

Expressions régulières

dans *grep*, *vi*, *sed*, ... les caractères `?`, `+`, `{`, `|`, `(`, `)` doivent être précédés d'un antislash (`\`)

```
. n'importe quel caractère
* caractère précédent 0 fois ou +
+ caractère précédent 1 fois au moins
? le caractère précédent 0 ou 1 fois
{n} le caractère précédent n fois
{m,n} le caractère précédent de m à n fois
{n,} le caractère précédent n fois ou plus
[a-z] un caractère en minuscule
[a-zA-Z] une lettre
[0-9] un chiffre
^/$ le début/la fin de ligne
| séparateur (ou logique)
[[:alnum:]] [[:alpha:]] ou [[:digit:]]
[[:digit:]] [0-9]
[[:alpha:]] [[:upper:]] ou [[:lower:]]
[[:space:]] espace ou tabulation
[[:lower:]] [a-z] ([[:upper:]] [A-Z])
[:print:] caractère imprimable ou espace
[:xdigit:] [A-Fa-f0-9]
```

Raccourcis bash

```
<ctrl-a> début de ligne
<ctrl-e> fin (end) de ligne
<ctrl-→> mot suivant
<ctrl-←> mot précédent
<ctrl-k> couper (kill) la fin de la ligne
<ctrl-y> coller (yank)
<ctrl-d> efface (delete) le caractère à droite du curseur
<esc>d efface le mot à droite du curseur
<ctrl-t> transposer deux caractères
<esc>t transpose le mot précédent et le suivant
<ctrl-r> rechercher dans l'historique
<ctrl-l> effacer l'écran (équivalent à clear)
<ctrl-d> envoie le caractère 'fin de fichier'
<ctrl-c> arrêter la commande en cours
<ctrl-z> stopper la commande en cours
<tab> complète l'élément
<tab><tab>affiche les complétions possibles
```

Historique et job control bash

```
!! ré-exécuter la dernière commande
!txte exécuter la dernière commande commençant par texte
!n exécuter la commande numéro n
cmd & exécuter la commande cmd en arrière plan
fg %n mettre le travail n en avant plan (sans argument : le dernier travail)
bg %n mettre le travail n en arrière plan (sans argument : le dernier travail)
kill %n terminer le travail n
jobs affiche la liste des travaux du shell
history affiche l'historique des commandes
```

Raccourcis less (ou man)

```
<espace> avance d'une page
<return> avance d'une ligne
n<return> avance de n lignes
b recule (back) d'une page
nb recule de n lignes
q quitter less
q /texte rechercher la chaîne texte
n rechercher l'occurrence suivante
> va au début du fichier
< va à la fin du fichier
e dite le fichier en cours
!command exécute command dans un shell
```

Vi

Modes

```
i (<ins>) mode insertion
<esc> mode commande
o ajouter une ligne après la ligne courante et passer en insertion
```

Sortir, sauver

```
:w sauver le fichier
:w nom sauver dans le fichier nom
:w>>nom sauver à la fin du fichier nom
:q quitter sans sauver
:q! quitter de force sans sauver
:x (:wq) sauver et quitter
:e nom ferme le fichier courant et édite le fichier nom
:e! nom ferme le fichier courant de force et édite le fichier nom
:e! annule tous les changements et revient au fichier original
```

Déplacements

```
h déplacement vers la droite
j déplacement vers le bas
k déplacement vers le haut
l déplacement vers la gauche
^ début de ligne
^^ fin de ligne
$ première ligne (ou 1G)
gg dernière ligne
nG ligne n (ou :n)
w mot suivant
b mot précédent
<ctrl-f> page suivante
<ctrl-b> page précédente
fx aller après le prochain caractère x de la ligne
tx aller avant le prochain caractère x de la ligne
H aller en haut de l'écran
M aller au milieu de l'écran
L aller en bas de l'écran
```

Copier, couper, coller

```
x coupe le caractère sous le curseur
YY copie la ligne courante entière
y^ copie la ligne du début au curseur
y$ copie la ligne du curseur à la fin
yw copie le mot suivant
dd coupe la ligne courante entière
d^ coupe la ligne du début au curseur
d$ coupe la ligne du curseur à la fin
dw coupe le mot suivant
P colle avant la position courante
p colle après la position courante
```

Numérations

vi permet d'appliquer une commande à n zones

```
nyy copie les n lignes suivantes
n dd coupe les n lignes suivantes
nyw coupe les n mots suivants
ndw coupe les n mots suivants
...
```

Rechercher

```
/motif recherche motif vers le bas
?motif recherche motif vers le haut
n recherche la prochaine occurrence
p recherche l'occurrence précédente
```

```
* aller à la prochaine occurrence du mot sous le curseur
# aller à l'occurrence précédente du mot sous le curseur
```

Remplacer

```
:m,n,s/a/b/g remplace a par b entre les lignes m et n
:1,$s/a/b/g remplace a par b dans tout le fichier
:~s/a/b/g idem
```

Signets

```
mx crée le signet x
'x va au signet x
d'x coupe jusqu'au signet x
'x aller à la dernière ligne éditée
```

Macros

```
qx enregistre la macro x
q termine l'enregistrement de la macro
@x invoque la macro x
n@x invoque n fois la macro x
```

