

# 1. function.bind / arrow functions

```
example.js — ~/Projects/temp

13   handleNoteDelete(noteId) {
14       this.setState({
15           notes: this.state.notes.filter(note => note.id !== noteId)
16       });
17   }
18
19   handleNoteAdd(newNote) {
20       this.setState({
21           notes: [ ...this.state.notes, newNote ]
22       });
23   }
24
25   render() {
26       return (
27           <div className="app">
28               <h2 className="app__header">NotesApp</h2>
29
30               <NoteEditor onNoteAdd={this.handleNoteAdd.bind(this)} />
31
32               <NotesGrid
33                   notes={this.state.notes} />

```

again,  
we create  
a new function  
every render

# 1. function.bind / arrow functions

```
example.js — ~/Projects/temp

example.js
8  export default class NotesApp extends Component {
9    constructor() {
10     super();
11
12     this.state = {
13       notes: []
14     };
15
16     this.handleNoteDelete = this.handleNoteDelete.bind(this);
17     this.handleNoteAdd = this.handleNoteAdd.bind(this);
18   }
19
20   handleNoteDelete(noteId) {
21     this.setState({
22       notes: this.state.notes.filter(note => note.id !== noteId)
23     });
24   }
25
26   handleNoteAdd(newNote) {
27     this.setState({
28       notes: [newNote, ...this.state.notes]
```

← better to  
bind 'this'  
in the  
constructor