

Logiciel

# Mémo des commandes Vim

Ceci est une liste des commandes principales regroupées sur un page.

Ces commandes servent tous les jours et cette page sert de pense-bête.

Vim dispose de trois modes :

- le mode normal ou commande (par défaut) ;
- le mode insertion, signalé par

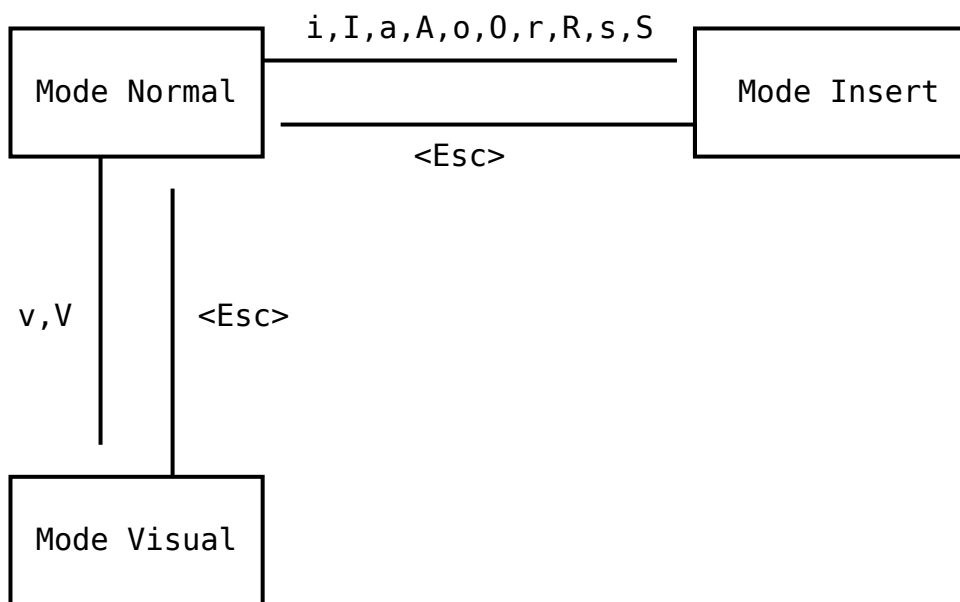
```
-- INSERT --
```

en bas de la fenêtre ;

- le mode visuel, signalé par

```
-- VISUAL {,LINE,BLOCK}--
```

Le passage d'un mode à l'autre est résumé par :



## Insertion

Le mode insertion permet d'éditer un fichier. On sort du mode insertion en appuyant sur Esc (on revient alors en mode commande)

- **i** : insère à la position du curseur

- **I** : insère au début de la ligne
- **a** : insère à la position à droite du curseur
- **A** : insère à la fin de la ligne
- **J** : Joint la ligne suivante au bout de la ligne courante

## Enregistrement et sortie

- **:w** : sauvegarde le fichier
- **:w nomfich** : sauvegarde le fichier sous le nom nomfich (ATTENTION APRES ON N'EDITE PAS nomfich !!! ON CONTINUE AVEC L'ANCIEN !!)
- **:q** quitte sans sauver (signale si fichier modifié avant de quitter)
- **:q!** quitte sans sauver et sans signalement de fichier modifié
- **:x** ou **:wq** : sauve et quitte

## Définition du type de fichier

Ceci redéfinit le type de fichier si nécessaire

- **:set fileformat=unix**
- **:set fileformat=dos**

## Déplacement

- **/mot** : Cherche le mot "mot"
- **:20** : Va à la ligne numéro 20
- **20|** : Va à la 20<sup>e</sup> colonne sur la ligne
- **k** : Déplace le curseur vers le haut (ou utiliser les flèches si elles fonctionnent)
- **l** : Déplace le curseur vers la droite (ou utiliser les flèches si elles fonctionnent)
- **j** : Déplace le curseur vers le bas (ou utiliser les flèches si elles fonctionnent)
- **h** : Déplace le curseur vers la gauche (ou utiliser les flèches si elles fonctionnent)
- **G** : Déplace le curseur à la fin du fichier
- **.'** : Retourne à la dernière ligne modifiée
- **`.** ou **g** : Retourne au dernier caractère modifié

## Définition des variables set

- **:set {no}qqchose** : définit (ou annule) une fonctionnalité de vim
- **:set ic** : Rend les recherches insensibles à la casse

