

# Mémo Vim

Adrian Poiget

7 juillet 2015

## Résumé

Ce document contient une liste non exhaustive des différents raccourcis sur VIM. Les raccourcis seront représenté comme dans le tableau suivant :

Raccourcis	Action
------------	--------

Pour les raccourcis, la casse est importante : les lettres en majuscule sont l'équivalent de la combinaison Maj-lettre. Par exemple, V correspond à Maj-v et v correspond uniquement à la touche v.

Il n'explique pas les différents mode de Vim, il faut connaître les bases.

J'espère qu'elle pourra vous être utile.

## Table des matières

<b>1</b>	<b>Se déplacer</b>	<b>3</b>
1.1	Déplacement de base . . . . .	3
1.2	Déplacement par mots . . . . .	3
1.3	Déplacement par ligne . . . . .	3
1.4	Déplacement dans le fichier . . . . .	3
1.5	Déplacement par occurrence . . . . .	4
1.6	Déplacement par écran . . . . .	4
1.7	Déplacement sur une ligne de "plusieurs lignes" visuellement . . . . .	4
1.8	Déplacement dans du texte . . . . .	4
1.9	Déplacement dans du code . . . . .	5
1.10	Sauts . . . . .	5
<b>2</b>	<b>Rechercher</b>	<b>5</b>
<b>3</b>	<b>Information</b>	<b>5</b>
<b>4</b>	<b>Insertion/Édition</b>	<b>6</b>
4.1	Insertion . . . . .	6
4.2	Édition . . . . .	6
4.3	Suppression (couper) . . . . .	7
4.4	Copier . . . . .	7
4.5	Coller . . . . .	8
4.6	Changer la casse . . . . .	8
4.7	Action . . . . .	8
4.8	Substitution . . . . .	8
<b>5</b>	<b>Ouvrir, sauvegarder et quitter</b>	<b>9</b>
5.1	Ouvrir depuis la ligne de commande . . . . .	9
5.2	Depuis Vim . . . . .	9
5.3	Les onglets . . . . .	10
<b>6</b>	<b>Macro</b>	<b>10</b>
<b>7</b>	<b>Marqueurs</b>	<b>10</b>
<b>8</b>	<b>Complétion</b>	<b>10</b>

<b>9 Les splits</b>	<b>11</b>
9.1 Ouverture . . . . .	11
9.2 Déplacement . . . . .	11
9.3 Arrangement . . . . .	11
<b>10 Indentation</b>	<b>12</b>
<b>11 Insertion de fichier</b>	<b>12</b>
<b>12 Divers</b>	<b>12</b>

# 1 Se déplacer

Pour commencer nous allons avoir les nombreux manières de se déplacer avec vim.

## 1.1 Déplacement de base

h	Aller sur la caractère de gauche
j	Aller sur le caractère du bas
k	Aller sur le caractère du haut
l	Aller sur le caractère de droite

ou alors on peut aussi utiliser les flèches.

## 1.2 Déplacement par mots

w	Aller au début du mot suivant
W	Aller au début du MOT suivant
e	Aller à la fin du mot suivant
E	Aller à la fin du MOT suivant
b	Aller au début du mot précédent
B	Aller au début du MOT précédent
ge	Aller à la fin du mot précédent
gE	Aller à la fin du MOT précédent

mot vs MOT : mot est une chaîne de lettres, chiffres et underscores, MOT est une chaîne de caractères sans espace

## 1.3 Déplacement par ligne

0 (zero)	Aller au début de la ligne
\$	Aller à la fin de la ligne
^ (accent circonflexe)	Aller au premier caractère non vide de la ligne
g_	Aller au dernier caractère non vide de la ligne
+	Aller au premier caractère non blanc de la ligne suivante
-	Aller au premier caractère non blanc de la ligne précédente
"Entrée"	Aller au premier caractère non blanc de la ligne suivante
:0	Aller au début du fichier (ligne 0)
gg	Aller au début du fichier (ligne 0)
1G	Aller au début du fichier (ligne 0)
:\$	Aller à la fin du fichier
G	Aller à la fin du fichier
CTRL-d	Aller à la fin du fichier
:n	Aller à la ligne n
nG	Aller à la ligne n
ngg	Aller à la ligne n

