

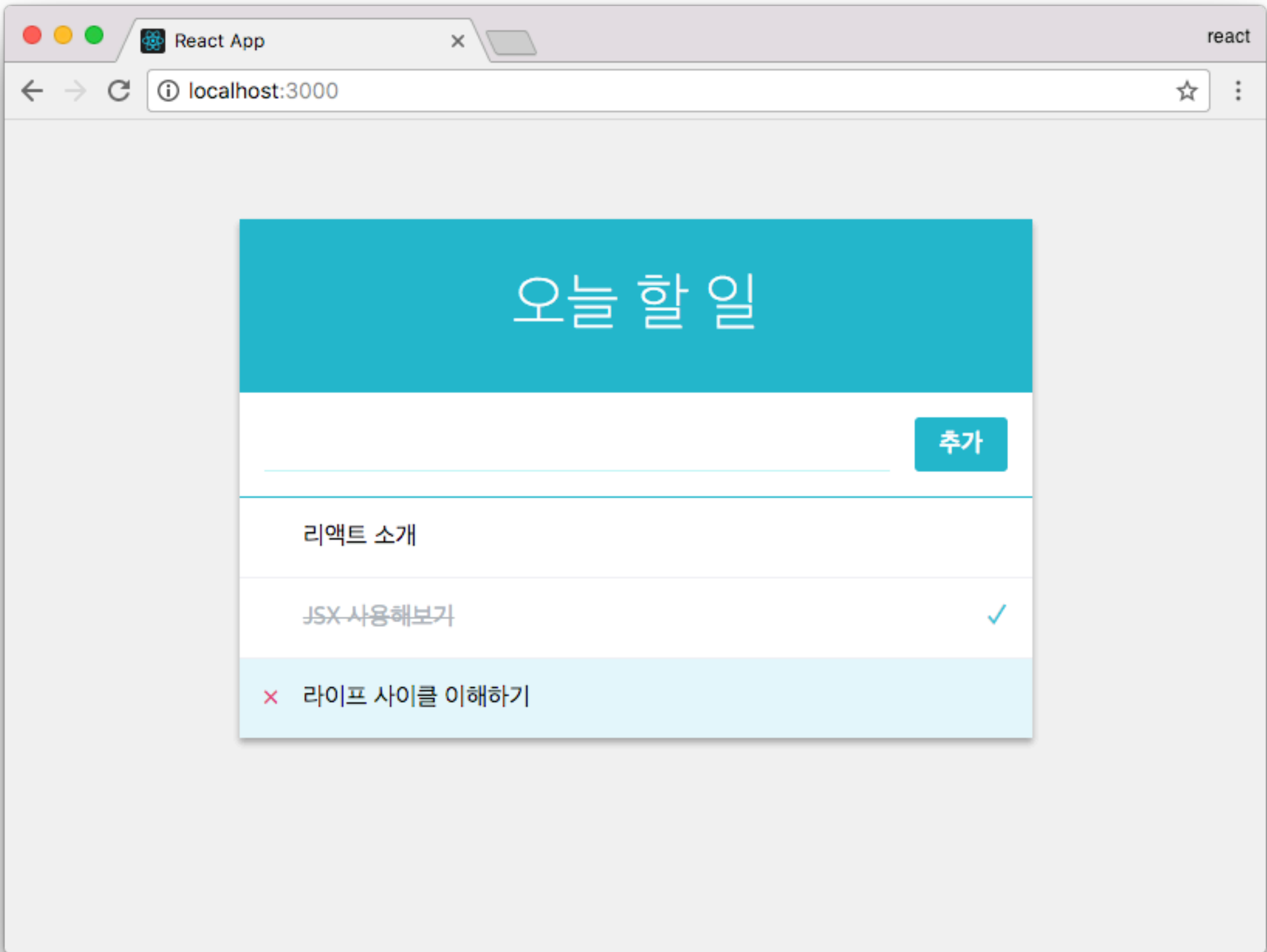
리액트 상태관리의 꽃 리덕스

왜 쓰는가?

