

Vim : mes notes perso

- Objet : Ce que j'ai compris, ma config, etc.,
- Niveau requis :
débutant, avisé
- Références :
 - <https://debian-facile.org/doc:editeurs:vim:vim?&#les-modes>
 - <http://docs.oracle.com/cd/E19620-01/805-1608/6j1io9lha/index.html>

Introduction

Registres (buffers), fenêtres (windows) et onglets (tabs)

Par comparaison avec l'expérience d'un navigateur web, ou d'un éditeur de texte comme notepad++ où un onglet correspond à un fichier ouvert ou à une page tant qu'on ne l'a pas fermé, vim fonctionne complètement différemment. Pour comprendre son fonctionnement, il faut prendre en compte les trois niveaux d'abstraction que sont registres, fenêtres et onglets sous vim.

- **Registre (appelé aussi tampon)**

Un registre dans vim peut être représenté par une instance de mémoire ouverte pour un fichier. Le contenu mémorisé d'un fichier ouvert dans un registre n'est pas forcément visible à l'écran.

Si ce registre est non modifiée (ou si vous utilisez `hidden`), et que vous ouvrez un autre fichier `:edit`, ce registre va dans l'arrière-plan et vous commencez à travailler avec le nouveau fichier.

Le registre précédent ne disparaît lorsque on supprime explicitement avec un appel à `:quit` de `:bdelete` `:quit` ou `:bdelete`, et même alors, il est encore récupérable à moins que vous faites `:bwipe`.

Par défaut, un registre ne peut être en arrière-plan comme celui-ci, sans avoir été modifié. On peut supprimer cette restriction si vous le souhaitez avec l'option `:set hidden`.

On peut obtenir une liste rapide des registres dans une session Vim avec `:ls`. Tout cela signifie que l'idée familière d'onglets dans un éditeur de texte comme notepad++, correspond plus à un registre sous Vim qu'à un véritable onglet.

- **Fenêtres**

Sous vim, en faisant `:split` ou `vsplit`, il est possible d'avoir

1. plusieurs fenêtres du même registre
2. "diviser" une fenêtre pour afficher plusieurs registres
pour cela après ouverture d'un fichier (un registre = affichage d'un registre pour un fichier)
on en ouvre un autre en faisant `:split /chemin/autre/fichier` ou `vsplit /chemin/autre/fichier`.

- **Onglets**

Une page de l'onglet est une collection d'une ou plusieurs fenêtres. Il y a toujours au moins une page d'onglet dans l'instance. (En fonction de votre showtab réglage qui permet de voir une liste des onglets lorsqu'il y en a plus d'un d'ouvert.)



La mémoire registre 5 est visible dans deux fenêtres à la fois, ce qui peut être utile pour faire référence à deux sections d'un long fichier simultanément, tandis que les registre 4 et 6 ne sont pas visibles dans toutes les fenêtres.

Ce qui semble déroutant au début, c'est que les registre ne sont pas liés à un élément d'interface utilisateur; ils peuvent aller et venir dans une fenêtre.

De plus, certaines commandes qui opèrent sur des registre tels que `:bdelete`, peuvent également affecter la disposition des fenêtres.

Commandes de base

Mode insertion

Commandes d'insertion	Actions
i	Insérer du texte à gauche du curseur
I	Insérer du texte en début de ligne
a	Insérer du texte à la droite du curseur
A	Insérer du texte à la fin de la ligne courante
o	Passer en mode insertion sur une nouvelle ligne au-dessous la ligne courante
O	Passer en mode insertion sur une nouvelle ligne au-dessus la ligne courante
cw	Remplacer la fin d'un mot du curseur jusqu'à la fin d'un mot (curseur au début du mot à remplacer puis taper cw puis écrire la correction)
ciw	Remplacer le mot entier où que soit le curseur sur ce mot
cc	Remplacer une ligne entière où que soit le curseur sur cette ligne (curseur sur la ligne à remplacer puis écrire cc la ligne est remplacée par une ligne blanche sur laquelle on écrit
C	Remplacer une ligne à partir du curseur jusqu'en fin de cette ligne
s	Substituer un ou plusieurs caractères (On place le curseur sur le caractère à substituer puis on ajoute un ou plusieurs caractères)
Échap	Sortir du mode commande

Mode commande

- Mode commande sans ex :

Commandes de déplacements	Actions
0	Aller en début de ligne
\$	Aller en fin de ligne

Commandes de déplacements	Actions
j	Se déplacer vers le bas
k	Se déplacer vers le haut
l	Se déplacer vers la droite
h	Se déplacer vers la gauche
b	("back") pour déplacer le curseur mot par mot vers la gauche
nb	pour déplacer le curseur de n mots vers la gauche
w	("word") pour déplacer le curseur mot par mot vers la droite
nw	pour déplacer le curseur de n mots vers la droite
W ou B	Se déplacer le curseur du signe de ponctuation le plus proche au blanc suivant ou précédent
e	("end") pour positionner le curseur sur le dernier caractère du mot en cours
^	Amener le curseur au début de la ligne en cours
\$	Amener le curseur à la fin de la ligne en cours
H	("high") pour amener le curseur en haut de l'écran
M	("middle") pour amener le curseur au centre de l'écran
L	("low") pour amener le curseur en bas de l'écran
Ctrl-F	Page-écran vers le bas (Maintenir la touche Ctrl enfoncée et appuyez sur la touche F)
Ctrl-D	Demi-écran vers le bas (Maintenir la touche Ctrl enfoncée et appuyez sur la touche D)
Ctrl-B	Page-écran vers le haut (Maintenir la touche Ctrl enfoncée et appuyez sur la touche B)
nG	Aller à la ligne n
G	Aller à la dernière ligne
gg	Aller à la première ligne
Commandes de modification de texte	Actions
u	Annuler la dernière opération
Ctrl-r	Rétablir la dernière opération annulée (Maintenir la touche Ctrl enfoncée et appuyez sur la touche r)
.	Répéter la dernière opération de modification de texte
x	Effacer le caractère
nx	Effacer n caractère (y compris celui sous le curseur)
dw	Effacer le texte jusqu'à la fin du mot
ndw	Effacer les n prochains mots (y compris celui sous le curseur)
diw	Effacer le mot sous le curseur
.	Répéter la dernière opération d'édition
v	Commence une sélection en mode caractères
nv	surligne (mode visuel) les n lignes horizontalement du curseur vers le bas de la page
V	Sélection horizontalement toute la ligne
nV	Sélection horizontalement les n lignes à partir de celle où est le curseur
Ctrl-v	Commence une sélection verticalement
nyy	Copier la ligne (4yy = 4 lignes)
ndd	Couper la ligne (4dd = 4 lignes)

