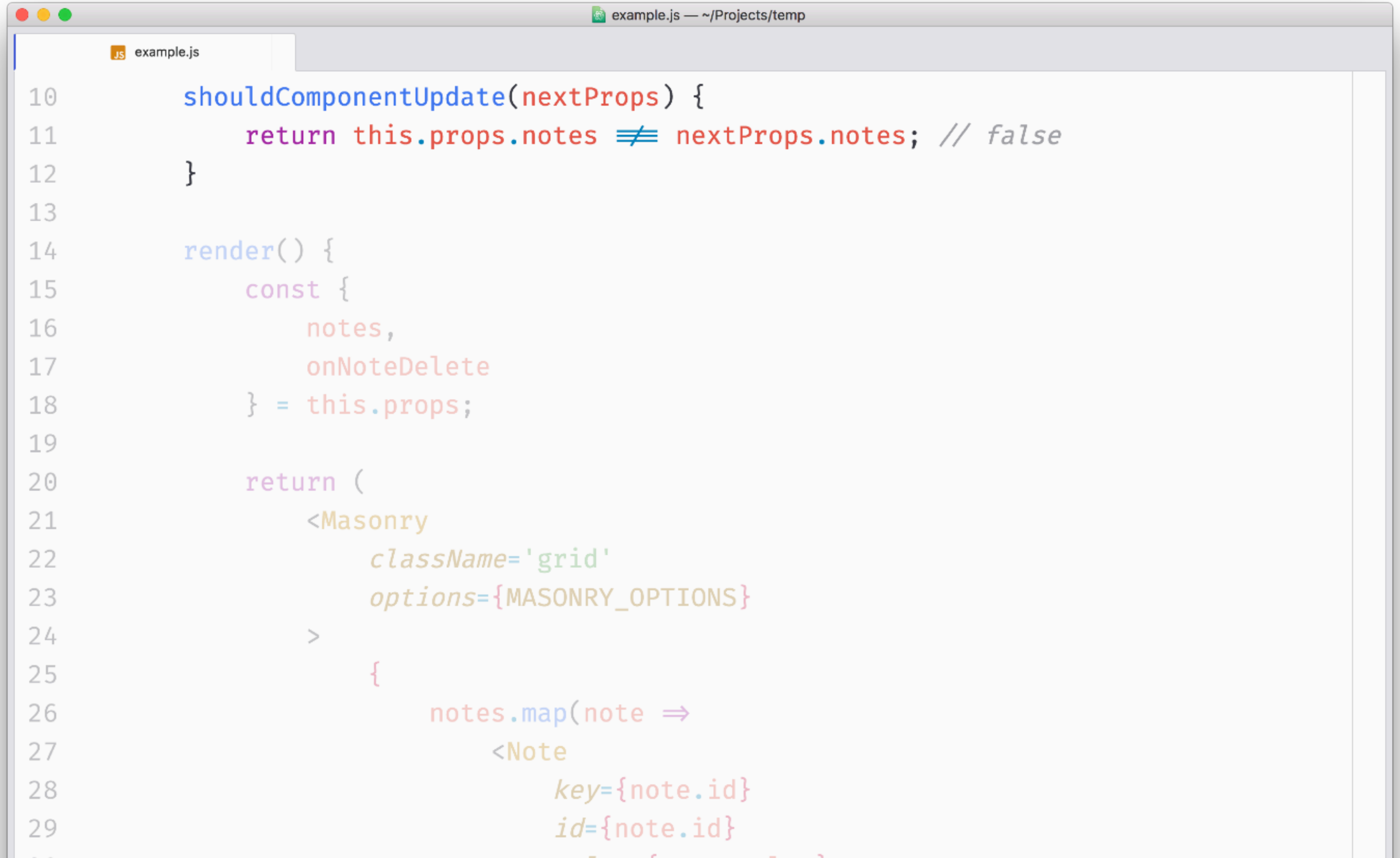
3. modification by reference



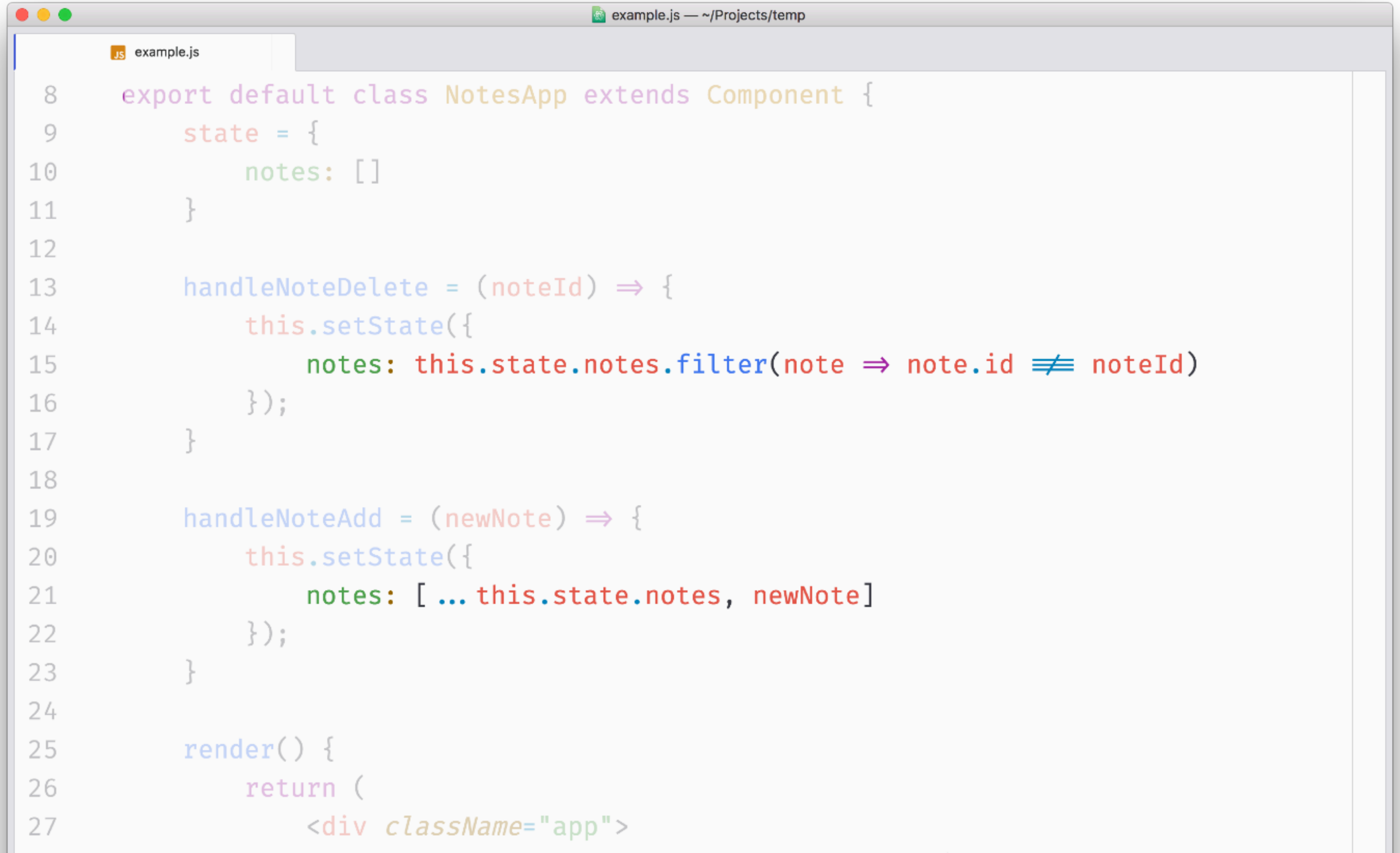
```
    shouldComponentUpdate(nextProps) {
        return this.props.notes !== nextProps.notes; // false
    }

    render() {
        const {
            notes,
            onNoteDelete
        } = this.props;

        return (
            <Masonry
                className='grid'
                options={MASONRY_OPTIONS}
            >
                {
                    notes.map(note =>
                        <Note
                            key={note.id}
                            id={note.id}
```

↑ so, shallow comparison produces 'false'

3. modification by reference



```
export default class NotesApp extends Component {
    state = {
        notes: []
    }

    handleNoteDelete = (noteId) => {
        this.setState({
            notes: this.state.notes.filter(note => note.id !== noteId)
        });
    }

    handleNoteAdd = (newNote) => {
        this.setState({
            notes: [ ...this.state.notes, newNote]
        });
    }

    render() {
        return (
            <div className="app">
```

use ES2015 spread operator or map, filter etc. methods