





VINCENT JOUSSE

# VIM - LE GUIDE POUR LES ÊTRES HUMAINS

VINCENT JOUSSE

Copyright © 2012 Vincent Jousse

PUBLIÉ PAR VINCENT JOUSSE

STYLE L<sup>A</sup>T<sub>E</sub>X <http://tufte-latex.googlecode.com>

Si vous n'avez pas payé cette copie, bah tant pis pour moi ;) Mais sachez que vous auriez dû !

# *Table des matières*

*Rendre Vim utilisable*      17

*Index*      23



## Table des figures

- 1 Le thème *Solarized* en sombre et en clair. <http://ethanschoonover.com/solarized> 17
- 2 Nouveau fichier vide. 18
- 3 Mon premier commentaire. 19
- 4 Position de repos, clavier QWERTY. *Illustration par Cy21 - CC-BY-SA-3.0* ([www.creativecommons.org/licenses/by-sa/3.0](http://www.creativecommons.org/licenses/by-sa/3.0)) ou *GFDL* ([www.gnu.org/copyleft/fdl.html](http://www.gnu.org/copyleft/fdl.html)), via *Wikimedia Commons* <http://commons.wikimedia.org/wiki/File:Typing-home-keys-hand-position.svg> 20





## *Liste des tableaux*



*Merci à ma femme et mes enfants qui ont  
permis à ce livre de voir le jour.*



# Introduction

LORSQUE LE BESOIN d'écrire ou de coder se fait se sentir, le choix d'un éditeur de texte est primordial. Il en existe énormément sur le "marché", mais peu d'entre eux peuvent se targuer d'environ 40 ans d'existence. C'est le cas d'*Emacs*<sup>1</sup>, de *Vi* et de son "successeur" *Vim*<sup>2</sup>. Ils ont été créés dans les années 70 et sont toujours très utilisés actuellement<sup>3</sup>. Comme vous avez sans doute pu le voir, ce n'est pas grâce à la beauté de leur site internet ou à l'efficacité de leur communication. Voici quelques raisons de leur succès :

1. <http://www.gnu.org/software/emacs/>

2. <http://www.vim.org/>

3. À noter que *Vim* n'est arrivé qu'en 1991

## *Pour la vie*

Ils s'apprennent une fois et s'utilisent pour toujours. Dans un monde où les technologies/langages changent tout le temps, c'est une aubaine de pouvoir investir sur du long terme.

## *Partout*

Ils sont disponibles sur toutes les plateformes possibles et imaginables (et l'ont toujours été).

## *Enlarge votre productivité*

Ils présentent un rapport temps investi / gain de productivité fabuleux.

## *Couteaux Suisses*

Ils permettent d'éditer tout et n'importe quoi. Quand vous changez de langage de programmation, vous n'aurez pas à changer d'éditeur. À noter que ce livre a bien sur été écrit avec *Vim*.

Et pourtant, ils restent difficiles à apprendre. Non pas qu'ils soient plus compliqués qu'autre chose, non pas que vous ne soyez pas à la hauteur, mais plutôt à cause d'un manque de pédagogie des différentes documentations.

Ce livre a pour but de pallier à ce manque en vous guidant tout au long de votre découverte de *Vim*<sup>4</sup>.

Si vous aussi vous en avez marre d'attendre la release de Text-Mate 2 ? D'essayer le n-ième éditeur à la mode et de devoir tout réapprendre et ce jusqu'à la prochaine mode ? De devoir changer

4. Je laisse *Emacs* à ceux qui savent. Pour un bref comparatif c'est ici : [http://fr.wikipedia.org/wiki/Guerre\\_d'éditeurs](http://fr.wikipedia.org/wiki/Guerre_d'éditeurs). Les goûts et les couleurs ...

d'éditeur quand vous passez de votre Mac, à votre Linux, à votre serveur ? Vous aussi, rejoignez la communauté des gens heureux de leur éditeur de texte. Le changement, c'est maintenant.

### *Pour qui ?*

Toute personne étant amenée à produire du texte (code, livre, rapports, présentations, ...) de manière régulière. Les développeurs sont bien sur une cible privilégiée, mais pas uniquement.

