

Extracted from:

Programming Elm

Build Safe and Maintainable Front-End Applications

This PDF file contains pages extracted from *Programming Elm*, 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 © 2019 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

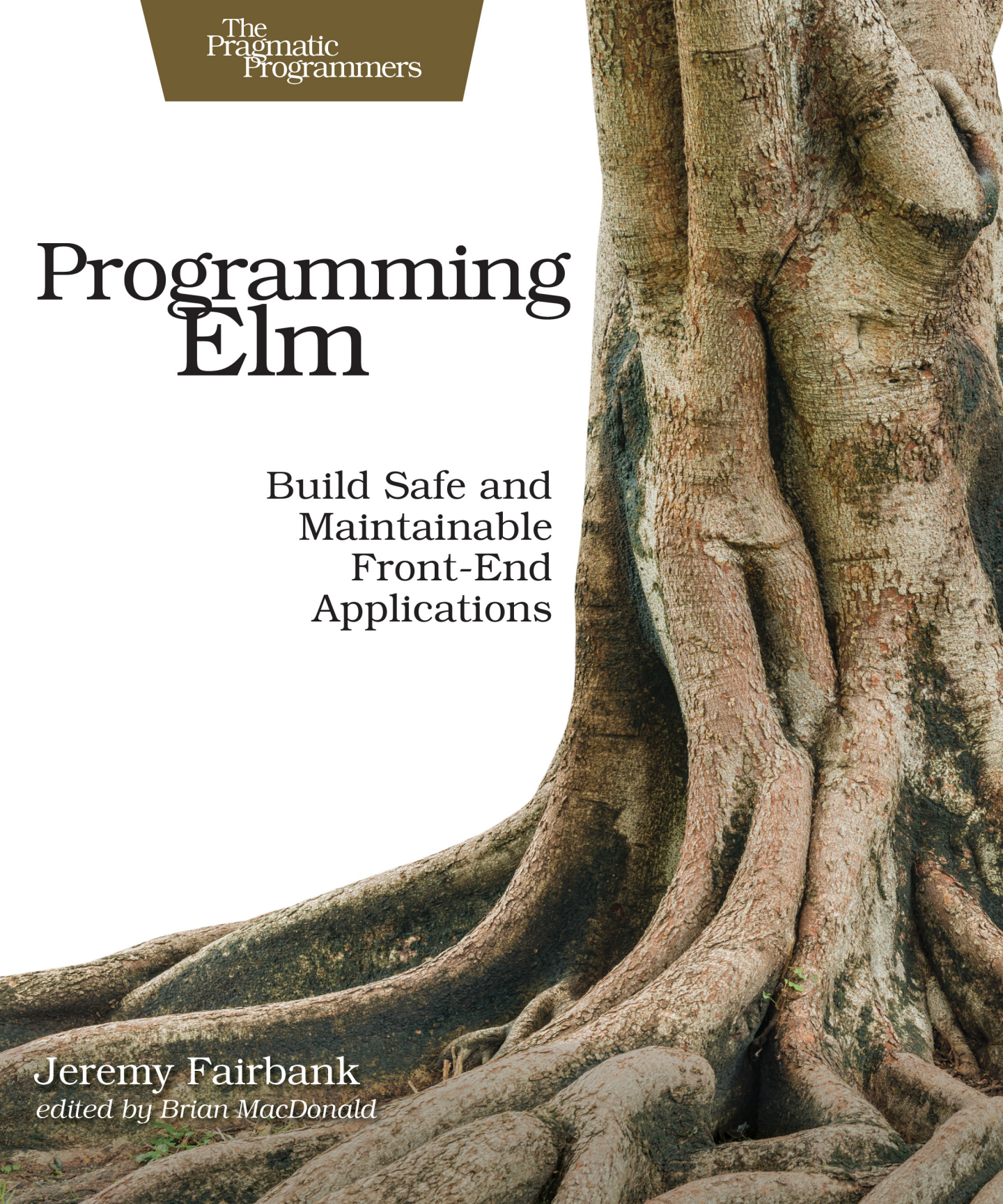
Raleigh, North Carolina

The
Pragmatic
Programmers

Programming Elm

Build Safe and
Maintainable
Front-End
Applications

Jeremy Fairbank
edited by Brian MacDonald



Programming Elm

Build Safe and Maintainable Front-End Applications

Jeremy Fairbank

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

Managing Editor: Susan Conant

Development Editor: Brian MacDonald

Copy Editor: Sean Dennis

Indexing: Potomac Indexing, LLC

Layout: Gilson Graphics

For sales, volume licensing, and support, please contact support@pragprog.com.

For international rights, please contact rights@pragprog.com.

Copyright © 2019 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.

ISBN-13: 978-1-68050-285-5

Book version: P1.0—June 2019

Preface

Don't worry; you haven't picked up the latest gardening book (however, I can teach you how to grow some great tomatoes). Elm is a statically typed, functional programming language made for building safe front-end web applications. It compiles down to minimal JavaScript for easy deployment of your applications to the web.

If you're a front-end developer tired of the JavaScript framework churn or want to build more resilient and maintainable applications, then you need to learn Elm. This book will take you from no knowledge of Elm to creating complex single-page applications.

Why Elm?

More and more front-end developers are choosing Elm to build applications for benefits such as:

- *No runtime exceptions in practice:* Elm's compiler catches problems early to prevent exceptions at runtime for your users.
- *No null or undefined errors:* Elm offers versatile types for representing null. The compiler also ensures you handle all possible nulls in your application.
- *No JavaScript fatigue:* You don't have to choose and wire up different frameworks and libraries to build an application. Elm has a built-in framework for creating applications, the Elm Architecture.
- *Predictable code:* All Elm code is free from side effects, you can trust your functions to always produce the same result based on their arguments.
