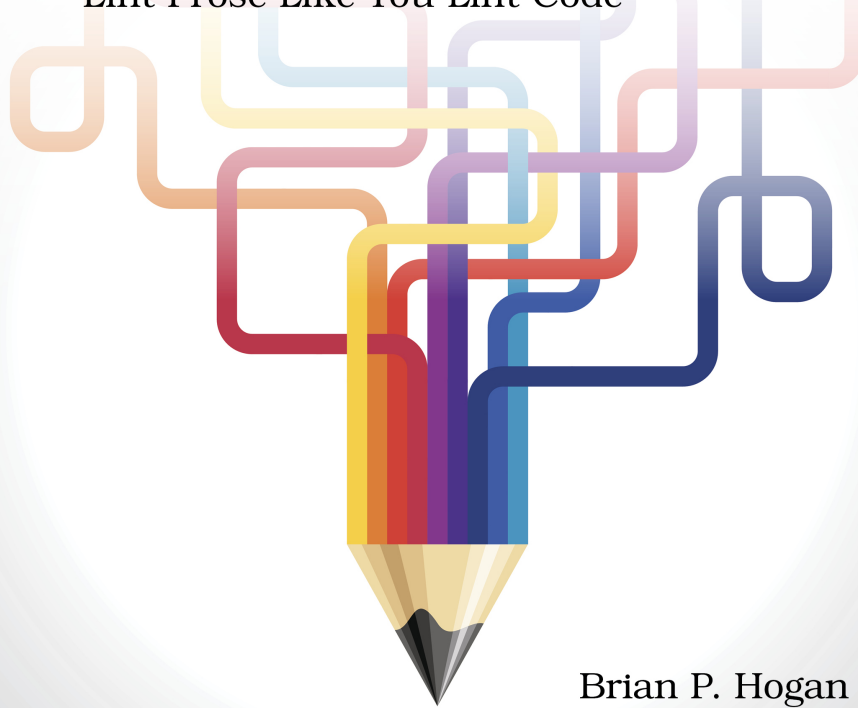


The
Pragmatic
Programmers

Pragmatic
exPress

Write Better with Vale

Automate Your Style Guides and
Lint Prose Like You Lint Code



Brian P. Hogan

edited by Susannah Davidson

This extract shows the online version of this title, and may contain features (such as hyperlinks and colors) that are not available in the print version.

For more information, or to purchase a paperback or ebook copy, please visit <https://www.pragprog.com>.

Copyright © The Pragmatic Programmers, LLC.

Preface

Content is a crucial part of your product's user experience, whether it's your product documentation, blog, use cases, or even customer education material. Your technical content is often the first place technical people go to evaluate a product, and clear, clean, and consistent content creates a great experience.

Style guides lay out the rules for that clarity and consistency. Whether you're writing for your English class or your company's blog, it's a good bet there's a style you're supposed to follow, like making sure all your headings use title-casing, or ensuring that you avoid certain words. Sometimes the style guide lives on a series of web pages or internal knowledge base. Other times it's held in the head space of your senior editors.

If you write code, you've no doubt come across "code linters", tools that enforce code style. You can get the same experience for your prose.

Vale¹ lets you turn a style guide into a set of rules that ensure your writing is consistent. Vale is a prose linter. You can use Vale to check for spelling errors, typos, heading consistency, profanity, grammar issues, and much more. You can hook Vale into your workflow and have it lint your writing while you work, and incorporate it into your continuous integration process. Using Vale, you can automate your style guide.

Here's how it works. You define rules that tell Vale what to look for. You can tell Vale to flag certain words or phrases, suggest substitutions to ensure authors use words consistently, or even ensure your content meets a certain word count. Vale parses your content into components like headings, lists, paragraphs, and sentences, and applies your rules. That means you can look for things in the entire document, or focus on specific parts, like the number of characters in your front-matter's description field, or whether your headings use title case. It knows to ignore things like code blocks or URLs, so you don't have to worry about false-positives in technical documentation.

1. <https://vale.sh>

Best of all, you don't have to start from scratch. If you based your style guide on Microsoft or Google's style guides, you can import Vale's implementations of those guides, or many others, into your project and extend them or change them to suit your needs.

In this short book you'll get hands-on with Vale, customizing its features, using rules from the community, and building your own style you can use to improve the user experience for your documentation.

What's in This Book?

This book is your guide to setting up and using Vale on your content project. You'll get comfortable with Vale's features as you implement a style guide against a collection of inconsistent Markdown documents. You can use Vale with other formats like reStructuredText and AsciiDoc, but in this book you'll focus on constructing style rules and using Vale in a practical use case.

If you use a different format for your own content, you'll find that most of the examples in this book will still work for you, although you'll have to modify some regular expressions when working with rules that look at markup rather than prose. Those cases are rare. Vale's documentation will help you work with other formats.

In [Chapter 1, Getting Started with Vale, on page ?](#), you'll start out by installing Vale and using it to look for basic spelling issues in Markdown and HTML documents. You'll explore some of Vale's command-line options and build a minimal configuration.

In [Chapter 2, Using Existing Styles, on page ?](#), you'll incorporate some existing community style rules into your configuration as you work to improve a small collection of tutorials. You'll make two overlapping style guide implementations work together, and then you'll make content more inclusive and readable, all with pre-existing rules.

In [Chapter 3, Building Your Own Style, on page ?](#), you'll build your own rules from scratch, along with your own vocabulary so you can make sure that your company's technical jargon isn't flagged as spelling errors, but is also represented correctly and consistently.

Finally, in [Chapter 4, Integrating Vale into Your Workflow, on page ?](#), you'll integrate Vale with existing tools, including your editor, your GitHub workflow, and formats that Vale doesn't directly understand. And you'll create a package for your style so you can share it with other projects or other teams.

This book is not meant to replace Vale’s official documentation, so you won’t find every feature covered. This is a focused guide that helps you implement Vale in your project. When you’re done, you’ll be able to confidently investigate other Vale use cases.

Who Should Read This Book

This book is for anyone who works in text-based documents who wants to automatically check that content against a style guide. Anyone with experience with Markdown and command-line tools will be able to follow the examples to build a style guide. If you can write text in a text editor and run some basic commands in the terminal, you’ll have no problems.

Technical writers who use “docs-as-code” workflows will feel at home, as will software developers who use linting tools like ESLint.²

What You Need

You’ll need your trusty text editor, some command-line knowledge, and some idea of the style guide you’d like to implement.

You should also have some experience with regular expressions. You’ll see and write lots of regular expressions in this book. You’ll find [Regex101.com](https://regex101.com)³ helpful as you write and debug your rules.

Where to Go for Help

Visit the book’s web page⁴ and the book’s companion web site⁵ for example code or to report an issue with the book’s examples. You’ll also find a link to the book’s discussion forum where you can interact with other readers.

Vale is a powerful tool that helps you write better quality, consistent documentation. Ready to get started?

Start by getting things set up.

-
2. <https://eslint.org/>
 3. <https://regex101.com>
 4. <https://pragprog.com/titles/bhvale>
 5. <https://thevalebook.com>