Par exemple vous êtes :

*Étudiant* Si vous voulez faire bien sur un CV, c'est un must (en plus d'être un attrape geekette en puissance <sup>5</sup>). Vous aurez un outil unique pour écrire tout ce que vous avez à écrire (et que vous pourrez réutiliser tout au long de votre carrière) : vos rapports en L<sup>A</sup>T<sub>E</sub>X, vos présentations, votre code (si vous avez besoin d'OpenOffice ou de Word pour écrire vos rapports, il est temps de changer d'outil et d'utiliser L<sup>A</sup>T<sub>E</sub>X).

5. À confirmer.

*Prof* Il est temps de montrer l'exemple et d'apprendre à vos étudiants à bien utiliser un des outils qui leur servira à vie, bien plus qu'un quelconque langage de programmation.

*Codeur* Investir dans votre outil de tous les jours est indispensable. Quitte à apprendre des raccourcis claviers, autant le faire de manière utile. Si cet investissement est encore rentable dans 10 ans, c'est juste l'investissement parfait, c'est *Vim*.

*Administrateur système Unix* Si vous utilisez emacs vous êtes pardonnable, si vous utilisez nano/pico vous êtes pendable, sinon il est grand temps de s'y mettre les gars, c'est un des cas d'utilisation parfait (un éditeur de texte surpuissant ne nécessitant pas d'interface graphique).

*Écrivain* Si vous écrivez en markdown/RST/WikiMarkup ou en L<sup>A</sup>T<sub>E</sub>X, *Vim* vous fera gagner beaucoup de temps. Vous ne pourrez plus repasser à un autre éditeur, ou vous voudrez le "Vimifier" à tout prix.

Faites moi confiance, je suis passé et repassé par ces 5 rôles, mon meilleur investissement a toujours été *Vim*, et de loin.

### *Ce que vous apprendrez (entre autres choses)*

- Comment utiliser *Vim* comme un éditeur "normal" d'abord (vous savez, ceux qui permettent d'ouvrir des fichiers, de cliquer avec la souris, qui ont une coloration syntaxique ...). En somme, la démystification de *Vim* qui vous permettra d'aller plus loin.

- Comment passer de l'édition de texte classique à la puissance de *Vim*, petit à petit (c'est là que l'addiction commence).
- Comment vous passer de la souris et pourquoi c'est la meilleure chose qu'il puisse vous arriver quand vous programmez/tapez du texte.
- Comment vous pouvez facilement déduire les "raccourcis claviers" avec quelques règles simples.

Si je devais le résumer en une phrase : puisque vous vous considérez comme **un artiste, passez du temps à apprendre votre outil**, comme un artiste, une bonne fois pour toute.

### *Ce que vous n'apprendrez pas (entre autres choses)*

- Vous n'apprendrez pas comment installer/configurer *Vim* pour Windows. Pas que ce ne soit pas faisable, mais je n'ai que très peu de connaissances sous Windows. Ça viendra peut-être, mais pas tout de suite. On couvrira ici Linux/Unix (et donc Mac Os X aussi).
- Vous n'apprendrez pas comment utiliser *Vi* (notez l'absence du "m"). Je vais vous apprendre à être productif pour coder/produire du texte avec *Vim*, pas à faire le beau devant les copains avec *Vi*. Pour ceux qui ne suivent pas, *Vi* est "l'ancêtre de *Vim* (qui veut dire *Vi* - *IMproved*, *Vi* amélioré)" et est installé par défaut sur tous les Unix (même sur votre Mac OS X).
- Vous n'apprendrez pas à connaître les entrailles de *Vim* par cœur : ce n'est pas une référence, mais un guide utile et pragmatique.
- Vous n'apprendrez pas comment modifier votre *Vim* parce que vous préférez le rouge au bleu : je vous ferai utiliser le thème [solarized](<http://ethanschoonover.com/solarized>), il est juste parfait pour travailler.

### *Le plus dur, c'est de commencer (et de continuer à commencer)*

Alors, prêt pour l'aventure ? Prêt à sacrifier une heure pour débiter avec *Vim*, une semaine pour devenir familier avec la bête, et le reste de votre vie pour vous féliciter de votre choix ? Alors c'est parti !





