

# React

◆ 배열 컴포넌트

정수아

# Contents

**01** 영화 목록 앱 구현

**02** `map()` 함수

**03** 영화 목록 앱 수정

**04** 배열 컴포넌트

**05** `key props`



01

## 영화 목록 앱 구현

# 영화 목록 앱 구현해보기

## ❖ 예) 영화 목록 앱 구현

- MovieComponent.js

```
import React from 'react';

function MovieComponent(props){
  const { title, open } = props;
  return (
    <div>
      <h1>제목 : {title} </h1>
      <h1>개봉 : {open} </h1>
      <hr/>
    </div>
  );
};

export default MovieComponent;
```

# 영화 목록 앱 구현해보기

- App.js

```
import React from 'react';
import MovieComponent from './MovieComponent';

function App(){
  return (
    <div>
      <MovieComponent title="공조2: 인터내셔날" open="2022.09.07"/>
      <MovieComponent title="인생은 아름다워" open="2022.09.28"/>
      <MovieComponent title="정직한 후보2" open="2022.09.28"/>
    </div>
  );
}

export default App;
```

# 영화 목록 앱 구현해보기

## ❖ 실행 결과



제목 : 공조2: 인터내셔널
개봉 : 2022.09.07
제목 : 인생은 아름다워
개봉 : 2022.09.28
제목 : 정직한 후보2
개봉 : 2022.09.28

# 영화 목록 앱 구현해보기

## ❖ 문제점

- 1. 만약 출력해야 할 영화 정보가 1,000개라면?
  - 1,000개를 반복해서 코드 작성
  - 각 태그마다 서로 다른 데이터 값 입력
- 2. 서버에서 데이터를 받아오는 경우
  - 몇 개의 데이터가 넘어오는지 알 수 없음



02

**map() 함수**



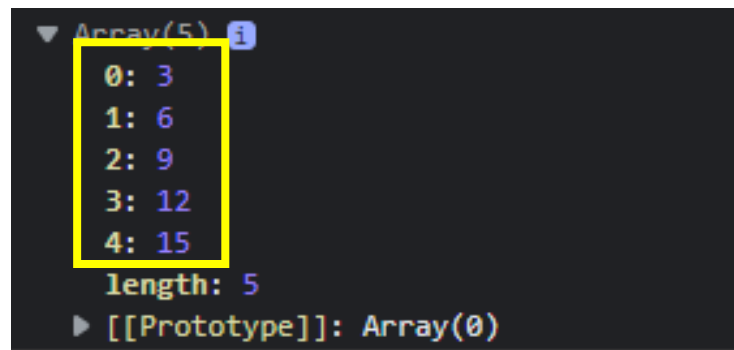
# map() 함수

## ❖ map() 함수란?

- 배열 안의 모든 요소들의 값을 변경해서 새로운 배열을 생성하는 함수

```
const arr = [1, 2, 3, 4, 5];  
const modifiedArr = arr.map(function(element){  
    return element * 3;  
});  
  
console.log(modifiedArr);
```

- 실행 결과



# map() 함수

## ❖ map() 함수 동작 과정

```
const arr = [1, 2, 3, 4, 5];
```

```
const modifiedArr = arr.map(function(element){  
  return element * 3; 작업  
});
```

1  
2  
3  
4  
5

결과물 = 배열



# 03

**map() 함수를 이용하여  
영화 목록 앱 수정**

# 영화 목록 앱 수정

## ❖ map() 함수를 이용하여 영화 목록 앱 수정하기

- 데이터는 서버에서 전달 받는다고 가정
- 데이터는 객체 형식으로 배열에 저장

```
function App(){  
  const movieList = [  
    {  
      title : "공조2: 인터내셔날",  
      open : "2022.09.07"  
    },  
    {  
      title : "인생은 아름다워",  
      open : "2022.09.28"  
    },  
    {  
      title : "정직한 후보2",  
      open : "2022.09.28"  
    }  
  ];  
  return (<></>);  
}  
export default App;
```