복잡한 상태관리

컴포넌트

state



App
{ input, todos }

자식들은 부모를 통해서만 대화한다

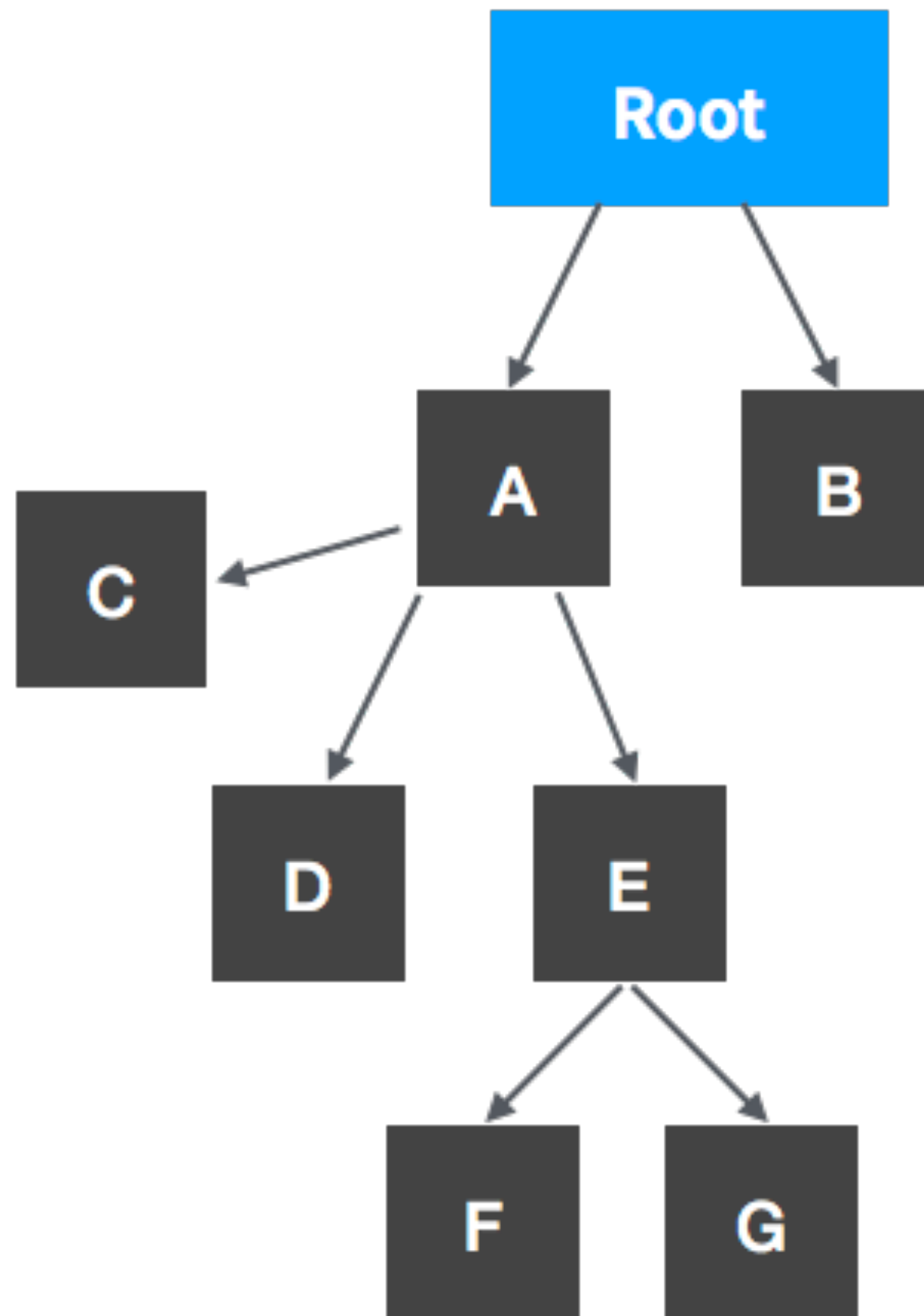
인풋 값과, 변경시키는 함수를
props 로 전달

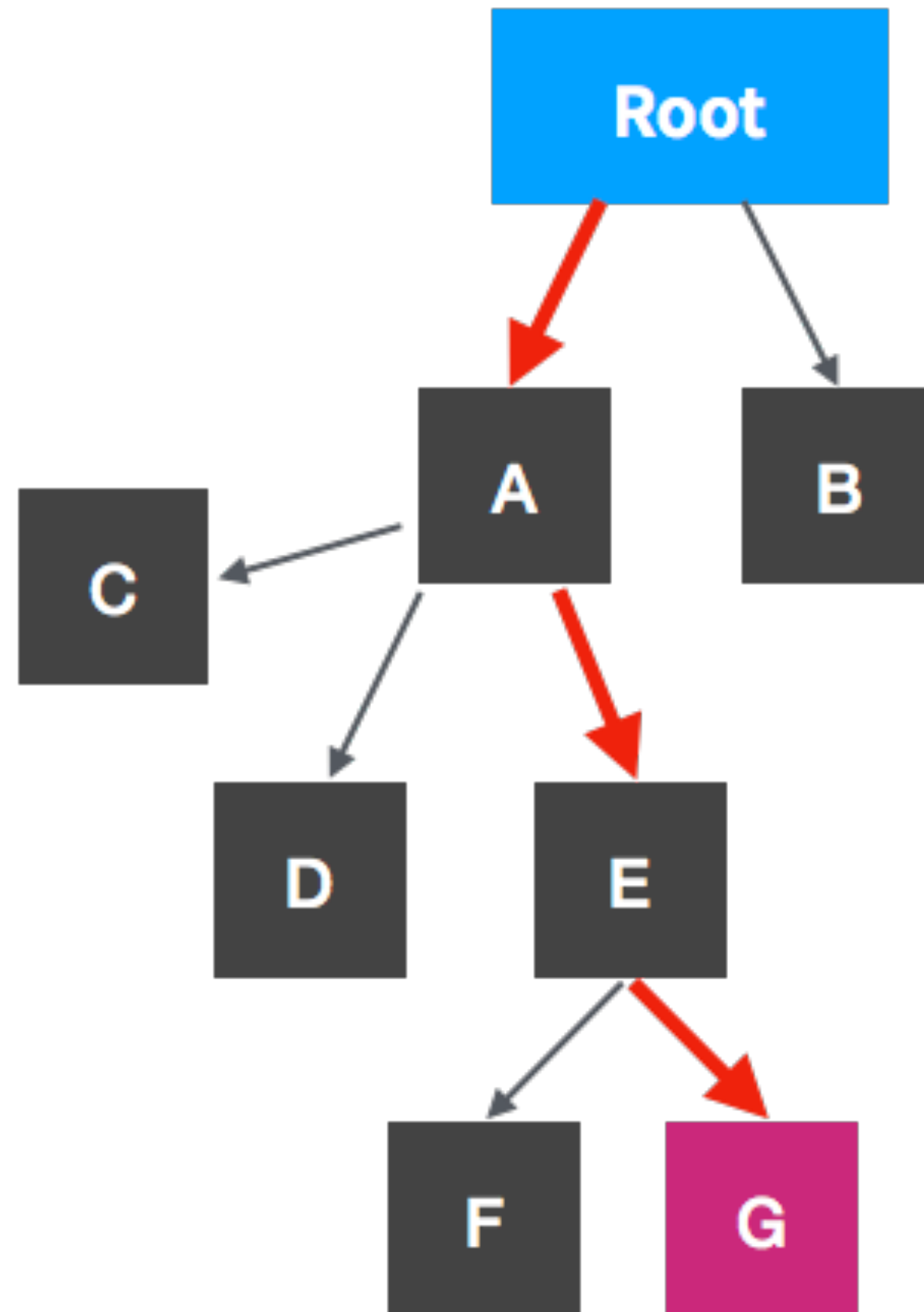
todos 목록과