## 1.4 Déplacement dans le fichier

50%	Aller à 50% du fichier.(milieu du fichier)
100l	Aller au 100ème caractère depuis la position actuelle
100 (100 suivi d'un espace)	Aller au 100ème caractère depuis la position actuelle
25	Aller au 25ème caractère de la ligne en cours
:goto 25	Aller au 25ème caractère depuis le début du fichier

## 1.5 Déplacement par occurrence

*	Aller à la prochaine occurrence du mot sous le curseur
#	Aller à la précédente occurrence du mot sous le curseur
fx	Déplace le curseur à la prochaine occurrence du caractère x
Fx	Déplace le curseur à la précédente occurrence du caractère x
tx	Déplace le curseur juste avant la prochaine occurrence du caractère x
Tx	Déplace le curseur juste avant la précédente occurrence du caractère x

Pour ce déplacer à l'occurrence suivante, il suffit d'appuyer sur ; et pour aller à l'occurrence précédente c'est le raccourcis ,

Exemple 4fx recherche la 4ième occurrence de x.

dt" supprime tout jusqu'au prochain ".

2dfx supprime tout jusqu'à la 2eme prochaine occurrence de x.

## 1.6 Déplacement par écran

CTRL-n	Descendre le curseur d'une ligne
CTRL-p	Monter le curseur d'une ligne
CTRL-f	Descendre le curseur d'une page
CTRL-b	Monter le curseur d'une page
CTRL-d	Descendre le curseur d'une demi-page
CTRL-u	Monter le curseur d'une demi-page
CTRL-e	Descendre l'affichage d'une ligne
CTRL-y	Monter l'affichage d'une ligne
L	Aller au bas de l'écran
xL	Placer le curseur à x lignes du bas de l'écran
H	Aller en haut de l'écran
xH	Placer le curseur à x lignes du haut de l'écran
M	Aller au milieu de l'écran

z<Enter> or zt Redraw the screen with the current line under the cursor at the top of the screen.

z. or zz Redraw the screen with the current line under the cursor at the middle of the screen.

z- or zb Redraw the screen with the current line under the cursor at the bottom of the screen.

z<Enter> (screen line on the top)

88z<Enter> positions line 88 at the top.

z- (scrolls line to the end of the screen)

z. (Center of the screen)

## 1.7 Déplacement sur une ligne de "plusieurs lignes" visuellement

gj	Descendre d'une "ligne"
gk	Monter d'une "ligne"
g0	Aller au début de la "ligne"
gm	Aller au milieu de la "ligne"
g\$	Aller à la fin de la "ligne"
g^	Aller au premier caractère non vide de la ligne

## 1.8 Déplacement dans du texte

{	Aller au début du paragraphe
}	Aller au début du paragraphe suivant
(	Aller au début de la phrase
)	Aller à la phrase suivant

## 1.9 Déplacement dans du code

%	Aller à la parenthèse, accolade, crochet correspondante : {}, [], (), or <>
[i	sur une variable : Affiche la ligne contenant la définition de la variable
[d	sur une macro : Affiche la définition de la macro
[I	sur une variable : Affiche toutes les lignes contenant la variable
gd	Aller à la définition de la variable locale (1ère occurrence de la variable dans la fonction)
gD	Aller à la définition de la variable locale (1ère occurrence de la variable du fichier)
[(	Aller à précédente (
)	Aller à prochaine )
{	Aller à précédente {
}	Aller à prochaine }
[[	Aller au précédent bloc "primaire" defini par des accolades
]]	Aller au prochain bloc "primaire" defini par des accolade
[<CTRL-i>	Aller à la première correspondance (1ère ligne du fichier contenant le terme)
]<CTRL-i>	Aller à la prochaine correspondance (1ère ligne du fichier après le curseur contenant le terme)

CTRL-i est Tab, donc, [<Tab> va aussi à la première correspondance. v% Select between braces

## 1.10 Sauts

Lorsque vous vous déplacer dans le fichier grâce aux raccourcis précédents, vous pouvez revenir à la position d'avant avec les raccourcis suivants :

CTRL-o	Revenir à l'emplacement précédent
CTRL-i ou tab	Aller à l'emplacement suivant
5CTRL-o	Sauter au 5ème emplacement précédent
5CTRL-i	Sauter au 5ème emplacement suivant
:jumps	Afficher les positions de jumps

## 2 Rechercher

/chaine	Recherche le mot vers le bas
?chaine	Recherche le mot vers le haut

Aller à la prochaine occurrence de la chaîne dans la même direction en tapant n.