Commandes de modification de texte	Actions
p	Colle le presse-papiers à la ligne au-dessus de celle de la position courante
P	Colle le presse-papiers à la ligne au-dessous de celle la position courante
c	Supprime la sélection et passe en mode insertion
d	Supprime la sélection et la copie dans le presse-papiers (reste en mode cmd sans ex)
Esc	Quitte le mode insertion u Annule la dernière commande
r	Remplace le caractère sous le curseur par le prochain caractère tapé
nr	Remplace les n caractères à partir du curseur par le caractère tapé (exemple toto ; 3r ; on tape y ; donne yyyy)
~	Convertit le caractère sous le curseur en majuscule si c'est une minuscule et vice-versa
n~	Convertit les n caractères à partir du curseur en majuscule si c'est une minuscule et vice-versa
V~	Convertit tous les caractères minuscules de la ligne surlignée en majuscules, sauf ceux déjà en majuscule, et vice-versa
xp	Inverser les deux caractères après le curseur exemple, on a écrit "lse" on place le curseur sur le l on tape xp on obtient "les" le curseur se place sur la 1ière des deux lettres inversées
n	Rechercher l'occurrence suivante
N	Rechercher l'occurrence précédente

Aller-retour entre mode visuel et mode commande

Ajout/Supprimer de bloc de caractères	Actions
Ctrl-v-↑ Shift-I-CaractèreAjouté-Échap	Ajout de caractères identiques devant un bloc de texte On pose le curseur devant le bloc de texte on passe en mode visuel bloc (Ctrl-v) On passe en mode insertion en début de ligne (↑ Shift-I) On tape le caractère à ajouter On appuie sur Échap
caractère-Ctrl-v-x	Suppression un bloc de caractères (pas forcément identiques) verticalement On pose le curseur sur le caractère à supprimer On passe en mode visuel bloc (Ctrl-v) On tape x

Mode dernière ligne

Commandes pour quitter/enregistrer	Actions
:q	Quitter
:q!	Quitter sans enregistrer
:w	Enregistrer le registre courant
:wq	Enregistrer et quitter

Commandes pour quitter/enregistrer		Actions
:x		Enregistrer (seulement en cas de modification) et quitter
Commandes de recherche		Actions
/texte		Rechercher de la chaîne "texte"
/^texte		Recherchée la chaîne "texte" en début de ligne
/texte\$		Rechercher la chaîne "texte" en fin de ligne
/\<texte\>		Rechercher de le mot "texte"
/?texte		Recherche en arrière du texte indiqué (dans ce cas n et N sont inversés)

Remarque sur les recherches de mots :

- la commandes de recherche / ne tient pas compte des mots :

/xxx trouve les chaînes "xxx"; "aaaxxx" ; "xxxbbb" ; "aaaxxxbbb"

- Il faut utiliser \< et \> pour signifier le début et la fin d'un mot :

/\<xxx trouve "xxx" et "xxxbbb"
 /xxx\> trouve "xxx" et "aaaxxx"
 /\<xxx\> trouve seulement "xxx"



Remarque sur les options de recherches :

- **set hlsearch** : activer la surbrillance des mots trouvés
- **:nohlsearch** : effacer la surbrillance des mots trouvés
- **set ignorecase smartcase** :
 ignorecase permet fait des recherches insensibles à la casse (pour une chaîne recherchée où tous les caractères sont d'une même casse)
 /texte trouve "texte", "TEXTE" et "Texte"
 smartcase permet de repasser en mode sensible à la casse si la chaîne recherchée comporte une majuscule
 /Texte trouve "Texte" mais pas "TEXTE".

Rechercher/remplacer	Actions
:%s/A/B/g	Remplacer tous (g) les A par des B
:%s/A/B/gc	Remplacer tous (g) les A par des B et demander confirmation pour le remplacement (c)
:g/A/s//B/g	(idem :%s/A/B/g) Remplacer tous (g) les A par des B
:g/A/s//B/gc	(idem :%s/A/B/gc) Remplacer tous (g) les A par des B et demander confirmation pour le remplacement (c)
:%s/chercher/remplacer	Recherche en arrière du texte indiqué

Remarque sur les recherches et remplacements : Certains caractères spéciaux (/ & ! . ^ * \$ \ ?) ont une signification particulière pour le processus de recherche et doivent être inhibés lorsqu'ils sont utilisés pour une recherche. Pour inhiber un caractère spécial, faites-le précéder d'une barre oblique inversée (\)



- Utilisation des expressions régulières pour les recherches :



