

# React*Redux*

2반  
김혜진



# 목차

1

Redux를 사용하는 이유

2

Redux 핵심개념

# Redux란?



대규모 웹 애플리케이션에  
널리 사용되는 상태 관리 패턴



# Redux 사용하는 이유

1. props 문법의 귀찮음
2. state 변경 관리가 용이



# 1. props 문법의 귀찮음



다른 컴포넌트에서 state에 접근이 불가

props를 통해 전달

만약 컴포넌트 100개가 중첩되어 있다면?



# 1. props 문법의 귀찮음

Redux 사용한다면?

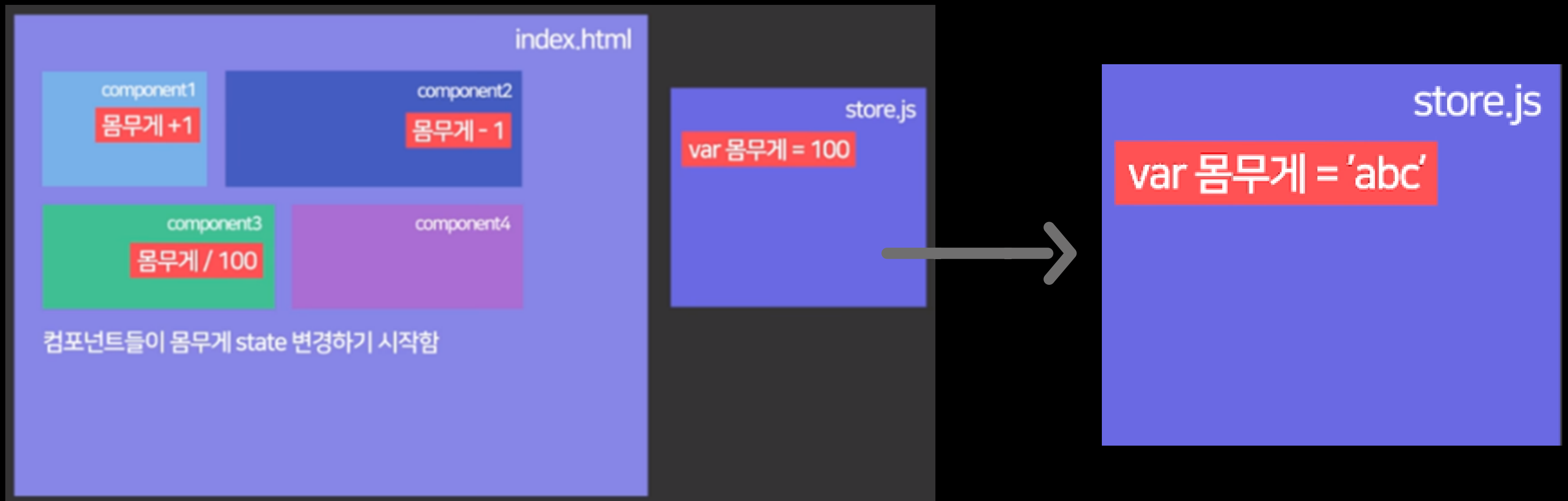
store.js에 state들을 보관



모든 컴포넌트가 props 없이 state에 직접 접근이 가능해짐



## 2. state 변경 관리가 용이

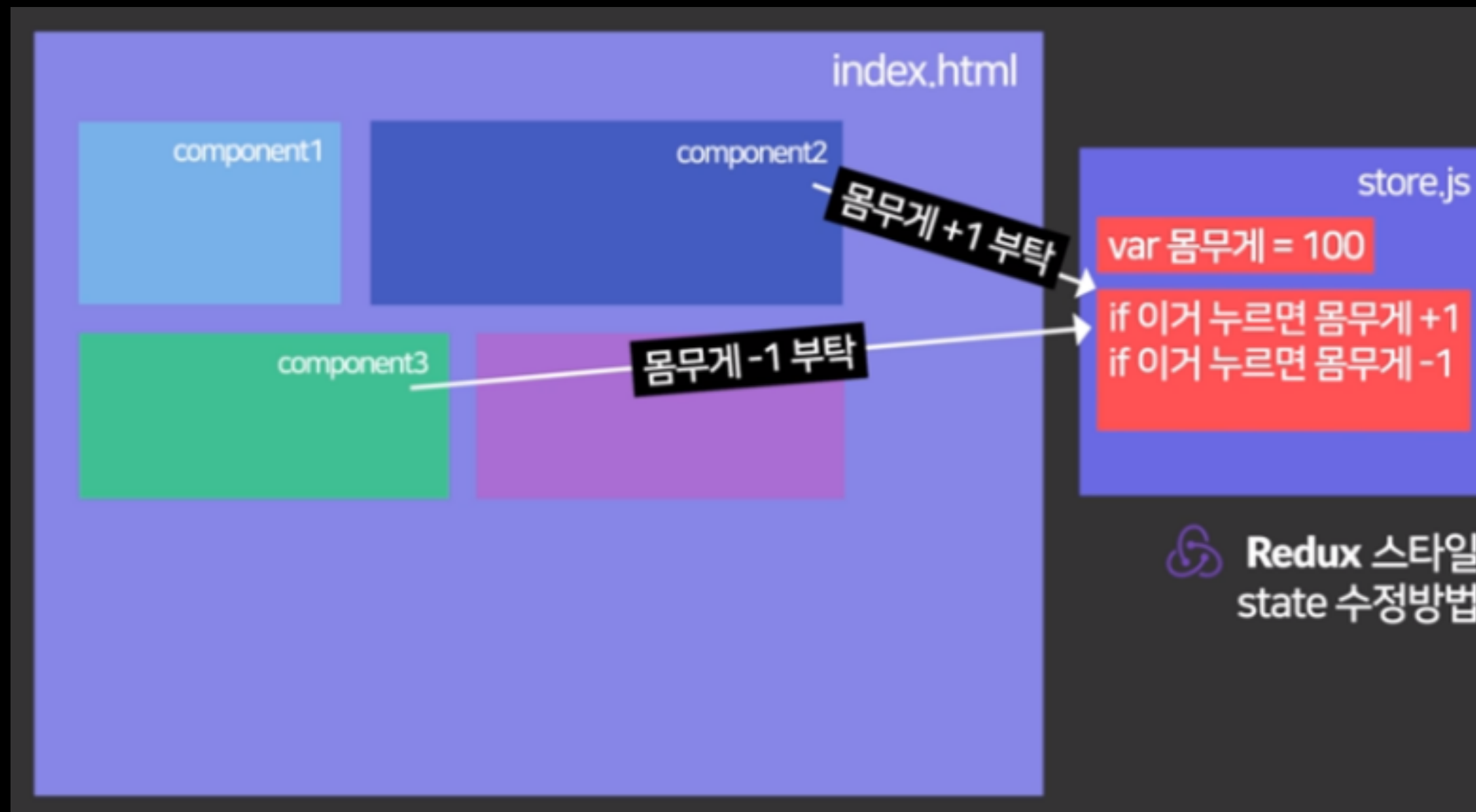


의심가는 컴포넌트부터 모두 조사해야 함



## 2. state 변경 관리가 용이

Redux 사용한다면?



state 수정은 store.js에서만  
버그가 발생해도 추적이 쉬움

state 관리가 편하다

규모가 큰 프로젝트에서는  
거의 반 필수?