# Rendre Vim utilisable

ÇA PEUT PARAÎTRE ÉTONNANT comme approche, mais c’est pour moi la première chose à faire : rendre *Vim* utilisable par un humain lambda. Si tout le monde semble s’accorder sur le fait que *Vim* est un **éditeur très puissant**, tout le monde pourra aussi s’accorder sur le fait que de base, il est juste **imbitable**. Soyons honnête, sans une configuration par défaut minimale, utiliser *Vim* est **contre-productif**.

C’est à mon avis le premier obstacle à surmonter avant toute autre chose. C’est ce que les autres éditeurs “Mainstream” comme Textmate, Sublimetext, Notepad++ ou Netbeans proposent, c’est à dire un environnement à minima utilisable tel quel, même si l’on en n’exploite pas la totalité.

Voici donc ce qui manque à un *Vim* nu (et ce qui est pour moi une **cause d’abandon pour beaucoup** d’entre vous) :

*Configuration par défaut* *Vim* est configurable grâce à un fichier nommé *vimrc*, qui est bien entendu vide par défaut. La première étape va être d’avoir un fichier *vimrc* avec une configuration minimale.

*Coloration syntaxique* De base, *Vim* est tout blanc et tout moche.

On va utiliser le thème *Solarized* (<http://ethanschoonover.com/solarized>). Si votre but est d’être efficace, c’est le meilleur thème disponible actuellement (tout éditeur de texte confondu). La figure 1 vous donne une idée des deux look disponibles (clair ou sombre). Pour ma part j’utilise le thème sombre.

*Explorateur de fichiers* Si vous utilisez *Vim* avec une interface graphique (ce qui est le cas de 99% d’entre vous je suppose) vous avez par défaut un menu Fichier vous permettant d’ouvrir un fichier. C’est certes un bon début, mais avoir à disposition un explorateur de projet à la Netbeans ou à la Textmate peut s’avérer très pratique. Pour obtenir le même comportement, nous utiliserons Nerdtree. À savoir qu’à la fin de ce livre, vous n’aurez plus besoin de la souris (et donc des menus et autres boutons).

Ce chapitre est indispensable si vous n’avez que peu d’expérience

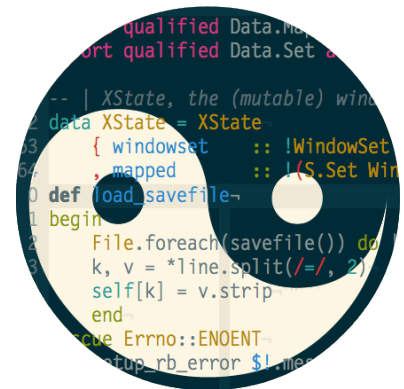


FIGURE 1: Le thème Solarized en sombre et en clair. <http://ethanschoonover.com/solarized>

(voir pas du tout) avec *Vim*. À la fin de ce chapitre, vous aurez un *Vim* dont vous pourrez commencer à vous servir pour vos tâches de tous les jours. Cela devrait être suffisant pour vous permettre d'apprendre le reste petit à petit. Car il n'y a pas de secret, il vous faudra pratiquer pour apprendre *Vim*, alors autant commencer de suite et le moins douloureusement possible.

En revanche, si vous êtes déjà familier avec *Vim* et n'utilisez déjà plus la souris, vous pouvez sagement sauter ce chapitre (soyez sûr tout de même de donner sa chance au thème *Solarized*).

### *Préambule indispensable : le mode insertion*

Prenons le pari de créer le fichier *vimrc* avec *Vim* lui-même. Comme je vous le disais, le plus tôt vous commencerez, le mieux ce sera. Vous devrez certainement commencer par installer une version de *Vim*. Si vous utilisez un Mac, essayez MacVim<sup>6</sup> sans aucune hésitation. Si vous utilisez GNU/Linux ou tout autre système "Unix" vous devriez sûrement avoir gVim à votre disposition (ou tout du moins facilement installable grâce à votre gestionnaire de logiciels). Pour Windows, il semblerait y avoir une version disponible sur le site officiel de *Vim*<sup>7</sup>, mais je ne l'ai pas testée.