Aller à la prochaine occurrence de la chaîne dans la même direction en tapant N

Les expressions régulières peuvent être utilisées pour rechercher . (dot) un caractère sauf "aller à la ligne" \* zero ou plusieurs occurrences [...] n'importe quel caractère de l'ensemble [...] n'importe quel caractère qui n'est pas dans l'ensemble ^Anchor - début de la ligne \$ Anchor - fin de la ligne < Anchor - début du mot > Anchor - fin du mot

/jo[ha]n Search john or joan /\< the Search a word beginning by " the" /the\> Search a word ending by "the" /\< the\> Search the /\< Š.\> Search all words of 4 letters??? /\ / Search fred but not alfred or frederick???? /fred\|joe Search fred or joe :bufdo /searchstr/ Search in all open files

:set ic pour ignorer la casse :set hlsearch pour surligner le mot correspondant :set incsearch pour commencer à chercher avant d'avoir tapé l'entrée :set hls is :nohlsearch

## 3 Information

Voici maintenant quelques commandes pour obtenir des informations sur la ligne ou le caractère

: =	Affiche le nombre de ligne
CTRL-g	Affiche des informations sur le ligne en cours
g CTRL-g	Affiche des informations détaillées sur le ligne en cours
:. =	Affiche le numero de la ligne en cours
:#	Affiche le numero de la ligne en cours
:num	Affiche le numero de la ligne en cours
:/foo/=	Affiche le numero de la prochaine ligne contenant foo
ga	Affiche la valeur hex et ascii du caractère
g8	Affiche la valeur hex du caractère

## 4 Insertion/Edition

### 4.1 Insertion

i	Passer en mode insertion avant le curseur
I	Passer en mode insertion au début de la ligne non vide
gI	Passer en mode insertion au début de la ligne
a	Passer en mode insertion après le curseur
A	Passer en mode insertion à la fin de la ligne
o	Ajouter une ligne en dessous et passer en mode insertion
O	Ajouter une ligne au dessus et passer en mode insertion
CTRL-a en mode insertion	Insert le dernier texte écrit
CTRL-@ en mode insertion	Pareil que CTRL-a mais quitte le mode insertion à la fin

Par exemple pour écrire à la fin d'un mot on peut faire ea

### 4.2 Edition

r	Remplacer un caractère (5ra remplace les 5 caractère par a)
R	Remplacer plusieurs caractères
cw	Effacer le mot et passer en mode insertion (changement)
cc ou S	Effacer la ligne entière et passer en mode insertion (changement)
c\$ ou C	Effacer jusqu'à la fin de la ligne et passer en mode insertion (changement)
cl ou s	Effacer le caractère et passer en mode insertion
xp	transposer deux caractères
J	joindre 2 lignes (nJ pour joindre n lignes)
gJ	joindre 2 lignes sans l'espace

select all lines ggVG or ggHG (gggHG?)

### 4.3 Suppression (couper)

x ou dl	supprimer le caractère sous le curseur (4x en supprime 4)
X ou dh	supprimer le caractère placé avant le curseur
dd ou :.d	supprimer la ligne entière (3dd pour supprimer 3 lignes)
dw	supprimer la fin du mot (d4w delete 4 words)
de	supprimer la fin du mot mais pas l'espace de fin
db	supprimer le début du mot mais pas l'espace de début
dge	supprimer le début du mot et l'espace de début
d\$ or D	supprimer jusqu'à la fin de la ligne
d0	supprimer le début de la ligne
diw	supprimer le mot (excluant l'espace de fin)
daw	supprimer le mot (incluant l'espace de fin)
dG	supprimer la fin du fichier
dgg	supprimer le début du fichier
:\$d	Supprimer la dernière ligne
: 'ad	Supprimer la ligne avec le marqueur a
:n,md	Supprimer les lignes n à m
:%d	Supprimer toutes les lignes
:nd"x	couper la ligne n dans le registre x
"xd	couper la sélection visual dans le registre x
CTRL-w (i)	comme pour bash, coupe le mot avant le curseur
CTRL-u (i)	comme pour bash, coupe du début de la ligne jusqu'au curseur

d2l supprimer 2 caractères à la droite du curseur

d4} supprimer 4 paragraphes

### 4.4 Copier

Pareil que pour les commandes pour supprimer en remplaçant d par y (yank)

