

Extracted from:

# Practical Vim, Second Edition

Edit Text at the Speed of Thought

This PDF file contains pages extracted from *Practical Vim, Second Edition*, 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 © 2015 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

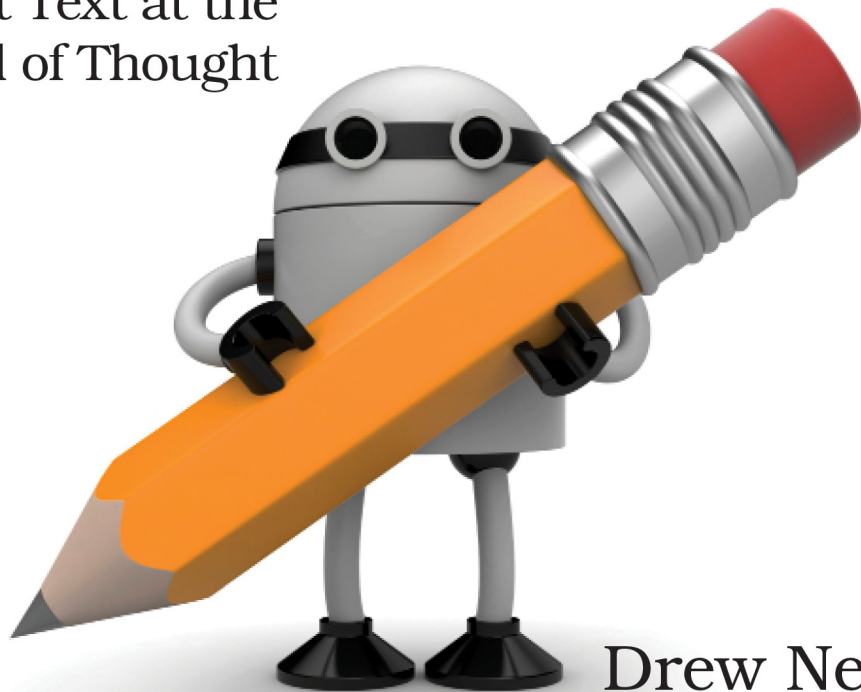
The  
Pragmatic  
Programmers

FOR  
Vim 8

# Practical Vim

Second Edition

Edit Text at the  
Speed of Thought



Drew Neil

Foreword by Tim Pope



# Practical Vim, Second Edition

Edit Text at the Speed of Thought

Drew Neil

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:

Katharine Dvorak (editor)  
Potomac Indexing, LLC (index)  
Cathleen Small (copyedit)  
Dave Thomas (layout)  
Janet Furlow (producer)

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 © 2015 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: 978-1-68050-127-8

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

Book version: P2.0—January 2017

---

# Foreword to the First Edition

Conventional wisdom dictates that Vim has a steep learning curve. I think most Vim users would disagree. Sure, there's an initial hump, but once you run through `vimtutor` and learn the basics of what to put in your `vimrc`, you reach a point where you can actually get work done—a sort of hobbled productivity.

What comes next? The Internet's answer to this is the “tip”—a recipe for solving a specific problem. You might search for specific tips when your current solution to a problem feels suboptimal, or you might proactively read some of the more popular tips. This strategy works—it's how I learned, after all—but it's slow. Learning that `*` searches for the word under the cursor is helpful, but it hardly helps you think like a Vim master.

You can understand my skepticism, then, when I found out *Practical Vim* was using a tips format. How could a couple of hundred tips accomplish what took me thousands? A few pages in I realized my definition of “tip” was narrow-minded. In contrast to the problem/solution pattern I had expected, *Practical Vim* tips teach lessons in thinking like a proficient Vim user. In a sense, they are more like parables than recipes. The first few tips are lessons about the wide applicability of the `.` command. This is a staple of any proficient Vim user's repertoire, yet without guidance it was years before I came to realize this on my own.

It is for this reason that I am excited about the publication of *Practical Vim*. Because now when Vim novices ask me what's the next step, I know what to tell them. After all, *Practical Vim* even taught me a few things.

**Tim Pope**

Vim core contributor

April 2012