

고급 소프트웨어 실습 7주차 보고서

4분반

20171693 조병화

□각 주요 컴포넌트들 간의 상호관계 및 구조에 대해 설명

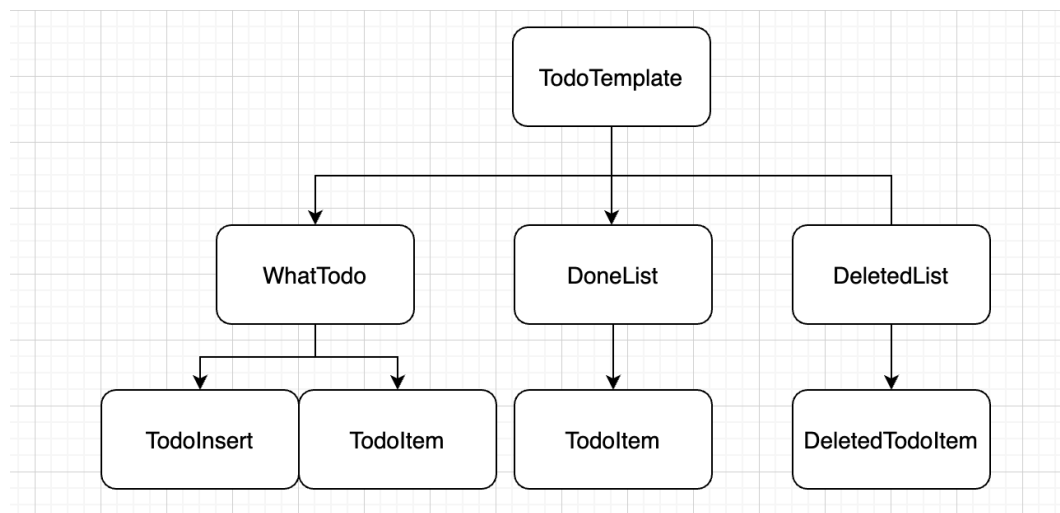
우선 각 컴포넌트를 감싸고 있는 전체 컴포넌트, TodoTemplate 는 3 종류의 컴포넌트들을 import 하고 있는데 그 종류는 WhatToDo, DoneList, DeletedList 컴포넌트 이다.

WhatToDo 컴포넌트는 TodoInsert, TodoItem 컴포넌트를 import 하고 있다.

DoneList 컴포넌트는 TodoItem 을 import 하고 있다.

DeletedList 의 경우에는 DeletedTodoItem 을 import 하고 있다.

상호관계를 나타내 보면 다음과 같은 형태를 가진다 할 수 있다.



□실습 프로젝트의 주요 함수들의 기능 및 코드내용설명

TodoTemplate 함수는 전체적인 3 개 컴포넌트 간 교환, 전달등이 이루어지는 가장 바깥쪽의 함수이다. 내부의 onRemoveTodo, onRemoveDoneTodo, onToggleTodo, onToggleDoneToDo 객체의 기능은 다음과 같다.

onRemoveTodo : WhatToDo 의 리스트 항목 중 체크 된 항목을 찾아 Deleted List 로 옮겨준(추가한) 후, WhatToDo 리스트에서 삭제한다.

onRemoveDoneTodo : Done list 의 리스트 항목 중 선택 된 항목을 찾아 Deleted list 에 추가하고, Done list 에서 삭제해준다.

onToggleTodo : What To Do 의 리스트 항목 중 선택 된 항목을 찾아 Done list 로 옮겨준 후, WhatToDo 리스트에서 삭제한다.

onToggleDoneToDo : Done List 의 리스트 항목 중 선택 된 항목을 찾아 WhatToDo 리스트에 추가하고 doneList 에서 삭제해준다.

위 4 객체들은 이동시키고, 삭제되는 종류만 다를 뿐 형태는 모두 같기 때문에 대표적으로 onRemoveTodo 의 코드를 살펴보도록 하겠다.

```
// 주어진 What To Do 항목(체크안된 항목)의 id로 What To Do에서 해당 항목을 찾고 Deleted List에 추가한 후에, What To Do에서 제외
const onRemoveTodo = (id) => {
  // id에 해당하는 항목을 todos 배열에서 찾을
  const deletedTodo = todos.find((todo) => todo.id === id);
  // 삭제된 todo -> deletedTodos에 저장
  setDeletedTodos([...deletedTodos, deletedTodo]);
  // filter 함수 사용해서 id에 해당하는 항목만 todos 배열에서 제외
  setTodos(todos.filter((todo) => todo.id !== id));
};
```

Find()함수를 통해 Id 로 what to do 에서 선택한 항목을 찾아준다.

그 후 setDeletedTodos 로 deleted to do 에 항목을 저장해준다.

이후 filter 를 이용해 선택한 id 에 해당하는 항목을 what to do 리스트에서 삭제해준다.

WhatToDo 함수는 TodoInsert, TodoItem 함수들을 이용하여 TodoInsert 부분에 입력을 받아 리스트를 보여준다.

TodoInsert 함수 : 사용자로부터 입력을 받고, 빈 입력, submit 후 input 창을 clear 해주는 등 입력 전반적인 기능을 한다.

TodoItem 함수 : 입력으로 받은 객체를 한 줄로 표시해준다. 이 때 체크 버튼과 그 내용을 포함한다.

DoneList 함수 는 doneTodos 의 객체들을 가지고 TodoItem 컴포넌트를 생성해 Done list 부분에 리스트로서 보여준다.

```
function DoneList({ doneTodos, onRemove, onToggle }) {  
  return (  
    <Container>  
      <h1 style={{color: '#415e3e'}}>DONE LIST</h1>  
      <DoneListContainer>  
        {doneTodos.map((todo) => {  
          return <TodoItem todo={todo} key={todo.id} onRemove={onRemove} onToggle={onToggle}/>  
        })}  
      </DoneListContainer>  
    </Container>  
  );  
}
```

중간에 todoitem 을 생성하여 return 하는 모습을 볼 수 있다.

deletedList 함수는 인자로 받은 deleted To do list 의 객체를 이용해 deletedTodoItem 컴포넌트를 생성 해 Deleted List 창에 항목들을 보여주는 역할을 한다.

```
function DeletedList({ deletedTodos }) {  
  return (  
    <Container>  
      <h1 style={{color: '#415e3e'}}>DELETED LIST</h1>  
      <DeletedListContainer>  
        {deletedTodos.map((todo) => {  
          return <DeletedTodoItem todo={todo}/>  
        })}  
      </DeletedListContainer>  
    </Container>  
  );  
}
```

위의 코드에서는 DeletedTodos 의 항목들을 인자로 받아와 DeletedTodoItem 컴포넌트를 만들어 return 하는 것을 확인 할 수 있다.

□ 설명한 함수들 간의 상호 관계에 대해 설명

TodoTemplate 함수에서는 WhatTodo, DoneList, DeletedList 함수들을 통해 컴포넌트들을 생성하고, 리스트에 추가하며 기존 리스트에서 삭제하는 기능들을 총괄한다.

WhatTodo 함수에서는 TodoInsert 함수를 통해 입력을 받고, TodoItem 함수를 통해 컴포넌트를 생성해 리스트로 확인가능하게 보여준다.

DoneList 함수에서는 TodoItem 함수를 통해 TodoItem 컴포넌트를 생성하여 그 객체를 리턴해준다. 즉, 리스트로 확인할 수 있게 보여주는 역할을 한다.

DeletedList 함수에서는 DeletedTodoItem 함수를 이용하여 컴포넌트를 생성해, Deleted List에서 보여준다.