

Extracted from:

# React for Real

Front-End Code, Untangled

This PDF file contains pages extracted from *React for Real*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit <http://www.pragprog.com>.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printed versions; the content is otherwise identical.

Copyright © 2017 The Pragmatic Programmers, LLC.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

The Pragmatic Bookshelf

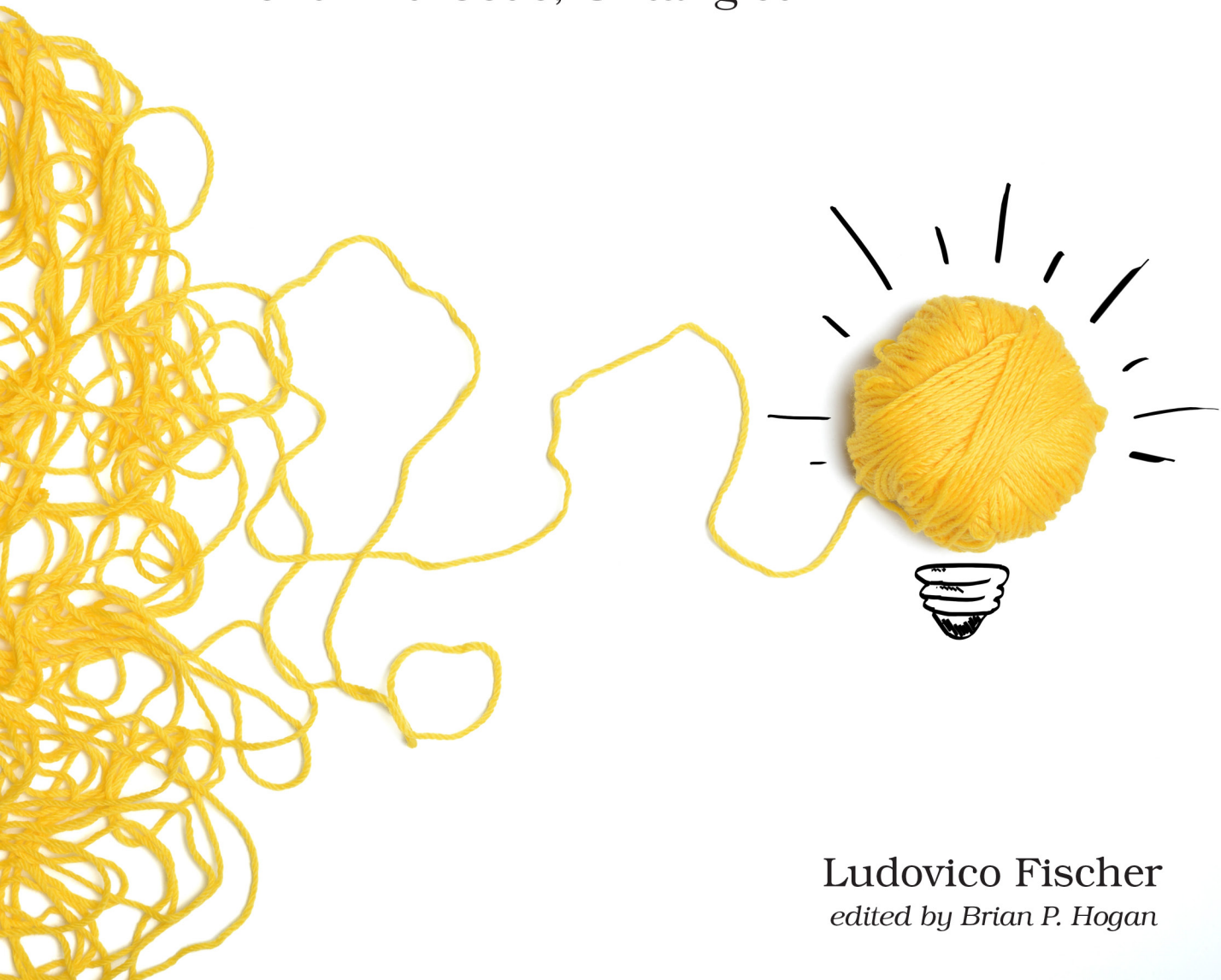
Raleigh, North Carolina

The  
Pragmatic  
Programmers

Pragmatic  
Express

# React for Real

Front-End Code, Untangled



Ludovico Fischer  
*edited by Brian P. Hogan*

# React for Real

Front-End Code, Untangled

Ludovico Fischer

The Pragmatic Bookshelf

Raleigh, North Carolina



Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and The Pragmatic Programmers, LLC was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals. The Pragmatic Starter Kit, The Pragmatic Programmer, Pragmatic Programming, Pragmatic Bookshelf, PragProg and the linking *g* device are trademarks of The Pragmatic Programmers, LLC.

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

Our Pragmatic books, screencasts, and audio books can help you and your team create better software and have more fun. Visit us at <https://pragprog.com>.

The team that produced this book includes:

Publisher: Andy Hunt  
VP of Operations: Janet Furlow  
Development Editor: Brian P. Hogan  
Copy Editor: Nicole Abramowitz  
Layout: Gilson Graphics

For sales, volume licensing, and support, please contact [support@pragprog.com](mailto:support@pragprog.com).