# 영화 목록 앱 수정

## ❖ App.js

```
const movieList = [ ...생략... ]

return (
  <>
    {
      movieList.map(movie =>
        <MovieComponent title={movie.title} open={movie.open} />)
    }
  </>
);
```

# 영화 목록 앱 수정

## ❖ 실행 결과



제목 : 공조2: 인터내셔널
개봉 : 2022.09.07
제목 : 인생은 아름다워
개봉 : 2022.09.28
제목 : 정직한 후보2
개봉 : 2022.09.28

# 영화 목록 앱 수정

## ❖ 영화 포스터 추가하기



# 영화 목록 앱 수정

## ❖ 영화 포스터 추가하기

- 이미지 데이터 추가

```
const movieList = [  
  {  
    title : "공조2: 인터내셔날",  
    open : "2022.09.07",  
    img : "https://search.pstatic.net/common?type=o&size~~~"  
  },  
];
```

- img 프로퍼티 추가

```
return <MovieComponent title={movie.title} open={movie.open}  
    img={movie.img} />
```



# 영화 목록 앱 수정

- MovieComponent 수정

```
function MovieComponent(props){  
  const { title, open, img } = props;  
  
  return (  
    <div>  
      <h3>제목 : {title} </h3>  
      <h3>개봉 : {open} </h3>  
      <img src = {img} />  
      <hr/>  
    </div>  
  );  
};
```



04

## 배열 컴포넌트

# 배열 컴포넌트를 위한 map() 함수 사용 방법

## ❖ 자바스크립트의 배열

- 다양한 자료형을 저장 가능

```
// 숫자형
const numberList = [1, 2, 3, 4, 5];

// 2개 이상의 자료형
const mixedList = [1, 'str', {}, function a() { }];
```

- XML과 JSX 또한 저장 가능

```
// JSX 배열
const componentList = [ <MyComponent />, <MySecondComponent />
                        <b>Hi</b> ];
```

# 배열 컴포넌트를 위한 map() 함수 사용 방법

## ❖ map() 함수를 이용하여 배열의 데이터를 JSX로 변환

- App.js

```
function App(){
  const todoList = [
    { taskName : '빨래하기', finished : false },
    { taskName : '공부하기', finished : true }
  ];

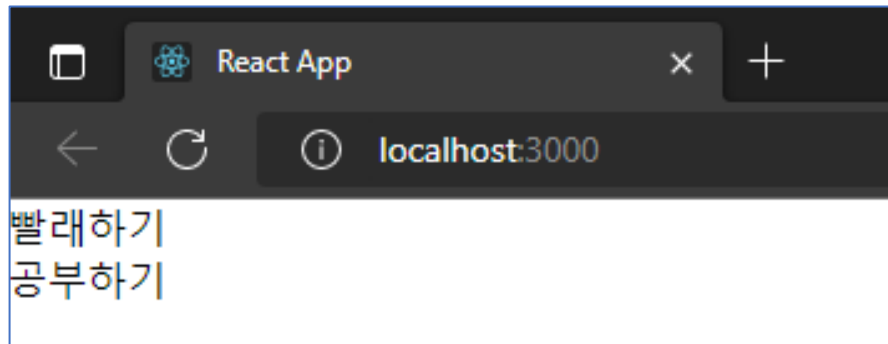
  return (
    <>
      {
        todoList.map(todo => <div>{ todo.taskName }</div>)
      }
    </>
  );
}
```

# 배열 컴포넌트를 위한 map() 함수 사용 방법

## ❖ 실행 결과

- JSX 배열 리턴

```
[<div>빨래하기</div>, <div>공부하기</div>]
```



# 배열 컴포넌트를 위한 map() 함수 사용 방법

## ❖ 배열 데이터를 컴포넌트의 프로퍼티에 전달하기

- App.js

```
function App(){
  const todoList = [
    { taskName : '빨래하기', finished : false },
    { taskName : '공부하기', finished : true }
  ];

  return (
    <>
    {
      todoList.map(todo =>
        <TodoTask taskName={todo.taskName}/>)
    }
    </>
  );
}
```

