

Introduction à vim

1. Commandes de base :

1.1 Complétez le tableau ci-dessous :

Mode	Raccourci ou commande	Raccourci alternatif	Description
visuel	Flèche gauche	h	Se déplacer vers la gauche
visuel	Flèche droite	l	Se déplacer vers la droite
visuel	Flèche haut	k	Se déplacer vers le haut
visuel	Flèche bas	j	Se déplacer vers le bas
commande	i	---	passage en mode insertion
commande	v	---	passage en mode visuel
insertion	échap	---	passage en mode commande
visuel	échap	---	passage en mode commande
commande	V	---	passage en mode visuel (ligne)
commande	:q	---	quitter
commande	:q!	---	quitter sans enregistrer
commande	:w	---	enregistrer le fichier
commande	:wq	---	enregistrer le fichier et quitter

2. Edition de fichiers textes

2.1 Complétez le tableau ci-dessous :

Mode	Raccourci ou commande	Raccourci alternatif	Description
commande	yy	---	copier la ligne courante
commande	cc	---	couper la ligne courante
commande	p	---	coller après
commande	P	---	coller avant
insertion	CTRL+dw	---	effacer le mot précédant
commande	gg	---	sauter au début de la première ligne du fichier
commande	G	---	sauter au début de la dernière ligne du fichier
commande	^	---	sauter au début de la ligne courante
commande	\$	---	sauter à la fin de la ligne courante
commande	:Ligne	---	sauter à la ligne n°
commande	u	---	undo
commande	CTRL+R	---	redo
commande	:set paste	---	passer en mode collage (paste)
commande	:set nopaste	---	sortir du mode collage (paste)

2.1 Expliquez l'intérêt du mode « paste »

Il n'y a pas d'auto indentation. Si a une ligne il y a un TAB au début et qu'on presse entrée pour passer à la ligne d'après on passera au vrai début de la ligne et pas au même endroit que le TAB

3. Rechercher / remplacer

3.1 Complétez le tableau ci-dessous :

Mode	Raccourci ou commande	Description
commande	/	rechercher une chaîne de caractère
commande	n	occurrence suivante
commande	N	occurrence précédente
commande	:%s/CHERCHE/REPLACE	remplacer une expression dans tout le fichier
commande	:#,#s/ancien/nouveau/g	Remplacer une expression dans une zone donnée

4. Edition de fichiers multiples

4.1 Expliquez la différence entre buffers et fenêtres

buffer est la mémoire (il n'est pas affiché) alors que la fenêtre c'est ce que l'on voit

4.2 Complétez le tableau ci-dessous :

Mode	Raccourci ou commande	Description
commande	:buffers	lister les buffers actuellement ouverts
commande	:edit [FICHIER]	ouvrir un nouveau buffer
commande	:bn	passer au buffer suivant
commande	:bp	passer au buffer précédent
commande	:buffer n°	passer au buffer numéro <n>
commande	:bd	Fermer le buffer courant
commande	:split	séparer l'écran (horizontalement) en deux fenêtres
commande	:vsplit	séparer l'écran (verticalement) en deux fenêtres
commande	CTRL+W+Flèche	changer de fenêtre
commande	CTRL+W puis c	fermer la fenêtre courante

5. Commandes pour le développement, l'édition de code

5.1 Complétez le tableau ci-dessous :

Mode	Raccourci ou commande	Description
commande	>>	augmenter l'indentation de la ligne
commande	<<	diminuer l'indentation de la ligne
commande	set autoindent	indenter automatiquement la ligne
commande		indenter automatiquement la selection
commande	=G	réindenter tout le fichier
commande	:! [commande]	executer une commande shell
commande	:! [commande] %	executer une commande shell sur le fichier courant (ex : wc -l)
commande	:read! [commande] %	executer une commande shell et récupérer la sortie dans le buffer

6. Intégration de plugins avec vundle

Installez, à l'aide de **pathogen**, les plugins suivants :

- nerdcommenter
- vim-airline
- vim-airline-themes
- vim-fugitive
- vim-gitgutter

ainsi qu'au moins deux autres plugins de votre choix, que vous trouvez utiles.

