

08 / 11 수요일 이론 강의

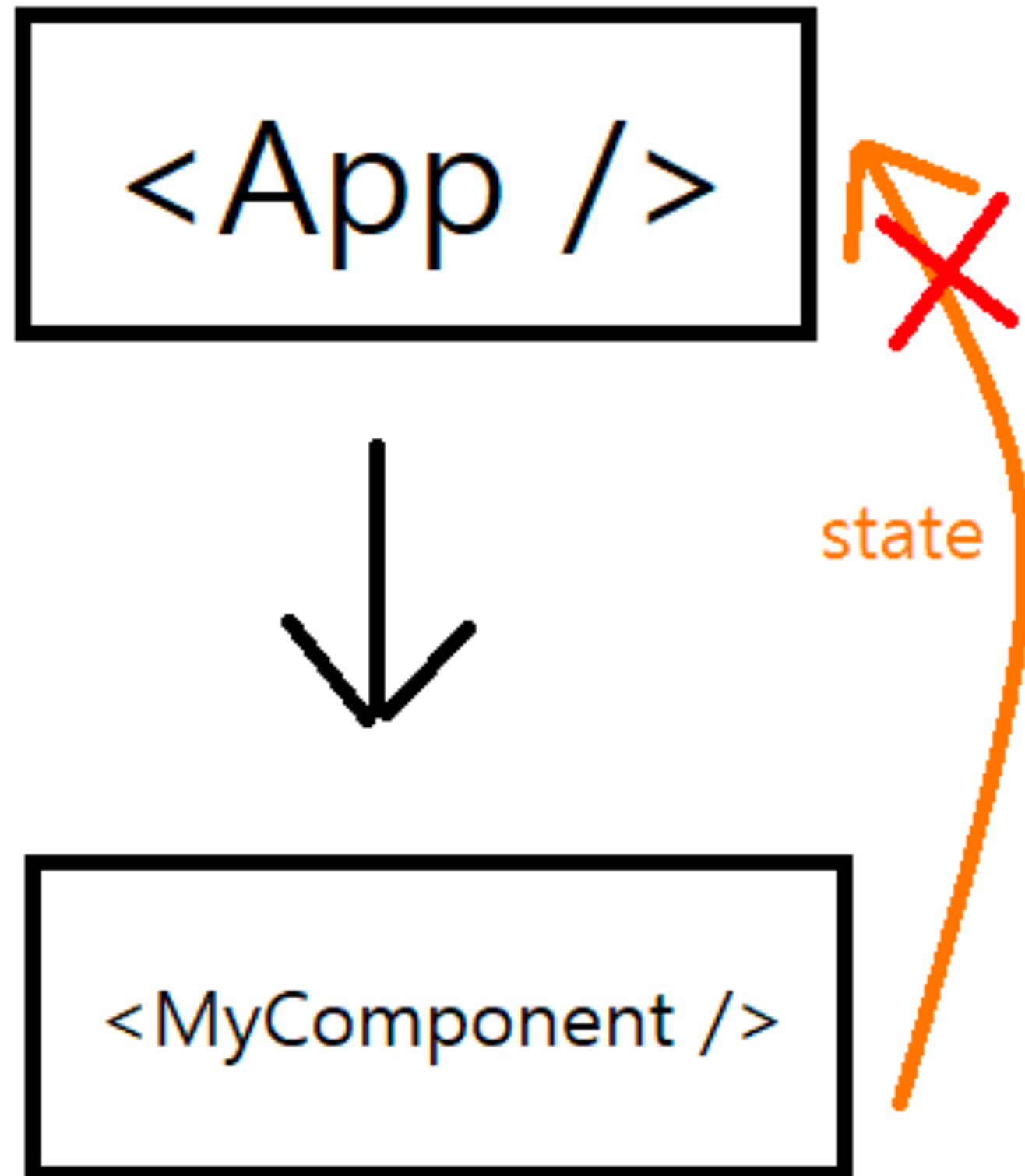


기존의 상태 관리 방식

- 함수 컴포넌트는 useState 사용
- 클래스 컴포넌트는 this.state 사용

이로 인한 문제점

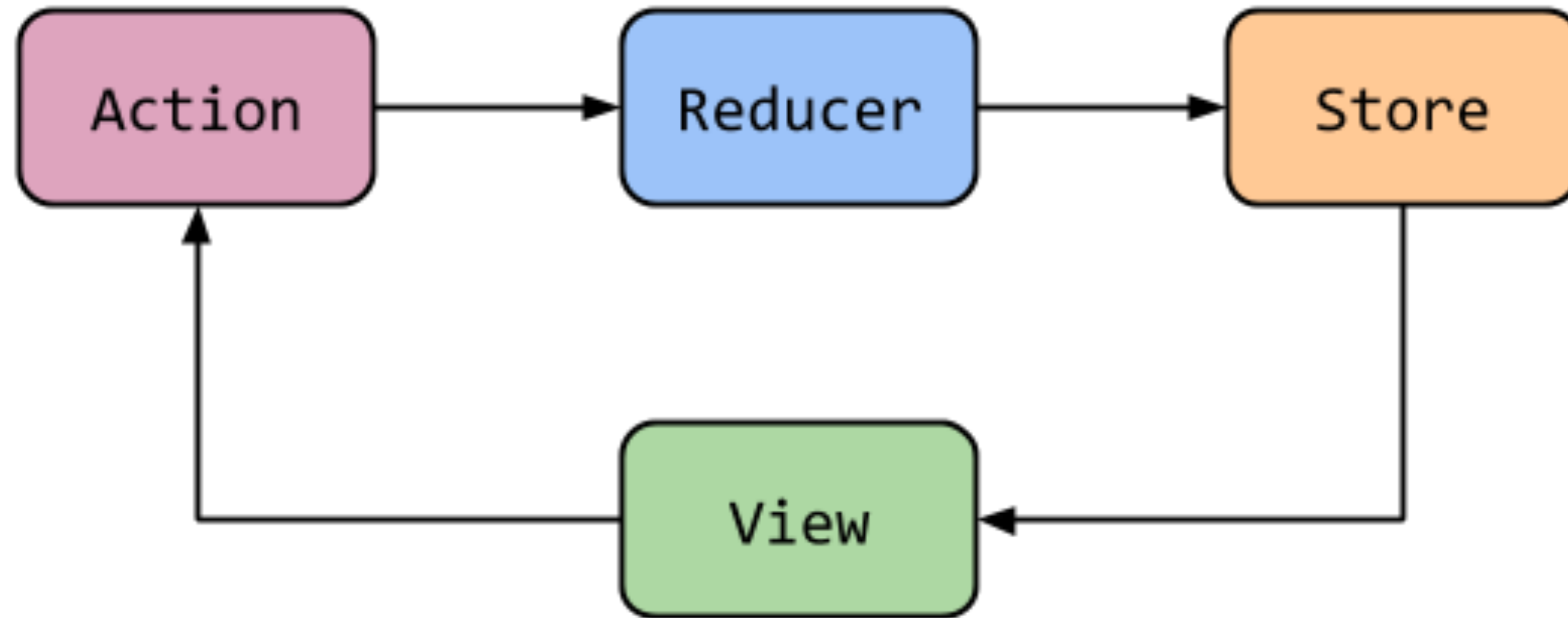
- 상태관리는 컴포넌트 내에서만 존재함
 - 자식 컴포넌트의 state의 값을 부모 컴포넌트는 알 수 없음



상태 관리를 전역으로 하게되면 문제 해결

Redux란?

- 전역 상태 관리를 위한 라이브러리



컴포넌트 => 액션 (값 전달) => 리듀서 (값 처리) => 스토어 (값 저장) => 컴포넌트

예시

```
const initState = {  
  객체 값  
}  
  
const mainReducer = (state = initState, action)  
=> {  
  switch(action.type) {  
    case [전달 받은 타입]:  
      // 변경  
      return 적용 값;  
  }  
}
```

Reducer 작성요령

- 초기값 설정
- 값을 처리하는 리듀서 정의

예시

```
const 액션이름 = (전달 받은 값) => ({  
  type: "리듀서로 전달할 액션 이름",  
  전달 받은 값  
});
```

Action 작성요령

- 액션 이름
- 전달 받은 데이터로 객체 만드는 함수 정의

예시

```
import { createStore } from "redux";  
  
const store = createStore(리듀서)  
  
...  
  
import { Provider } from "react-redux";  
function App() {  
  return (  
    <Provider store={store}>  
      ...  
    </Provider>  
  )  
}
```

store 작성

- createStore에 리듀서를 넣기
- 최상위 컴포넌트에 Provider 적용

Recoil이란?

- 상태관리 라이브러리는 Redux만 있는게 아닌 여러 라이브러리가 존재
- Recoil은 페이스북에서 내놓은 전역 상태관리 라이브러리
- reducer, store, action 방식이 아닌 atom이라고 부르는 하나의 객체만 사용
- useState와 비슷하게 사용 가능

Recoil 단점

- 엄청 오래된 라이브러리가 아니기 때문에 안정성 보장 X