
Vim for humans

Release 1.0

Vincent Jousse

August 12, 2015

1	Preamble	1
1.1	Disclaimer	1
1.2	Paid book but free price	1
1.3	Free license	1
2	Introduction	3
2.1	For who?	4
2.2	What you will be learning	4
2.3	What you will not be learning	4
2.4	The hardest part is to get started	5
3	Having a usable <i>Vim</i>	7
3.1	Essential preamble: the insert mode	8
3.2	Modes: the powerful <i>Vim</i> secrets	10
3.3	The lifesaver default configuration	12
3.4	And now, the color!	13
3.5	Our first plugin: the file explorer	17
3.6	Here we go	19
4	The text editor you’ve always dreamed of	23
4.1	Learning how to move: the copy/paste use case	23
4.2	Forgetting the directional keys	26
4.3	Doing without the Esc key	29
4.4	Combining keys and moves	29
4.5	Search / Move quickly	30
4.6	Visual mode	31
4.7	It’s your turn!	32
5	Essential plugins	33
5.1	Managing and switching between files : <i>Lusty Explorer</i>	33
5.2	Searching files on disk: <i>Ack</i>	36
5.3	Searching files on disk: Ctrlp	38
5.4	Advanced plugins	39
6	Cheatsheet & examples	41
6.1	Questions / Answers	41
6.2	Reminders	42
7	Indices and tables	45

PREAMBLE

1.1 Disclaimer

I'm not a native english speaker, so if you spot errors, feel free to tell me. Here is my email address: vincent@jousse.org

1.2 Paid book but free price

What does it mean? I am convinced that culture and education should be accessible for everyone. Asking people to pay a price fixed in advance would put a barrier that limits the access to the book. That's why I let you choose your price.

And what if you don't have any euro to give in exchange for this book? Well, just download it freely and make the promise you will use it for the best. But don't tell me you have nothing to offer in exchange. An email with a thank-you would be just fine.

Whatever you give, it will be a motivation for me to write more books like this one.

If you give in euros, 20% of the earnings are donated to Framasoft, an association promoting free software.

You can give on the official website <http://vimebook.com/en>

Thanks for listening, and have fun!

1.3 Free license

I'm releasing the book under the [Creative Commons Attribution 4.0 International Licence](#). So, you can do (almost) whatever you want with this ebook.



INTRODUCTION

When you need to write or to code, you have to choose a text editor, and a very good one. There are many text editors available out there, but very few of them are more than 40 years old. It's the case of *Emacs* (<http://www.gnu.org/software/emacs/>), *Vi* and its improved successor *Vim* (<http://www.vim.org>). They were created in the 70's and are still used a lot nowadays. You may have already noticed that it's not thanks to the beauty of their website or the efficiency of their communication. Here are some **reasons for their success**:

Forever You learn them once and you use them forever. In a world where languages and technologies are constantly changing, it's a real chance to be able to invest on the long term.

Everywhere They are available for each and every possible platform : Mac Os X, Windows, GNU/Linux, BSD, ... and it's always been the case.

Efficient Thanks to their features (like the extensive use of the keyboard), you can edit and write text as fast as your thoughts.

For everything They allow you to edit everything and anything. When you'll use another programming language, or another markup language, you'll not have to change your editor. Of course, this book has been written using *Vim* (and the [ReST Markup](#)).

Yet, these text editors are difficult to learn. Not that they are harder than anything else, not that you can't handle it, but rather because there is no smart way out there to learn them for now. So, here we are.

The aim of this book is to address this gap by guiding you through your discovery of *Vim*. I'll put *Emacs* aside from now and I'll focus on *Vim*. If you want to know more about this **Editor war**, be sure to check the [Wikipedia page](#). This book doesn't claim to be a reference book about *Vim*. There are already a lot of good references on the subject like [A byte of Vim](#). However, it claims to reduce the entry barrier to get used to *Vim*. To me, the hardest thing to do when learning *Vim* is not to discourage away and find a method allowing use to learn it step by step. We all have to get things done with our text editor on a daily basis, that's why losing all your productivity when switching to *Vim* is not possible.

I'm sure you'll find a lot of persons who will tell you: "Just do it cold turkey", "you'll see, it's hard at the beginning, but time will help". Sure. But you still have to be productive on a daily basis, the problem remains. The approach of this book is the following:

- Have a modern *Vim*: syntax highlighting and nice colors.
- Use *Vim* as any other text editor: easily edit code and switch between files using the mouse.
- Learn keyboard shortcuts and go without the mouse step by step.
- Install the *best* plugins to start using *Vim* to its full potential.

Starting from bullet number 2, you'll already be able to use *Vim* on a daily basis without losing a lot of productivity. It's where the magic will happen: if you can integrate *Vim* in your daily habits, you have won. You'll then have the rest of your life to learn all the shortcuts and the tip and tricks of *Vim*.