Cliquez sur Fichier (File) -> Nouveau (New). Le texte d'accueil par défaut de *Vim* devrait avoir disparu et vous devriez avoir une page blanche comme sur la figure 2.

6. MacVim : <http://code.google.com/p/macvim/>

7. Page de téléchargement officielle de *Vim* : <http://www.vim.org/download.php>

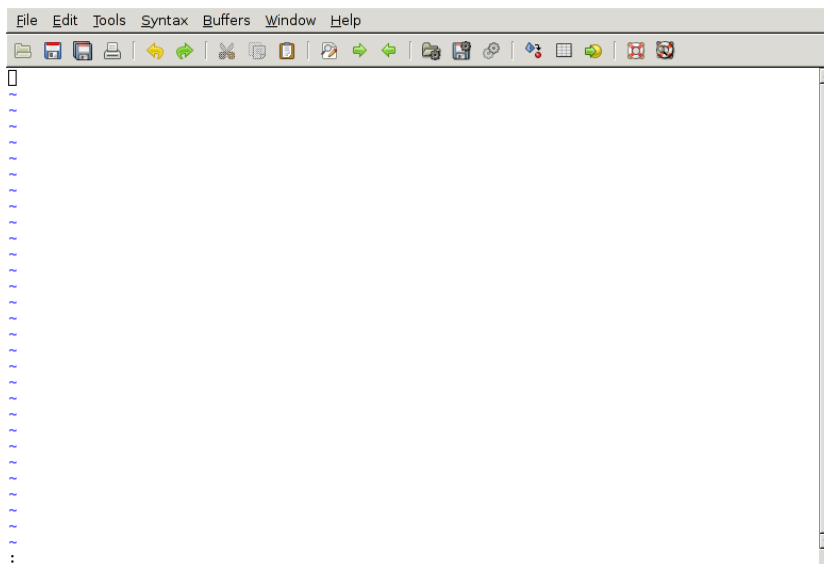


FIGURE 2: Nouveau fichier vide.

Commençons par entrer un commentaire dans l'entête du fichier pour y mentionner notre nom. Pour pouvoir entrer du texte appuyez

sur la touche **i** (le curseur devrait changer d’aspect) et entrez le commentaire ci-dessous<sup>8</sup>.

```
" VIM Configuration - Vincent Jousse
```

Vous aurez remarqué que les commentaires en *VimL* (le langage de configuration de *Vim*) commencent par un `"`. Appuyez ensuite sur la touche **Esc** (**Échap**) pour revenir au mode par défaut (le mode normal) de *Vim*. Et voilà le travail, cf figure 3.

8. Si vous ne savez pas trop ce que vous avez fait et que *Vim* vous affiche des trucs en rouge en bas à gauche au ne semble pas réagir comme il faut quand vous appuyez sur la touche **i**, appuyez plusieurs fois sur la touche **Esc** (**Échap**), ça devrait vous remettre au mode par défaut de *Vim*

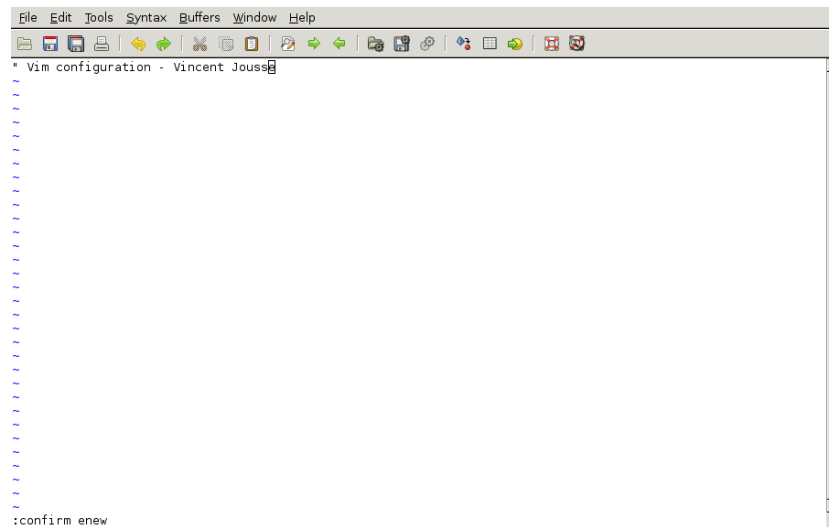


FIGURE 3: Mon premier commentaire.