## Alignement

- **:set textwidth=80** : Largeur d'une ligne du fichier (si la ligne est trop longue, vim renvoie a la ligne automatiquement EN MODE INSERTION !!)
- **:{range}center [width]** : aligne au centre selon la range choisie et width indique la largeur de la page.
- **:center** : aligne le texte tape au centre.
- **:right** : aligne le texte à droite
- **:left** : aligne le texte à gauche

## Indentation

- **:set nocindent** : Enlève l'indentation pour les programmes en C (Utile avant un copie/colle avec la souris)
- **:set noautoindent** : Enlève l'indentation pour les programmes en C (Utile avant un copie/colle souris)
- **:set nopaste** : Enlève l'indentation pour les programmes en C (Utile avant un copie/colle souris)
- **:set autoindent** : Indentation automatique
- **:set shiftwidth=4** : Largeur de l'indent (8 car. par défaut)
- **:set softtabstop=4** : Largeur de l'indent quand on pose un TAB
- **:set tabstop=3**
- **:retab 8** Réaligne un code en tabulation a la valeur 3 a la place de la valeur 8
- **/\* vim: set ts=4 sw=4: \*/** : Placé dans un fichier PHP, il est recallé avec 4 cars. par TAB!!!
- **1GVG=** : indente le code du fichier complet

## Remplacement

- **:[range]s/from/to/[flags]** :
  - flags intéressants :
    - **g** : pour toutes les occurrences sur la ligne
    - **c** : demande la validation de la modification
    - **from** : pattern
- 2. **:s/^the/these/** : recherche en début de ligne
- 3. range :
  - **:1,5s/this/that/g** : change sur les lignes 1 a 5
  - **:\$s/yes/no/** : change depuis la ligne courante jusqu'à la fin du fichier
- 4. définir des marques :
  - **mt** : mark top (marque le sommet du bloc)
  - **mb** : mark bottom (marque la fin du bloc)
  - **'t,'b** : range inclus dans les marques

## Effacement

- **dd** : efface (delete) ligne
- **dw** : efface mot (delete word)
- **d\$** : efface jusqu'a la fin de la ligne
- **d^** : efface jusqu'au debut de la ligne
- **x** : effacement du caractère sous le curseur
- **X** : effacement du caractère à gauche du curseur

## Copie / Colle

- **yy** : Sélectionne ligne
- **yw** : Sélectionne word
- **y\$** : Sélectionne jusqu'à la fin de la ligne
- **y^** : Sélectionne jusqu'au début de la ligne
- **p** : colle le contenu de la copie ou de l'effacement en dessous du curseur
- **P** : colle le contenu de la copie ou de l'effacement au dessus du curseur

## Remplacement

- **cd** : Change ligne
- **cw** : Change word
- **c\$** : Change jusqu'à la fin de la ligne
- **c^** : Changement jusqu'au début de la ligne

## Mode Visuel

- **v** : Passe en mode visuel (par mot)
- **V** : Passe en mode visuel (par ligne)
- **Ctrl+V** : Passe en mode visuel (par bloc)  
Sélectionne les lignes a partir du curseur  
Taper ensuite la touche selon l'action que vous souhaitez faire de la sélection. Ex. :  
d pour effacer, y pour copier ...

## Répétition

- **.(point)** : Répète la dernière action

## Affichage des caractères tab et retour chariot

- **:set list** : Affiche ^I pour les tabs et \$ pour les retour chariots

## Coloration Syntaxique

- **:syntax enable** : active la coloration syntaxique (si possible)  
Les diverses colorations peuvent être vues dans : /usr/share/vim/vimXX/syntax  
Si vi ne détecte pas le type automatiquement faire :
- **:set filetype=fortran** : si le type est fortran

## 80 Caractères par ligne dans le HTML

Mettre ces deux lignes quelque part. Attention, garder sur deux lignes !!

```
<!-- vim: tw=80:  
-->
```

## Augmenter/Diminuer un nombre

Si on appuie sur **CTRL+Alt**, on incrémente le chiffre sous le curseur et **CTRL+X** décrémente le chiffre sous le curseur, ce en mode commande.

## Réorganiser un paragraphe pour qu'il soit dans les 80 colonnes

- **:set tw=80**  
**gqap**

ATTENTION : Il s'agit de paragraphe, les listes numériques ne sont pas considérées comme paragraphe !