# 배열 컴포넌트를 위한 map() 함수 사용 방법

## ❖ TodoTask.js

```
import React from 'react';

function TodoTask(props){
  const { taskName } = props;

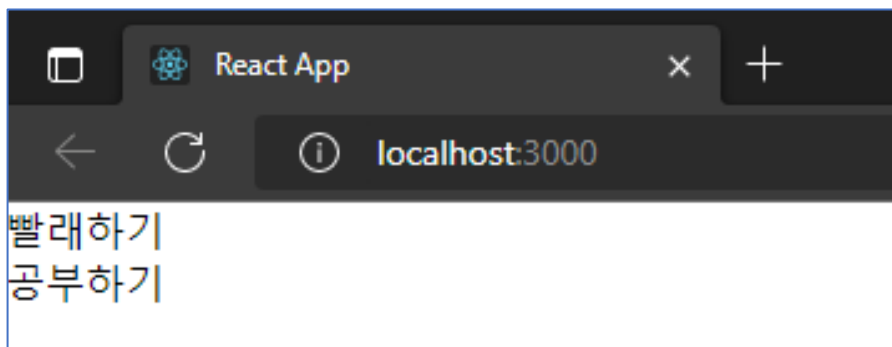
  return (
    <div>
      {taskName}
    </div>
  );
};

export default TodoTask;
```

# 배열 컴포넌트를 위한 map() 함수 사용 방법

## ❖ 실행 결과

```
[<ToDoTask taskName=“빨래하기”/>, <ToDoTask taskName=“공부하기” />]
```





# [실습] map() 함수를 활용한 가격표 목록 출력

## ❖ App.js

```
import React from 'react';
import PriceList from './PriceList';

function App(){
  const priceList = [1000, 2000, 3000, 4000];
  const prices = priceList.map(price => <div>가격 : {price}원</div>)

  return (
    <>
      <PriceList prices={prices}/>
    </>
  );
}

export default App;
```

# [실습] map() 함수를 활용한 가격표 목록 출력

## ❖ PriceList.js

```
import React from 'react';

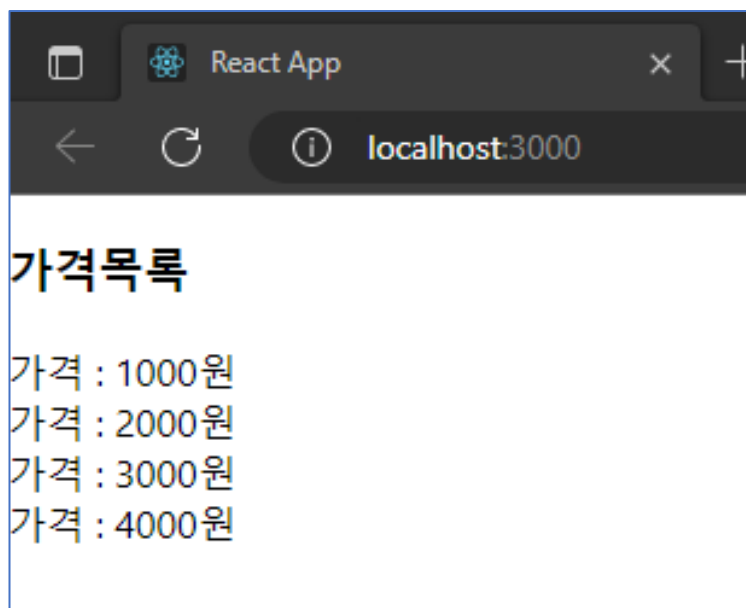
function PriceList(props) {
  const { prices } = props;

  return (
    <div>
      <h3>가격 목록</h3>
      {prices}
    </div>
  );
};

export default PriceList;
```

# [실습] map() 함수를 활용한 가격표 목록 출력

## ❖ 실행 결과






**05**

**key props**

# 배열 컴포넌트의 key props

## ❖ 개발자 도구를 보면 다음과 같은 경고 메시지가 나옴

- 배열 속성에 key 값이 없기 때문

```
✖ ▶ Warning: Each child in react-jsx-dev-runtime.development.js:87   
  a list should have a  
  unique "key" prop.  
  
  Check the render method of `App`. See https://reactjs.org/link/warning-keys for more information.  
    at div  
    at App
```

