

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* (<http://www.gnu.org/software/emacs/>), de *Vi* et de son "successeur amélioré" *Vim* (<http://www.vim.org/>). Ils ont été créés dans les années 70 et sont toujours très utilisés actuellement. 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** :

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).

Augmentent votre vitesse d'édition de texte

Grâce à leurs particularités (notamment l'utilisation du clavier), ils permettent d'éditer du texte très rapidement.

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 sûr été écrit avec *Vim*.

Et pourtant, ces éditeurs de texte 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 ce manque en vous guidant tout au long de votre découverte de *Vim*¹. Il ne prétend pas être un guide exhaustif, vous pouvez essayer *A Byte of Vim* pour cela : <http://www.swaroopch.org/notes/Vim>. En revanche, il prétend diminuer la marche à franchir pour s'habituer à *Vim*. À mon sens, le

1. Je laisse *Emacs* à ceux qui savent.
Pour un bref comparatif c'est ici :
http://fr.wikipedia.org/wiki/Guerre_d'%C3%A9diteurs. Les goûts et les couleurs ...

plus compliqué avec *Vim*, c'est de ne pas se décourager de suite et de trouver une méthode qui vous permette de l'apprendre au fur et à mesure. Nous avons tous un travail à effectuer au quotidien, et perdre toute sa productivité à cause d'un changement d'éditeur de texte n'est pas envisageable.

Vous trouverez beaucoup de personnes pour vous dire « mais tu n'as qu'à t'y mettre pour de bon », « tu verras après ça va aller mieux », certes, mais vous devez toujours être productif au jour le jour, ça ne change rien au problème. L'approche de ce livre est la suivante :

1. Disposer d'un *Vim* un temps soit peu moderne : coloration syntaxique et jolies couleurs.
2. Pouvoir utiliser *Vim* comme n'importe quel autre éditeur de texte : éditer facilement du code et naviguer entre les fichiers en utilisant la souris.
3. Apprendre des raccourcis clavier et se passer de la souris au fur et à mesure.
4. Installer un *best-of* des plugins pour commencer à tirer partie de la puissance de *Vim*.

À partir du point numéro 2, vous pourrez déjà utiliser *Vim* au quotidien sans perdre énormément de productivité. Et c'est là que la différence se fait : si vous pouvez intégrer *Vim* dans votre quotidien c'est gagné. Vous aurez ensuite toute votre vie pour connaître tous les raccourcis clavier.

Vous aussi vous en avez marre d'attendre la release de TextMate 2² ? D'essayer le n-ième éditeur à la mode et de devoir tout réapprendre et ce jusqu'à la prochaine mode ? De devoir changer d'éditeur quand vous passez de votre Mac, à votre Windows, à votre Linux ? Alors vous aussi, rejoignez la communauté des gens heureux de leur éditeur de texte. **Le changement, c'est maintenant !**

2. À noter que depuis l'écriture de ce livre, le code de TextMate 2 a été publié sous licence GPL : <https://github.com/textmate/textmate>

Pour qui ?

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

Par exemple vous êtes :

Étudiant Si vous voulez faire bien sur un CV, c'est un must. C'est en effet un gage de sérieux de voir qu'un étudiant a pris sur son temps personnel pour apprendre *Vim*. De plus, 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^AT_EX, vos présentations³, 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^AT_EX).

Enseignant 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 ce que l'on pourrait appeler l'investissement parfait, c'est *Vim*.

Administrateur système Unix Si vous utilisez *Emacs* vous êtes pardonnable. Si vous utilisez nano/pico je ne peux plus rien pour vous, sinon il est grand temps de s'y mettre les gars. Administrer un système Unix à distance est un des cas d'utilisation parfait pour *Vim* (un éditeur de texte surpuissant ne nécessitant pas d'interface graphique).

Écrivain Si vous écrivez en markdown/reStructuredText/WikiMarkup ou en L^AT_EX, *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** comment utiliser l'outil qui vous permet de vous exprimer, une bonne fois pour toute.

3. En utilisant *impress.js* par exemple : <http://bartaz.github.com/impress.js>. Basé sur du HTML/JS/CSS, je vous le recommande grandement pour des présentations originales et basées sur des technologies non propriétaires.