

# Vim en 3 pages ...

François Poulain

3 juin 2007

## Résumé

*Vi Improved* est un éditeur très puissant qui a trop de commandes pour pouvoir toutes les énoncer dans un document comme celui-ci, mais l'objectif de ce document est de rassembler l'ensemble des commandes efficaces de Vim en 3 pages, afin de vous permettre d'utiliser simplement et rapidement Vim.

## Table des matières

<b>1</b>	<b>La base :</b>	<b>2</b>
<b>2</b>	<b>Effacements, annulations :</b>	<b>2</b>
<b>3</b>	<b>Insertion :</b>	<b>2</b>
<b>4</b>	<b>Couper, copier, coller du texte :</b>	<b>3</b>
<b>5</b>	<b>Position, recherche et remplacement</b>	<b>3</b>
<b>6</b>	<b>Exécuter une commande externe</b>	<b>3</b>
<b>7</b>	<b>Accéder à l'aide en ligne :</b>	<b>4</b>
<b>8</b>	<b>Ré-indentation automatique d'un fichier</b>	<b>4</b>
<b>9</b>	<b>Pour aller plus loin ...</b>	<b>4</b>

## 1 La base :

1. Le curseur se déplace avec les touches fléchées ou les touches hjkl.  
gauche :  $\Leftrightarrow$  touche h  
bas :  $\Leftrightarrow$  touche j  
haut :  $\Leftrightarrow$  touche k  
droite :  $\Leftrightarrow$  touche l
2. Pour entrer dans Vim (à l'invite \$) tapez vim [Fichier] <Entrée>.
3. Pour quitter Vim sans enregistrer, tapez <Échap> :q! <Entrée>.
4. Pour quitter Vim en enregistrant, tapez <Échap> :wq <Entrée>, ou également <Échap> :x <Entrée>, ou encore <Échap> ZZ.
5. Pour effacer un caractère sous le curseur en mode **normal** tapez x.
6. Pour insérer du texte au niveau du curseur en mode **normal** tapez i et insérez votre texte. Vous êtes alors dans un mode **insertion** que vous pouvez quitter en tapant <Échap>.

**Note :** Appuyer <Échap> vous place en mode **normal** ou annule une commande partiellement tapée dont vous ne voudriez plus.

## 2 Effacements, annulations :

1. Pour effacer du curseur jusqu'à la fin d'un mot tapez dw.
2. Pour effacer du curseur jusqu'à la fin d'une ligne tapez d\$.
3. Pour effacer toute une ligne tapez dd.
4. Le format d'une commande en mode **normal** est :  
[nombre] commande objet, ou commande [nombre] objet, où :  
nombre : c'est combien de fois est répétée la commande.  
commande : c'est ce qu'il faut faire, par exemple d pour effacer  
objet : c'est ce sur quoi la commande devrait agir, par exemple 1 (*letter*), w (*word*), \$ (jusqu'à la fin de la ligne), etc.
5. Pour annuler des actions précédentes, tapez u minuscule, comme *undo*.  
Pour annuler tous les changements sur une ligne tapez U majuscule.  
Pour refaire tapez <Ctrl>-R comme *redo*.

## 3 Insertion :

1. Taper o ouvre une ligne **sous** le curseur et y place celui-ci en mode **insertion**. Taper un O majuscule ouvre une ligne **au dessus** de la ligne où se trouve le curseur.
2. Tapez un a (*append*) pour insérer du texte **après** le caractère où se trouve le curseur. Taper un A majuscule ajoute du texte **à la fin** de la ligne.
3. Pour remplacer le caractère sous le curseur, tapez r (*replace*) suivi du caractère qui remplacera l'original.
4. Taper un R majuscule active le mode **remplacement** jusqu'à ce que la touche <Échap> soit appuyée pour en sortir.

## 4 Couper, copier, coller du texte :

1. Les commandes d'effacement, comme `dd` (*delete*), en réalité **coupent** le texte.
2. Pour **coller** le texte en mémoire **après** la position courante du curseur, tapez `p`. Si une ligne complète a été effacée, elle sera placée **sous** la ligne du curseur. `P` majuscule colle le texte **avant** (ou **au dessus**) la position courante du curseur.
3. Le changement vous permet de changer l'objet spécifié, du curseur jusqu'à la fin de l'objet. Par exemple, tapez `cw` (comme *change word*) pour changer du curseur jusqu'à la fin du mot, `c$` pour changer jusqu'à la fin d'une ligne.
4. Le format d'utilisation pour le changement est : `[nombre] c [objet]`, ou encore `c [nombre] [objet]`.
5. Pour **copier** du texte tapez `yw`, ou `y$`, ou `yy` (*yank*), avec le même format que précédemment indiqué.