Rechercher	Actions
<code>/.exte</code>	Rechercher la chaîne commençant par par n'importe quel caractère et finissant par "exte"
<code>/te*</code>	Rechercher une chaîne commençant par "te" et finissant par zéro ou plusieurs caractères ("te" ; "texte"; "tenant"...)
<code>/[a-z]exte</code>	Rechercher une chaîne commençant par une lettre minuscule de l'alphabet et se terminant par "exte"

Pour aller plus loin pour l'utilisation des ER avec vim :

1. <http://vimregex.com/>
2. [vim et ERB](#)
3. [vim, classe, et ERb](#)

Commandes de fenêtrages	Actions
<code>:n nom-fichier</code>	Ouvrir le fichier "nom-fichier" depuis un fichier édité
<code>:sp nom-fichier</code>	Ouvrir le fichier "nom-fichier" en splittant horizontalement
<code>:sp</code>	Splitte horizontalement le fichier en cours
<code>Ctrl-w-↓</code>	Passer à la fenêtre du bas (fenêtres splittées horizontalement)
<code>Ctrl-w-↑</code>	Passer à la fenêtre du haut (fenêtres splittées horizontalement)
<code>:vsp nom-fichier</code>	Ouvrir le fichier "nom-fichier" en splittant verticalement
<code>Ctrl-w-→</code>	Passer à la fenêtre de droite (fenêtres splittées verticalement)
<code>Ctrl-w-←</code>	Passer à la fenêtre de gauche (fenêtres splittées verticalement)
<code>Ctrl-w-w</code>	Passer d'une fenêtre à l'autre (fenêtres splittées horizontalement ou verticalement)
<code>Ctrl-w-n</code>	Ouvrir un fichier vierge dans une nouvelle fenêtre
<code>:qall</code>	Quitter vim (Aucun des buffers n'a été modifiés)
<code>:qall!</code>	Quitter vim (quitte sans enregistrer les modifications de l'un ou tous les registres ouverts)
<code>wqall</code>	Enregistre chaque buffer modifié et quitte



Redimensionnement des fenêtres 

Commandes EX

Les commande Ex présente le format suivant :

```
:n°ligne,n°ligne cmdEX n°ligne
```

- **co** : Copier

Le n° de ligne où l'on veut copier doit exister ; si la dernière ligne est la ligne n°5 `1,2 co 6` ne fonctionnera pas.

- **m** : Déplacer

:1,5 m 10 déplace le bloc de la ligne 1 à la ligne 5 et place ce bloc à la ligne 10.

- **d** : Supprimer

2,4 d supprime de la ligne 2 à la ligne 4

- **r** : Copier d'un fichier dans l'autre

```
:n°ligne r autre_fichier
```

Dans le fichier en cours à la ligne "n°ligne" copie de "autre_fichier" en totalité.

Autres commandes	Actions
J	Fusionne la ligne courante et la suivante
:w fichier	Écrit le document dans le fichier indiqué
:r fichier	Inclut le fichier indiqué à partir de la position courante
ctrl+n	Complétion de vim (Le dictionnaire de complétion est fourni par l'ensemble des tampons ouverts.)
ctrl+x+f	Complétion des chemins inscrit dans le fichier édité
ctrl+x-ctrl+f	Idem avec proposition de tous les chemins commençant par ce qui est déjà inscrit
:! ls ~/<tab+tab>	Complétion des commandes shell comme dans le terminal

Range

Pour travailler avec des blocs de lignes

- **1** : la première ligne
- **\$** : la dernière ligne
- **.** : la ligne courante
- **1,\$** : de la ligne jusqu'à la dernière ligne
- **.,2** : de la ligne courante jusqu'à la deuxième ligne
- **%** : l'ensemble des lignes (texte complet)

Exemples

:23d4 : supprimer les lignes 23, 24, 25 et 26.

:+1,\$d : supprimer de la ligne courante +1 jusqu'à la fin.

:2,5p : afficher de la ligne 2 jusqu'à la ligne 5 (p = print).

1,2move5 : déplacer les lignes 1 et 2 à la ligne 5

Utilisation des registres nommés et du presse-papier

- **Vim et les registres nommés :**

Il y a autant de registre que de lettres de l'alphabet; "ayy copie la ligne dans le registre nommé a. Pour coller ce qui est dans le registre nommé a; on fait "ap... On peut copier ou couper dans un registre nommé ce qu'on veut d'un fichier, et le coller dans un autre fichier ouvert.

Il existe aussi un registre poubelle qui supprime réellement en envoyant la suppression dans "null"; c'est "_
Si l'on fait "_dd toute la ligne est réellement supprimée.

- **Notion de presse-papier sous le serveur X11 :**

Le serveur X11 dispose de trois sélections appelées : LA sélection PRIMAIRE, la SECONDAIRE et le PRESSE-PAPIERS(clipboard).

1. La sélection primaire : est normalement utilisée pour mettre en oeuvre le copier/coller via le bouton central de la souris
2. voir : <http://linux.die.net/man/1/xsel>

- **Avec Vim et presse-papiers :**

Références :

- <http://vim.wikia.com/wiki/VimTip21>
- http://vim.wikia.com/wiki/Accessing_the_system_clipboard
- <http://stackoverflow.com/questions/11489428/how-to-make-vim-paste-from-and-copy-to-system-s-clipboard>

Vim est composé de plusieurs registres (= buffers) nommés * et +

Le presse-papiers PRIMARY (contenant la dernière sélection à la souris) est en commun avec les autres applications graphiques

le CLIPBOARD (contenant la dernière sélection explicitement copiée, yy, dd, x...) peut être associé au presse-papiers primaire afin d'être en commun avec les autres applications.