yl	Copier le caractère sous le curseur
yh	Copier le caractère à gauche du curseur
yy , :y ou :.y ou Y	Copier la ligne
nyy ou :ny	Copier n ligne
:\$y	Copier la dernière ligne
: 'ay	Copier la ligne avec le marqueur a
:n,my	Copier les lignes n à m
:%y	Copier toutes les lignes
yw	Copier la fin du mot
ye	Copier la fin du mot mais sans l'espace de fin
yb	Copier le début du mot
yge	Copier le début du mot avec l'espace de début
y\$	Copier la fin de la ligne
y0	Copier le début de la ligne
yiw	Copier le mot (excluant l'espace de fin)
yaw	Copier le mot (incluant l'espace de fin)
yG	Copier la fin du fichier
ygg	Copier le début du fichier
y4}	Copier 4 paragraphes
" +y	Copier la sélection dans le presse papier du système
"ay	Copier la sélection dans le registre a

y3h : copy three characters to the left

## 4.5 Coller

Mode normal	
p	Coller après le curseur
P	Coller avant le curseur
]p	comme p en indentant
]P	comme P en indentant
gp	comme p en gardant le curseur après le texte copié
gP	comme P en gardant le curseur après le texte copié
"xp ou "xP	Coller le registre x avant/après le curseur
Mode insertion	
CTRL-R a	Coller le registre a
CTRL-R + ou CTRL-R *	Coller le presse papier du système

CTRL-R=5\*5 : insert 25 into text

## 4.6 Changer la casse

~	Changer la casse d'une lettre
guu ou Vu	Toute la ligne est mise en minuscule
gUU ou VU	Toute la ligne est mise en majuscule
g~~ or g~g~	Inverser la casse de la ligne
vEU	Mettre le mot en majuscule
vEu	Mettre le mot en minuscule
vE~	Inverser la casse du mot
ggguG	Tout le fichier est mis en minuscule
gggUG	Tout le fichier est mis en majuscule
ggg~G	Inverser la casse de tout le fichier
<selection>U	La sélection est mise en majuscule)
<selection>u	La sélection est mise en minuscule
g~motion	(It does not depend on tildeop)
gUmotion	(All uppercase)
gumotion	(All uppercase)
gUw	Mettre le mot en majuscule (de même avec W pour le MOT)
guw	Mettre le mot en minuscule (de même avec W pour le MOT)
g~w	Inverser la casse du mot
:%s/\<./\u&/g	Mettre la 1ère lettre de chaque mot en majuscule
:%s/\<./\l&/g	Mettre la 1ère lettre de chaque mot en minuscule
:%s/.*\u&	Mettre la 1ère lettre de chaque ligne en majuscule
:%s/.*\l&	Mettre la 1ère lettre de chaque ligne en minuscule

:set ignorecase Ignore case in searches

:set smartcase Ignore case in searches excepted if an uppercase letter is used

## 4.7 Action

u	Annulation
U	Annulation sur la ligne actuelle
CTRL-r	Restauration
.	Répéter la dernière commande

## 4.8 Substitution

:s/regex/Substitute/flags flag g (global) replace all matches c ask confirmation i insensitive case number , replace the number th occurrence no flag replace only the first occurrence

:s/foo/bar/g all occurence on the current line. :s/foo/bar/gthelastline. : %s/foo/bar/gChangeeach'foo'to'bar'inalllines. : 5,12s/foo/bar/gChangeeach'foo'to'bar'foralllinesfromline5toline12inclusive. :' a,' bs/foo/bar/gChangeeach'foo'to'bar'for., \$s/foo/bar/gChangeeach'foo'to'bar'foralllinesfromthecurrentline(.)tothelastline() inclusive. :.,+2s/foo/bar/g



Change each 'foo' to 'bar' for the current line (.) and the two next lines (+2). :g/^baz/s/foo/bar/g Change each 'foo' to 'bar' in each line starting with 'baz'. if no 'g' at the end , replace only the first or the 'n'th occurrence :g/regex1/s/regex2/string/g Replace all lines matching a regular expression from the set that matches another regular expression For example, to replace all occurrences of York with Orleans on every line that matches New York, type :g/New York/s/York/Orleans/g :%s/^hello/g Replace the beginning of each line by hello :%s/\$/Harry/g Replace the end of each line by Harry :%s/onward/forward/gi Replace onward by forward, case unsensitive :%s/\*\$/g Delete all white spaces :g/string/d Delete all lines containing string :v/string/d Delete all lines containing which didn't contain string :

