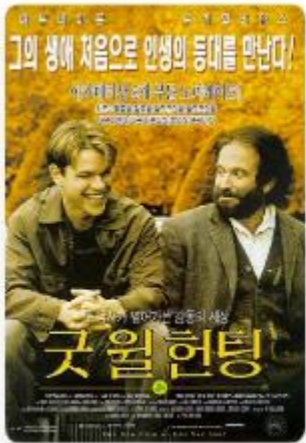


# 응용 실습

리스트 렌더링과 컴포넌트 합성

# 이미지 리스트 렌더링 + Card 컴포넌트 합성 실습



```
const images = [
  { src: "1.jpg", title: "Movie 1" },
  { src: "2.png", title: "Movie 2" },
  { src: "3.jpg", title: "Movie 3" },
  { src: "4.jpg", title: "Movie 4" },
];
```

# 이미지 리스트 렌더링 + Card 컴포넌트 합성 실습

## 영화 소개



### 굿윌헌팅

비상한 머리와 재능을 가졌지만 어린시절의 아픔으로 인해 질풍노도의 시기를 보내는 월의 재능을 알아본 MIT 교수는 자신의 대학 동기인 심리학 교수 손에게 월을 부탁한다. 월은 심리학 교수 손과의 만남을 통해 상처를 치유하며 점차 변화하기 시작한다.



### 이미테이션 게임

세계 2차대전 중, 독일군은 문장을 이해할 수 없는 글자 배열로 바꾸어 무한대에 가까운 암호 조합을 만들어내는 에니그마를 사용해 소통한다. 이를 해독하기 위해 연합군은 천재 수학자 앨런 튜링을 암호 해독팀에 채용하고, 앨런은 에니그마 해독을 위해 특별한 기계를 발명한다.



### 인셉션

국제적 수배자로 쫓기고 있는 특수 요원 코브는 타인의 꿈에 잠입하여 생각을 훔치는 일을 하며 숨어산다.

## 영화 소개



### 굿윌헌팅

비상한 머리와 재능을 가졌지만 어린시절의 아픔으로 인해 질풍노도의 시기를 보내는 월의 재능을 알아본 MIT 교수는 자신의 대학 동기인 심리학 교수 손에게 월을 부탁한다. 월은 심리학 교수 손과의 만남을 통해 상처를 치유하며 점차 변화하기 시작한다.



### 이미테이션 게임

세계 2차대전 중, 독일군은 문장을 이해할 수 없는 글자 배열로 바꾸어 무한대에 가까운 암호 조합을 만들어내는 에니그마를 사용해 소통한다. 이를 해독하기 위해 연합군은 천재 수학자 앨런 튜링을 암호 해독팀에 채용하고, 앨런은 에니그마 해독을 위해 특별한 기계를 발명한다.



### 인셉션

국제적 수배자로 쫓기고 있는 특수 요원 코브는 타인의 꿈에 잠입하여 생각을 훔치는 일을 하며 숨어산다.

```
const profiles = [  
  {  
    image: "./images/1.jpg",  
    name: "굿윌헌팅",  
    intro: "내용 1",  
  },  
  {  
    image: "./images/2.png",  
    name: "이미테이션 게임",  
    intro: "내용 2",  
  },  
  {  
    image: "./images/3.jpg",  
    name: "인셉션",  
    intro: "내용 3",  
  },  
  {  
    image: "./images/4.jpg",  
    name: "인터스텔라",  
    intro: "내용 4",  
  },  
];
```

# 이미지 리스트 렌더링 + Card 컴포넌트 합성 실습

```
<div className="profile">
  <h1>영화 소개</h1>
  <div className="profile-list">
    {profiles.map((profile, index) => (
      <Card key={index}>
        <img
          src={profile.image}
          alt={`${profile.name}'s profile`}
          className="profile-image"
        />
        <h1>{profile.name}</h1>
        <p>{profile.intro}</p>
      </Card>
    ))}
  </div>
</div>
```

# 이미지 리스트 렌더링 + Card 컴포넌트 합성 실습

```
function Card({}) {  
  return (  
    <div className="card">  
        
    </div>  
  );  
}
```

# 이미지 리스트 렌더링 + Card 컴포넌트 합성 실습

## Profiles.css

```
.profile {
  font-family: Arial, sans-serif;
  padding: 20px;
}

h1 {
  text-align: center;
  color: #333;
  font-size: 20px;
}

.profile-list {
  display: grid;
  grid-template-columns: repeat(auto-fill, minmax(200px, 1fr));
  gap: 16px;
  justify-items: center;
  margin-bottom: 20px;
}
```

```
.card {
  width: 200px;
  padding: 16px;
  border: 1px solid #ddd;
  border-radius: 16px;
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
  text-align: center;
}

.profile-image {
  width: 100%;
  max-width: 150px; /* 가로 크기 제한 */
  height: auto;
  border-radius: 10%;
  object-fit: cover;
  margin-bottom: 12px;
}

.card:hover {
  background-color: #f5ec47db;
}
```

# 미니 프로젝트

코드 분석 및 내용 완성하기

# 할일 앱 만들기

- 주어진 코드 실행 후 필요한 기능을 추가하여 완성해 보세요.