Vous détaillerez ci-dessous les commandes utilisées pour déployer ces plugins et fournirez l'adresse d'un repo GIT accessible à l'enseignant contenant :

- Un tableau des commandes qui semblent utiles dans ces plugins
- Votre repertoire .vim
- Votre fichier vimrc

La notation portera sur le travail réalisé afin de personnaliser votre vim et sur son originalité par rapport aux tutoriaux disponibles.

Pour commencer j'installe Pathogen grâce aux commandes suivantes :

```
mkdir -p ~/.vim/autoload ~/.vim/bundle
```

```
sudo apt install curl
```

```
curl -Lo ~/.vim/autoload/pathogen.vim https://raw.githubusercontent.com/tpope/vim-pathogen/master/autoload/pathogen.vim
```

```
cd ~/.vim/
```

Ensuite dans le fichier .vimrc j'y ajoute les lignes suivantes :

```
vim ~/.vimrc  
execute pathogen#infect()  
syntax on  
filetype plugin indent on
```

Maintenant Pathogen est fonctionnel et il ne me reste plus qu'à installer des plugins dans le dossier bundle :

```
cd ~/.vim/bundle/
```

Pour les plugins on peut utiliser le site de vim pour les trouver et les installer :

<https://vimawesome.com/>

Plugin nerdkommenter :

```
git clone https://github.com/preservim/nerdkommenter.git
```

Plugin nerdtree :

```
git clone https://github.com/preservim/nerdtree.git
```

Plugin vim-airline :

```
git clone https://github.com/vim-airline/vim-airline
```

Plugin vim-fugitive :

```
git clone https://github.com/tpope/vim-fugitive
```

Plugin vim-gitgutter :

```
git clone https://github.com/airblade/vim-gitgutter
```

Plugin vim-colors-solarized :

```
git clone https://github.com/altercation/vim-colors-solarized
```

Commandes utiles pour le plugin nerdtree :

Ce plugin permet via vim d'avoir une arborescence de tout le système et de se déplacer dedans tout en restant dans vim. Ainsi je peux éditer des fichiers et les sauvegarder puis continuer de me déplacer dans le système

	Commande sur les fichiers
double-clic, o	ouvrir un fichier
go	aperçu du fichier
t	ouvrir un fichier dans un onglet

	Commande sur les répertoires
double-clic, o	ouvrir/fermer le répertoire
O	ouvrir le répertoire et tous les sous-répertoires

Commandes utiles pour le plugin nerdcommenter :

	" Add spaces after comment delimiters by default let g:NERDSpaceDelims = 1
	" Use compact syntax for prettified multi-line comments let g:NERDCompactSexyComs = 1
	" Align line-wise comment delimiters flush left instead of following code indentation let g:NERDDefaultAlign = 'left'
	" Set a language to use its alternate delimiters by default let g:NERDAltDelims_java = 1
	" Add your own custom formats or override the defaults let g:NERDCustomDelimiters = { 'c': { 'left': '/*', 'right': '*/' } }
	" Allow commenting and inverting empty lines (useful when commenting a region) let g:NERDCommentEmptyLines = 1
	" Enable trimming of trailing whitespace when uncommenting let g:NERDTrimTrailingWhitespace = 1
	" Enable NERDCommenterToggle to check all selected lines is commented or not let g:NERDToggleCheckAllLines = 1

Commandes utiles pour le plugin vim-airline :

NORMAL	vimrc	vim	utf-8[dos]	100%	h 826: 1
INSERT	vimrc	vim	utf-8[dos]	100%	h 826: 14
VISUAL	vimrc	vim	utf-8[dos]	99%	h 825: 1

On peut paramétrer les couleurs du plugin dans le fichier vimrc

Commandes utiles pour le plugin vim-fugitive :

:Gedit - éditer un fichier dans l'index et l'écrire pour mettre en scène les modifications
:Gstatus - équivalent du git status
:Gblame - fait apparaître une division verticale de la sortie de git blame
:Gmove - pour git mv
:Gremove - pour git rm
:Git - lance n'importe quelle commande

Commandes utiles pour le plugin vim-gitgutter :

Eteindre le Gitgutter - :GitGutterDisable
Allumer le Gitgutter - :GitGutterEnable

Dépôt GIT avec les dossiers : <https://github.com/CooKiiieS/TP-VIM-LP-IDO>