## Split Ecran

- **:vsp nomfich** : Partage l'écran en deux verticalement, avec chargement de nomfich
- **:hsp nomfich** : Partage l'écran en deux horizontalement, avec chargement de nomfich
- **[CTRL]+W** : Change de split

## Voir les différences entre deux fichiers en couleur

- **\$ vimdiff fichier1 fichier2**

: il s'agit de l'appel de vimdiff dans la console, pas dans vim...

## Édition de plusieurs fichiers

- **:bn** : aller sur le fichier (buffer) suivant
- **:bp** : aller sur le fichier précédent

## Développement

- **Ctrl+N** : affiche les variables déjà existantes dans le code pour une fonction
- **ga** : affiche les correspondances décimale, hexadécimale, octale du caractère sous le curseur (le code hexa peut être utilisé en HTML avec <sup>a</sup>)

## Folding

Permet de faire des sections dans le code pour le plier. Une section commence par {{{ et finit par }}}. En PHP, penser à mettre en commentaires avec /\* {{{ \*/ et /\* }}} \*/

- **zc** : Pour plier une section
- **za** : Pour changer le pliage (de fermé à ouvert ou de ouvert à fermé)
- **zo** : Pour déplier une section
- **zM** : Pour tout plier
- **zR** : Pour tout déplier

## Gestion des onglets

- **:tabnew** : crée un onglet vide
- **:tabnext** : passe à l'onglet suivant

## B.A. BA

### Mode normal

#### Commencer l'insertion de texte

- sur une nouvelle ligne sous la courante :

```
0
```

- sur une nouvelle ligne au-dessus de la courante :

```
0
```

## Enregistrer, quitter

- fermer tous les fichiers/fenêtres :

```
:qa
```

- ZZ
- enregistrer et fermer tous les fichiers :

```
:xa
```

## Écrire dans un fichier file

- tout le fichier, puis quitter :

```
:wq! {file}
```

- la sélection :

```
{range}w {file}
```

- ajouter la sélection au fichier :

```
{range}w >> file
```

Attention,



```
{range}w
```

sauvegarde l'intervalle dans le fichier courant.

## Éditer un fichier

- le fichier courant (recharger) :

```
:e
```

- le fichier courant, sans tenir compte des modifs :

```
:e!
```

- le fichier file :

```
:e {file}
```

## Insérer

- insère le fichier file sous le curseur :

```
:r file
```

- insère le résultat de la commande cmd sous le curseur :

```
:r! cmd
```

## Ouvrir un fichier

- ouvrir file :

```
:o file
```

## Enregistrer dans un fichier

- `:saveas file`

## Mode insertion

- Écrire du texte...
- Quitter, repasser en mode normal : `Esc`
- Complétion :
  - complète le mot avant le curseur : `Ctrl+N`
  - complète le mot après le curseur : `Ctrl+P`
  - complète la ligne : `Ctrl+XCtrl+L`
- 4. Complétion à partir d'un dictionnaire : `Ctrl+XCtrl+K` si

```
:set dictionary=/usr/share/dict/french
```

- 5. Indentation :
  - indente la ligne : `Ctrl+T`

- dédente la ligne :**Ctrl**+**D**
- 6. Suppressions
  - supprime le mot avant le curseur :**Ctrl**+**W**
  - supprime le texte de la ligne courante :**Ctrl**+**U**
- 7. Insérer une espace insécable
  - **Ctrl**+**K** Espace Espace
  - **Ctrl**+**K** ↑ **Shift**+**N** ↑ **Shift**+**Shift**
- 8. Digraphes :**Ctrl**+**K** c1 c2
  - taper

```
:digraphs
```

pour voir la liste complète.

## Mode visuel

Le mode visuel permet de sélectionner des objets.

## Mode commande, opérateurs, actions

## Rechercher, remplacer

## Commandes externes

## Outils d'édition

## Productivité

## Voir aussi

- **(fr)** <http://nmesnier.free.fr/vim.html>
- **(fr)** <https://connaissances.fournier38.fr/entry/M%C3%A9mo%20des%20commandes%20Vim>

---

Basé sur « [Mémo Vim](#) » par Nicolas Mesnier.

Last update:  
2022/08/13  
22:15

logiciel:programmation:vim:memo:start <https://www.nfrappe.fr/doc-0/doku.php?id=logiciel:programmation:vim:memo:start>

---

From:  
<https://www.nfrappe.fr/doc-0/> - **Documentation du Dr Nicolas Frappé**

Permanent link:  
<https://www.nfrappe.fr/doc-0/doku.php?id=logiciel:programmation:vim:memo:start>

Last update: **2022/08/13 22:15**