## 1. 초기화면

오늘은 📅

**Tue Nov 26 2024**

새로운 Todo 작성하기 ➡

추가

**Todo List 🌱**

검색어를 입력하세요

<input type="checkbox"/> React 공부하기	2024. 11. 26.	삭제
<input type="checkbox"/> 빨래 널기	2024. 11. 26.	삭제
<input type="checkbox"/> 노래 연습하기	2024. 11. 26.	삭제

## 2. 새로운 todo 추가

오늘은 📅

**Tue Nov 26 2024**

새로운 Todo 작성하기 ➡

추가

**Todo List 🌱**

검색어를 입력하세요

<input type="checkbox"/> 운동하기	2024. 11. 26.	삭제
<input type="checkbox"/> React 공부하기	2024. 11. 26.	삭제
<input type="checkbox"/> 빨래 널기	2024. 11. 26.	삭제
<input type="checkbox"/> 노래 연습하기	2024. 11. 26.	삭제

## 3. 검색

오늘은 📅

**Tue Nov 26 2024**

새로운 Todo 작성하기 ➡

추가

**Todo List 🌱**

react

---

<input type="checkbox"/> React 공부하기	2024. 11. 26.	삭제
-------------------------------------	---------------	----



# 미니프로젝트 1

## ■ App 페이지 구성

오늘은 📅  
**Sat Nov 23 2024**

Header

새로운 Todo 작성하기 ➡

TodoEditor

추가

Todo List 🌱

TodoList

검색어를 입력하세요

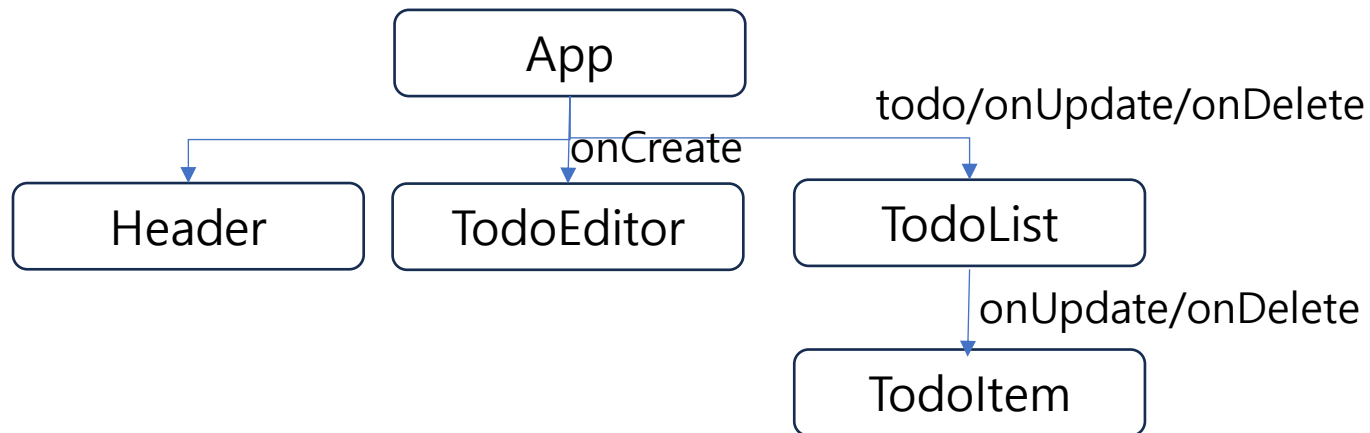
☐ React 공부하기 2024. 11. 23. 삭제

☐ 빨래 널기 2024. 11. 23. 삭제

☐ 노래 연습하기 2024. 11. 23. 삭제

App.js

```
<div className="App">  
  <Header />  
  <TodoEditor onCreate={onCreate} />  
  <TodoList todo={todo} onUpdate={onUpdate}  
    onDelete={onDelete} />  
</div>
```



```
const mockTodo = [  
  {  
    id: 0,  
    isDone: false,  
    content: "React 공부하기",  
    createdAt: new Date().getTime(),  
  },  
  {  
    id: 1,  
    isDone: false,  
    content: "빨래 널기",  
    createdAt: new Date().getTime(),  
  },  
  {  
    id: 2,  
    isDone: false,  
    content: "노래 연습하기",  
    createdAt: new Date().getTime(),  
  },  
];
```

```
function App() {  
  const [todo, setTodo] = useState(mockTodo);  
  const idRef = useRef(3);  
  
  const onCreate = (content) => {  
    const newItem = {  
      id: idRef.current,  
      content,  
      isDone: false,  
      createdAt: new Date().getTime(),  
    };  
    setTodo([newItem, ...todo]);  
    idRef.current += 1;  
  };  
}
```

```
const onUpdate = (targetId) => {
  setTodo(
    todo.map((it) =>
      it.id === targetId ? { ...it, isDone: !it.isDone } : it
    )
  );
};

const onDelete = (targetId) => {
  setTodo(todo.filter((it) => it.id !== targetId));
};

return (
  <div className="App">
    <Header />
    <TodoEditor onCreate={onCreate} />
    <TodoList todo={todo} onUpdate={onUpdate} onDelete={onDelete} />
  </div>
);
}
```

# 미니프로젝트 1

- Header.jsx

오늘은 📅

