

# Oleksandr Oleksiv

Frontend developer

 scalac



# React Hooks

# What are **React Hooks**?

*React Hooks are functions that allow to “power up” your functional components with class-only features (e.g. state, lifecycle hooks, etc.)*

# Functional

Great for presentation

Properties as input, JSX as output

Single purpose focus

# Class-based

Hold the business logic

Work with props and state

Components orchestration

# Problems

Wrapper hell

Extending components with new functionality

Lifecycle hooks

# Components with **Hooks**

Easier to maintain

Flexible

Easier to manage lifecycle hooks

Less code

useState(...)

```
export default class Todo extends Component {
  state = {
    inputValue: "",
    todoList: []
  };

  onChangeHandler = (event) => {
    this.setState({
      inputValue: event.target.value
    })
  };

  onClickHandler = () => {
    this.setState({
      todoList: this.state.todoList.concat(this.state.inputValue),
      inputValue: ""
    })
  };

  render() {
    const todos = this.state.todoList.map((item, index) => <li key={index}
      className="todo-list__item">{item}</li>);

    return (
      <Fragment>
        <input type="text"
          className="input-control"
          placeholder="Todo"
          value={this.state.inputValue}
          onChange={this.onChangeHandler}/>
        <button type="button"
          className="btn btn-default"
          disabled={!this.state.inputValue.length}
          onClick={this.onClickHandler}>Add to list</button>
        <ul className="todo-list">
          {todos}
        </ul>
      </Fragment>
    )
  }
}
```



```
3  const Todo = () => {
4    const [inputValue, setInputValue] = useState('');
5    const [todos, setTodos] = useState([]);
6
7    const onChangeHandler = (event) => {
8      setInputValue(event.target.value)
9    };
10
11   const onClickHandler = () => {
12     setTodos(todos.concat(inputValue));
13     setInputValue('');
14   };
15
16   const todoListItems = todos.map((item, index) => <li key={index} className="todo-list_item">{item}</li>);
17
18   return (
19     <Fragment>
20       <input type="text"
21         placeholder="Type here"
22         className="input-control"
23         value={inputValue}
24         onChange={onChangeHandler}/>
25       <button type="button"
26         className="btn btn-default"
27         onClick={onClickHandler}>Add</button>
28       <ul>
29         {todoListItems}
30       </ul>
31     </Fragment>
32   );
33 };
```

```
79  const Todo = () => {
80    const [todoState, setTodoState] = useState({
81      inputValue: '',
82      todos: []
83    });
84
85    const onChangeHandler = (event) => {
86      setTodoState({
87        inputValue: event.target.value,
88        todos: todoState.todos
89      })
90    };
91
92    const onClickHandler = () => {
93      setTodoState({
94        todos: todoState.todos.concat(todoState.inputValue),
95        inputValue: ''
96      });
97    };
98
99    const todoListItems = todoState.todos.map((item, index) => <li key={index} className="todo-list__item">{item}</li>);
100
101    return (
102      <Fragment>
103        <input type="text"
104          placeholder="Type here"
105          className="input-control"
106          value={todoState.inputValue}
107          onChange={onChangeHandler}/>
108        <button type="button"
109          className="btn btn-default"
110          onClick={onClickHandler}>Add</button>
111        <ul>
112          {todoListItems}
113        </ul>
114      </Fragment>
115    );
116  };
```

