

리스트와 map함수

안 화 수

리스트

❖ 리스트

1. 리스트는 같은 아이템을 순서대로 모아 놓은 것을 의미한다.
2. 리스트를 사용하기 위한 자료구조가 **배열**이다.
3. 배열은 자바스크립트에서 변수나 객체를 하나의 변수로 묶어 놓은 것이다.

```
const numbers = [1, 2, 3, 4, 5]
```

리스트

❖ 프로젝트 생성

```
npx create-react-app list01
```

map

❖ map함수

배열에 저장된 원소값을 반복적으로 구해오는 함수

```
const subject = ["자바","오라클","JSP","Spring","파이썬"]
```

```
subject.map((name, index)=>{  
    <li key={index}>{name}</li>  
})
```

리스트

❖ 리스트 예제1 : **src/App.js**

```
import Main from './Main';  
import AttendanceBook from './AttendanceBook';
```

```
function App() {  
  
  return(  
    <Main/>  
    // <AttendanceBook/>  
  )  
  
}
```

```
export default App;
```

리스트

❖ 리스트 예제1 : src/Main.js (1/2)

```
import React from "react";
```

```
//function Main(props){  
const Main=(props)=>{
```

```
  const subject = ["자바","오라클","JSP","Spring","파이썬"]
```

리스트

❖ 리스트 예제1 : src/Main.js (2/2)

```
return(  
  <ul>  
    { subject.map(function(name){  
      return <li>{name}</li>  
    }) }  
    <br></br>  
    { subject.map((name)=>{  
      return <li>{name}</li>  
    }) }  
    <br></br>  
    { subject.map((name, index)=>{  
      return <li key={index}>{name}</li>  
    })}  
  </ul>  
)  
}  
export default Main;
```

리스트

❖ 리스트 예제2 : src/AttendanceBook.js (1/2)

```
import React from "react";
```

```
function AttendanceBook(props) {
```

```
  const students = [
```

```
    {
```

```
      id: 1,
```

```
      name: "홍길동",
```

```
    },
```

```
    {
```

```
      id: 2,
```

```
      name: "홍길순",
```

```
    },
```

```
    {
```

```
      id: 3,
```

```
      name: "김길동",
```

```
    },
```

```
    {
```

```
      id: 4,
```

```
      name: "안화수",
```

```
    },
```

```
  ];
```


리스트

❖ 리스트 예제2 : src/AttendanceBook.js (2/2)

```
return (  
  <ul>  
    { students.map((student, index) => {  
      return <li key={student.id}>{student.name}</li>;  
    })}  
  </ul>  
);  
}
```

```
export default AttendanceBook;
```

리스트

❖ 프로젝트 생성

```
npx create-react-app list02
```

❖ 컴포넌트 나누기

```
index.js - App.js - Main1.js  
Main2.js  
Main3.js
```

리스트

❖ 배열 전달하기

부모 컴포넌트에서 자식 컴포넌트에게 배열값 전달하기

```
const names1 = ['홍길동','홍길순','김길동'];           // 1차원 배열
const names2 = [['홍길동', 19], ['홍길순', 29], ['김길동', 39]] // 2차원 배열
const names3 = [{userName:'홍길동',age:19},           // 2차원 배열(객체)
                 {userName:'홍길순',age:29},
                 {userName:'김길동',age:39}]
```

// 배열의 원소를 자식 컴포넌트에 전달하기

```
const nameList1 = names1.map((name)=>(<Main1 name={name}/>));
const nameList2 = names2.map((v)=>(<Main2 name={v[0]} age={v[1]}/>));
const nameList3 = names3.map((v)=>(<Main3 name={v.userName} age={v.age}/>));
```

리스트

❖ 리스트 예제2 : src/App.js (1/3)

```
import React from 'react';  
import Main1 from './Main1';  
import Main2 from './Main2';  
import Main3 from './Main3';
```

```
// 부모 컴포넌트
```

```
function App() {
```

```
  const names1 = ['홍길동','홍길순','김길동']; // 1차원 배열  
  const names2 = [['홍길동', 19], ['홍길순', 29], ['김길동', 39]] // 2차원 배열  
  const names3 = [{userName:'홍길동',age:19}, // 2차원 배열(객체)  
                  {userName:'홍길순',age:29},  
                  {userName:'김길동',age:39}]
```

리스트

❖ 리스트 예제 2 : src/App.js (2/3)

// 배열의 원소를 자식 컴포넌트에 전달하기

```
const nameList1 = names1.map((name)=>(<Main1 name={name}/>));
```

```
const nameList2 = names2.map((v)=>(<Main2 name={v[0]} age={v[1]}/>));
```

```
const nameList3 = names3.map((v)=>(<Main3 name={v.userName} age={v.age}/>));
```

// filter함수를 이용해서 age가 30보다 큰 데이터만 자식 컴포넌트에 전달하기

```
let nameList4 = names3.filter(v=>v.age > 30);
```

```
nameList4 = nameList4.map((v)=>(<Main3 name={v.userName} age={v.age}/>));
```

리스트

❖ 리스트 예제2 : src/App.js (3/3)

```
return (  
  <div>  
    {nameList1}<hr></hr>  
    {nameList2}<hr></hr>  
    {nameList3}<hr></hr>  
    {nameList4}<hr></hr>  
  </div>  
);  
}
```

```
export default App;
```

리스트

❖ 리스트 예제2 : src/Main1.js

```
import React from "react";
```

```
// 자식 컴포넌트
```

```
const Main1=(props)=>{
```

```
  return(
```

```
    <div>
```

```
      <h3>안녕 하세요? {props.name}입니다.</h3>
```

```
    </div>
```

```
  )
```

```
}
```

```
export default Main1;
```

리스트

❖ 리스트 예제2 : src/Main2.js

```
import React from "react";
```

```
// 자식 컴포넌트
```

```
const Main2=(props)=>{
```

```
  return(
```

```
    <div>
```

```
      <h3>안녕 하세요? {props.name}({props.age}세)입니다.</h3>
```

```
    </div>
```

```
  )
```

```
}
```

```
export default Main2;
```


리스트

❖ 리스트 예제2 : src/Main3.js

```
import React from "react";
```

```
// 자식 컴포넌트
```

```
const Main3=(props)=>{
```

```
  return(
```

```
    <div>
```

```
      <h3>안녕 하세요? {props.name}({props.age}세)입니다.</h3>
```

```
    </div>
```

```
  )
```

```
}
```

```
export default Main3;
```