8253jang@daum.net
장희정

JavaScript의 map() 함수

JavaScript 배열 메서드 중 하나로, 배열의 각 요소에 대해 주어진 함수를 실행하고 그 결과로 새로운 배열을 반환한다. React에서 map( ) 함수는 주로 **동적으로 리스트를 렌더링**할 때 사용된다.
예를 들어, 배열에 있는 데이터를 기반으로 여러 개의 컴포넌트를 생성할 때 유용하다.
map( )은 원본 배열을 변경하지 않고, **변환된 새로운 배열을 반환한다.**

기본적인 map()함수

```
const arr = [1, 2, 3, 4, 5];
const result = arr.map(num => num * 2);

console.log(result);  // [2, 4, 6, 8, 10]
```

React에서 map() 함수의 사용

React 에서 map() 함수는 **리스트 렌더링** 시에 많이 사용된다. 예를 들어배열에 있는 데이터를 기반으로 여러 개의 React 컴포넌트를 동적으로 생성하려면 map()을 사용하여 각 항목을 반복하면서 JSX 를반환다.

```
3   const Ex01_Map = () => {
4       const items = ['Apple', 'Banana', 'Orange'];
5       return (
6           <div>
7               <ul>
8                   {
9                       items.map((item,index)=>
10                          <li key={index}>{item}</li>
11                      )
12                  }
13              </ul>
14          </div>
15      );
16  };
17
18  export default Ex01_Map;
```

key는 각 항목을 식별 할수 있는 고유한 값이어야 하며, 배열을 랜더링 할때 React가 효율적으로 변경 사항을 추적 할수 있도록 돕는다.

8253jang@daum.net
장희정

객체 배열로 동적 리스트 렌더링

배열의 각 항목이 객체일 경우, 객체 속성을 사용하여 렌더링할 수 있다.

```
3   const EX02_Map = () => {
4       const products = [
5           { id: 1, name: 'Apple',  price: 2000 },
6           { id: 2, name: 'Banana', price: 1000 },
7           { id: 3, name: 'Orange', price: 3000 },
8         ];
9       return (
10          <div>
11            <ul>
12              {
13              products.map(product => (
14                  <li key={product.id}>
15                      {product.name} - {product.price}원
16                  </li>
17              ))}
18            </ul>
19          </div>
20      );
21  };
```

JSX를 사용하여 복잡한 컴포넌트 렌더링

map() 함수는 JSX 내부에서 컴포넌트를 렌더링하는 데에도 사용된다.

8253jang@daum.net
장희정

```jsx

const Product = ({ name, price }) =>{
    return (
      <div>
        <h2>{name}</h2>
        <p>Price: {price}원 </p>
      </div>
    );
  }

const Ex03_Map = () => {
    const products = [
        { id: 1, name: 'Apple', price: 2000 },
        { id: 2, name: 'Banana', price: 1000 },
        { id: 3, name: 'Orange', price: 3000 },
      ];
    return (
        <div>
        {products.map(product => (
          <Product key={product.id} name={product.name} price={product.price} />
        ))}
      </div>
    );
};

export default Ex03_Map;
```

주의

React 에서 리스트를 렌더링할 때 key 속성은 매우 중요하다. key 는 각 항목을 고유하게 식별할 수 있도록 돕고, React 가 변경된 항목을 효율적으로 업데이트할 수 있게 해준다.

key 값은 고유해야 하며, 일반적으로 항목의 id 값을 사용하는 것이 좋다.

인덱스를 key로 사용하는 것은 바람직하지 않다.특히, 배열이 동적으로 변할 때 문제가 발생할 수 있다.

