

# Vim Preface

## About Vim

[Vim][1] is a computer program used for writing which provides a range of features to help you **write better**.

## Why Vim?

Let's face it, it's very rare to produce your best work on the first attempt. Most likely, you will keep editing it frequently until it becomes 'good'.

As Louis Brandeis once said:

“There is no great writing, only great rewriting.”

Making these numerous rapid changes would be a lot easier if we had a capable editor to help us, and that is *exactly* where Vim shines, and is far better compared to most plain text editors and rich document editors.

## Why Write This Book?

I have been using the Vim editor ever since I learned to use the old vi editor during Unix classes in college. Vim is one of the few pieces of software that I use for nearly 10 hours a day. I knew there were many features which I didn't know about that could potentially be useful to me, so one day I started exploring Vim little by little.

To crystallize my understanding and to help others also explore Vim, I started writing this collection of notes, and called it a book.

Some of the principles I have tried to keep in mind while writing these notes are:

1. Simple literature. The importance of this should be reinforced again and again.
2. Emphasis on examples and how-to.
3. The one-stop shop for readers to learn Vim – from getting started to learning advanced stuff.
4. Get the user to understand how to do things the Vim way – from modes to buffers to customization. Most people learn only the basic vi commands and do not attempt to learn anything beyond that. Learning such concepts is the tipping point, they become hardcore Vim users, i.e., Vimmers, which means they extract the most out of Vim, which is the intent of this book.

5. A lot of things are documented and stored here as a reference for people such as how to use Vim as an IDE, etc. There are various ways of doing it and instead of the user struggling to figure out which plugins to try out, the book already has the basic background work already for the reader.
6. Just enough info to get you to understand and use, not everything required (Pareto principle)
7. Relatedly, the book shouldn't attempt to rewrite the reference manual. Where appropriate, it should simply point out the relevant parts. This way, there is no redundancy, the user learns to use the awesome built-in reference manual which is important, *and* the book can stand on its own strengths as well.

To summarize, the mantra is *Concepts. Examples. Pithy.*

## Something To Think About

Books aren't written – they're rewritten. Including your own. It is one of the hardest things to accept, especially after the seventh rewrite hasn't quite done it.

– Michael Crichton

Perfection is achieved, not when there is nothing more to add, but when there is nothing left to take away.

– Antoine de Saint-Exupery

---

[1]: <http://www.vim.org> [2]: <http://www.swaroopch.com/contact/> [3]: <http://creativecommons.org/licenses/by-sa/3.0/> [4]: <http://www.opensource.org/licenses/bsd-license.php> [5]: [http://en.wikipedia.org/wiki/Wikipedia:Text\\_of\\_the\\_GNU\\_Free\\_Documentation\\_License](http://en.wikipedia.org/wiki/Wikipedia:Text_of_the_GNU_Free_Documentation_License) [6]: <http://swaroopch.com/Swaroop> (User:Swaroop)