

## Récapitulatif vim

Le charme de `vim` : des années d'utilisations et toujours quelque chose de nouveau...

# Commandes générales

- Entrer en mode insertion.

```
i ou a
```

- Pour sortir du mode insertion ou de n'importe quel mode.

```
Echap
```

- Sauvegarder.

```
:w
```

- Quitter sans sauvegarder.

```
:q!
```

- Quitter.

```
:q
```

- Sauvegarder et quitter.

```
:wq
```

- Sauvegarder et quitter.

```
:x
```

- Afficher les numéros de ligne.

```
:set nu ou :set number
```

- Pour enlever l'affichage des numéros de lignes (inverse de `:set nu`).

```
:set nonumber
```

- Permettre le collage tel quel notamment depuis le Web.

```
:set paste
```

- Annuler une opération.

```
Echap puis u
```

- Rétablir une annulation.

```
Ctrl + R
```

- Utiliser la commande `view` au lieu de la commande `vim` ouvrir un fichier uniquement en lecture (écriture impossible).
- `vim` en lecture seule (`view`)

```
vim -R <fichier>
```

## Copier/coller/supprimer/sélectionner des lignes

- Copier/coller : **yy puis p**
- Couper/coller : **dd puis p**
- Copier/coller n lignes : **n yy puis p**
- Couper/coller n lignes : **n dd puis p**
- Coller 5 fois : **5 p**
- Supprimer 10000 lignes après le curseur.

```
10000 dd
```

- Supprimer 10000 lignes avant le curseur.

```
:. , -10000d  
Répondre Y et appuyer sur entrée.
```

- Supprimer une plage de lignes (ici de la 8 a la 14).

```
:8,14d
```

- Supprimer une ligne.

```
D
```

- Couper à droite du curseur : `d$`
- Coller à droite du curseur : `echap p`

## Sélection de texte

- Sélection par block.

```
Ctrl + v
```

- Sélection ligne par ligne (toute la ligne).

```
Shift + v ou V
```

- Indenter :

```
>>
```

- Indenter les 8 lignes suivantes :

```
8>>
```

- Désindenter :

```
<<
```

- Désindenter en combinant une sélection :

```
Ctrl + v et >>
```

- Couper un mot : **dw**
- Copie deux mots : **2 y w**
- Supprimer la fin de la ligne à partir du curseur :

```
Shift + d ou D
```

## Multiple ouverture de fichiers

```
vim -p fichier-1 fichier-2 fichier-n
```

Ouvrir un onglet supplémentaire

```
:tabedit nom-fichier
```

Passer d'un onglet à l'autre

```
g + t  
dans le sens inverse  
g + T
```

Aller directement à l'onglet n

```
:tabmove n
```

## Se déplacer dans le texte

### Sans les flèches avec la méthode vim

- Up : **k**

- Bas : **j**
- Left : **h**
- Right : **l**

## A une ligne précise ou dans les extrêmes

- Aller a la ligne 50.

```
:50
```

- Aller a la fin du fichier.

```
:$
```

- Aller tout en haut.

```
gg
```

- Aller tout en bas.

```
G
```

## Utilisation du shell

- Rester dans vi pour accéder a la commande Linux.

```
!:cdebash
```

- Mettre le résultat de la commande Linux dans vi à l'endroit du curseur.

```
:r!cdebash
```

## Utiliser des buffers

- Ouvrir un fichier supplémentaire.

```
:e fichiersupplementaireouvrir
```

## Se déplacer dans les buffers

- Buffer next.

```
:bn
```

- Buffer delete.

```
:bd
```

## Utiliser des onglets

- Ouvrir un nouveau fichier dans un autre onglet.

```
:tab new cheminnouveau fichier
```

- Se déplacer d'un onglet à l'autre.

```
:tabn** ou **:tabp
```

ou encore **gt**.

## Ecrans multiples

- Découper l'écran en deux.

```
:split
```

- Fermer un écran.

```
:close
```

- vim + :e <path\_file>
- Ensuite :bn ou :bp pour passer de l'un à l'autre
- :vsplit
- Ctrl + ww pour passer de l'un à l'autre
- Ctrl + n pour lister les options disponibles sur l'ensemble des fichiers ouverts

## Configuration vim via vimrc

On peut faire des gabarits dans /home/utilisateur/.vimrc. Ce fichier est en complément de /etc/vimrc qui est le fichier de configuration globale.