You're tired of trying a new editor each year? You're tired of having to relearn everything from scratch every time? You're tired to have to change your editor when you're using your Mac, Windows or Linux? So, just stop it, and join the community of people happy with their text editor!

2.1 For who?

Every guy having to produce text (code, book, reports, slideshows, ...) regularly. Developers are of course concerned, but it's not only about them.

For example, your are a:

Student If you want to impress your future boss with your resume, it's a must. It's a proof of seriousness to see that a student took the time to learn *Vim* on its own. Moreover, you'll have a unique tool to write all what you'll have to write (and that you'll be able to use for the rest of your career): your LaTeX reports, your slideshows, your code (if you need Word or LibreOffice to write your reports, it's time to use [LaTeX](#), [Markdown](#) or [reStructuredText](#)).

Friendly advice: for your slideshows, don't hesitate to use something like [impress.js](#). It's using HTML/JS/CSS and I highly recommend that you use it to do awesome presentations based on non-proprietary technologies. You can have a look at [reveal.js](#) too, and its online editor [slide.es](#).

Teacher It's time to set an example for your student and to learn them one of the tool the will use during their entire life. A lot more than any other programming language.

Coder To invest time in your daily tool is something essential. You'll anyway have to learn keyboard shortcuts, so you'd better do it for something useful. If this investment is style profitable 10 years from now, it's the perfect investment, it's *Vim*.

System and network administrator If you use *Emacs*, then I can forgive you. If you use nano/pico, there is nothing I can do for you, otherwise, it's time to get some job done guys! To remotely administrate a Unix system is the perfect use case for *Vim* (a powerful text editor without the need of a graphical interface).

Writer If you write using Markdown/reStructuredText/WikiMarkup or LaTeX, *Vim* will save you a lot of time. You'll not be able to go back to another editor after it, or you'll want to *Vimify* it at all costs.

Trust me, I have done and still do all these 5 roles, and my best investment has always been, by far, *Vim*.

2.2 What you will be learning

- How to use *Vim* as an “usual” editor first (you know, the type of text editors having syntax highlighting, allowing you to open files, to click using the mouse, ...). In short, the demystification of *Vim* that will allow you to go further.
- How to move from classical text editon to the power of *Vim*, baby step by baby step (it's where addiction begins).
- How to do without the mouse and why it's the best thing that can happen to you when you're programming/writing text.
- How you can easily deduce keyboard shortcuts with some simple rules.

To sum up: as you consider yourself a craftsman, act like one. Learn how to use your tool, once and for all.

2.3 What you will not be learning

- You'll not be learning how to install and to configure *Vim* for Windows. It's doable, but I have very limited knowledge about Windows. It may happen, but not yet. Only Linux/Unix will be discussed (and by extension

Mac OS X).

- You'll not be learning how to use *Vi* (notice the lack of “*m*”). I'll only teach you how to be productive writing text with *Vim*, I'll not teach you how to impress your friends with *Vi* (and anyway, *Vim* is enough for that). For those who don't get what I'm talking about, *Vi* is the “ancestor of *Vim* (which stands for *Vi* - *IM*proved)” and is installed by default on all Unix-like systems (even on Mac OS X).
- You'll not be learning to know *Vim* by heart: this book is not a reference it's a pragmatic smart way to learn *Vim*.
- You'll not learn how to pimp the colors of your *Vim*: I'll use the [Solarized](#) theme, it's the best theme for your eyes.

2.4 The hardest part is to get started

So, you are ready for the adventure? Ready to sacrifice one hour to start using *Vim*, one week to be familiar with it, and the rest of your life to be happy with your choice? So here we go! Well, almost, we need to talk a little bit before.

With *Vim* you'll have to struggle. No matter how big your willpower is, you will struggle. Be prepared. The goal of this guide is to diminish this struggle as much as possible, but be aware that you will struggle anyway. No pain, no gain. Here is the method I recommend to tame the beast:

- Try to make using *Vim* a habit. Be sure to follow this guide until the chapter about *The NERD Tree* (the file explorer). Then you'll be able to use *Vim* as you would do with Notepad++, Textmate or Sublime Text for example. You'll be using only 1% of the capacities of *Vim*, but whatever. What really matters is to use *Vim* on a daily basis.
- Be sure to have a printed sheet with all the main *Vim* shortcuts near you. The goal here is not to learn them by heart, but only to have somewhere to look when you'll ask yourself: “it surely exists a better way to do this”.
- Keep the faith. At the beginning you'll be sceptical regarding the usefulness of learning everything from scratch with *Vim*. And then, one day, you'll have a “a ha!” moment. You'll be asking yourself why all the softwares you are using can't be controlled using *Vim* shortcuts.
- Keep in mind that it's an investment for your next 20 years. And everybody knows that investment are rarely profitable immediately.

So, enough talking, let's get started!