**Sat Nov 23 2024**

```
<div className="Header">
  <h3>오늘은 📅 </h3>
  <h1>{new Date().toLocaleDateString()}</h1>
</div>
```

# 미니프로젝트 1

## ▪ TodoEditor.jsx

새로운 Todo 작성하기 ➡

새로운 Todo...

추가

```
const TodoEditor = ({ onCreate }) => {
  const [content, setContent] = useState("");
  const inputRef = useRef();

  const onChangeContent = (e) => {
    setContent(e.target.value);
  };

  const onSubmit = () => {
    if (!content) {
      inputRef.current.focus();
      return;
    }
    onCreate(content);
    setContent("");
  };
};
```

```
const onKeyDown = (e) => {
  if (e.keyCode === 13) { // <Enter 키>
    onSubmit();
  }
};

return (
  <div className="TodoEditor">
    <h4>새로운 Todo 작성하기 ➡ </h4>
    <div className="editor_wrapper">
      <input
        ref={inputRef}
        value={content}
        onChange={onChangeContent}
        onKeyDown={onKeyDown}
        placeholder="새로운 Todo..."
      />
      <button onClick={onSubmit}>추가</button>
    </div>
  </div>
);
```

# 미니프로젝트 1

## ▪ TodoList.jsx

**Todo List** 🌱

검색어를 입력하세요

<input type="checkbox"/> React 공부하기	2024. 11. 23.	삭제
<input type="checkbox"/> 빨래 널기	2024. 11. 23.	삭제
<input type="checkbox"/> 노래 연습하기	2024. 11. 23.	삭제

```
const TodoList = ({ todo, onUpdate, onDelete })  
=> {  
  const [search, setSearch] = useState("");
```

search 상태 값 갱신

```
  const getSearchResult = () => {  
    return search === ""  
      ? Todo :  
      todo.filter((it) =>  
        it.content.toLowerCase()  
          .includes(search.toLowerCase())  
      );  
  };  
};
```

# 미니프로젝트 1

## ▪ TodoList.jsx

### Todo List 🌱

검색어를 입력하세요

☐ React 공부하기

2024. 11. 23.

삭제

☐ 빨래 널기

2024. 11. 23.

삭제

☐ 노래 연습하기

2024. 11. 23.

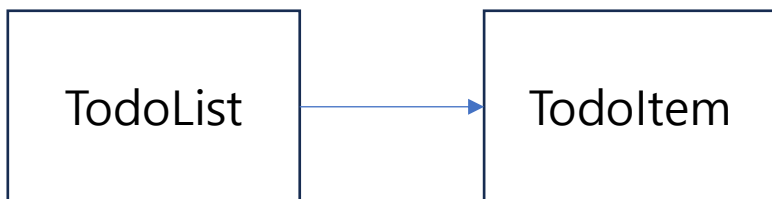
삭제

```
return (  
  <div className="TodoList">  
    <h4>Todo List 🌱 </h4>  
    <input  
      value={search}  
      onChange={onChangeSearch}  
      className="searchbar"  
      placeholder="검색어를 입력하세요"  
    />  
    <div className="list_wrapper">  
      {getSearchResult().map((it) => (  
        <TodoItem  
          key={it.id}  
          {...it}  
          onUpdate={onUpdate}  
          onDelete={onDelete}  
        />  
      ))}  
    </div>  
  </div>  
)  
);
```



# 미니프로젝트 1

- TodoItem.jsx



```
<TodoItem  
  key={it.id}  
  {...it}  
  onUpdate={onUpdate}  
  onDelete={onDelete}  
>
```

☐ React 공부하기

2024. 11. 23.