## 5 Ouvrir, sauvegarder et quitter

### 5.1 Ouvrir depuis la ligne de commande

vim -R fichier	Ouvre le fichier en lecture seule
vim +143 fichier	Ouvre le fichier et positionne le curseur à la ligne 143
vim +/recherche fichier	Ouvre le fichier et positionne le curseur à la première correspondance du terme recherché depuis la ligne 1
vim +?recherche fichier	Ouvre le fichier et positionne le curseur à la première correspondance du terme recherché depuis la ligne 1
vim + fichier	Ouvre le fichier et positionne le curseur à la fin du fichier

### 5.2 Depuis Vim

:e ou :edit	Ouvrir un fichier (éditer) dans un nouveau buffer
:ls	liste les buffers
:bnext ou :bn	aller dans le prochain buffer
:bprec ou :bp	aller dans le buffer précédent
:bd	supprimer le buffer (fermer le fichier)
:saveas foo ou :sav	sauvegarde dans foo et ouvre foo
gf	Ouvre le fichier dont le nom est le mot sous le curseur
:Sex	Ouvrir un exploreur de fichier dans un nouveau split
:Ex	Ouvrir un exploreur de fichier
:browse e	Ouvre un exploreur de fichier graphique
:e! ou :edit!	Remet la dernière version sauvegarder du fichier (Actualise le fichier si modifier depuis l'extérieur)
ZZ ou :wq ou :x	Sauvegarder le fichier et quitter
ZQ ou :q!	Quitter sans sauver
:w [nomfichier]	Sauvegarde dans nomfichier
:wq!	Sauvegarde lorsqu'un fichier a été ouvert en lecture seule (vim -R file)
:n,mw foo	Sauvegarde les ligne n à m dans 'foo'
:w »foo	Ajoute le fichier courant au fichier 'foo'
:q	quitter le fichier
:q!	force la fermeture du fichier sans sauvegarder

:cd .. Move to parent directory :args List files :args \*.php Open file list :grep expression \*.php Returns a list of .php files containing expression

### 5.3 Les onglets

:tabe ou :tabedit fichier	ouvrir un fichier (éditer) dans un nouvel onglet
:tabnew	ouvrir un nouvel onglet
:tabclose ou :tabc	fermer l'onglet
:tabonly ou :tabo	fermer tous les onglets sauf l'onglet ouvert
:tabnext ou :tabn ou gt	Aller à l'onglet suivant
:tabprev ou :tabp ou gT	Aller à l'onglet précédent
#gt	Aller à l'onglet numéro #
:tabmove #	Déplacer l'onglet à la position # (l'index commence à 0)

vim -p file1 file2 ouvrir chaque fichier dans un onglet :tabfind file open a new tab with filename given, searching the 'path' to find it :tabclose i close i-th tab :n Go to next file :p Go to previous file

## 6 Macro

qa	Enregistre dans le registre a
@a	Rejoue la macro enregistrée dans le registre a
@@	Rejoue la dernière macro exécutée
10@a	Rejoue 10 fois la macro du registre a

Exemple : Pour créer une séquence de nombre

Ecrire 1. sur la première ligne, sortir du mode insertion avec <Esc> puis taper qaYp<CTRL-a>q

Explication :

- qa : début de l'enregistrement.
- Yp : copier cette ligne.
- <CTRL-a> : incrémente le nombre.
- q : arrête d'enregistrer.

Écrivez 100@@ ou 100@a pour créer une liste de nombre croissants jusqu'à 102.

