

Extracted from:

Build Websites with Hugo

Fast Web Development with Markdown

This PDF file contains pages extracted from *Build Websites with Hugo*, 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 © 2020 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

Build Websites with Hugo

Fast Web Development
with Markdown



Brian P. Hogan
edited by Tammy Coron

Build Websites with Hugo

Fast Web Development with Markdown

Brian P. Hogan

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

Executive Editor: Dave Rankin

Development Editor: Tammy Coron

Copy Editor: Jasmine Kwityn

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 © 2020 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: 978-1-68050-726-3

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

Book version: P1.0—May 2020

Preface

Are you using a database to serve content that rarely changes? If you're using WordPress, Ghost, or other solutions for your content site, you probably are. Or at best, you've set up caching strategies to speed things up and reduce database calls.

Database-driven content management systems like WordPress let you define a common theme shared across your site, making it a breeze to publish new content. However, since most content doesn't change in real time, you're sacrificing speed and scalability for features that benefit content creators and developers instead of the people who want to read your content. And that additional complexity means you need more resources in production too: more servers to handle the traffic, standby database servers, a caching layer you have to manage, and more.

There's no better way to make a snappy content site than by serving static pages from a traditional web server or a content delivery network (CDN). But you don't have to give up the rapid development features you've come to expect. Static site generators, like Hugo,¹ give you a fantastic middle ground. You get the theming and content management features of a database-driven site without the bloat, security vulnerabilities, or complexities associated with caching.

Hugo gives you a framework for building a fast, organized content site using many of the skills you already have. You define your layouts in HTML and your content in Markdown. Hugo has built-in support for tagging, categorization, related content, multiple output formats, image optimization, and asset handling. While you're developing your site, Hugo gives you a web server for testing, and automatically refreshes your pages as you change them. When you're done building your site, Hugo generates a static site that you can upload to your web server.

1. <https://gohugo.io>

There are other static website generators available, like Jekyll² and Gatsby,³ two popular and well-known choices. Hugo has a few benefits over those other choices. First, Hugo uses a single binary with no dependencies, which means you don't need a complex build chain or additional runtimes like Node or Ruby to get started or use its features. Unlike Jekyll, Hugo doesn't assume you're creating a blog. And unlike Gatsby, you don't need to learn React to build your themes or content; if you already know HTML and Markdown, you can be productive right away. Finally, Hugo is incredibly fast when it comes to building the site. It can generate thousands of pages quickly, which comes in handy if you're using Hugo to build something like a documentation website.

In this book, you'll use Hugo to build a personal portfolio site that you can use to showcase your skills and thoughts to the world. You'll build the basic skeleton, develop a custom theme, and use content templates to generate new pages quickly. You'll use internal and external data sources to embed content into your site, and render some of your content in JSON and RSS. You'll add a blog section with posts and integrate Disqus with your site, and then make your site searchable. While this isn't a web design book, you'll integrate modern CSS into the site, and use Hugo's asset management features to process styles, images, and scripts. Then you'll integrate Hugo with Node.js and Webpack for those situations where you need a little more flexibility. Finally, you'll explore deployment with Netlify,⁴ cloud storage, and traditional web hosts so you can share your work with the world.

What's in This Book

Each chapter of the book covers a specific part of the development process with Hugo.

In [Chapter 1, Kicking the Tires, on page ?](#), you'll build the home page and a few supporting pages, along with a skeleton layout.

Then, in [Chapter 2, Building a Basic Theme, on page ?](#), you'll transform the layout into a custom theme for the site. You'll break things up into reusable pieces and put everything in its place.

In [Chapter 3, Adding Content Sections, on page ?](#), you'll create the projects section of the site. You'll work with custom content types and build out more customized layouts for your new content type.

2. <https://jekyllrb.com/>

3. <https://www.gatsbyjs.org/>

4. <https://www.netlify.com/>

Then, you'll integrate data into the site in [Chapter 4, Working with Data, on page ?](#). You'll leverage front matter from your docs, data from external files, and data from external sites. You will also render some of your content as JSON data.

In [Chapter 5, Adding a Blog, on page ?](#), you'll use what you've learned so far to use Hugo to build a static blog. You'll create a content template for blog posts, create a custom layout to control how your posts display, and add support for commenting. You'll generate lists of posts, including a main list of posts in reverse chronological order, as well as lists for tags and categories.

In [Chapter 6, Adding Search to Your Site, on page ?](#), you'll get search working with your site using a search index along with some client-side JavaScript.

Next, in [Chapter 7, Managing Assets with Pipes, on page ?](#), you'll set up a proper asset pipeline so that you can manage your JavaScript and CSS files more easily. You'll switch to using Sass for your stylesheets and explore some of Hugo's built-in features for handling images. You'll also look at integrating Webpack with Hugo, and using npm to manage tasks that make building your site easier.

In [Chapter 8, Deploying the Site, on page ?](#), you'll explore a few methods to move your site to production for the world to see. You'll deploy to Netlify, then look at deploying to Amazon S3, and finally, how to deploy to more traditional web servers.

Finally, you'll find [Appendix 1, Migrating to Hugo, on page ?](#), which explores the process of what it takes to move an existing site to Hugo.

At the end of each chapter, you'll find a few additional exercises that will help cement your knowledge of the concepts.

What You Need

You'll need your trusty text editor and some experience building things for the web with HTML and CSS. This book doesn't go into much detail about those things and assumes you've built websites before. You won't use any CSS frameworks for the layout of your site, but you will use a small amount of modern CSS. The CSS you'll use in this book is intended to demonstrate concepts, not to substitute for careful and professional design. However, if you're an experienced web developer, you'll be able to see exactly where you can apply your existing knowledge as you build out your design.

Hugo runs on Windows, macOS, and various flavors of Linux and BSD operating systems, and it's a single binary file with no dependencies. You'll

download and install Hugo in the first chapter. The examples of this book use Hugo version 0.68.3.

Because Hugo is a command-line tool, you should be comfortable using the command-line interface (CLI). You'll use the `hugo` command throughout the book to build your site, launch a development server, and generate files. If you want to get more comfortable with the CLI, you'll find [Small, Sharp Software Tools \[Hog19\]](#) helpful.

The examples in this book will show CLI commands for copying and moving files. These commands will work in the macOS and Linux terminals, and they will also work on Windows machines using the Windows Subsystem for Linux if you've configured that. Alternatively, you can use your graphical environment or text editor to manage and create files.

Additionally, some experience with JavaScript will be helpful when integrating search into the site.

Finally, in [Chapter 5, Adding a Blog, on page ?](#) and [Chapter 7, Managing Assets with Pipes, on page ?](#), you'll use Node.js⁵ for some additional tooling. You should have Node.js installed on your system by following the official installation instructions for your platform.

Conventions

Throughout the book you'll see commands like this:

```
$ hugo new site portfolio
```

This is a command that you'll type at your command-line interface. The dollar sign indicates the prompt. You won't type that character.

You'll also see code listings, like this:

```
<div class="container">
</div>
```

Sometimes, you'll see highlighted sections when adding new lines of code, like this:

```
<div class="container">
➤   <header>
➤     <h1>{{ .Site.title }}</h1>
➤   </header>
</div>
```

5. <https://nodejs.org>

You'll find instructions on where to add the code to your project with each listing, along with details on what the code does.

Online Resources

The book's website⁶ has links to submit errata for the book as well as the source code for the sites you'll build in this book. The downloadable source code is there as a reference and contains the site as it exists at the end of each chapter.

6. <http://pragprog.com/book/bhhugo>