변경시키는 함수를 props 로 전달

Form

TodoItemList

내부 컴포넌트 끼리는
대화하지 않는다!





```
// App.js 에서 A 렌더링
```

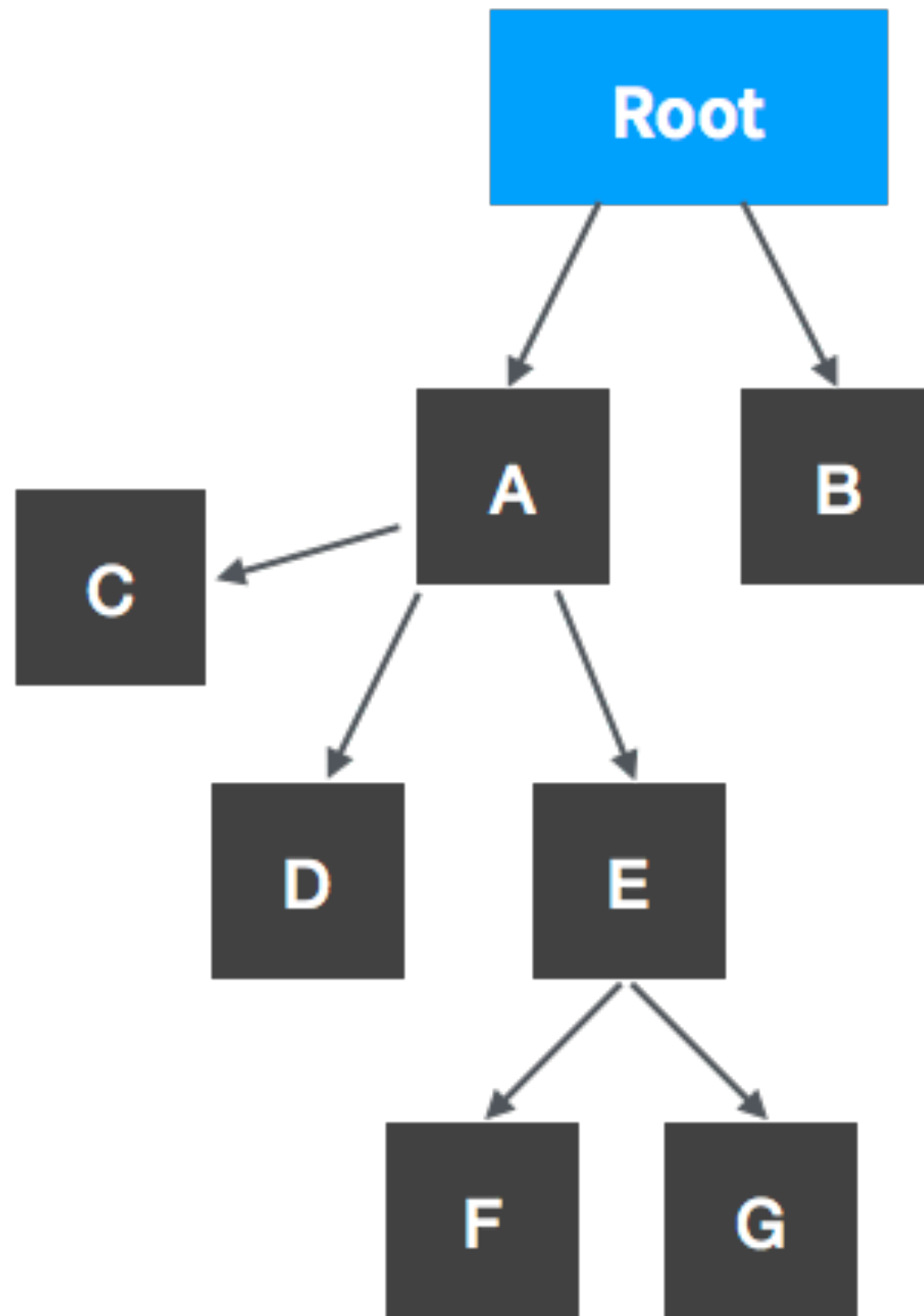
```
<A value={5}>
```

```
// A.js 에서 E 렌더링
```