For international rights, please contact [rights@pragprog.com](mailto:rights@pragprog.com).

Copyright © 2017 The Pragmatic Programmers, LLC.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

Printed in the United States of America.

ISBN-13: 978-1-68050-263-3

Encoded using the finest acid-free high-entropy binary digits.

Book version: P1.0—August 2017

# Preface

---

Many web applications allow users to browse and edit content without reloading the page—for example, editing a document, selecting people to email from a contact list, or even browsing a list of events and buying a ticket. While web pages used to be just about the display of information, now a fair amount of business logic might run in the browser. To handle this logic, you often create specialized code—the data model for your application.

In all these applications, you must update the user interface and data model consistently. Since every web page is a tree of elements, called the DOM, you could apply a change directly to the DOM on every action, but this makes it difficult to distinguish business logic from visual adjustments. When you make visual adjustments, you manipulate DOM elements to change the appearance of the user interface. That’s why web developers have started to prefer to store the data in a separate data structure, like a JavaScript object. But now you have a new problem: you need to communicate the changes from the data structure to the DOM and back.

You could add DOM elements with native browser functions such as `appendChild()` and change element contents by overwriting the `innerHTML` attribute on DOM nodes. This forces you to spend time on boilerplate code, away from your application’s core logic. Since this boilerplate turns out similarly across different web applications, it makes sense to offload DOM manipulation to a library.

React is a JavaScript library that automatically updates the DOM when the data changes. It makes writing highly interactive web pages faster and more reliable. It’s easy to use and integrates with a wealth of techniques and other libraries.

In this book, you’ll learn the fundamentals of building user interfaces with React and how to prepare an application for production so that it loads efficiently. You’ll also test components, use React with the Redux library for more complex applications, and work with other libraries. You’ll get hands-on practice as you work through complete examples.

## Who Should Read This Book

If you already have some web development experience in languages like PHP, JavaScript, Ruby, or Java, and would like to learn React, this is the book for you. It's not necessary that you have front-end web development experience, although you should be familiar with HTML. You can use a number of languages to write front-end code. In this book, we use JavaScript, as it's the most used language in the browser. We'll explain novel JavaScript concepts as we go along.

## What's in This Book

This book introduces you to the fundamentals of React and teaches you strategies to build applications, starting from the basics and building up to more advanced cases. Once you've completed this book, you can learn more online through tutorials or other books.

The book focuses on applications that run inside a web browser. It uses hands-on examples to get you started from basic building blocks, to more advanced architectures. We'll cover building the application, testing, and integrating with other libraries.

In [Chapter 1, \*An Introduction to Components\*, on page ?](#), you'll create React components to build your first application using functions, classes, and JSX.

In [Chapter 2, \*Work with State and Events\*, on page ?](#), you'll make your application respond to input and make calls to a web server.

[Chapter 3, \*Create a Production Build\*, on page ?](#) shows you how to create an optimized build so that your application loads quickly in production. It also introduces modules to help you handle large code bases.

You'll test business logic as well as user interface behavior in [Chapter 4, \*Test Your React Components\*, on page ?](#), as you explore different techniques to test your React applications.

Sometimes you'll need to create more complex applications where different components all interact with the same data. In [Chapter 5, \*Using Redux as a Central Data Store\*, on page ?](#), you'll use Redux, a library to handle applications where React's basic model can become inadequate.

Finally, not all JavaScript code out there plays well with React's automatic DOM updates. In [Chapter 6, \*Work Well with Others\*, on page ?](#), you'll integrate React with libraries that have conflicting mechanisms.

Although you can use React to render applications on the server, we won't cover that in this book, as it is a rapidly evolving area, and competing solutions depend on the web framework you use server-side. We won't cover using React to write native applications for iOS and Android either.

## What You Need

You'll need a computer running Windows 10 or a recent version of Linux or macOS, with a recent release of either the Chrome or Firefox web browser. The code you'll write will work in all recent browsers, but sometimes I'll suggest you install some developer tools, which only work in Chrome and Firefox, to inspect the application while it's running. You'll need to be able to install software on your computer, and you'll need an Internet connection.

You'll need a text editor or an IDE. If you're unsure what to pick, Visual Studio Code<sup>1</sup> supports React well. You'll also need to use a terminal to type in commands.

## Conventions

When you'll have to type a command in the command prompt or terminal, it will look like this:

```
$ npm
```

The dollar sign represents the terminal prompt. It's there to tell you that you should type the code in your terminal, but you should just type whatever comes after the dollar sign, without the dollar sign itself.

Other times, you'll see code snippets. The file name above the code sample tells you where to find the file in the book's source code download.

If you read the electronic version of the book, clicking the file name above the code lets you see the full source for that file on the Pragmatic Bookshelf web site.

## Where to Go for Help

You can ask questions and provide feedback in the book's forum linked on the Pragmatic Bookshelf page for this book.<sup>2</sup> On the same page, you'll find links to download the source code for the book's examples and to the errata.

Let's start by learning about React's basic building block, components.

---

1. <https://code.visualstudio.com/>

2. <https://pragprog.com/book/lfreact/>