

1. <글 발행> 버튼에 대한 event handler

글 발행 버튼을 누르면 title array의 복사본이 형성된다. 바뀐 title array 복사본으로 title 상태 변수가 업데이트되어 title 리스트의 상단에 새로운 입력 내용이 추가된다.

- `onClick={() => {...}}` : 버튼을 눌렀을 때 실행되는 event handler
- `let copy = [...title];` : `[...title]`을 이용하여 title array의 복사본 만들기
- `copy.unshift(inputValue);` : copy array에 inputValue 붙이기 => 리스트의 상단에 새로운 제목 추가하기
- `setTitle(copy);` : 바뀐 copy array로 title 상태 변수 업데이트하기

2. 👍 이모티콘을 누르면 좋아요 수가 1씩 증가

각 제목의 👍 버튼을 클릭하면 좋아요 수가 1씩 증가한다. `colorIndex` 상태 변수 때문에 제목을 누르면 제목의 색깔이 변한다. 제목 아래엔 날짜를 보여준다.

3. modal 상태 변수에 따라 <Modal> 컴포넌트를 보여줄지 결정하는 conditional rendering statement

- `{modal == true? ... : null}` : modal 상태 변수가 true인지 확인한다. True인 경우, 첫번째 괄호 안 (...)이 실행되고, 그렇지 않은 경우 null값이 반환된다.
- `<Modal title={title} changeTitle={changeTitle}></Modal>` : modal 상태 변수가 true인 경우, <Modal> 컴포넌트를 title과 changeTitle로 렌더링한다.
- `: null` : modal 상태 변수가 true가 아닌 경우, 아무것도 렌더링되지 않는다.

4. Modal이라는 functional component는 props를 argument로 가지며 JSX를 반환한다.

- `function Modal(props) {...}` : Modal은 props를 argument로 가지고 props는 부모 컴포넌트로부터 Modal 컴포넌트로 데이터를 전달한다.
- `<div className = 'modal'> ... </div>` : className attribute는 CSS class 'modal'을 이 요소에 할당하여 CSS를 이용하여 이 modal의 스타일을 지정할 수 있도록 한다.
- `<h4>{props.title[props.title[props.changeTitle]]}</h4>` : title content를 modal에 display한다.