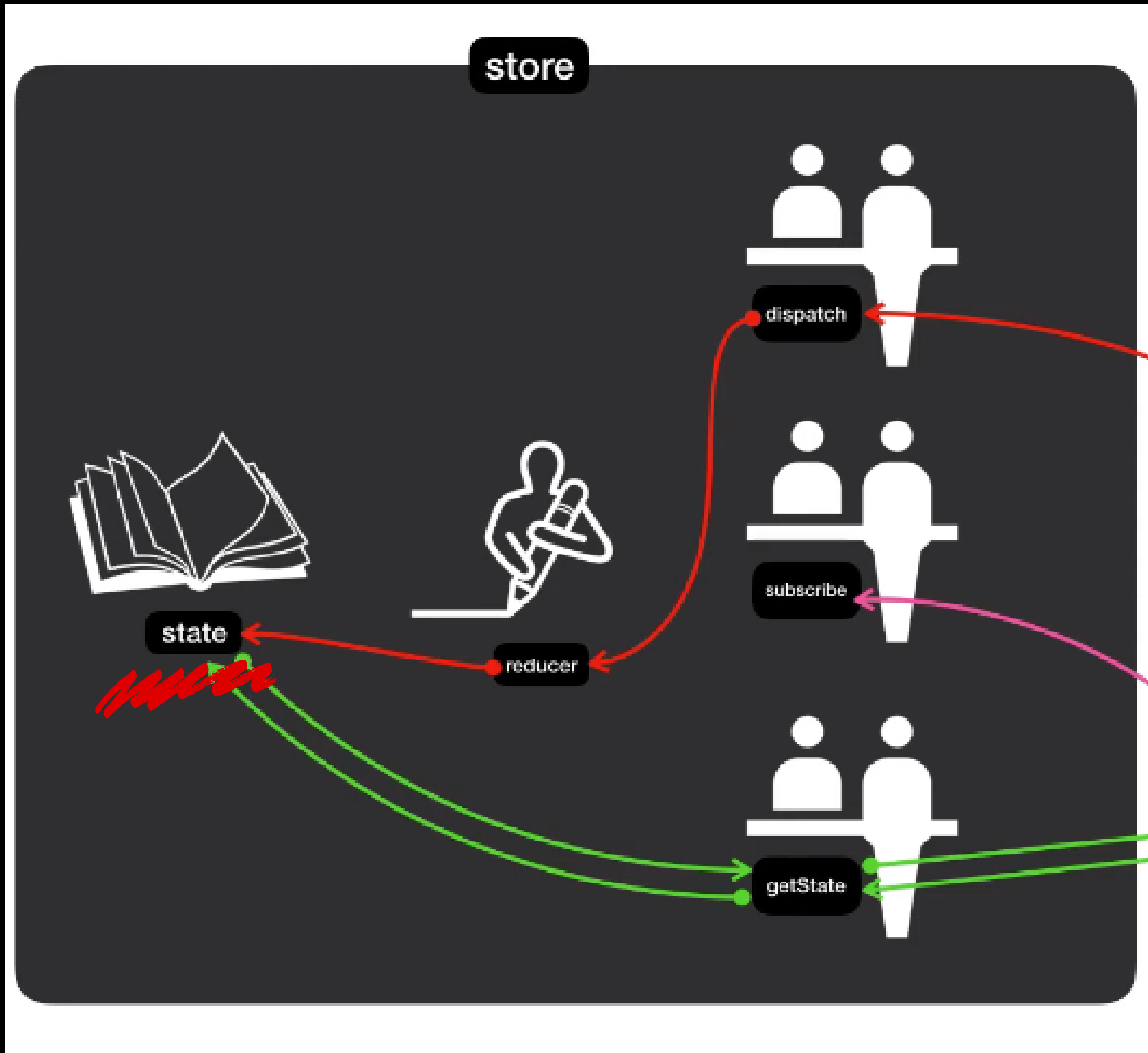
```
import { createStore } from 'redux';
import rootReducer from './reducers';
```

```
const store = createStore(rootReducer);
```



