



리액트와 함께 장고 시작하기 / 리액트

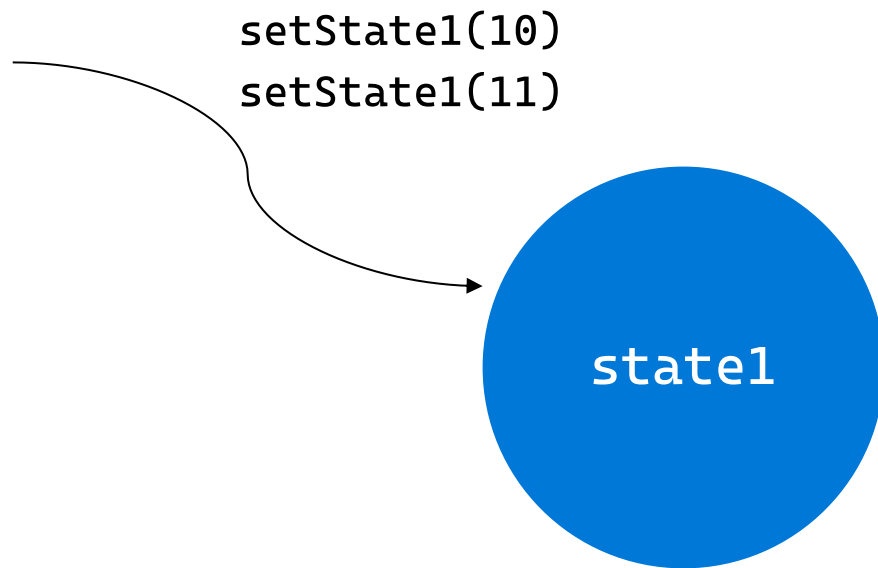
리듀서와 useReducer 훅

여러분의 파이썬/장고 페이스메이커가 되겠습니다.

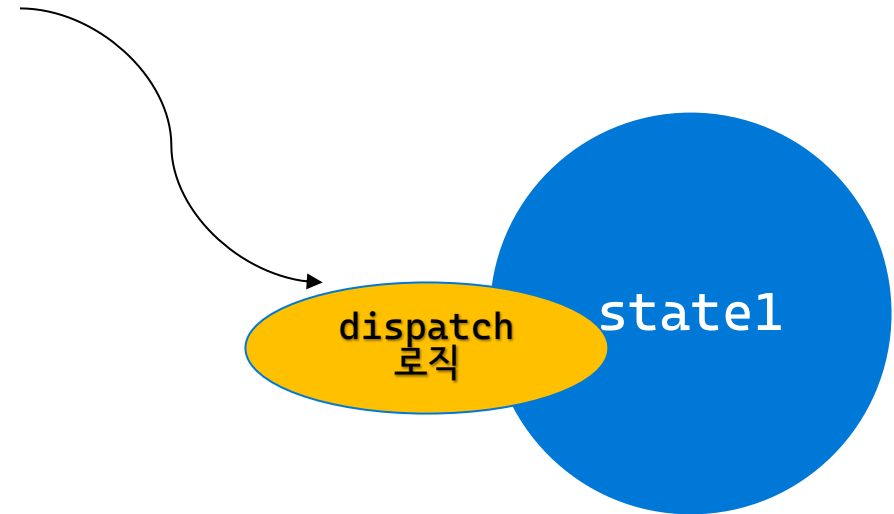
리액트에서 상태값을 변경하는 2가지 방법

1. setter 함수를 직접 제공 → setter 로직이 다수 혼재
2. setter 함수를 제공하지 않고, dispatch 함수를 제공 → Redux에서의 방식

setter 로직을 dispatch를 통해서만 처리



```
dispatch({ type: 'INCREMENT', payload: 1 })  
dispatch({ type: 'INCREMENT', payload: 1 })
```



useReducer 훅

컴포넌트의 상태값을 리덕스의 리듀서처럼 관리 가능

```
const INITIAL_STATE = { name: 'Tom', age: 10 };
```

```
const reducer = (state, action) => {  
  switch ( action.type ) {  
    case 'setName': return { ...state, name: action.name };  
    case 'setAge': return { ...state, age: action.age };  
    default: return state;  
  }  
};
```

```
const Person = () => {  
  const [state, dispatch] = React.useReducer(reducer, INITIAL_STATE);  
  return (  
    <div>  
      <p>name : {state.name}</p>  
      <p>age: {state.age}</p>  
      <input type="text" value={state.name} onChange={e => dispatch({ type: 'setName', name: e.currentTarget.value })} />  
      <input type="text" value={state.age} onChange={e => dispatch({ type: 'setAge', age: e.currentTarget.value })} />  
    </div>  
  );  
};
```

Context API를 통해,
깊은 트리의 자식 컴포넌트에
dispatch 전달을 할 수 있겠다.

참고) Redux에서의 dispatch 함수 생성

```
import { handleActions } from "redux-actions";

const SET_NAME = 'setName';
const SET_AGE = 'setAge';

const actions = handleActions({
  [SET_NAME]: (state, { payload: name }) =>
    produce(state, (draft) => {
      draft.name = name;
    }),
  [SET_AGE]: (state, { payload: age }) => {
    produce(state, (draft) => {
      draft.age = age;
    });
  },
});
```

Life is short.
You need Python and Django.

I will be your pacemaker.