useEffect(...)

```
38 export default class Todo extends Component {
39   state = {
40     inputValue: "",
41     todoList: []
42   };
43
44   componentDidMount() {
45     axios.get('api/v1/todos.json')
46       .then(todos => {...})
51     .catch(error => console.error(error));
52   }
53
54   onChangeHandler = (event) => {
55     this.setState({
56       inputValue: event.target.value
57     });
58   };
59
60   onClickHandler = () => {
61     axios.post('api/v1/todos.json', {name: this.state.inputValue})
62       .then(todos => {...})
68     .catch(error => console.error(error));
69   };
70
71   componentWillUnmount() {
72     // cleanup stuff goes here
73   }
74
75   render() {
76     const todos = this.state.todoList.map((item, index) => <li key={index}
77                                               className="todo-list_item">{item}</li>);
78
79     return (
80       <Fragment>
81         <input type="text"
82               className="input-control"
83               placeholder="Todo"
84               value={this.state.inputValue}
85               onChange={this.onChangeHandler}/>
86         <button type="button"
87                className="btn btn-default"
88                disabled={!this.state.inputValue.length}
89                onClick={this.onClickHandler}>Add to list</button>
90         <ul className="todo-list">
91           {todos}
92         </ul>
93       </Fragment>
94     )
95   }
}
```

```
4  const Todo = () => {
5    const [inputValue, setInputValue] = useState('');
6    const [todos, setTodos] = useState([]);
7
8    useEffect(() => {
9      axios.get('api/v1/todos.json')
10         .then(todos => {
11           setTodos(todos);
12         })
13         .catch(error => console.error(error));
14     }, []);
15
16     const onChangeHandler = (event) => {
17       setInputValue(event.target.value);
18     };
19
20     const onClickHandler = () => {
21       axios.post('api/v1/todos.json', {name: inputValue})
22         .then(todos => {
23           setTodos(todos);
24           setInputValue('');
25         })
26         .catch(error => console.error(error));
27     };
28
29     const todoListItems = todos.map((item, index) => <li key={index} className="todo-list__item">{item}</li>);
30
31     return (
32       <Fragment>
33         <input type="text"
34           placeholder="Type here"
35           className="input-control"
36           value={inputValue}
37           onChange={onChangeHandler}/>
38         <button type="button"
39           className="btn btn-default"
40           onClick={onClickHandler}>Add</button>
41         <ul>
42           {todoListItems}
43         </ul>
44       </Fragment>
45     );
46   };
47
```

```
useEffect(() => {  
  window.addEventListener('mousemove', onMouseMoveHandler);  
  return () => {  
    window.removeEventListener('mousemove', onMouseMoveHandler);  
  }  
});
```

```
const onMouseMoveHandler = event => {  
  console.log(event.clientX, event.clientY);  
};
```

`useContext(...)`

```
const App = () => {
  const [view, setView] = useState('todos');
  const [isLoggedIn, setIsLoggedIn] = useState(false);

  const onViewChangeHandler = (viewName) => {
    setView(viewName);
  };

  return (
    <div className="App">
      <AuthContext.Provider value={{status: isLoggedIn, login: setIsLoggedIn}}>
        <Header onLoadTodos={() => onViewChangeHandler('todos')}
          onLoadAuth={() => onViewChangeHandler('auth')} />
        {view === 'todos' ? <Todo /> : <Auth />}
      </AuthContext.Provider>
    </div>
  );
};
```



```
const Auth = props => {
  const auth = useContext(AuthContext);
  return (
    <div>
      <h1>This is Auth component</h1>
      <button onClick={auth.login}>Login</button>
    </div>
  );
};
```

```
const Header = props => {
  const auth = useContext(AuthContext);
  return (
    <header>
      {auth.status}
      ? <button type="button" onClick={props.onLoadTodos}>Todos</button>
      : <button type="button" onClick={props.onLoadAuth}>Authenticate</button>
    </header>
  );
};
```

useMemo(...)

```
33 <Fragment>
34   <input type="text"
35     placeholder="Type here"
36     className="input-control"
37     value={inputValue}
38     onChange={onChangeHandler}/>
39   <button type="button"
40     className="btn btn-default"
41     onClick={onClickHandler}>Add</button>
42   <ul>
43     {useMemo(() => <TodoList items={todos} />, [todos])}
44   </ul>
45 </Fragment>
```

# Rules of Hooks

All hooks should be used at top level

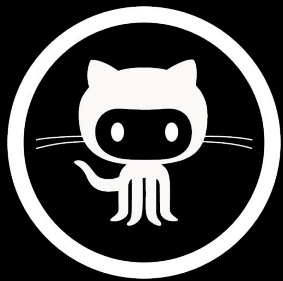
Hook's can't be declared conditionally

Use hooks **ONLY** inside functional components and custom hooks

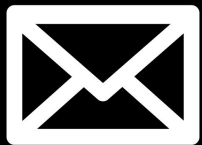
The order matters

THANK YOU

# Contact **me**



[github.com/OleksivO](https://github.com/OleksivO)



Oleksandr.Oleksiv  
[@scalac.io](mailto:Oleksandr.Oleksiv@scalac.io)