# Redux 핵심 개념 - state

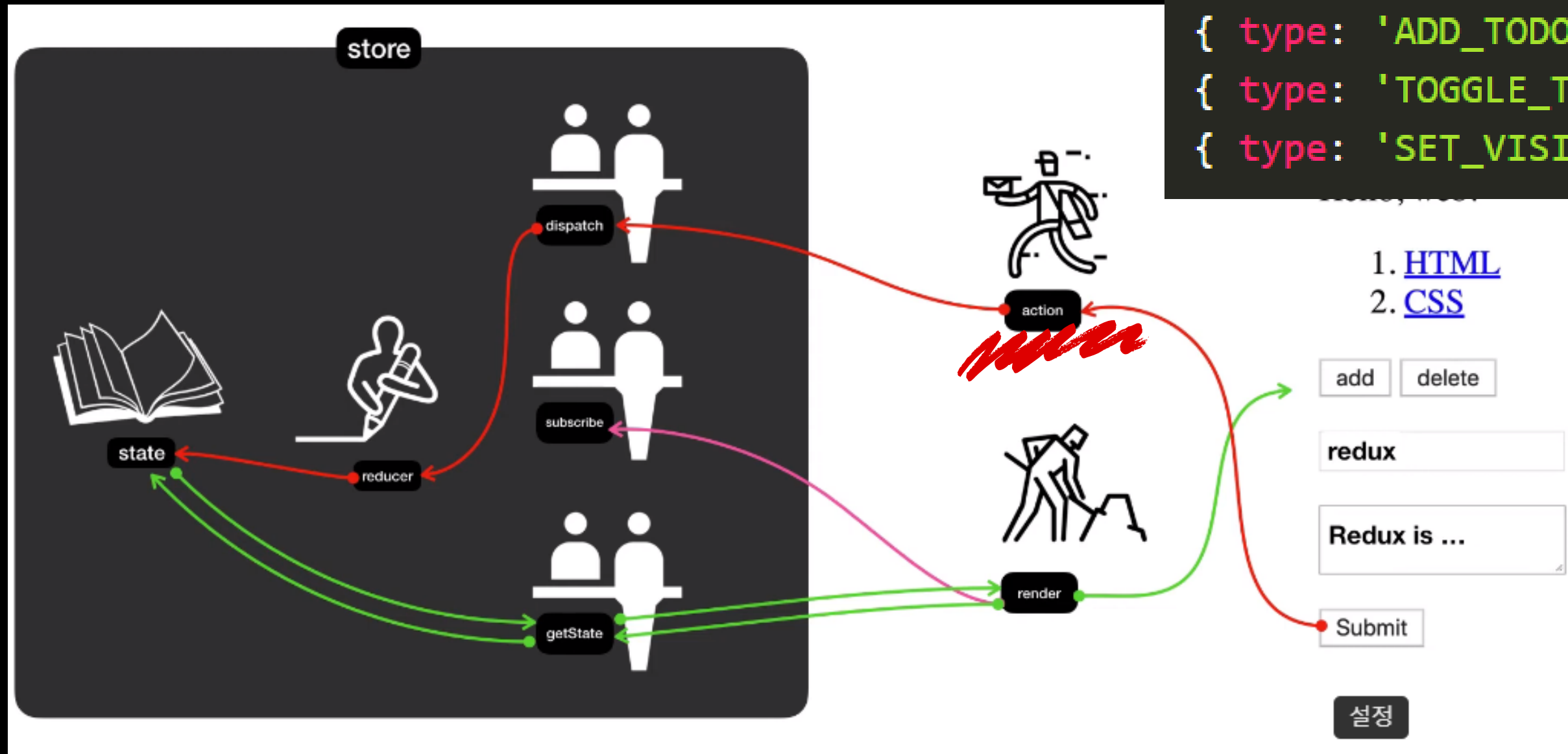
외부에서 state에 직접 접근은 불가



```
{
  todos: [{
    text: 'Eat food',
    completed: true
  }, {
    text: 'Exercise',
    completed: false
  }],
  visibilityFilter: 'SHOW_COMPLETED'
}
```