- Lorsqu'on tape m il remplacera par la commande :

```
map m :!clear; grep -E 'class \|def ' %
```

- Lorsqu'on tape ,em en mode insertion, le contenu défini est remplacé par {\em } :

```
imap ,em {\em }<ESC>ha
```

- Idem avec une page XHTML préconfigurée accessible avec ,xhtml en insertion :

```
imap ,xhtml <?xml version="1.0" encoding="UTF-8"?><CR><!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN"><CR>"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<CR><html xmlns="http://www.w3.org/1999/xhtml"><CR> <head><CR> <meta
http-equiv="Content-Type" content="text/html; charset=UTF-8"
/><CR><title>Hello world</title><CR><ESC>i </head><CR><body><CR>
<CR><ESC>i </body><CR><ESC>i</html><ESC>kkA
```

- Configurer vi pour qu'il fasse un espace a la place des tabulations (important pour Python basé sur les indentations) :

```
set sts=2
```

- To allow the selected copy/paste in vim on Debian 9, edit the following lines in the /usr/share/vim/vim80/defaults.vim to change mouse=a by mouse=r

```
" Set the mouse mode to 'r'
if has('mouse')
    set mouse=r
endif
```

## Fichier vimrc

- Exemple de fichier vimrc utilisé notamment pour le développement Web avec Python : [vimrc](#)

# Substituer/traitement par lot

## Options courantes de traitement par lot

- Toutes les lignes : %
- Substitute : **s**
- Gourmant (tout le texte) : **g**
- Demander confirmation (ex : %s/blanc/noir/gc) : **c**

## Substitution et ajout par regex

- Substituer dans tout le texte un mot par un autre (ici blanc par noir).

```
:%s/blanc/noir/g
```

- Substituer un mot par un autre sur une plage de ligne (ici de la 4 a la 12).

```
4,12s/blanc/noir/gc
```

Le "c" sert à demander confirmation.

- Ajouter un "#" devant toutes les lignes.

```
:%s/^/#/
```

- Ajouter un "#" devant la plage de lignes spécifiée (ici de la 5 a la 11).

```
:5,11s/^/#/
```

- Supprimer les "#" en début de ligne sur toutes les lignes.

```
:%s/^#//
```

- Ajouter un commentaire "#" en début de ligne sur des lignes sélectionnées.

```
Faire un Ctrl+v verticalement au début sur toutes les lignes (les  
sélectionner simplement)
```

```
Shift+i
```

```
#
```

```
Echap
```

- Remplacer des zones de texte par un seul caractère.

```
Faire un Ctrl+v sur tout le texte qu'on veut (le sélectionner  
simplement)
```

```
r + lettre qu'on veut mettre à la place
```

- Supprimer toutes les lignes qui commencent par ";".

```
g/^;/d
```

- Supprimer les lignes vides.

```
g/^$/d
```

## Supprimer les caractères MS-DOS ^M avec vim

Pour convertir un fichier ouvert avec vim au format UNIX, il suffira d'utiliser la commande suivante.

```
:set ff=unix
```

Ou celle-ci, si l'on souhaite utiliser le format MS-DOS :

```
:set ff=dos
```

Pour ceux qui veulent en savoir plus, ff est un raccourci pour fileformat. La commande suivante marchera aussi.

```
:set fileformat=unix
```

## Options intéressantes

Couleur pour un fichier ini ou yaml

```
:set ft=dosini  
:set ft=yaml
```

Appliquer le retour a la ligne pour voir tout le texte

```
:set wrap pour retour a la ligne
```

Afficher visuellement les colonnes en déplaçant le curseur

```
:set cuc
```

Commande pipée directement à vim avec des options sur la ligne de commande

```
oc export svc docker-registry | vim - :set ft=yaml nowrap
```

## Help

- Accéder a l'aide interactive : **:help**
- Pour cliquer : **Ctrl ]**
- Pour revenir : **Ctrl T**

From:

<https://wiki.ouieuhoutca.eu/> - **kilsufi de noter**

Permanent link:

<https://wiki.ouieuhoutca.eu/vim?rev=1777987745>

Last update: **2026/05/05 13:29**