Pour que le contenu du clipboard soit collable dans d'autres applications du système

Il faut :

- 1) que vim soit compilé avec l'option +clipboard et non -clipboard (pour le savoir faire dans le terminal vim -version, ou avoir 1 au retour de la commande (dans vim) :echo has('clipboard')).
- 2) ajouter la configuration suivante dans ~/.vimrc :

```
set clipboard=unnamedplus
```

- 3) si on accède au terminal qui lance vim en ssh, il faut se connecter avec l'option -X : ssh -X utilisateur@ipMachineDistante 😊

Exemples:

"*yy ou "+yy (selon les applications du système) : la ligne est copiée peut être collée dans une autre application via la souris ou shift+insert

"*%yy : copie du fichier ouvert dans sa totalité

:71,72y * : pour copier les lignes 71 et 72 afin de les coller dans une autre application via la souris ou shift+insert

:%y * : pour copier tout le fichier afin de les coller dans une autre application via la souris ou shift+insert

“*p : pour coller avant le curseur du fichier de vim le contenu du presse-papier (venant ou pas d'une autre application du système)

:76put * : pour coller à la ligne 77 du fichier de vim le contenu du press-papier (venant ou pas d'une autre application du système)

Avec le paramétrage de vim avec set clipboard=unnamedplus



Tout est les registres sont communs avec le presse-papiers primaire, ainsi,

1. si l'on supprime une ligne dans vim on remplace le contenu du presse-papier par une ligne vide
2. on ne peut utiliser les registres nommés

Commandes de modification de texte	Actions
:n fichier_1	Ouvrir le fichier “nom-fichier” depuis un fichier édité
“ayy-Ctrl-w-↓-p	Copie une ligne du fichier courant ; passe dans fichier_1 ; copie la ligne

Divers

- % sur un } permet de trouver l'autre {
- curseur sur un mot : * occurrence suivante ; # occurrence précédente

Configurations de vim ide avec pathogen

- Voir : http://vim.wikia.com/wiki/Use_Vim_like_an_IDE

Références des plugins gérés avec pathogen

- références commandes à connaître pour paramétrer ~/.vimrc:
 - Comment configurer vim : <http://www.guckles.net/vim/setup.html>
- références des plugins installés
 - Pour la gestion des plugins : <https://github.com/tpope/vim-pathogen>
 - Mise en place d'arborescence des dossiers et fichiers : <https://github.com/scrooloose/nerdtree>
 - Améliorer la gestion de l'arbre nerdtree : <https://github.com/jistr/vim-nerdtree-tabs>
 - Pour améliorer le “status line” en autre naviguer entre les onglets par buffers : <https://github.com/vim-airline/vim-airline>
 - Mode “noncompatible” amélioré, bref du posix pour config vim : <https://github.com/tpope/vim-sensible>
 - Recherche dans fichiers, tags, buffers : <https://github.com/kien/ctrlp.vim>
 - Configuration python : <https://github.com/klen/python-mode>
 - Coloration langage de programmation vim-polyglot : <https://github.com/sheerun/vim-polyglot>
 - Coloration pour vim-polyglot : complétion html5 : <https://github.com/othree/html5.vim>

- Complétion : voir <https://github.com/vim-scripts/AutoComplPop>
- Débugger :
 - <https://mutelight.org/minimal-guide-to-debugging-php-with-xdebug-and-vim>
 - <https://www.octopuce.fr/debugger-php-en-cli-avec-xdebug-vim/>

Installation

- On installe git-core :

```
apt-get install git-core
```

- On télécharge les fichiers suivant :



On vérifie que son navigateur envoie les téléchargements dans le dossier ~/Téléchargements

