Bash - Edition

^k	Coupe du curseur jusqu'à la fin de ligne
^u	Coupe du curseur jusqu'en début de ligne
^w	Coupe le mot avant le curseur
^y	Coller une chaîne précédemment coupée
%Backspace	Supprime un mot jusqu'à un symbole de type tiret...
%d	Supprime le mot suivant
^h	Remplace Backspace
%c	Met la première lettre en maj et avance d'un mot
%u	Met le mot en majuscule
%l	Met le mot en minuscule
%.	Réécrit le paramètre de la dernière commande
%t	Inverse la position des deux mots avant le curseur
^t	Inverse la position des deux caractères avant le curseur

^ : Ctrl
% : Alt

Bash - Historique

!!
Relancer la dernière commande

!p
Relancer la dernière commande commençant par p

!l:p
Afficher la dernière commande commençant par l

!$
Récupérer le dernier argument de la commande précédente

!^
Récupérer le premier argument de la commande précédente

Bash - Historique (cont)

!*
Tous les arguments de la dernière commande

!*:p
idem mais l'affiche

!n
Execute la nième commande

history -c
Vider l'historique

!-2
Execute la nième commande en partant de la fin

!?pattern
Execute la dernière commande contenant pattern

pattern1 pattern2
Remplace pattern1 par pattern2 dans la dernière commande

Bash - Divers

%r	Vide la ligne
^r	Recherche une commande déjà tapée
^c	Arrête la commande en court
^d	Quitte le shell en court
^l	Efface le contenu de l'écran
^o	Valide la ligne en cours
tab	Complétion
%*	Affiche les complétions disponibles

Bash - Variables

${var}
Valeur de var

${var:-word}
Affiche word si var est nulle ou unset

${var:=word}
Affiche word si var est nulle ou unset et set assigne word à var

${var:?}
Affiche une erreur si VAR est nulle ou unset

${var:+word}
Affiche word si var est différente de nulle

Bash - Variables (cont)

${var:offset}
Affiche var à partir de l'offset

${var:offset:lenght}
Affiche var à partir de l'offset sur lenght de longeur

${!tab[@]}
Liste les id du tableau tab

${tab[#_ID]}
Affiche la valeur du #_ID de tab

${#var}
Affiche la taille de var

${var#pattern}
var amputé du pattern mini en prefix

${var##pattern}
var amputé du pattern max en prefix

${var%pattern}
var amputé du pattern mini en suffix

${var%%pattern}
var amputé du pattern max en suffix

${var/pattern/string}
Substitution

${var^}
Maj du premier caractère

${var^^}
Maj de tous les caractères

${var,}
Min du premier caractère

${var,,}
Min de tous les caractères

Bash - Deplacement

^a	Aller en début de ligne
^e	Aller en fin de ligne
%b	Aller au mot précédent
%f	Aller au mout suivant
^xx	Alterne le curseur avec sa position précédente
^p	Historique précédent

Bash - Deplacement (cont)

^n	Historique suivant

^ : Ctrl
% : Alt ou Esc

Bash - IO Redirections

cmd > file	Redirige stdout de cmd dans file
cmd 2> file	Redirige stderr de cmd dans file
cmd &> file	Redirige stdout et sterr de cmd dans file
cmd < file	Envoi le contenu de file dans cmd
cmd 2> /dev/null	Redirige sterr dans un trou noir
cmd > file.out 2> file.err	Redirige stdout dans file.out et stderr dans file.err
cmd1 | cmd2	Redirection stdout de cmd1 dans stdin de cmd2. stderr n'est pas transmit dans les |
cmd1 | cmd2 | cmd3 | cmd4; echo ${PIPESTATUS[@]}	Suite de redirections et récupération des
> file	Vide et/ou crée un fichier
cmd | tee cmd.out | sort | tee sort.out | uniq -c | tee uniq.out	Un fichier de sortie par cmd
(cmd1; cmd2) > file	stdout des 2 cmd dans file (via sous shell)
{ cmd1; cmd2; } > file	stdout des 2 cmd dans file (sans sous shell)
cmd1; cmd2	Execution cmd1 puis cmd2
cmd1 && cmd2	Execution de cmd2 si cmd1 est OK
cmd1 || cmd2	Execution de cmd2 si cmd1 est non OK

>> ajoute au lieu de rediriger.