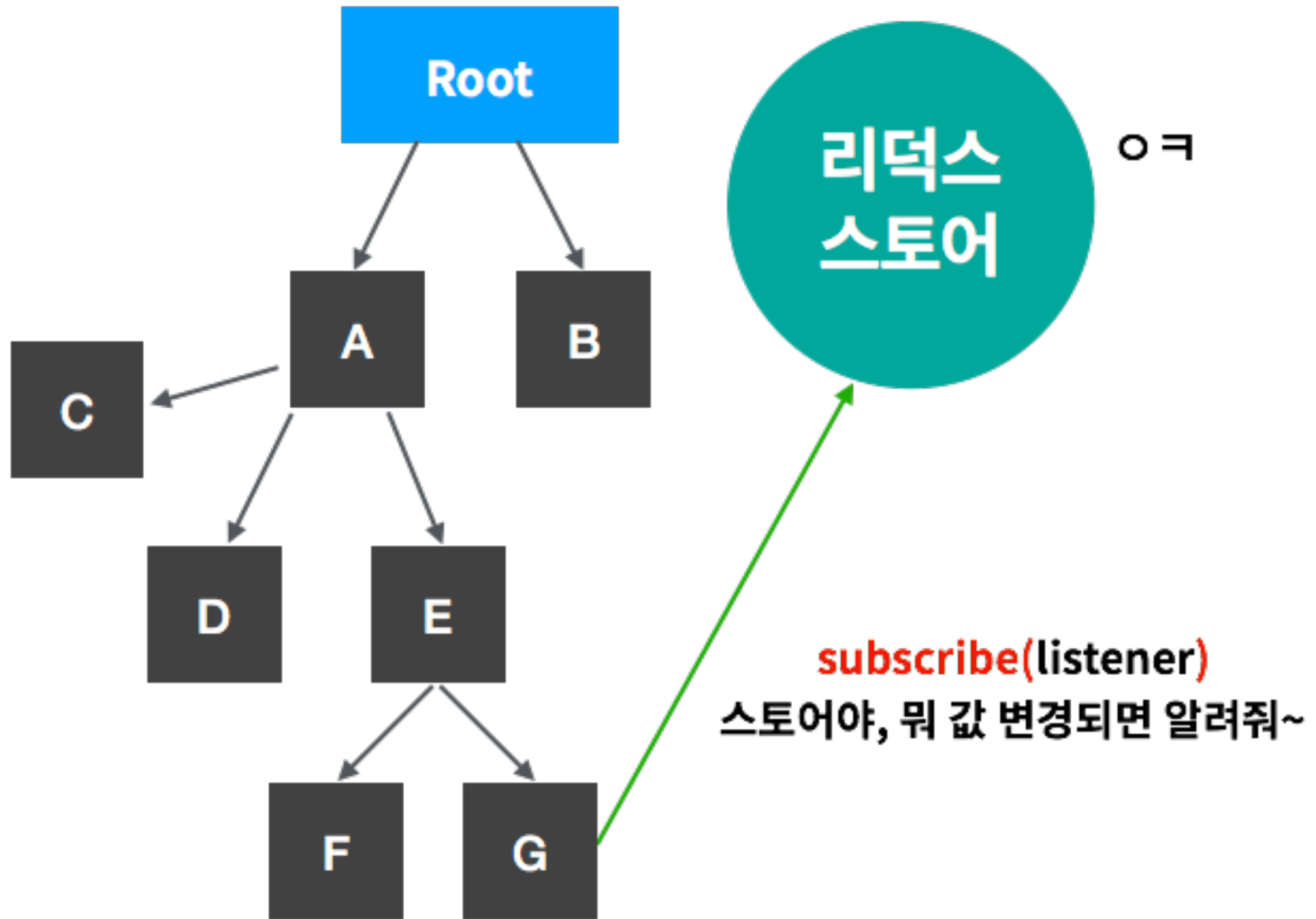
```
<E value={this.props.value} />
```