vimrc

MonVimrc

```
" .vimrc fichier modifié par Hypathie
"
=====
====
" Configuration générales
"
=====
====
"if a line is already longer than textwidth when inset mode is started,
the line may not be wrapped even if text is added to the
"line.

filetype plugin indent on
syntax on

" ## Encodage

" Encodage par défaut des buffers en utf-8
set encoding=utf-8
" Encodage par défaut des fichiers en utf-8
set fileencoding=utf-8

" Couleur
" J'ai choisi le thème wombat256mod
" mkdir -p ~/.vim/colors && cd ~/.vim/colors
```

```
" wget -O wombat256mod.vim
http://www.vim.org/scripts/download_script.php?src_id=13400
" Pour avoir plusieurs couleurs au choix voir :
http://vim.wikia.com/wiki/Switch_color_schemes
set t_Co=256
color wombat256mod

" Recharger ~/.vimrc automatiquement lorsqu'il est édité
autocmd! bufwritepost .vimrc source %

" retour-chariot
set bs=2 " make backspace behave like normal again

" Gestion du mapping. Voir :
" http://vim.wikia.com/wiki/Mapping_keys_in_Vim_-_Tutorial_%28Part_1%29
" http://vim.wikia.com/wiki/Mapping_keys_in_Vim_-_Tutorial_%28Part_2%29
" http://vim.wikia.com/wiki/Mapping_keys_in_Vim_-_Tutorial_%28Part_3%29
" Pour que la touche <Leader> soit la virgule
let mapleader = ","

" Pour supprimer le surlignage issu d'une recherche Ctrl-n
noremap <F4> :nohl<CR>
vnoremap <F4> :nohl<CR>
inoremap <F4> :nohl<CR>

" Indenter un bloc
" Sélectionner un bloc en mode visual et utiliser > ou < pour déplacer
le bloc
" vers la droite ou vers la gauche
vnoremap < <gv " better indentation
vnoremap > >gv " better indentation

" Afficher les numéros de ligne
set number " Afficher les numéros*
" Faire disparaître/réapparaître les numéros de ligne

" touche F3
map <silent> <F3> :set invnumber<cr>

" La longueur des lignes
set tw=129 " tw=79 par défaut
" tw=0 pour désactiver
" correspond à textwidth voir :
http://vim.wikia.com/wiki/Automatic_word_wrapping
"set nowrap " 'wrap : envelopper la ligne'
" set nowrap : ne pas mettre en place la taille lors du
chargement du fichier édité
"set fo+=t " mise en place du tag 't'. Voir
http://vim.wikia.com/wiki/Word_wrap_without_line_breaks
```

```
" Colonnes
set colorcolumn=130 " Avoir une colonne verticale à 80 caractères du
bord gauche
highlight ColorColumn ctermbg=233 " couleur grise de la colonne

" Mise en forme des paragraphes
vmap Q gq
nmap Q gqap

" Historique des commandes et de suppression des dernières
modifications
set history=700
set undolevels=20 "http://vim.wikia.com/wiki/Undo_and_Redo
" Utiliser quatre espaces à la place d'une tabulation
set tabstop=4
set softtabstop=4
set shiftwidth=4
set shiftround
set expandtab

" Recherche insensible à la case
set hlsearch
set incsearch
set ignorecase
set smartcase

" Fait une copie de sauvegarde lors de l'écrasement d'un fichier
set backup
" Dossier contenant la sauvegarde. N'oubliez pas de le créer et de lui
faire un
" chmod 700
set backupdir=~/.cache/vim/backup

" " ## Sessions et Views
" set viewdir=~/.local/var/vim/views

" Pour ne pas contrarier pathogen
filetype off " voir http://vim.wikia.com/wiki/Filetype.vim

" Collage et presse-papier
" Avec l'option set paste vim ajoute des indentations lors du collage
dans le fichier vim
" Avec set nopaste il respecte les indentations déjà existantes du
fichier
" copié
" Avec l'option set pastetoggle=<F2> on passe de l'option nopaste à
l'option
" paste en appuyant sur F2, il faut alors commenter les options set
paste
" et set nopaste
" voir http://vim.wikia.com/wiki/Toggle_auto-indenting_for_code_paste
```

```

"set paste
"set nopaste
set pastetoggle=<F2>

" Pour que le presse-papier primaire (celui du système) soit le buffer
sans nom (évite de taper par
set clipboard=unnamedplus
map <C-c> "+y<CR>

" gestion du statusline voir "
http://vim.wikia.com/wiki/Displaying_status_line_always
set laststatus=2

" Neocomplete native dans vim 7.3.885+ voir vim --version
" voir https://github.com/Shougo/neocomplete.vim
let g:neocomplete#enable_at_startup = 1

" ou pour completion commande vim omnifunc
" voir : http://vim.wikia.com/wiki/Omni_completion
set omnifunc=syntaxcomplete#Complete

" Pour afficher un menu de completion
" Voir :
http://vim.wikia.com/wiki/Great_wildmode/ildmenu_and_console_mouse
set wildmenu
set wildmode=list:longest,full

"
=====
=====
" gestion des plugins
"
=====
=====

"$ VIMRUNTIME pointe vers un répertoire où la plupart des fichiers
requis par
"Vim sont placés
"Voir : http://vim.wikia.com/wiki/Understanding_VIMRUNTIME
runtime ~/.vim/bundle/pathogen/autoload/pathogen.vim

" #### pathogen ####

" Charger pathogen
call pathogen#infect()
"call pathogen#helptags()

" Mise en place de pathogen pour la gestion des plugins
" mkdir -p ~/.vim/autoload ~/.vim/bundle
" curl -so ~/.vim/autoload/pathogen.vim

```

```

" https://raw.githubusercontent.com/tpope/vim-pathogen/HEAD/autoload/pathogen.vim

" On peut installer maintenant les plugins dans le dossier
" .vim/bundle/plugin-name/

" ##### nerdtree #####

" Pour que l'arbre de nerdtree soit à droite
let g:NERDTreeWinPos = "right"
"" remarque ctrl-w-x pour basculer nerdtree de droite à gauche de la
fenêtre

" Pour lancer nerdtree à l'ouverture de vim
" autocmd VimEnter * NERDTree
" Pour que le curseur soit placé dans l'arbre de nerdtree au démarrage
de vim
" autocmd VimEnter * wincmd p
" Donc pour que le curseur soit dans l'arbre de nerdtree si on lance
vim sans
" ouvrir un fichier mais que le curseur soit dans la fenêtre du fichier
ouvert
" avec vim
autocmd VimEnter * NERDTree " lancer NERDTree au démarrage de vim
autocmd VimEnter * if argc() | wincmd p | endif
nnoremap <leader>q :bp<cr>:bd #<cr> " quitter un onglet (buffer)
                                "sans quitter vim totalement

" ##### vim-airline #####
" git clone https://github.com/vim-airline/vim-airline.git
" ~/.vim/bundle/vim-airline
" mettre une ligne d'onglet au dessus (1 onglet=1 buffer)
let g:airline#extensions#tabline#enabled = 1

" use mouse in nerdtree :
" voir si on a : ":set mouse=a"
"let g:NERDTreeMouseMode=3

" raccourcit pour naviguer entre les fichiers (fichier = buffer)
nmap <Leader>n :bn<CR> " map mode normal(nmap) puis ,
                        "-g ou ,-l (cmd vim pour naviguer entre
buffers),
                        "<CR> retour à la ligne
nmap <Leader>p :bp<CR>

set mouse=a

" ##### vim-nerdtree-tabs #####

" configuration de vim-nerdtree-tabs
" voir https://github.com/jistr/vim-nerdtree-tabs