- 리액트의 원소가 리스트에 포함되면서 리액트 원소들의 유일성이 사라짐
- 배열 속성에 key 값을 추가하여 해결 가능함

# 배열 컴포넌트의 key props

## ❖ 배열 컴포넌트에 key props 추가

- 배열 컴포넌트는 배열 원소의 개수만큼 반복하므로 성능에 영향을 많이 줌
- 배열에 key 값을 추가하면 성능 향상에 도움이 됨
  - key 값을 정의하여 출력한 배열 컴포넌트를 다시 출력해야 하는 경우, 리액트 엔진이 기존 컴포넌트를 재활용하여 성능을 높일 수 있음
- key props는 리액트 내부에서 사용되는 특수한 props이기 때문에 하위 컴포넌트에는 전달되지 않음

# 배열 컴포넌트의 key props

## ❖ 기존 코드 수정하기

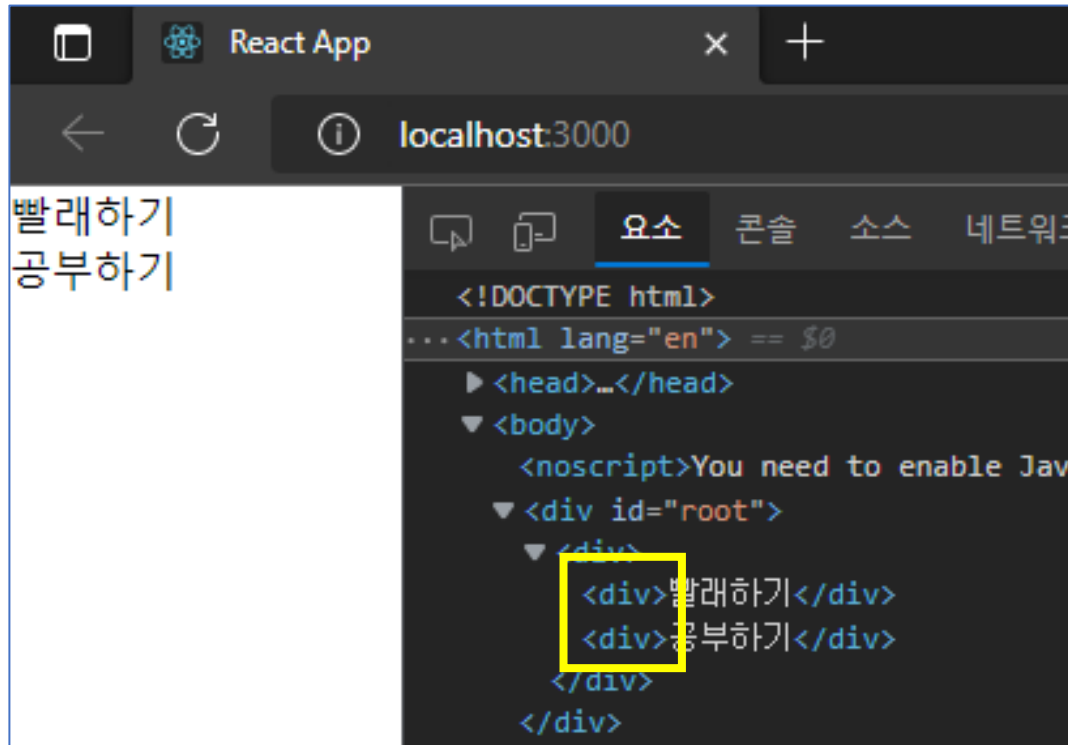
- App.js

```
function App(){
  const todoList = [
    { taskName : '빨래하기', finished : false },
    { taskName : '공부하기', finished : true }
  ];

  return (
    <>
    {
      todoList.map(todo =>
        <div key={todo.taskName}>{todo.taskName}</div>
      )
    }
    </>
  );
}
```

# 배열 컴포넌트의 key props

## ❖ 실행 결과



- key props가 전달되지 않은 것을 확인!



**THANK 😊 YOU**