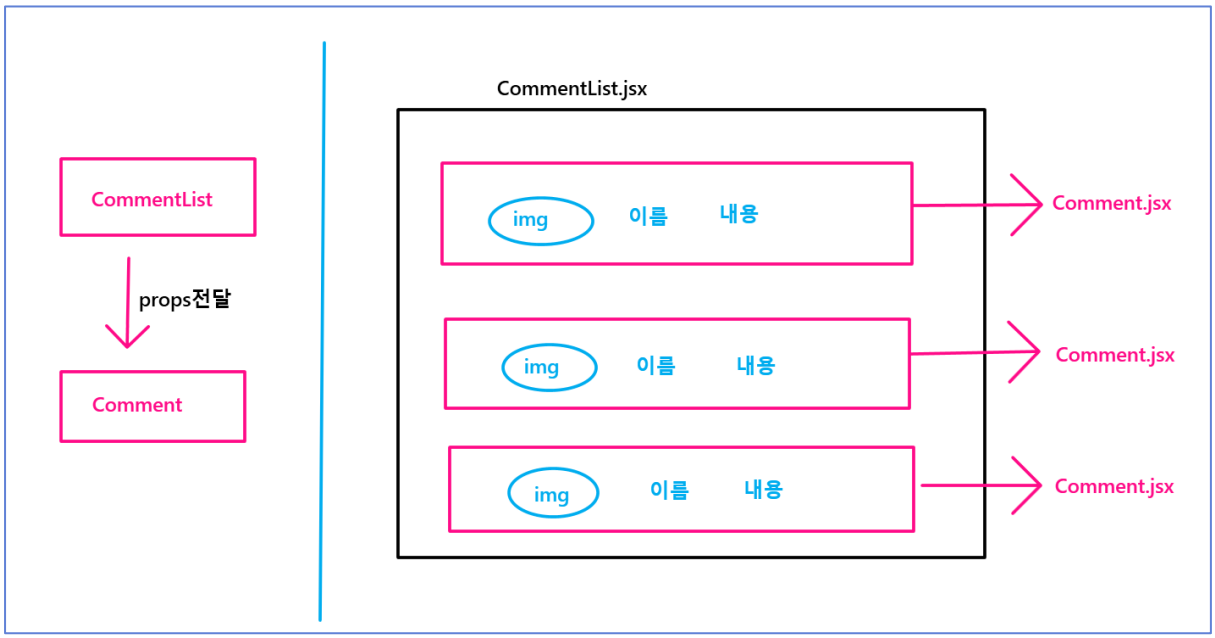
.

댓글 컴포넌트 만들기

src>comments >Comment.jsx | CommentList.jsx

8253jang@daum.net
장희정

전체구조



Comment.jsx에 추가

```
const styles = {
    wrapper: {
        margin: 8,
        padding: 8,
        display: "flex",
flexDirection: "row",
        border: "1px solid grey",
borderRadius: 16,
    },
imageContainer: {},
    image: {
        width: 50,
        height: 50,
borderRadius: 25,
    },
contentContainer: {
marginLeft: 8,
        display: "flex",
flexDirection: "column",
justifyContent: "center",
    },
nameText: {
        color: "black",
```

```
fontSize: 16,
fontWeight: "bold",
    },
commentText: {
        color: "black",
fontSize: 16,
    },
};
```

commentList.jsx에 추가

```
const comments = [
    {
        name: "장희정",
        comment: "안녕하세요, Best Programmer입니다.",
    },
    {
        name: "이가현",
        comment: "리액트 재미있어요~!",
    },
    {
        name: "이찬범",
        comment: "저도 리액트 배워보고 싶어요!!",
    },
];
```