```

```

" Pour faire apparaître/diparaître l'arbre de nerdtree
map <Leader>t <plug>NERDTreeTabsToggle<CR>

"set hidden

" Settings for ctrlp
" cd ~/.vim/bundle
" git clone https://github.com/kien/ctrlp.vim.git
let g:ctrlp_max_height = 30
set wildignore+=*.pyc
set wildignore+=*_build/*
set wildignore+=*/coverage/*

" ##### HTML + CSS
#####
" ##### Vim-polygot
" ##### cd ~/.vim/bundle
" ##### git clone https://github.com/sheerun/vim-polyglot.git
" ##### doc voir : https://github.com/sheerun/vim-polyglot
" ##### Exemple pour désactiver le css
" ##### vim-polygot
" ##### let g:polyglot_disabled = ['css']
"
#####
#####

" Configuration pour html5
" "
" avec polyglot dans ~/.vim/bundle/html5.vim.git
" cd ~/.vim/bundle/html5.vim.git
" git clone https://github.com/othree/html5.vim.git

" " Settings for css
" "
" avec polygot dans ~/.vim/syntax/

" ##### PHP
#####
"
" Configuration native : A mettre en toute fin de .vimrc

"function! PhpSyntaxOverride()
" hi! def link phpDocTags phpDefine
" hi! def link phpDocParam phpType
"endfunction
"
"augroup phpSyntaxOverride

```

```

" autocmd!
" autocmd FileType php call PhpSyntaxOverride()
"augroup END

" ### ma configuration développement php :

"
http://blogs.fsfe.org/dervishe/2012/09/27/un-environnement-de-developpe
ment-php-avec-vim/

" bloc de commentaires php :
" http://www.vim.org/scripts/script.php?script_id=1355
" CTRL+o en mode édition ajoutera un bloc commentaire avec des infos à
" compléter.
"For example include into your .vimrc:

"source ~/.vim/php-doc.vim
"inoremap <C-P> <ESC>:call PhpDocSingle()<CR>i
"nnoremap <C-P> :call PhpDocSingle()<CR>
"vnoremap <C-P> :call PhpDocRange()<CR>

"source ~/.vim/plugin/php-doc.vim
"imap <C-o> <ESC>:set paste<CR>:call PhpDoc()<CR>:set nopaste<CR>

" complete.vim pour la complétion php : fichier word_complte.vim
" voir http://www.vim.org/scripts/script.php?script_id=73 :

" mkdir ~/.vim/plugin/
" cd ~/.vim/plugin/
" wget http://www.vim.org/scripts/download_script.php?src_id=6504
" mv download_script.php\?src_id\=6504 word_complete.vim

" On modifie ~/.vim/plugin/word_complete.vim ainsi :
"          " ===== file word_complete.vimrc
=====
"          " User Configuration file for word_complete.vim .
"          " To use this, uncomment and change the defaults.
"
"          " Do not complete words shorter than this length:
"          let g:WC_min_len = 3
"          " Use this key to accept the offered completion:
"          let g:WC_accept_key = "<Tab>"
"          " ===== end: word_complete.vimrc
=====

" http://www.vim.org/scripts/script.php?script_id=3171 : dans
~/.vim/autoload/

" http://www.scintilla.org/php.api : dans ~/.vim/syntax/

```



```
"
"set complete=.,w,b,u,t,i,k~/.vim/syntax/php.api
"autocmd FileType php set omnifunc=phpcomplete#CompletePHP
"source ~/.vim/plugin/word_complete.vim
"call DoWordComplete()

" la vérification de votre code avec php:
"on va redéfinir la commande makeprg (make programme) de vim pour la
faire
"pointer vers la version cli de php avec l'option -l:

"set makeprg=/usr/lib64/php5.4/bin/php\ -l\ %
"nmap <F9> :make<ENTER>:copen<ENTER><CTRL>L

" debug php
" utiliser une extension de PHP xdebug. Celle-ci s'installe donc sur le
" serveur et se déclenche automatiquement (selon sa configuration)
lorsque le
" serveur reçoit une requête http avec le paramètre
XDEBUG_SESSION_START=1.

" installation de xdebug depuis les dépôts debian:
" command apt-get -y install php5-xdebug

"Rechargez la configuration:
" test -e '/etc/init.d/php5-fpm' && command service 'php5-fpm'
'restart'
" test -e '/etc/init.d/apache2' && command service 'apache2' 'force-
reload'

" éditer /etc/php5/mods-available/xdebug.ini
"
" ajouter à la fin
"xdebug.remote_enable=on
"xdebug.remote_handler=dbgp
"xdebug.remote_host=localhost
"xdebug.remote_port=9000

" les 2 autres fichiers :
" /etc/php5/apache2/conf.d/20-xdebug.ini
" /etc/php5/cli/conf.d/20-xdebug.ini
" sont modifiés après enregistrement des ajouts de
" /etc/php5/mods-available/xdebug.ini

" puis : service apache2 restart

"
=====
```

```

=====
" Python IDE Setup
"
=====
=====
"" Settings for vim-powerline
" cd ~/.vim/bundle
" git clone git://github.com/Lokaltog/vim-powerline.git
"set laststatus=2
"
"" Settings for ctrlp
"" cd ~/.vim/bundle
"" git clone https://github.com/kien/ctrlp.vim.git
"let g:ctrlp_max_height = 30
"set wildignore+=*.pyc
"set wildignore+=*_build/*
"set wildignore+=*/coverage/*
"
"" Settings for python-mode
"" cd ~/.vim/bundle
"" git clone https://github.com/klen/python-mode
"" map <Leader>g :call RopeGotoDefinition()<CR>
"" let ropevim_enable_shortcuts = 1
"" let g:pymode_rope_goto_def_newwin = vnew
"" let g:pymode_rope_extended_complete = 1
"" let g:pymode_breakpoint = 0
"" let g:pymode_syntax = 1
"" let g:pymode_syntax_builtin_objs = 0
"" let g:pymode_syntax_builtin_funcs = 0
"" let g:pymode_rope_lookup_project = 0
"map <Leader>b Oimport ipdb; ipdb.set_trace() # BREAKPOINT<C-c>
"
"" Better navigating through omnicomplete option list
"" See http://stackoverflow.com/questions/2170023/how-to-map-keys-for-popup-menu-in-vim
"set completeopt=longest,menuone
"function! OmniPopup(action)
"  if pumvisible()
"    if a:action == 'j'
"      return "\<C-N>"
"    elseif a:action == 'k'
"      return "\<C-P>"
"    endif
"  endif
"  return a:action
"endfunction
"
"ignoremap <silent><C-j> <C-R>=OmniPopup('j')<CR>
"ignoremap <silent><C-k> <C-R>=OmniPopup('k')<CR>
"
"" Python folding

