냐고요?

**변수 말고 state에 데이터 저장해서 쓰는 이유**

힘들게 state를 만들어 써야하는 이유는 이겁니다.

**"변수가 변경될 때 자동으로 관련된 HTML을 재렌더링되게 만들고 싶으면"**

그럴 때 변수 말고 state에 저장해서 데이터바인딩 하셔야합니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

여러분 지금 state로 글제목을 하나 만드셨죠?

근데 여러분이 버튼을 눌러서 **글제목을 수정하거나** 그런 짓거리를 할 때가 있습니다. (지금은 없지만)

그럴 때 수정사항이 자동으로 웹페이지에 스무스하게 반영되게 만들고 싶으면

글제목을 state로 만들어놓고 사용하셔야합니다.

**리액트는 state가 수정이 일어나면**

**state가 포함된 HTML을 자동으로 재렌더링 해줍니다.**

사용자가 버튼을 누르든 해서 글제목이라는 state가 '여자 코트 추천' 이런 식으로 변경되면

**<h3> 남자 코트 추천</h3>** 이 부분을

**<h3> 여자 코트 추천</h3>** 으로 새로고침 없이 샤샥 재렌더링해준다는 소리입니다.

그럼 OT강의에서 설명드렸던 스무스한 앱들과 유사하게 동작하겠죠?

웹앱을 만들고 싶으면 좋은말 할 때 state에 중요데이터들을 저장해 사용하시길 바랍니다.

(일반 변수는 변경이 발생해도 자동으로 재렌더링 해주지 않습니다.)

다음 시간에 state 데이터를 변경하는 법을 한번 배워서 진짜 앱처럼 스무스하게 바뀌는지 실험해봅시다.

**Q. 그럼 블로그 로고명 같은 그런 데이터도 state로 만들어서 데이터바인딩할까요?**

그러셔도 됩니다. 하지만 블로그 로고명은 거의 바뀌지 않죠?

바뀌지 않는 데이터들은 state로 굳이 저장할 필요 없습니다.

state의 가장 큰 장점은 state가 변경될 때마다 자동으로 state와 관련된 HTML이 재렌더링이 된다는 것인데

로고명은 아예 바뀔 일이 없으니 의미가 없겠죠.

state는 상품명, 글제목, 가격 이런것 처럼 자주 변하는 데이터들을 저장하시는게 좋은 관습입니다.

import React, {useState} from 'react';

  // 리액트에 있는 {내장함수} 하나만 쓰겠습니당

let [ 글제목,글제목변경] = useState('남자 코트 추천');

    // 데이터 저장하는 공간 만들었움

    // 실행하면 [a,b] 가 남음

    // a에는 '남자 코트 추천' 이 들어가고

    // b에는 '남자 코트 추천' state 정정해주는 함수가 들어감

    // [state데이터, state데이터 변경 함수]

  // 신문법 하나만 짚고 넘어감

  // destructuring : array, object 에 있던 자료를 변수에 쉽게 담고 싶을 때

  // var [a,b] = [10,100];

  let [ 글제목1,글제목변경1] = useState(['남자 코트 추천', ' 강남 우동 맛집']);

    // 얘는 새로고침안해도 됨 이걸 자동으로 재랜더링된다고 함

  let posts = '강남 고기 맛집'; // 얘는 새로고침해야하는데

  return (

    <div className="App">

      <div className="black-nav">

        <div>개발 Blog</div>

      </div>

      <div className='list'>

        <h3>{ 글제목 }</h3>

        <p>2월 17일 발행</p>

        <hr/>

      </div>

      <div className='list'>

        <h3>{ 글제목1[1] }</h3>

        <p>2월 17일 발행</p>

        <hr/>

      </div>

    </div>

  );

.black-nav {

    background: black;

    width: 100%;

    display: flex;

    color: white;

    padding: 20px;

    font-weight: 600;

    font-size: 20px;

    box-sizing: border-box;

}

.list {

    margin-top: 30px;

    text-align: left;

    padding-left: 20px;

    padding-right: 20px;

}