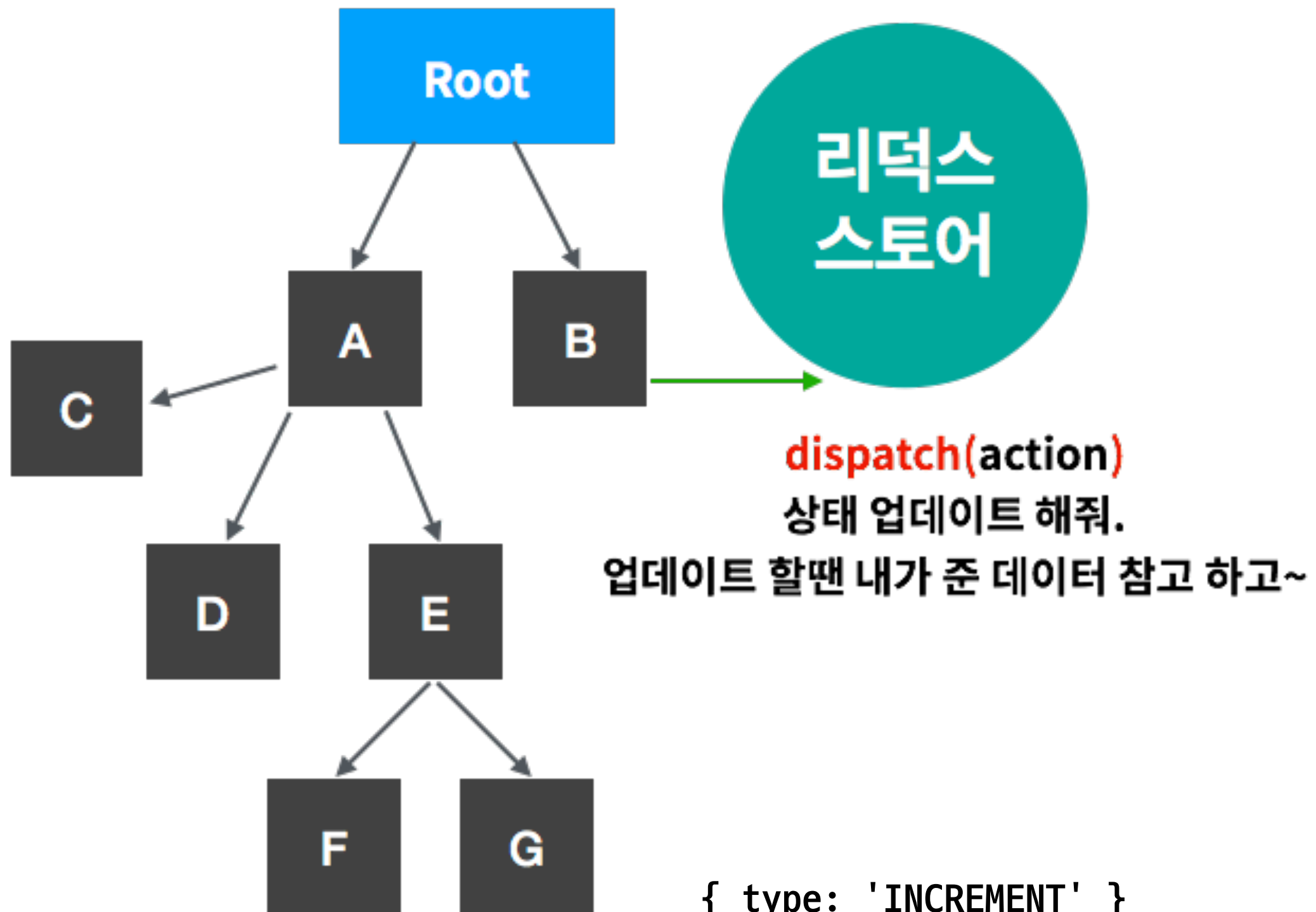
```
// B.js 에서 G 렌더링
```

```
<G value={this.props.value} />
```