```

```

"" mkdir -p ~/.vim/ftplugin
"" wget -O ~/.vim/ftplugin/python_editing.vim
http://www.vim.org/scripts/download_script.php?src_id=5492
"set nofoldenable
"
"
"" turn off auto complete
"" let g:pymode_rope_completion = 0
"" let g:pymode_rope_complete_on_dot = 0

```

configVim.sh

configVim.sh

```

#!/bin/sh

#####
# script d'installation pour la configuration de vim en tant qu'IDE #
# créé par Hypathie _ Mars 2016                                     #
#####

lancerInstall ()
{
    if [ -f /home/$USER/Téléchargements/MonVimrc ]; then
        if [ -d /home/$USER/.vim/ ]; then
            printf "le dossier /home/$USER/.vim/ existe déjà !\n"
            while ([ "$repl" != "0" ] && [ "$repl" != "o" ] && [
"$repl" != "N" ] && [ "$repl" != "n" ] );
            do
                read -p "Voulez-vous continuer: oui (0/o) ou non
(N/n) ?" repl;
                case $repl in
                    N|n)
                        printf "Arrêt de l'installation.\n"
                        exit 1
                        ;;
                    0|o)
                        printf "Déplacement des dossiers
/home/$USER/.vim/ dans le dossier /home/$USER/recup\n"
                        mkdir -p /home/$USER/recup/
                        mv /home/$USER/.vim/
/home/$USER/recup/vim/
                        printf "Votre dossier /home/$USER/.vim/
a été sauvegardé avec succès\n"
                        mkdir -p /home/$USER/.vim/autoload/
/home/$USER/.vim/bundle/ /home/$USER/.vim/colors/
/home/$USER/.vim/ftplugin/ /home/$USER/.vim/syntax/
                        esac

```

```

done
else
    mkdir -p /home/$USER/.vim/autoload/
/home/$USER/.vim/bundle/ /home/$USER/.vim/colors/
/home/$USER/.vim/ftplugin/ /home/$USER/.vim/syntax/
fi

if [ -f /home/$USER/.vimrc ] ; then
    printf "Un fichier /home/$USER/.vimrc existe déjà !\n"
    while ([ "$rep2" != "0" ] && [ "$rep2" != "o" ] && [
"$rep2" != "N" ] && [ "$rep2" != "n" ]);
    do
        read -p "Voulez le remplacer ? O/o ou N/n ?" rep2
        case $rep2 in
            N|n)
                printf "Arrêt de l'installation.\n"
                mv /home/$USER/recup/vim/ ~/.vim/
                cd /home/$USER/
                rm -rf recup/
                cd /home/$USER/

                exit 1
                ;;
            O|o)
                mv /home/$USER/.vimrc ~/recup/vimrcOLD
                cp /home/$USER/Téléchargements/MonVimrc
/home/$USER/.vimrc
                esac
        done
    else
        cp /home/$USER/Téléchargements/MonVimrc /home/$USER/.vimrc
    fi
else
    printf "Veuillez télécharger le fichier MonVimrc ou vérifier qu'il
est dans ~/Téléchargements avant d'exécuter ce script.\n"
    exit 2
fi

if [ -d /home/$USER/.vim/autoload/ ] ; then
    printf "Installation de pathogen dans
/home/$USER/.vim/autoload/.....\n"
    cd /home/$USER/.vim/autoload/
    wget https://raw.githubusercontent.com/tpope/vim-
pathogen/master/autoload/pathogen.vim
    printf
".....
.\n"
    cd /home/$USER/
fi

if [ -d /home/$USER/.vim/bundle/ ] ; then
    printf "Installation de nerdtree dans ~/.vim/bundle/