이미지 경로

<imgsrc="./assert/profile.png" alt="" style={styles.image} />

Comment.jsx 파일의 일 부분

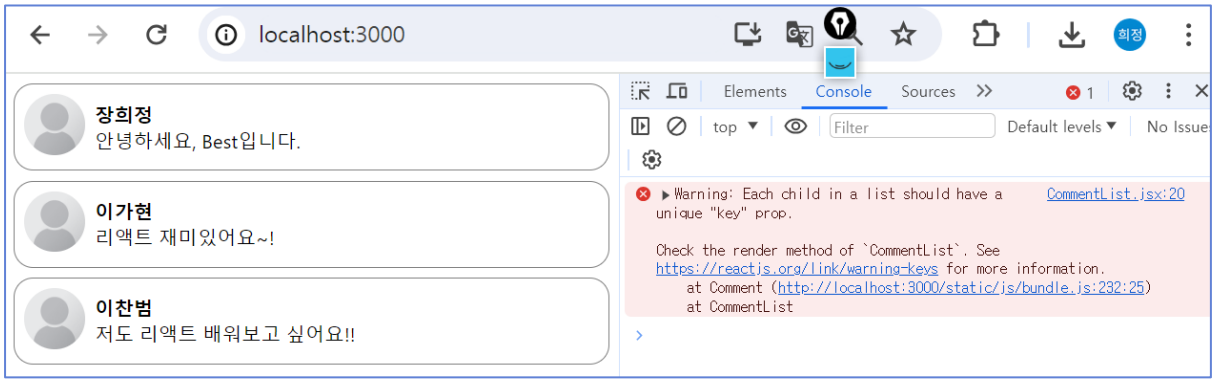
```
38      return (
39          <div style={styles.wrapper}>
40              <div style={styles.imageContainer}>
41              <img src={profile} alt="" style={styles.image} />
42              </div>
43              <div style={styles.contentContainer}>
44                  <span style={styles.nameText}>{props.name}</span>
45                  <span style={styles.commentText}>{props.text}</span>
46              </div>
47          </div>
48      );
49  };
50
51  export default Comment;
```

CommentList.jsx 일 부분

```jsx
    return (
        <div>
        {
            comments.map((com, index)=>{
                return <Comment name={com.name}  text={com.comment} />
            })
        }
        </div>
    );
};

export default CommentList;
```

실행결과



위 그림의 오른쪽에 오류가 나는 이유?

만약, map함수를 사용했을 때 key를 설정하지 않으면 위의 오른쪽 같은 오류가 발생한다.

리액트에서**key는 컴포넌트 배열을 렌더링했을 때 어떤 원소에 변동이 있었는지 알아내려고 사용한다.** 유동적인 데이터를 다룰 때는 key가 없다면 Virtual DOM을 비교하는 과정에서 리스트를 순차적으로 비교하면서 변화를 감지한다. 하지만 key가 있다면 이 값을 이용하여 어떤 변화가 일어났는지 더욱 빠르게 알아낼 수 있다.

key값은 언제나 유일해야한다. 따라서 데이터가 가진 고유값을 key값으로 설정해야 한다.

단, map의 index로 key를 설정하면 안된다!

아래의 코드를 실행해 보고 index를 key로 설정했을때의 문제점을 확인하고 수정해보자.

수정코드 - key추가

```
26      return (
27          <div>
28          {
29              comments.map((com, index)=>{
30                  return <Comment name={com.name}  text={com.comment} key={com.name}/>
31              })
32          }
33          </div>
34      );
35  };
36
37  export default CommentList;
```

Ex04_MapKeyTest.jsx 파일 참고

```
1   import React, { useState } from 'react';
2
3   const Ex04_MapKeyTest01 = () => {
4
5     // const menu = ["짬뽕","짜장"];
6     const [list , setList] = useState( ["짬뽕","짜장"] );
7     const [inputValue, setInputValue ] = useState("")
8
9     //추가
10    const addList = ()=>{
11       // console.log("inputValue" + inputValue);
12
13       // text박스의 값을 조회
14       setList((preList)=>[inputValue, ...preList]);
15
16       //추가후에 text박스 지우기
17       setInputValue("");
18    }
19
20    const inputChangeValue = (e)=>{
21        setInputValue( e.target.value )
22    }
23
24      return (
25        <div>
26            <input type='text'  onChange={inputChangeValue} value={inputValue}/>
27            <button onClick={addList}>추가</button>
28            <ul>
29               {
30                list.map((menu, index)=>{
31                    return (<li key={index}>{menu}</li>)
32                })
33               }
34
35            </ul>
36        </div>
37    );
38  };
39
40  export default Ex04_MapKeyTest01;
```