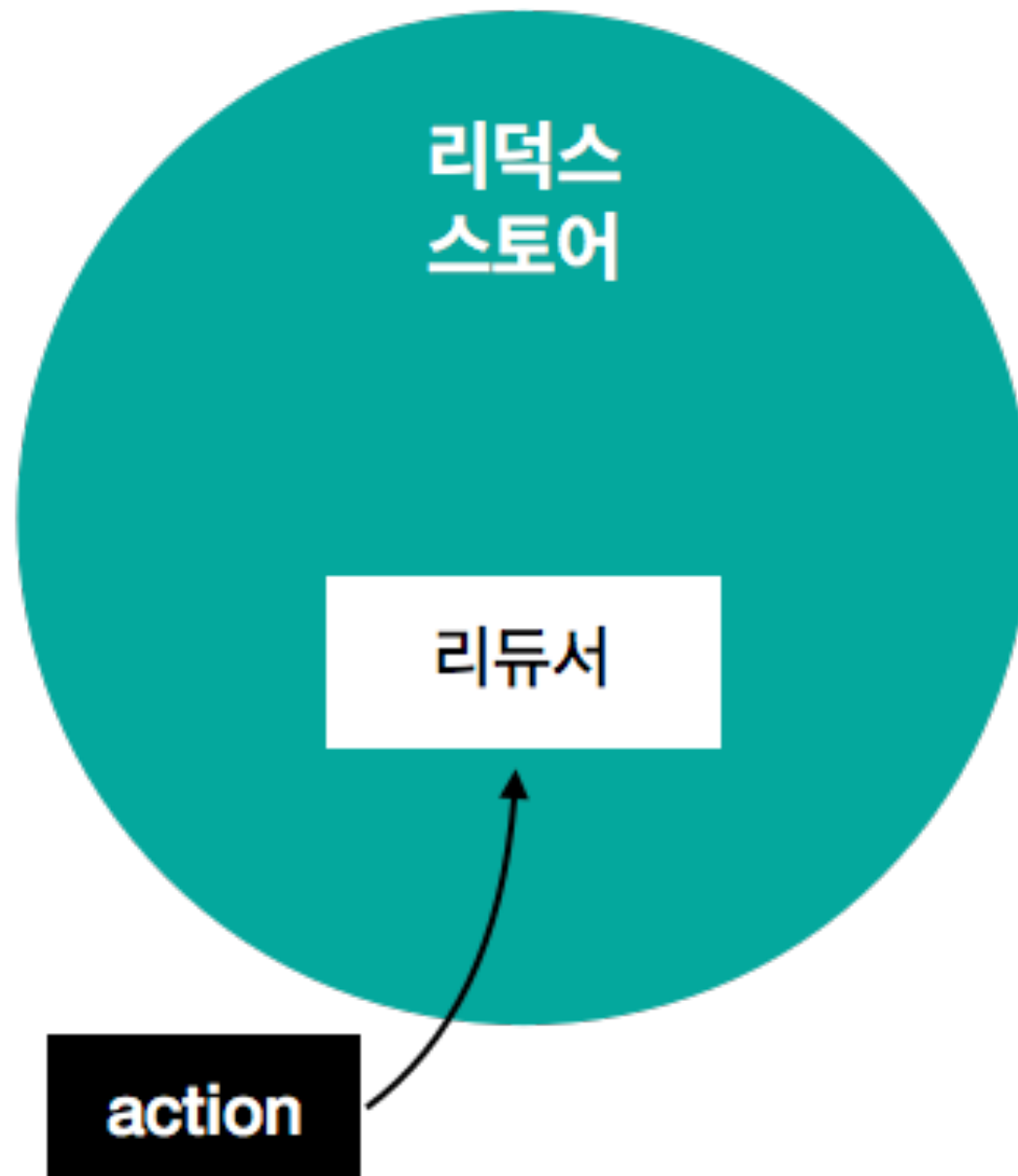
B 에서 호출된 것이
G 에 영향을 끼치는 상황



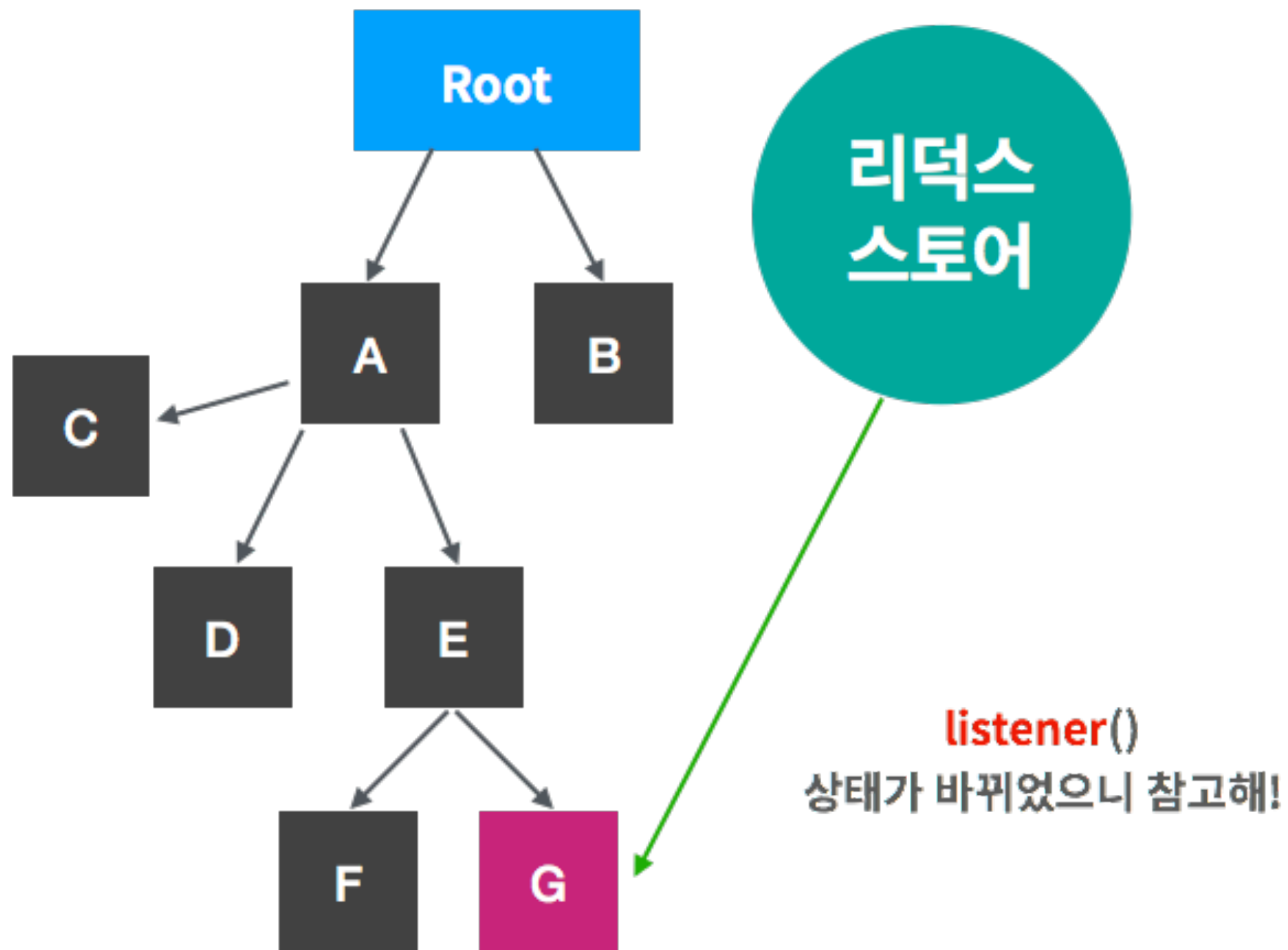


```
{ type: 'INCREMENT' }
```

```
{ type: 'INCREMENT', diff: 2 }
```