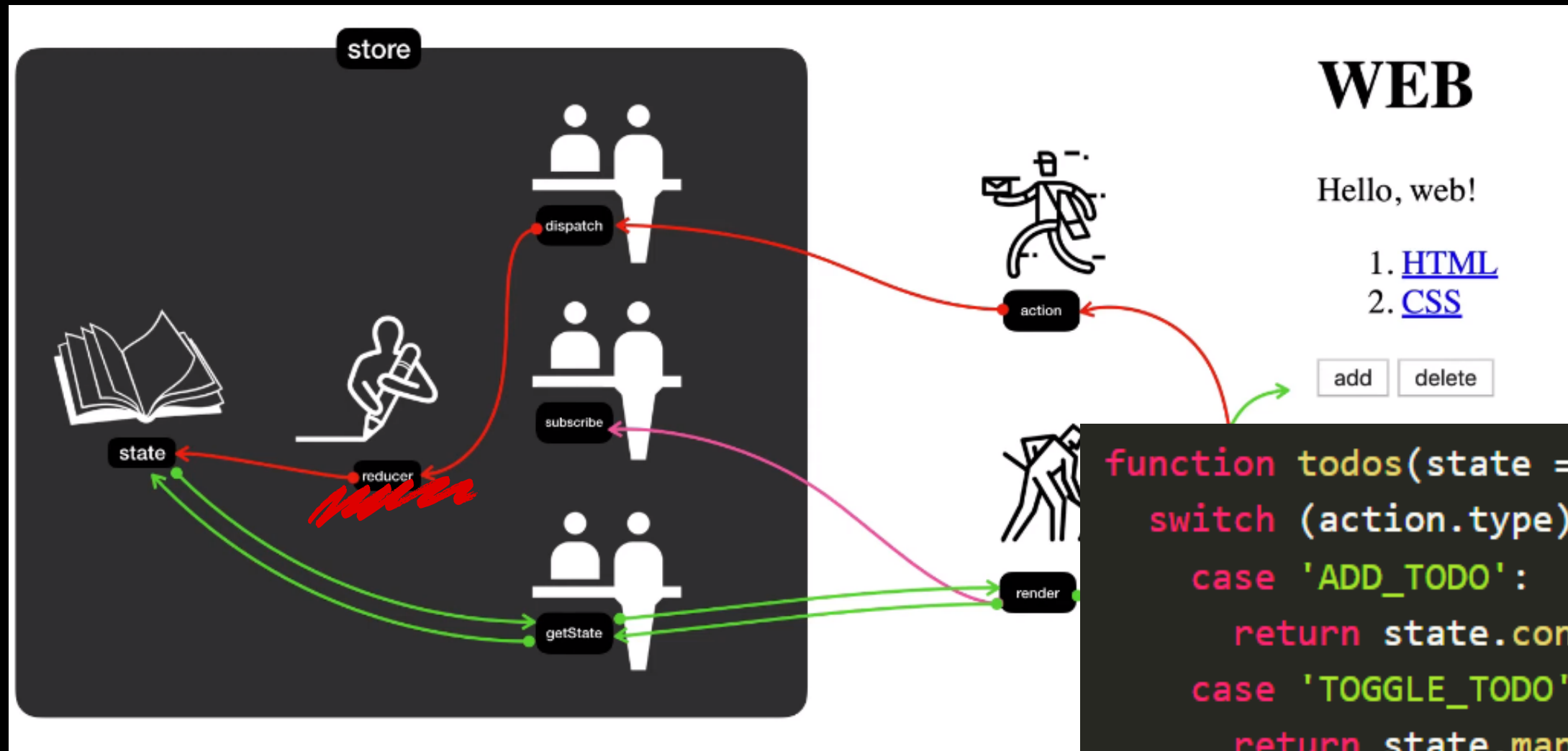
# Redux 핵심 개념 - action



state 가 업데이트될 때, 어떻게 업데이트 할지를 정의해주는 객체  
어플리케이션에서 일어나는 모든 사건들 중에서 상태 변화가 필요한 사건들



# Redux 핵심 개념 - reducer



```
function todos(state = [], action) {  
  switch (action.type) {  
    case 'ADD_TODO':  
      return state.concat([{ text: action.text, completed: false }])  
    case 'TOGGLE_TODO':  
      return state.map((todo, index) =>  
        action.index === index  
          ? { text: todo.text, completed: !todo.completed }  
          : todo  
      )  
    default:  
      return state  
  }  
}
```