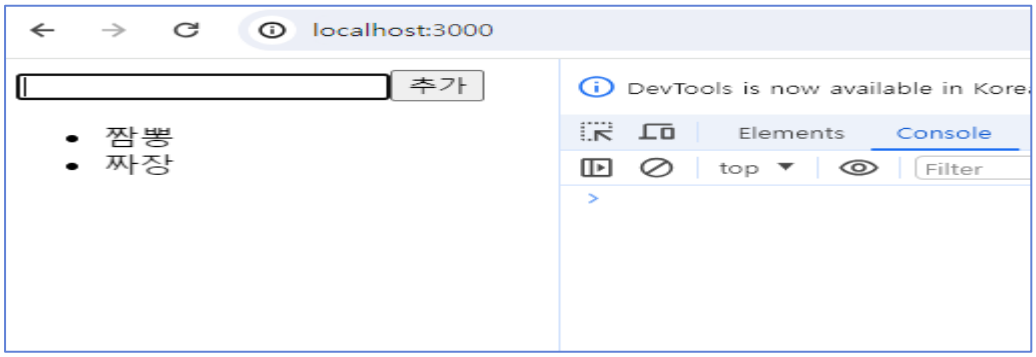
key를 index로 설정했을 때의 문제점?

8253jang@daum.net
장희정

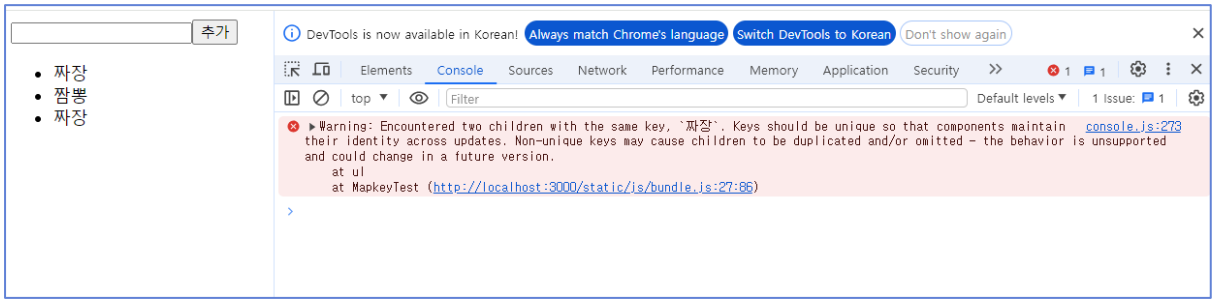
key가 중복 되었을 때의 문제점을 확인해보자!

```
<ul>
    {
     list.map((menu, index)=>{
         return (<li key={menu}>{menu}</li>)
     })
    }

</ul>
```



동일한 메뉴를 추가해보자!



동일한 key를 넣으면 경고메시지가 나온다.

오류를 해결 해 보자.(Ex05_MapKeyTest.jsx 파일참고)

8253jang@daum.net
장희정

```jsx
const Ex05_MapkeyTest02 = () => {
    console.log("MapkeyTest02 call..... - rendering....")
    //useState에 의해 관리되는 변수의 상태가 변경되면 화면을 다시 그린다.
    // - rerendering된다. --> 컴포넌트 = 함수가 다시 호출된다..(함수안에 있는 모든 변수는초기화!!!)

    //inputValue는 text박스에 값이 변경되면 setInputValue를 이용해서 값을 수정한다.
    const [inputValue, setInputValue] = useState('');

    const [list, setList] = useState([
        {id:1, value:"짬뽕"},
        {id:2, value:"짜짱"},
    ]);

    const inputValueChange = (e)=>{
        setInputValue( e.target.value)
    }
    const addList = ()=>{
        setList((preList)=>[
            {id: list.length + 1, value: inputValue},
            ...preList
            ]);
        setInputValue("");
    }
    return (
        <>
          <input type='text' onChange={inputValueChange} value={inputValue}/>
          <button onClick={addList}>추가</button>
          <ul>
            {
                list.map((menu,i)=>{
                    return(
                    <li key={menu.id}>{menu.value}</li>
                    )
            })
            }
          </ul>
        </>
    );
};

export default Ex05_MapkeyTest02;
```