- *Immutable data types:* You don't have to worry about your code or third-party code changing data unexpectedly and causing bugs. Your data will be consistent and safe.

- *Strong static types*: Elm’s compiler uses static types to ensure you call functions with the right types of arguments. You won’t run into subtle type-coercion bugs.
- *Custom types*: Elm’s custom types let you create entirely new types for clearly modeling your business domain. Powerful pattern matching prevents undefined situations by ensuring you handle your custom types consistently.
- *Advanced tools*: Elm’s Debug module makes it easy to inspect data to catch bugs, and add placeholders to your code until you’re ready to implement it. Third-party tools such as create-elm-app let you quickly bootstrap Elm applications and offer powerful development servers for immediate development feedback.

Who Is This Book For?

This book is for front-end developers new to Elm who want to quickly learn how to build maintainable applications with it. You’ll start with basics such as Elm’s syntax and creating functions and advance all the way to building a single-page application.

Before you read this book, you should know HTML and how to nest HTML elements. Elm’s syntax for building UIs closely mimics HTML. You should also have a good grasp of JavaScript. This book compares some Elm code to JavaScript code—you should know basic JavaScript syntax, objects, arrays, and how to create functions.

In a later chapter, you’ll add Elm code to an existing JavaScript application, so you should be familiar with how to process events with callbacks, bind functions to objects, interact with the DOM, deal with JSON, use promises, and add methods to ES2015 classes.

What’s In This Book?

The first five chapters of this book focus on how to build applications. You will create a photo sharing application called Picshare and add new functionality in each chapter.

[Chapter 1, Get Started with Elm, on page 7](#) introduces you to Elm, explains some of the basics of functional programming, and lets you create a basic Picshare application.

[Chapter 2, Create Stateful Elm Applications, on page ?](#) explains Elm's framework for building applications, the Elm Architecture. You'll use the Elm Architecture to manage state and events in the Picshare application.

[Chapter 3, Refactor and Enhance Elm Applications, on page ?](#) expands on the Picshare application. You'll learn patterns for refactoring code and how to add new features to the Picshare application.

