**고정적 배열 렌더링**

//UserList.js

import React from ‘react’;

function User({user}) {

return (

<div>

<b>{user.username}</b> <span>({user.email})</span>

</div>

);

}

function UserList() {

const users = [

{

id: 1,

username: 'velopert',

email: 'public.velopert@gmail.com'

},

{

id: 2,

username: 'tester',

email: 'tester@example.com'

},

{

id: 3,

username: 'liz',

email: 'liz@example.com'

}

];

return (

<div>

<User user={users[0]} />

<User user={users[1]} />

<User user={users[2]} />

</div>

);

}

export default UserList;

**설명**

**-**자바스크립트 배열 선언

const users = [

{

id: 1,

username: 'velopert',

email: 'public.velopert@gmail.com'

},

{

id: 2,

username: 'tester',

email: 'tester@example.com'

},

{

id: 3,

username: 'liz',

email: 'liz@example.com'

}

];

-같은 파일 내 함수를 호출하여 배열을 간단히 사용

function User({user}) {

return (

<div>

<b>{user.username}</b> <span>({user.email})</span>

</div>

);

}

…

return (

<div>

<User user={users[0]} />

<User user={users[1]} />

</div>

);

**동적인 배열 렌더링**

-위처럼 할 시 크기가 고정적인 배열에서만 사용가능하다.

-크기가 증가할 시, 코드를 다시 수정해야함.

-다음처럼 map을 사용함

return (

<div>

{users.map(user=> (

<User user={user}/>

))}

</div>

);

-위처럼 할 시 에러발생. 그 이유는 리액트에서 배열을 렌더링할 때는 key라는 props를 설정해야하므로.

-key값은 각 원소들마다 가지고 있는 고유값으로 설정을 해야함

-지금의 경우는 id가 고유값

{users.map(user=> (

<User user={user} key={users.id} />

))}

-하지만, 배열안에 원소가 가지는 고유값이 없다면, map() 함수를 사용할 때,

설정하는 콜백함수의 두번재 파라미터 index를 key로 사용하면 됨.

{user.map((user, index) => (

<User user={user} key={index} />

))}

-배열 안에 map()함수가 있다는 식으로 이해하면 될 듯.

-map() 함수는 배열의 각 원소에 대하여 어떠한 작업을 하고 새로운 배열로 만드는 함수

-> 따라서 컴포넌트를 소환하지만, 사실은 반환되어 오는 게 새로운 배열 원소이고

-> 이게 원래는 받는 변수로 들어가야하지만, 받는 변수가 없으므로 바로 실행되어

-> 화면에 표시되는 것??

**key 존재 유무에 따른 업데이트 방식**

const array =[‘a’, ‘b’, ‘c’, ‘d’];

array.map(item=> <div>{item}</div>);

위 같이 했을 때 위 배열의 b와 c사이에 z를 삽입하게 된다면, 리렌더링을 하게 될 때 <div>b</div>와

<div>c</div> 사이에 새 div 태그를 삽입하게 되는 것이 아니고, 기존의 c가 z로 바뀌고, d는 c로

바뀌고 마지막에 d가 새로 삽입됨.

이는 매우 비효율적. 하지만 각 배열 원소에 각각의 key값이 있으면 기존의 값은 그대로 두고 원하는 곳에 내용을 삽입하거나 삭제함

array.map(item => <div key={item.id}> {item.text}</div>);

따라서 배열을 렌더링할 때는 고유한 key값이 있는 것이 중요하며, 만약에 배열 안에 중복되는 key가 있을 때는 렌더링시에 오류메세지가 나타나고 업데이트가 이루어지지 않음.