삭제

```
const TodoItem = ({ id, content, isDone, createdAt, onUpdate, onDelete }) => {  
  const onChangeCheckbox = () => {  
    //해당 체크박스 상태 변경  
  };  
  const onClickDelete = () => {  
    //해당 todo 항목 삭제  
  };  
  
  return (  
    <div className="TodoItem">  
      <div className="checkbox_col">  
        <input onChange={onChangeCheckbox} checked={isDone}  
type="checkbox" />  
      </div>  
      <div className="title_col">{content}</div>  
      <div className="date_col">  
        {new Date(createdAt).toLocaleDateString()}  
      </div>  
      <div className="btn_col">  
        <button onClick={onClickDelete}>삭제</button>  
      </div>  
    </div>  
  );  
};
```

# {...it}

it 객체의 모든 속성을 TodoItem의 props로 전달한다.

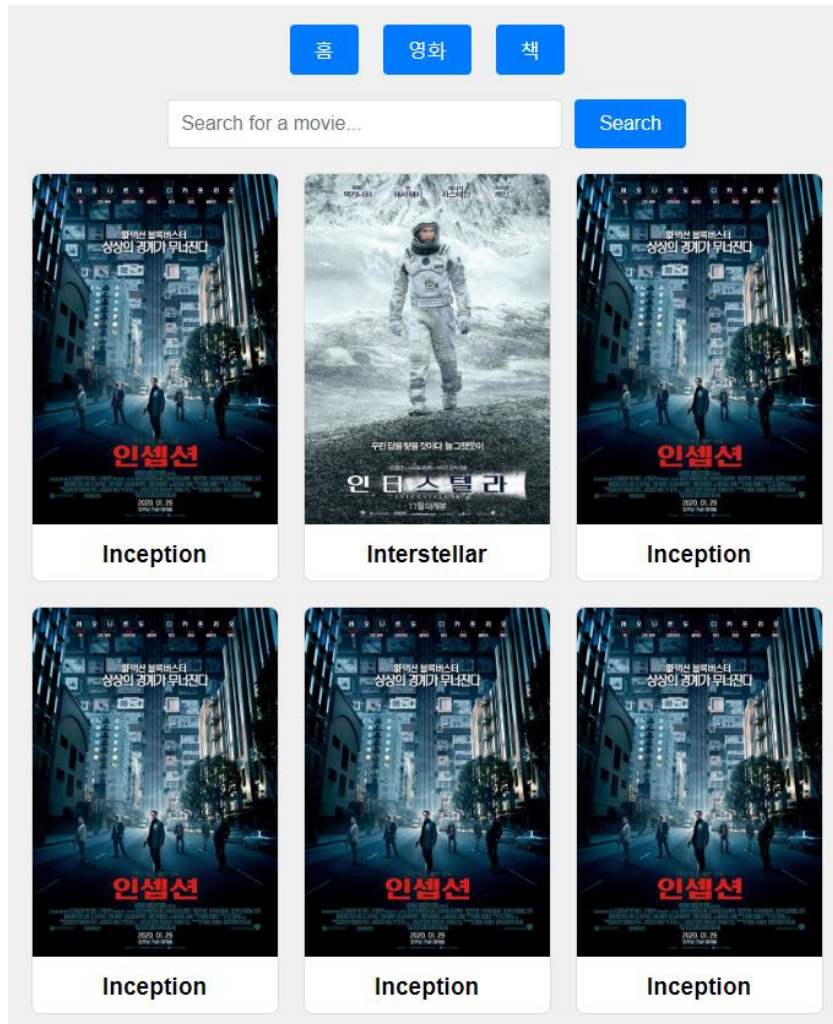
```
it = {  
  id: 0,  
  isDone: false,  
  content: "React 공부하기",  
  createdAt: new Date().getTime(),  
},
```

```
<TodoItem  
  key={it.id}  
  {...it}  
  onUpdate={onUpdate}  
  onDelete={onDelete}  
/>
```

```
<TodoItem  
  id={it.id}  
  content={it.content}  
  isDone={it.isDone}  
  createdAt={it.createdAt}  
  onUpdate={onUpdate}  
  onDelete={onDelete}  
/>
```

# 미니프로젝트 2 - 2인 1조

## 1. 초기 화면



## 2. 이미지 클릭시 상세 설명 화면



# 미니프로젝트 2

## ➤ 코드 분석 후 조건에 맞게 내용 완성하세요.

1. '홈' 버튼을 'About Me' 로 수정하기
2. About Me 화면을 본인 프로필 소개 화면으로 수정하기
3. '책' 페이지 추가하기

