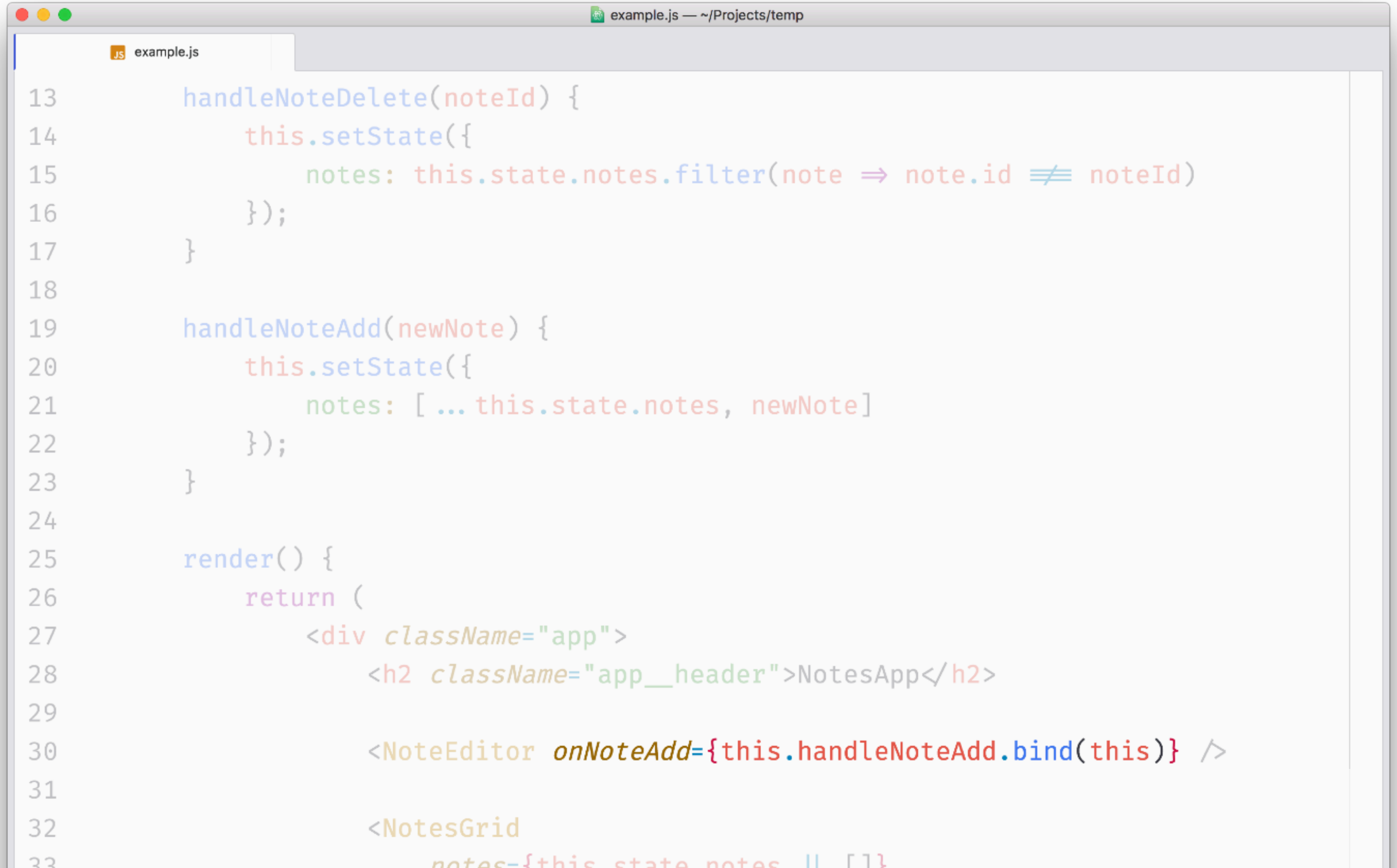
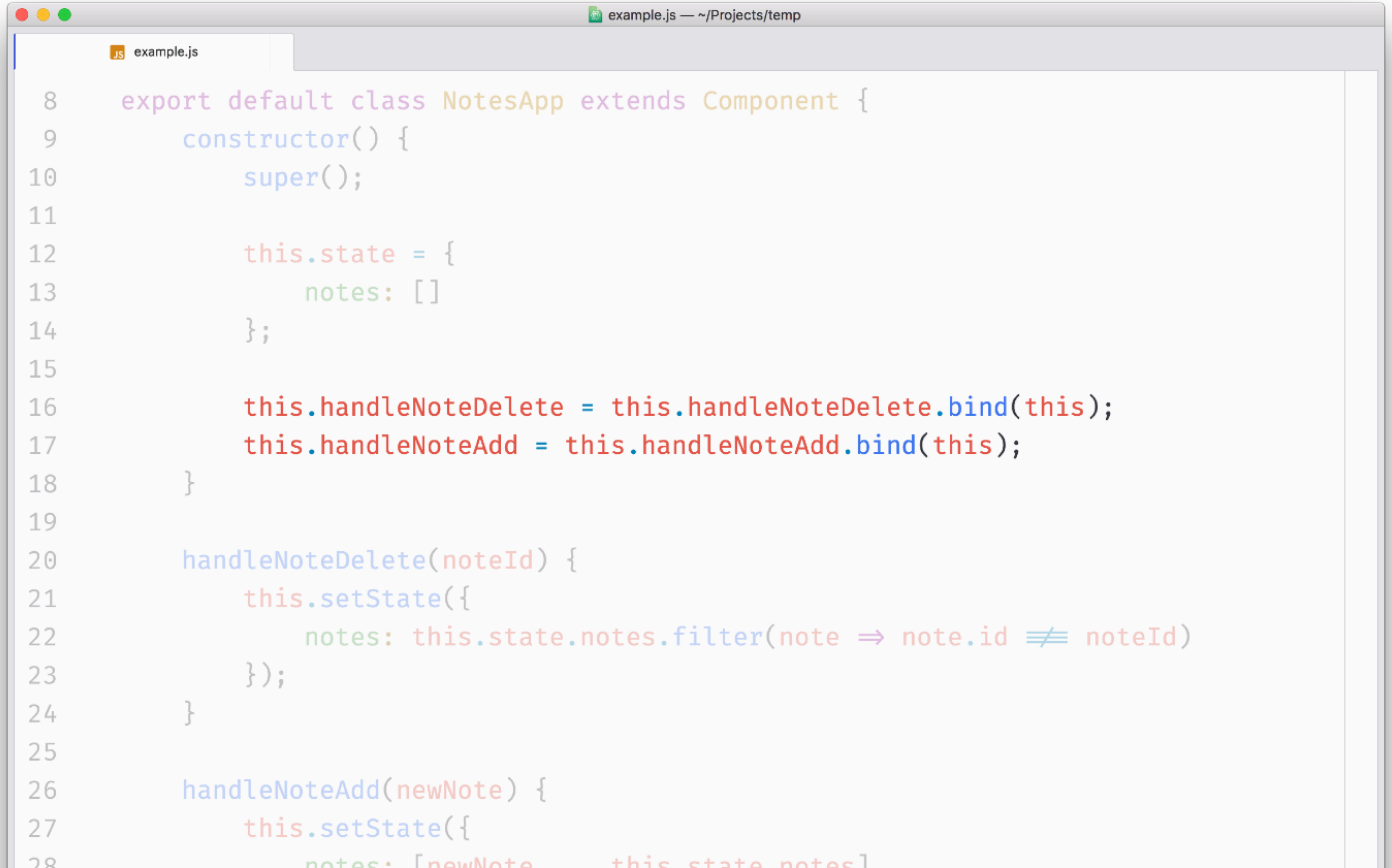
1. function.bind / arrow functions



```
    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">
                <h2 className="app__header">NotesApp</h2>

                <NoteEditor onNoteAdd={this.handleNoteAdd.bind(this)} />

                <NotesGrid
                    notes={this.state.notes || []}
```

again, we create a new function every render

1. function.bind / arrow functions



```
export default class NotesApp extends Component {
    constructor() {
        super();

        this.state = {
            notes: []
        };

        this.handleNoteDelete = this.handleNoteDelete.bind(this);
        this.handleNoteAdd = this.handleNoteAdd.bind(this);
    }

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

    handleNoteAdd(newNote) {
        this.setState({
            notes: [newNote, ...this.state.notes]
```

← better to bind `this` in the constructor