

# 3. modification by reference

```
example.js — ~/Projects/temp

example.js
10  shouldComponentUpdate(nextProps) {
11      return this.props.notes !== nextProps.notes; // false
12  }
13
14  render() {
15      const {
16          notes,
17          onDelete
18      } = this.props;
19
20      return (
21          <Masonry
22              className='grid'
23              options={MASONRY_OPTIONS}
24          >
25              {
26                  notes.map(note =>
27                      <Note
28                          key={note.id}
29                          id={note.id}
```

↑ so, shallow comparison produces 'false'

# 3. modification by reference

```
example.js — ~/Projects/temp
example.js
8  export default class NotesApp extends Component {
9    state = {
10     notes: []
11   }
12
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">
```

use  
ES2015 spread  
operator or  
map, filter etc.  
methods