## 5 Position, recherche et remplacement

1. `Ctrl-G` affiche votre position dans le fichier et l'état de celui-ci. `G` vous place à la fin du fichier. Un numéro de ligne suivi de `G` vous place à cette ligne.
2. Taper `/` suivi d'un texte recherche ce texte vers l'**avant**.  
Taper `?` suivi d'un texte recherche ce texte vers l'**arrière**.  
Après une recherche tapez `n` pour trouver l'occurrence suivante dans la même direction ou `Ma j-N` pour rechercher dans la direction opposée.
3. Taper `%` lorsque le curseur est sur des délimiteurs de bloc `(, )`, `[, ]`, `{ ou }` déplace celui-ci sur le caractère correspondant.
4. Pour remplacer :

Le premier <code>aa</code> par <code>bb</code> sur une ligne tapez :	<code>s/aa/bb</code>
Tous les <code>aa</code> par <code>bb</code> sur une ligne tapez :	<code>s/aa/bb/g</code>
Du texte entre deux numéros de ligne tapez :	<code>#, #s/aa/bb/g</code>
Toutes les occurrences dans le fichier tapez :	<code>%s/aa/bb/g</code>
Avec une confirmation à chaque fois ajoutez <code>c</code> :	<code>%s/aa/bb/gc</code>

## 6 Exécuter une commande externe

1. `: !<commande>` exécute une commande externe.  
`: !ls` affiche le contenu du dossier courant.  
`: !rm <Fichier>` efface `<Fichier>`.
2. `:w <Fichier>` enregistre le fichier Vim courant avec le nom `Fichier`.
3. `:#, #w <Fichier>` enregistre les lignes `# à #` dans `Fichier`.
4. `:r <Fichier>` lit et inclus `Fichier` dans le fichier courant à partir de la position du curseur.

## 7 Accéder à l'aide en ligne :

1. Vim a un système complet d'aide en ligne. Pour y accéder :

- Appuyez la touche `<F1>`
- Sinon tapez `:help <Entrée>`  
Tapez `:q <Entrée>` pour fermer la fenêtre d'aide.

Vous pouvez accéder à l'aide sur à peu près n'importe quel sujet en donnant des arguments à la commande `:help`. Essayez par exemple (n'oubliez pas d'appuyer sur `<Entrée>`) :

- `:help w`
- `:help c_<T`
- `:help insert-index`
- `:help user-manual`

## 8 Ré-indentation automatique d'un fichier

Il existe une commande particulièrement sympathique quand on récupère des codes sources C/C++/Java : en tapant `gg=G`, vous ré-indentez automatiquement l'ensemble du fichier.

## 9 Pour aller plus loin ...

Ceci conclut ce document. Le but est de posséder un résumé des commandes, afin de permettre d'utiliser l'éditeur relativement facilement. Il est loin d'être complet, vu que Vim a beaucoup beaucoup plus de commandes. Un Manuel de l'utilisateur est disponible en anglais : `:help user-manual`.

Pour continuer à découvrir et à apprendre Vim, il existe un livre traduit en français. Il parle plus de Vi que de Vim, mais pourra vous être utile :

L'éditeur Vi - Collection Précis et concis - par Arnold Robbins. Éditeur : O'Reilly France, ISBN : 2-84177-102-4.

Ce document est en fait un résumé du **tutoriel Vim**, qui a été écrit par Michael C. Pierce et Robert K. Ware de l'École des Mines du Colorado, et reprend des idées fournies par Charles Smith, Université d'État du Colorado. Modifié pour Vim par Bram Moolenaar. Traduit en Français par Adrien Beau, en avril 2001.

Vous pouvez retrouver le tutoriel original en tapant `vimtutor` dans une console.

Pour continuer à découvrir et à apprendre Vim, vous pouvez retrouver la [documentation en ligne, et en français](#). Ce document a été créé avec [LATEX](#). Vous êtes encouragés à lire, distribuer et améliorer ce document dans les termes de la licence GNU/GPL.