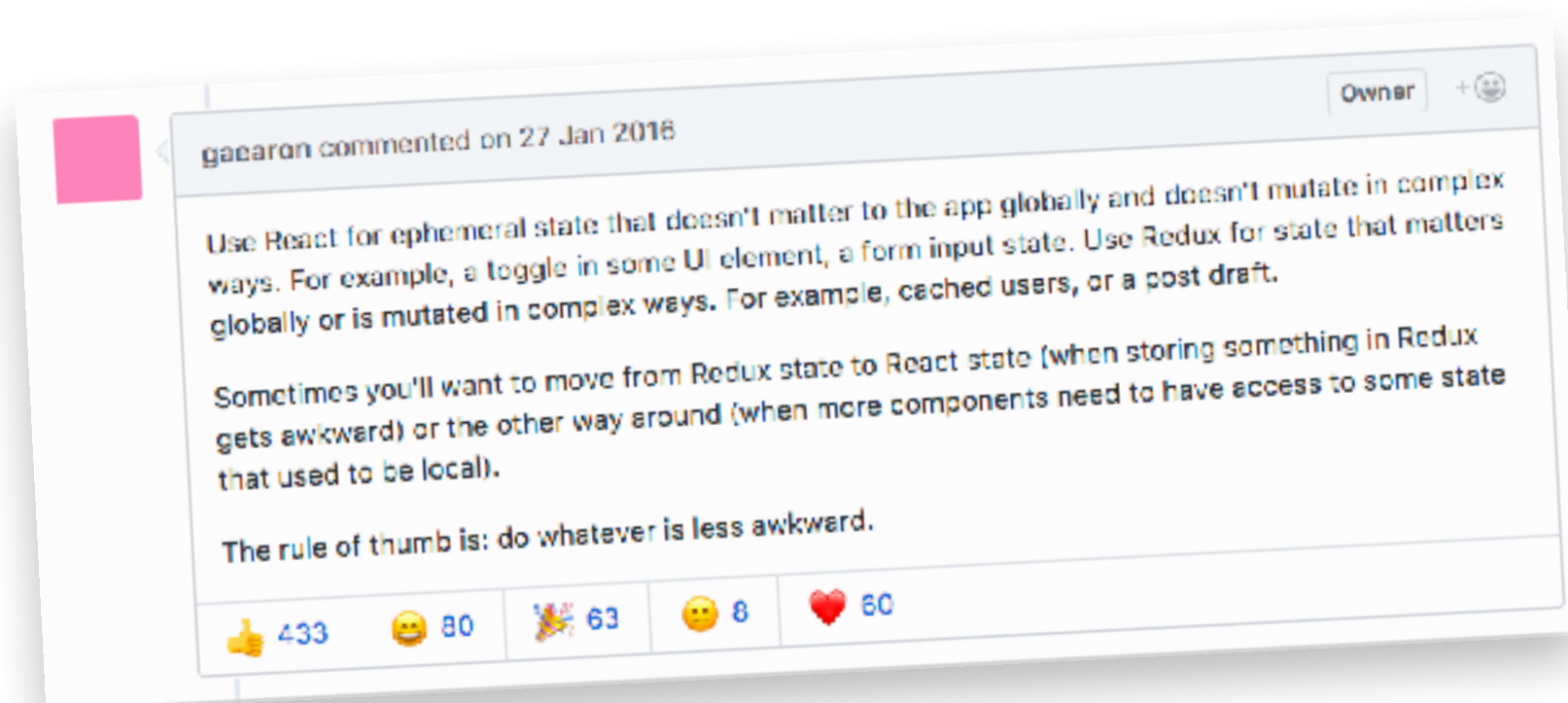
```
function reducer(state, action) {  
  return updatedState;  
}
```



오우, 새로운 상태? 그러면 리렌더링 할게



Redux 를 꼭 사용해야 할까?
state 는 사용하지 말아야 할까?



라프텔 랭킹 순 ▼



라프텔 랭킹 순 ▼

라프텔 랭킹 순

최신 순

평가 많은 순

평점 높은 순

컴포넌트 내부에서만 필요할 때
= state 만으로 충분