Tout ça pour ça me direz vous, et vous avez bien raison. Mais tout cela a une logique que je vais vous expliquer. L’avantage de *Vim* est qu’il est généralement logique, quand vous avez compris la logique, tout vous semble limpide et tomber sous le sens.

Par défaut, *Vim* est lancé dans un mode que l’on appelle le mode “Normal”. C’est à dire que ce mode n’est pas fait pour écrire du texte (ça, ça sera le mode “Insert”) mais juste pour se déplacer et manipuler du texte. C’est la présence de ces 2 différents modes (il y en a d’autres mais ce n’est pas le sujet pour l’instant) qui fait toute la puissance de *Vim*. Il vous faudra un certain temps pour vous rendre compte de cette puissance par vous même, alors faites moi juste confiance pour l’instant.

Si vous vous demandez pourquoi ces modes, pourquoi on semble se compliquer la vie (on se la simplifie en fait) et en quel honneur, dans le mode par défaut, il n’est même pas possible d’insérer du texte, lisez attentivement la section qui suit.

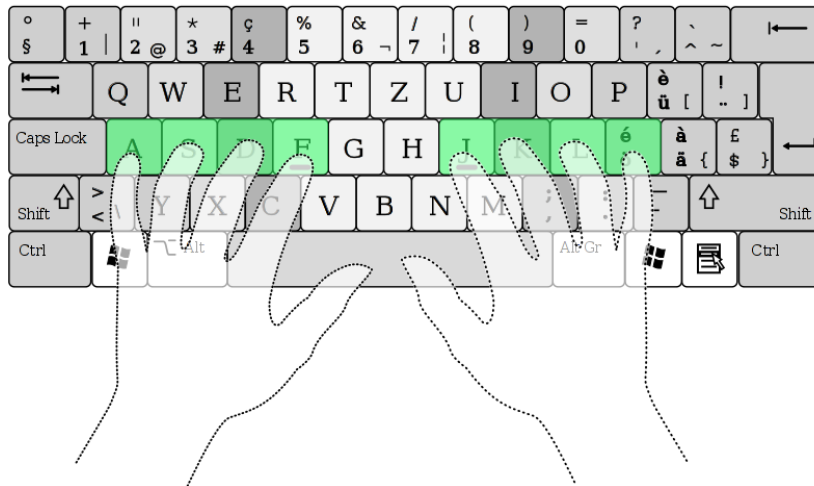


FIGURE 4: Position de repos, clavier QWERTY. Illustration par Cy21 - CC-BY-SA-3.0 ([www.creativecommons.org/licenses/by-sa/3.0](http://www.creativecommons.org/licenses/by-sa/3.0)) ou GFDL ([www.gnu.org/copyleft/fdl.html](http://www.gnu.org/copyleft/fdl.html)), via Wikimedia Commons <http://commons.wikimedia.org/wiki/File:Typing-home-keys-hand-position.svg>

*Les modes : d'où Vim tire sa puissance*

*La configuration par défaut : indispensable*

*La coloration syntaxique : le futile indispensable*

**TODO!**

*L'explorateur de fichiers : parce qu'ouvrir des fichiers, ça peut être pratique*

```
" VIM Configuration - Vincent Jousse
" Some links : http://nvie.com/posts/how-i-boosted-my-vim/

set nocompatible                " Must come first because it changes other options.

" Use pathogen to easily modify the runtime path to include all
" plugins under the ~/.vim/bundle directory
call pathogen#helptags()
call pathogen#runtime_append_all_bundles()

" Quickly edit/reload the vimrc file
nmap <silent> <leader>ev :e $MYVIMRC<CR>
nmap <silent> <leader>sv :so $MYVIMRC<C

" Colorscheme
set t_Co=256
```

**TODO!**





## *Index*