## 7 Marqueurs

mx	Add a marker to the character x
'x	Jump to marker x
'x	Jump to the first non-blank character on the line with marker

## 8 Complétion

Les commandes suivantes se font en mode insertion

CTRL-p	Recherche pour completion du mot dans le texte avant le curseur
CTRL-n	Recherche pour completion dans mot le texte après le curseur
CTRL-x CTRL-l	Recherche pour completion de la ligne

## 9 Les splits

### 9.1 Ouverture

<b>Lors du lancement de vim</b>	
vim -o foo bar	Ouvre foo et bar en split
vim -O foo bar	Ouvre foo et bar en vsplit
<b>Depuis vim</b>	
:sp tux ou :split tux	Ouvrir le fichier tux dans un split horizontal
:vsp tux ou :vsplit tux	Ouvrir le fichier tux dans un split vertical
:sview tux	Pareil que :sp mais en lecture seule
<b>Ouvrir de nouveaux splits</b>	
CTRL-w s	Partager l'écran en deux horizontalement
CTRL-w v	Partager l'écran en deux verticalement
CTRL-w w	Changer de split
CTRL-w q	Fermer le split

### 9.2 Déplacement

CTRL-w j	sélectionne le split d'en bas
CTRL-w k	sélectionne le split d'en haut
CTRL-w h	sélectionne le vsplit de gauche
CTRL-w l	sélectionne le vsplit de droite

CTRL-w fleche fonctionne aussi pour se déplacer. Si vous n'avez que deux splits, CTRL-w w vous permettra de basculer de l'un à l'autre.

### 9.3 Arrangement

<b>Splits horizontals</b>	
CTRL-w +	Agrandir le split actif d'une ligne
CTRL-w -	Réduire le split actif d'une ligne
CTRL-w _	Maximise la taille du split
<b>Splits vertivals</b>	
CTRL-w >	Agrandir le split actif d'une colonne
CTRL-w <	Réduire le split actif d'une colonne
CTRL-w	Maximise la taille du split

CTRL-w =	Égalise la taille des splits
CTRL-r r	Échange la position des splits Fonctionne aussi avec "R" majuscule pour échanger en sens inverse
CTRL-w q	Fermer le split courant
CTRL-w o	Fermer tous les splits sauf le split courant

## 10 Indentation

«	indenter la ligne à gauche (un seul < si on est en visual mode)
CTRL-d (i)	indente la ligne à gauche
»	indenter la ligne à droite (un seul > si on est en visual mode)
CTRL-t (i)	indente la ligne à droite
==	autoindentation d'une ligne
=	auto-indentation de la sélection
Vdeplacement=	pour réindenter la sélection
=i{	pour réindenter l'intérieur du bloc d'accolades courant
=a{,	les lignes avec les accolades aussi
=ap	pour réindenter le paragraphe courant
=%	(re)indenter le bloc des accolades { ... }
G=gg	auto (re)indenter le fichier entier

Comme la majorité des commande on peut mettre un nombre devant l'indentation pour répéter cette commande.

## 11 Insertion de fichier

On va utiliser la commande `:read` pour insérer dans notre fichier un autre fichier ou le résultat d'une commande

:r foo.txt	Insérer le fichier foo.txt sous le curseur
:0r foo.txt	Insérer le fichier foo.txt avant la première ligne
:r !ls	Insérer le résultat de la commande <code>ls</code> sous le curseur
:nr !ls	Insérer le résultat de la commande <code>ls</code> à la ligne n
:’ar !ls	Insérer le résultat de la commande <code>ls</code> à la ligne avec le marqueur a
:\$r !pwd	Insérer le résultat de la commande <code>ls</code> à la fin du fichier
:% !cmd	Remplacer tout le fichier par le résultat de la commande <code>cmd</code>
:n,m !cmd	Remplacer la ligne n à m par le résultat de la commande <code>cmd</code>
:n !cmd	Remplacer la ligne n par le résultat de la commande <code>cmd</code>
:. !cmd	Remplacer la ligne actuelle par le résultat de la commande <code>cmd</code>
:’a !cmd	Remplacer la ligne avec le marqueur a par le résultat de la commande <code>cmd</code>

## 12 Divers

:ab test test@foo.info	l'abréviation test sera remplacé par test@foo.info
:!cmd ou !!cmd	Lancer la commande externe <code>cmd</code>
:sh	Lance le shell (retourner à vim avec CTRL-d)
CTRL-a	sur un nombre Ajoute 1 au nombre
CTRL-x	sur un nombre Enleve 1 au nombre
ggg ?G	rot13 le fichier entier
ggVGg ?	Changer le texte to Rot13
:w !sudo tee %	modifier un fichier système lorsqu'il a été ouvert sans être root
CTRL-k (i)	digraph, Insérer non-ascii caractère
:dig	visualiser les digraph définies
CTRL-[ (i) ou Esc	Quitter le mode édition