[Chapter 4, Communicate with Servers, on page ?](#) lets you create a more realistic Picshare application. Front-end applications typically need to communicate with servers to be useful. You'll learn how to call APIs and safely decode JSON into static types.

[Chapter 5, Go Real-Time with WebSockets, on page ?](#) takes Picshare's interactivity further. You'll use Elm subscriptions with WebSockets to receive updates in real time.

The next six chapters focus on advanced patterns for scaling, debugging, integrating, and maintaining Elm applications.

[Chapter 6, Build Larger Applications, on page ?](#) addresses the problem of scaling complex applications containing lots of code. You'll use patterns such as reusable helper functions, extensible records, and message wrappers to refactor an application into a more maintainable state.

[Chapter 7, Develop, Debug, and Deploy with Powerful Tooling, on page ?](#) introduces Elm's tooling. Although Elm's compiler prevents tons of bugs through static types, bugs can still occur from logic errors. You'll use Elm's Debug module to debug values at runtime. You'll also bundle and deploy an application with powerful third-party tools.

[Chapter 8, Integrate with JavaScript, on page ?](#) covers interacting with JavaScript code, which is important for accessing impure APIs or migrating existing JavaScript applications to Elm. You'll learn how to add a new feature with Elm to an existing JavaScript application.

[Chapter 9, Test Elm Applications, on page ?](#) introduces testing to ensure your code is correct. You'll use elm-test to create a module with test-driven development, test properties of your code with fuzz testing, and test an Elm application with elm-html-test.

[Chapter 10, Build Single-Page Applications, on page ?](#) teaches you how to build modern single-page applications with Elm. You'll learn how to handle routes and coordinate different page components.

[Chapter 11, Write Fast Applications, on page 7](#) concludes with speeding up your code. You'll learn about common performance issues, how to measure performance, and how to optimize applications with efficient algorithms, lazy design patterns, and the `Html.Lazy` module.

How to Read This Book

If you're an Elm novice, then you should read chapters 1–5 in order to learn the basics and how to create applications with the Elm Architecture. Each of these chapters also builds upon the previous one by using the same application as an example.

If you already know the basics of building applications but want to learn how to interact with servers, then you could skip ahead to chapter 4. Each chapter has code downloads with a version of the application from the previous chapter, so you don't have to go through all of them to catch up.

This book is intended to be read from start to finish, but if you're already pretty familiar with Elm basics, you can skip around after chapter 5. If you're completely new to Elm, you can skip around too, but be forewarned: I introduce some general concepts and built-in Elm functions that might not make sense in later chapters if you skip a previous chapter.

Online Resources

You can visit this book's web page¹ to download the source code examples from this book and provide feedback through an errata-submission form.

Acknowledgements

I would like to thank the following people for reviewing this book and providing invaluable feedback to help improve it: Nick Capito, Jacob Chae, Joel Clermont, Elliot Davies, Zulfikar Dharmawan, Scott Ford, Matt Margolis, Nick McGinness, Luca Mezzalana, Nouran Mhmoud, Daivid Morgan, Eoghan O'Donnell, Emanuele Origgi, Will Price, Noel Rappin, Sam Rose, Dan Sheikh, Kim Shrier, Gianluigi Spagnuolo, Stefan Turalaki, Mitchell Volk, and Stephen Wolff.

I would like to thank my editor Brian MacDonald for all his advice and direction on this book. You helped me become a better writer and teacher. This book wouldn't have been possible without you.

I would like to thank my wife Emily Fairbank for supporting me throughout this journey. You are the most compassionate and understanding person I

1. <https://pragprog.com/book/jfelm/programming-elm>

know. Thank you for bearing with me as I sacrificed so much time to write this book. I love you.

Thank you Tucker for being the most loving and loyal dog I ever knew and the inspiration behind many code samples in this book. Until we meet again.

Finally, thank you reader for choosing to explore this book. I'm humbled by all the kind words others have shared about it. I hope you find it useful and inspiring to embark on your own Elm journey.

Let's get this Elm party started.