현재 state, action



새로운 state 리턴



# Redux 핵심 개념 - useSelector

```
import React from 'react'
import { useSelector } from 'react-redux'
import TodoListItem from './TodoListItem'

const selectTodos = state => state.todos

const TodoList = () => {
  const todos = useSelector(selectTodos)

  // since `todos` is an array, we can loop over it
  const renderedListItems = todos.map(todo => {
    return <TodoListItem key={todo.id} todo={todo} />
  })

  return <ul className="todo-list">{renderedListItems}</ul>
}

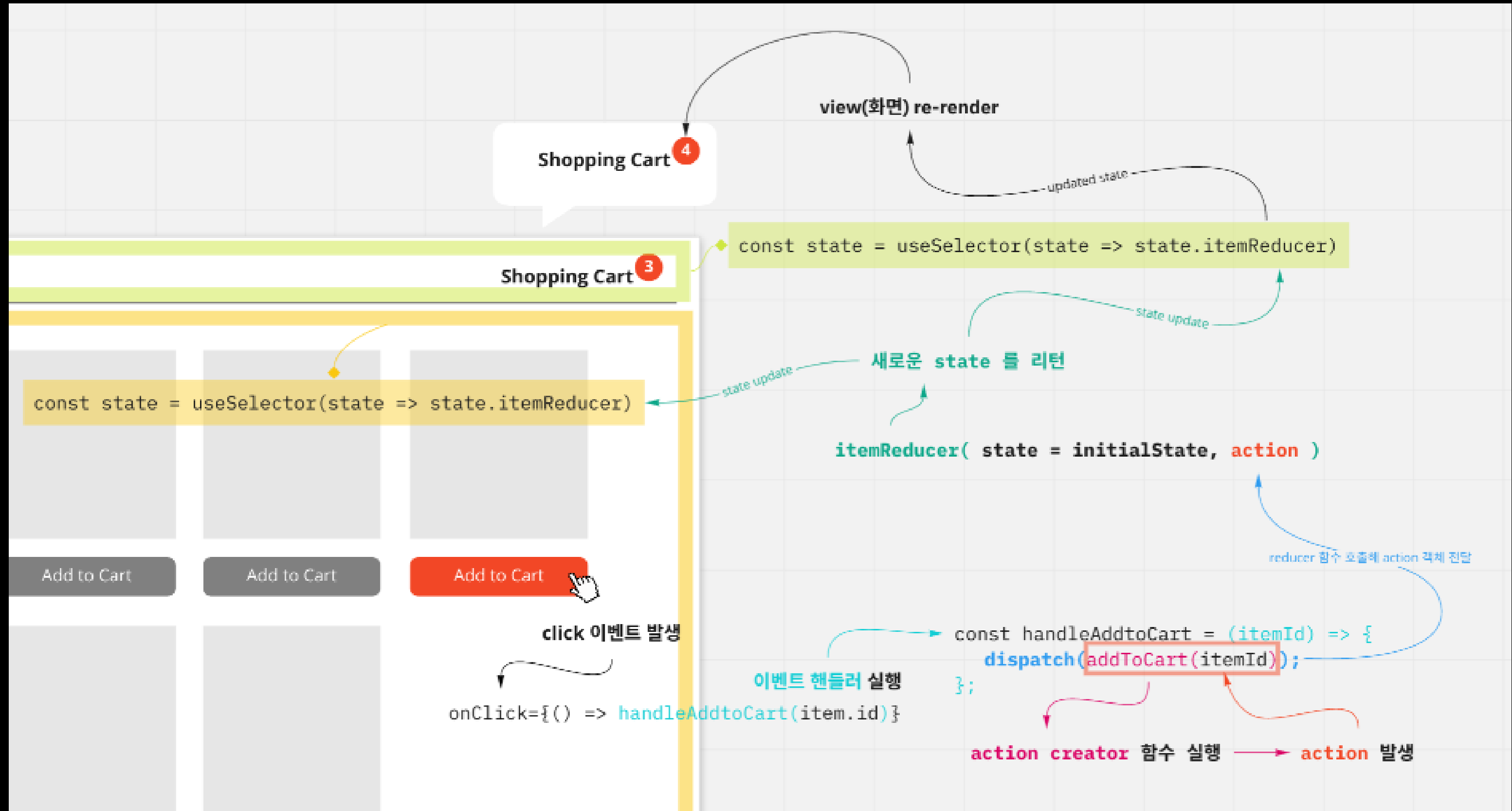
export default TodoList
```

useSelector로 state 값 가져옴

reducer로 state가 변경되면  
자동으로 리렌더링



# 쇼핑몰 application 예시





## 참고 링크

<https://ko.redux.js.org/tutorials/fundamentals/part-1-overview>

<https://velog.io/@seungsang00/React-Redux-%EA%B8%B0%EC%B4%88#store%EC%8A%A4%ED%86%A0%EC%96%B4>

<https://www.youtube.com/watch?v=QZcYz2NrDIs>