Registres

```
:reg liste les registres et leur contenus
"n sélectionne le registre n
"nd$ coupe la fin de la ligne courante et la sauve dans le registre n
"np colle le contenu du registre n
```

Shell

```
!cmd exécuter cmd dans un shell
:!cmd remplace la totalité du buffer par le résultat de la commande cmd exécutée sur le buffer
:r!cmd insère le résultat de cmd
!'xcmd exécute la commande shell cmd sur le texte jusqu'au signet x
:shell démarre un shell sous vi
```

Fenêtres (splits)

les commandes :q, :w, ... s'appliquent dans chaque fenêtre

```
:sp fic coupe (split) l'écran en deux et charge fic dans la 2° moitié
:vsp fic coupe l'écran en deux verticalement et charge fic dans la 2° moitié
```

les commandes ci-dessous doivent être précédées de <ctrl-w>

```
<ctrl-w> passe au split suivant
k passe au split supérieur
j passe au split inférieur
+ augmente la taille du split d'1 ligne (n<ctrl-w>+ augmente la taille du split de n lignes)
- diminue la taille du split d'1 ligne
_ maximise le split courant
s coupe l'écran en deux avec le contenu de la fenêtre courante
ajoute un split vide
```

Divers

```
u annuler
<ctrl-r> refaire
<ctrl-l> redessiner l'écran
<ctrl-g> affiche des informations sur le fichier
<ctrl-n> propose une complétion sur un mot
```

(en mode insertion)

```
<ctrn-x><ctrl-l> propose une complétion pour la ligne entière (en mode insertion)
<ctrl-r>=op insère le résultat de l'opération (mode insertion)
J fusionner deux lignes
K voir la page de man du mot sous le curseur
ga affiche la valeur ascii du caractère
gf ouvre le fichier sous le curseur
<ctrl-a> incrémente le nombre sous le curseur
<ctrl-x> décrémente le nombre sous le curseur
:help aide
c comme d, mais passe en mode insertion
. répète la dernière commande
:Ex ouvre l'explorateur de fichiers
:Sex ouvre l'explorateur de fichiers dans un split
```

Invoquer vi

Apprendre vi
vimtutor fr

Ouvrir un fichier distant

```
vi scp://user@host//path/to/file
vi ftp://user@host/path/to/file
```

Multi-file edit (passer d'un fichier au suivant par :n)
vi fichier1 fichier2 ...

Utiliser la version graphique de vim
gvim

Variables .vimrc

Utiliser l'indentation automatique
set autoindent

Taille de l'indentation
set tabstop=4
set shiftwidth=4

Affichage de la parenthèse/ du crochet/de l'accolade correspondant
set showmatch

Utilise la cloche visuelle plutôt que le beep
set noerrorbells
set vb

Affiche/n'affiche pas les correspondances de recherches en surbrillance
set hls
set nohls

Recherche incrémentale en cours de frappe
set incsearch

Permet à vim de fonctionner en mode non-compatible vi
set nocompatible

Utilise la coloration syntaxique
syntax on

Définit les raccourcis <ctrl-j> et <ctrl-k> pour passer d'une fenêtre à l'autre en la maximisant
map <C-J> <C-W>j<C-W>_
map <C-K> <C-W>k<C-W>_

Définit le raccourci <F10> pour enregistrer le fichier et quitter
map <F10> :wq<CR>

Emacs

Sortir, sauver

les commandes de cette section sont précédées de <ctrl-x>

```
<ctrl-f> ouvrir un fichier
<ctrl-s> sauver le fichier
<ctrl-w>fi enregistrer sous fi
<ctrl-c> quitter
```

Déplacements

```
<ctrl-a> début de ligne
<ctrl-e> fin (end) de ligne
<ctrl-→> mot suivant
<ctrl-←> mot précédent
<esc> < début du fichier
<esc> > fin du fichier
```

Copier, couper, coller

```
<ctrl-espace> pose une marque de début
<ctrl-k> couper (kill) la fin de la ligne
<ctrl-y> coller (yank)
<ctrl-w> coupe de la marque au curseur
<ctrl-d> efface (delete) le caractère à droite du curseur
<esc>d efface le mot à droite du curseur
```

Numération

```
<ctrl-u>n cmd exécute n fois cmd
```

Rechercher

```
<ctrl-s>motif rechercher (search) motif
<ctrl-s> recherche l'occurrence suivante
<ctrl-r> motif recherche arrière
```

Remplacer

```
<esc>% remplacer
en réponse à la question : y remplace, n ne remplace pas, ! remplace toutes les occurrences
```

Shell

```
<esc>xshell démarre un shell
<esc>!cmd exécute cmd
<esc>|cmd exécute cmd sur le contenu de la région marquée
```

Buffers

```
<ctrl-x>b passe au buffer suivant
<ctrl-x>bnom passer au buffer nom
<ctrl-x>k supprimer le buffer courant
```

Fenêtres

```
<ctrl-x>2 diviser l'écran en 2 fenêtres
<ctrl-x>0 aller dans l'autre (other) fenêtre
<ctrl-x>1 1 seule fenêtre dans l'écran
```

Divers

```
<ctrl-x>u annuler la dernière commande
<ctrl-g> annuler une commande en cours
<ctrl-t> transposer deux caractères
<esc>t transpose le mot précédent et le suivant
<ctrl-l> centre l'écran sur la ligne courante
```