

# Introduction

On revient aux fondamentaux !

Présentation sur les principes généraux de vi orientée manipulation de texte, déplacement, raccourcis clavier

Non exhaustive, vi est énorme !

## Syntaxe générale

S'applique à toutes les commandes ( même les insertions ) !

[ nombre d'occurrence ] déplacement / action  
par défaut 1

Observation :

vi est fait pour les clavier qwerty ( chiffre, quotes, point... ), comment contourner ?

## Déplacements

### caractères

h : back a char

l : forward a char

### mots

w : mot suivant ( word )

b : mot précédent ( backword )

### lignes

j : ligne suivante

k : ligne précédente

^|0|< deb : début ligne

\$|> fin : fin de ligne

) : prochaine fin de phrase

### flèches

```
←↑→↓  
= char + line
```

## ecran

```
H : premiere ligne de l'écran ( High )  
M : ligne milieuécran ( Middle )  
L : dernière ligne écran ( Low )
```

## Goto line ( action de déplacement)

```
# gg|G  
G  
:#
```

## Actions d'éditions

```
d : cut  
y : copie  
p : paste près curseur  
P : paste avant curseur  
x : supprimer caractère  
r : remplacer caractère  
u : undo  
U : undo all for line
```

```
Y = yy  
D = dd
```

## Mode insertion

```
i : insert before char  
I : insert en debut de ligne  
a : insert after char  
A : insert en fin de ligne  
c = d + i => toutes les possibilités de d  
o : insert un ligne après  
O : insert une ligne avant
```

# Exemples

Rappel : number occurrence / déplacement / action

```
4yj  
10DG  
4w  
10g
```

Human more intuitive ( mais pas pour tout )

action number deplacement ex : y4j

exemple en édition :

10o édition esc => copier 10 fois le texte saisie en dessous de la ligne courante

## Les Buffers

règles générale :

buffer("a-"z) action(y d p) déplacement

```
"z6y : copy 6 lines to buffer z  
"Z2yw : copy 2 words to the end of buffer Z
```

## Cursor Mark

Comme pour les buffers, il y a 24 mark ( de A à Z majuscule )

mX : garde la position de la ligne dans le mark X

'X : va au mark

( remplacer X par la lettre de votre choix)

## Mode ligne de commande

### Syntaxe

:[adress]ACTION

### Address components

```
. Current line  
n Line number n  
.+m Current line plus m lines  
$ Last line  
/string/ A line that contains "string" % Entire file
```

```
[addr1],[addr2] Specifies a range  
% the whole file
```

## Exemples

### Search

```
/toto  
n : next occurence  
N : previous occurence
```

### Search et replace

```
:.,.+10s/Banana/Kiwi  
:%s/apple/pear/g  
:%s./$/ ( suppression des ^M de fin de lignes)
```

## Paramètres

```
:set list : Show invisible characters  
:set nolist Don't show invisible characters  
  
:set number : Show line numbers  
:set nonumber Don't show line numbers  
  
:set autoindent : Indent after carriage return  
:set noautoindent Turn off autoindent  
  
:set showmatch Show matching sets of parentheses as they are typed  
:set noshowmatch Turn off showmatch  
  
:set showmode Display mode on last line of screen  
:set noshowmode Turn off showmode  
  
:set all Show values of all possible parameters
```

## Tips

File header : # vim : set noai ts=2 sw=2 nu syntax=yaml  
(noautoindent tabstop=2 shiftwidth=2 number )  
Copyright Olivier

. : répète la dernière action

split :

<https://www.tecmint.com/split-vim-screen/>

<https://gist.github.com/Starefossen/5957088>

memento : [https://www.atmos.albany.edu/daes/atmclasses/atm350/vi\\_cheat\\_sheet.pdf](https://www.atmos.albany.edu/daes/atmclasses/atm350/vi_cheat_sheet.pdf)