

### Bash - Edition

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

### Bash - Historique

<b>!!</b>	Relancer la dernière commande
<b>!p</b>	Relancer la dernière commande commençant par p
<b>!!:p</b>	Afficher la dernière commande commençant par l
<b>!\$</b>	Récupérer le dernier argument de la commande précédente
<b>!^</b>	Récupérer le premier argument de la commande précédente

### Bash - Historique (cont)

<b>!* </b>	Tous les arguments de la dernière commande
<b>!*:p</b>	idem mais l'affiche
<b>!n</b>	Exécute la nième commande
<b>history -c</b>	Vider l'historique
<b>!-2</b>	Exécute la nième commande en partant de la fin
<b>!?pattern</b>	Exécute la dernière commande contenant pattern
<b>pattern1 pattern2</b>	Remplace pattern1 par pattern2 dans la dernière commande

### Bash - Divers

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

### Bash - Variables

<b>\${var}</b>	Valeur de var
<b>\${var:-word}</b>	Affiche word si var est nulle ou unset
<b>\${var:=word}</b>	Affiche word si var est nulle ou unset et set assigne word à var
<b>\${var:?}</b>	Affiche une erreur si VAR est nulle ou unset
<b>\${var:+word}</b>	Affiche word si var est différente de nulle

### Bash - Variables (cont)

<b>\${var:offset}</b>	Affiche var à partir de l'offset
<b>\${var:offset:length}</b>	Affiche var à partir de l'offset sur length de longueur
<b>\${!tab[@]}</b>	Liste les id du tableau tab
<b>\${tab[#_ID]}</b>	Affiche la valeur du #_ID de tab
<b>\${#var}</b>	Affiche la taille de var
<b>\${var#pattern}</b>	var amputé du pattern mini en prefix
<b>\${var##pattern}</b>	var amputé du pattern max en prefix
<b>\${var%pattern}</b>	var amputé du pattern mini en suffix
<b>\${var%%pattern}</b>	var amputé du pattern max en suffix
<b>\${var/pattern/string}</b>	Substitution
<b>\${var^}</b>	Maj du premier caractère
<b>\${var^^}</b>	Maj de tous les caractères
<b>\${var,}</b>	Min du premier caractère
<b>\${var,,}</b>	Min de tous les caractères

### Bash - Deplacement

<b>^a</b>	Aller en début de ligne
<b>^e</b>	Aller en fin de ligne
<b>%b</b>	Aller au mot précédent
<b>%f</b>	Aller au mot suivant
<b>^xx</b>	Alterne le curseur avec sa position précédente
<b>^p</b>	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 stderr de cmd dans file

`cmd < file`

| Envoi le contenu de file dans cmd

`cmd 2> /dev/null`

| Redirige stderr 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.