```

```
.....\n"
  cd /home/$USER/.vim/bundle/
  git clone https://github.com/scroolose/nerdtree
  printf
" .....\n"
  printf "Installation de vim-nerdtree-tabs.git dans
~/vim/bundle/.....\n"
  git clone https://github.com/jistr/vim-nerdtree-tabs.git
  printf
" .....\n"
  printf "Installation de vim-airline dans ~/.vim/bundle/
.....\n"
  git clone https://github.com/vim-airline/vim-airline.git
  printf
" .....\n"
  printf "Installation de vim-sensible.git dans ~/.vim/bundle/
.....\n"
  git clone https://github.com/tpope/vim-sensible.git
  printf
" .....\n"
  printf "Installation de ctrlp.vim dans ~/.vim/bundle/
.....\n"
  git clone https://github.com/kien/ctrlp.vim.git
  printf
" .....\n"
  printf "Installation de vim-polyglot.git dans ~/.vim/bundle/
.....\n"
  git clone https://github.com/sheerun/vim-polyglot.git
  printf
" .....\n"
  printf "Installation de Html5.vim.git dans ~/.vim/bundle/
.....\n"
  git clone https://github.com/othree/html5.vim.git
  printf
" .....\n"
  printf "Installation de python-mode.git dans ~/.vim/bundle/
.....\n"
  git clone https://github.com/klen/python-mode.git
  printf
" .....\n"
  printf "Installation de php.vim.git dans ~/.vim/bundle/
.....\n"
  git clone https://github.com/StanAngeloff/php.vim.git
  printf
" .....\n"
  printf "Installation de la couleur wombat256.vim
.....\n"
  cd ~/.vim/colors
  wget
https://raw.githubusercontent.com/thesheff17/youtube/master/vim/wombat2
56mod.vim
```

```

printf
".....\n"
printf "Installation de python_editing.vim dans ~/.vim/bundle/
.....\n"
cd ~/.vim/ftplugin
wget
https://raw.githubusercontent.com/thesheff17/youtube/master/vim/python_
editing.vim
printf
".....\n"
printf "Installation de css.vim dans ~/.vim/syntax/
.....\n"
cd ~/.vim/syntax/
wget https://github.com/JulesWang/css.vim.git
cd ~
fi

printf "Configuration système pour le cache de vimrc\n"
printf "Création du dossier caché /home/$USER/.cache/vim/backup\n"
if (! [ -d /home/$USER/.cache/vim/backup/ ] ) ; then
mkdir -p /home/$USER/.cache/vim/backup/
printf "Le dossier caché /home/$USER/.cache/vim/backup a été
créé avec succès.\n"
printf
".....\n"
else
printf "Le dossier caché /home/$USER/.cache/vim/backup existait
déjà, il a été laissé en l'état.\n"
fi
printf
".....\n"
printf ".....Suppression des fichiers
d'installation.....\n"
rm ~/Téléchargements/configVim.sh && rm ~/Téléchargements/MonVimrc
printf "Les fichiers d'installation ont été supprimés avec
succès.....\n"
printf
".....\n"
printf "..... Installation terminée
.....\n"
printf "Informations ..... \n"
printf "Pour supprimer cette installation lancer dans un terminal
en utilisateur : \n"
printf "rm -rf ~/.vim/ && rm -r ~/.vimrc\n"
printf ".....\n"
printf "Si vous avez sauvegardé une ancienne configuration pendant
l'installation \n"
printf "Pour la récupérer, lancer en utilisateur : \n"
printf "mv ~/recup/vim/ ~/.vim/ && mv ~/recup/vimrcOLD ~/.vimrc\n"

```

```

    printf "Puis pour supprimer le dossier de sauvegarde de l'ancienne
configuration :\n"
    printf "rm -rf ~/recup/\n"
    printf
".....\n"
}

if [ -d /home/$USER/recup/ ]; then
    printf
".....\n"
    printf
".....\n"
    printf " Attention il semblerait que vous avez déjà exécuté ce
script !\n"
    printf " Si vous avez déjà lancé ce script et que vous continuez la
sauvegarde de votre ancienne configuration pourrait être définitivement
perdue !\n"
    printf
".....\n"
    while ( [ "$rep" != "0" ] && [ "$rep" != "o" ] && [ "$rep" != "N" ]
&& [ "$rep" != "n" ] );
    do
        read -p "Voulez-vous continuer: oui (O/o) ou non (N/n) ?" rep;
        case $rep in
            N|n)
                printf
".....\n"
                printf ".....Arrêt de
l'installation.....\n"
                printf "\n"
                printf " Avant d'exécuter à nouveau ce script
renommez\n"
                printf " le dossier /home/$USER/recup\n"
                printf " afin de conserver une ancienne
configuration\n"
                printf "\n"
                printf
".....\n"
                exit 3
                ;;
            O|o)
                printf "suppression de /home/$USER/recup.....\n"
                rm -rf /home/$USER/recup/
                printf ".....\n"
                lancerInstall
                esac
        done
    else
        lancerInstall

```

```
fi
exit 0
```

- On installe cette configuration

```
cd ~/Téléchargements && chmod +x configVim.sh
```

```
./configVim.sh
```

A voir : Se passer de plugin

- Pour l'utilisation native de vim équivalente à NerdTree voir : <https://blog.mozhu.info/vimmers-you-dont-need-nerdtree-18f627b561c3#.bj6o4k1os>

Mes raccourcis clavier

mapleader

```
let mapleader = ","
```

- faire apparaître/disparaître l'arbre de nerdTree : ,+t
- passer d'un onglet à l'autre (ouvert depuis l'arbre de nerdtree : ,+n (:bn) ou ,+p (:bp)
- fermer une fenêtre-onglet sans fermer vim : ,+q

utilisations des touches de fonction

- F2 : mode collage
- F3 : numéros de ligne
- F4 : effacer le surlignage

CTRL

- passer du fichier édité à l'arbre de nerdtree : ctrl+w

Utilisation de nerdtree

créer, modifier, supprimer un fichier ou dossier

- m : faire apparaître le menu
- a : ajouter un fichier
si nom fichier terminé par / ça crée un dossier
- m : modifier le nom (= mv)
- d : supprimer le fichier sélectionner

- c : copier le fichier sélectionner

quelques commandes nertree

- r : sur le noeud pour rafraîchir le répertoire courant
- o : ouvrir un répertoire

voir : <http://ngressier.developpez.com/articles/vim/vim-plugin-nerdtree/>

From:

<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:

<http://debian-facile.org/utilisateurs:hypathie:tutos:vim>



Last update: **16/04/2017 09:20**