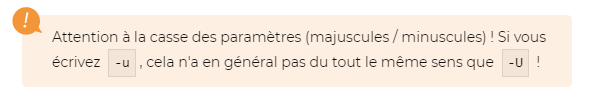
**$** : ce dernier symbole est très important ; il indique votre niveau d'autorisation sur la machine. Il peut prendre deux formes différentes :

**$** : signifie que vous êtes en train d'utiliser un compte utilisateur « normal », avec des droits limités (il ne peut pas modifier les fichiers système les plus importants). Mon compte mateo21 est donc un compte normal avec des droits limités ;

**#** : signifie que vous êtes en mode superutilisateur, c'est-à-dire que vous êtes connectés sous le pseudonyme « root ». Le root est l'utilisateur maître qui a le droit de tout faire sur sa machine (même de la détruire !). Nous verrons le mode root plus en détail plus tard ; pour l'instant nous restons dans un compte utilisateur limité, ainsi nous ne risquons pas de faire de bêtise.

date : date du jour

ls : liste des fichier/répertoire dans le répertoire courant



Les paramètres peuvent être combinés :

commande -d

commande -d -a -U -h

commande -daUh

Les paramètres longs donc commencent par -- :

commande --parametre

commande --parametre1 --parametre2

commande -daUh --autreparametre

Un paramètre long a son équivalent court : ls -a ou ls –all

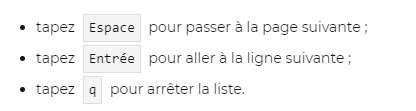
Associé une valeur à un paramètre :

commande -p 14

commande --parametre=14

Autocomplétion : taper les premières lettres puis TABULATION Tab.

Historique des commandes : flèche haut ou bas. Ou commande history.



CTR + R + première lettre, recherche dans l’historique la commande commençant par la première lettre.

**Ctrl + L** : efface le contenu de la console.

**Ctrl + D** : envoie le message EOF (fin de fichier) à la console.

**Shift + PgUp** : vous permet de « remonter » dans les messages envoyés par la console

**Shift + PgDown** : pareil, mais pour redescendre.

**Ctrl + A** : ramène le curseur au début de la commande.

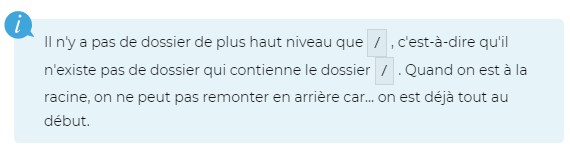
**Ctrl + E** : ramène le curseur à la fin de la ligne de commandes.

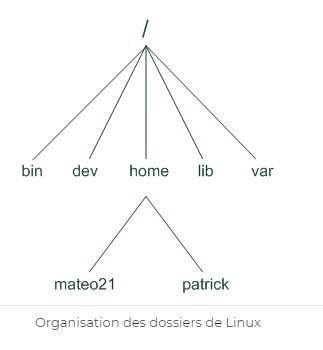
**Ctrl + U** : supprime tout ce qui se trouve à gauche du curseur.

**Ctrl + K** : supprime tout ce qui se trouve à droite du curseur.

**Ctrl + W** : supprime le premier mot situé à gauche du curseur.

**Ctrl + Y** : si vous avez supprimé du texte avec une des commandes Ctrl + U, Ctrl + K ou Ctrl + W qu'on vient de voir, alors le raccourci Ctrl + Y « collera » le texte que vous venez de supprimer.





pwd : afficher le dossier actuel

which : connaître l'emplacement d'une commande ex which pwd

ls : lister les fichiers et dossiers

Si la couleur ne s'affiche pas, vous pouvez rajouter le paramètre--color=auto, comme ceci :ls --color=auto.  
Si vous ne voulez pas de la couleur au contraire, essayez le paramètre--color=none.

En bleu : les dossiers

ls -a : afficher tous les fichiers et dossiers cachés

ls -F : indique le type d'élément

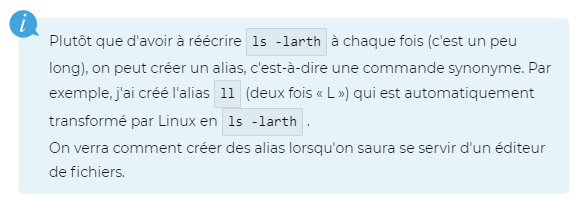
ls -l : liste détaillée

ls -h : afficher la taille en Ko, Mo, Go…

ls -t : trier par date de dernière modification

ls -r qui renverse l'ordre d'affichage des fichiers.

Combinaison des paramètres : ls -larth



cd: changer de dossier

cd / : aller à la racine

cd ../.. / : remonter de 2 répertoires parents

**Les chemins relatifs**

Un chemin relatif est un chemin qui dépend du dossier dans lequel vous vous trouvez

Les chemins absolus

Contrairement aux chemins relatifs, les chemins absolus fonctionnent quel que soit le dossier dans lequel on se trouve.

Un chemin absolu est facile à reconnaître : il commence toujours par la racine (/). Vous devez ensuite faire la liste des dossiers dans lesquels vous voulez entrer.

Revenir au répertoire home :

cd ~

ou cd

utiliser la TABULATION.

du: taille occupée par les dossiers

du -h : la taille pour les humains

du -a : afficher la taille des dossiers ET des fichiers

du -s : avoir juste le grand total

cat & less (more) : afficher un fichier

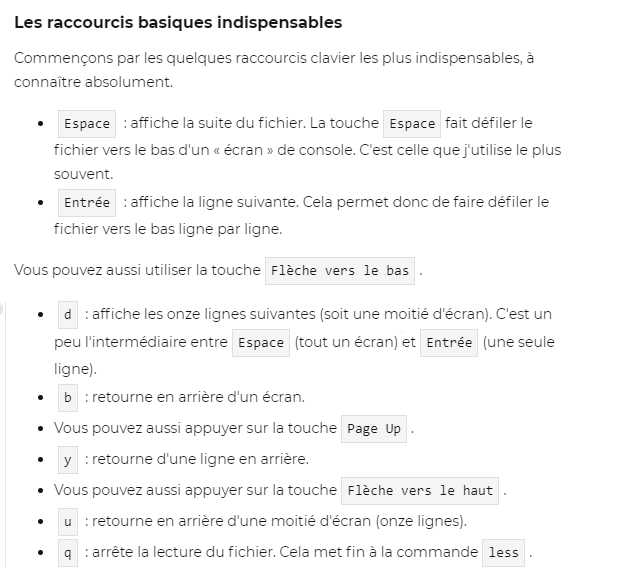
cat -n : permet d'afficher les numéros de ligne

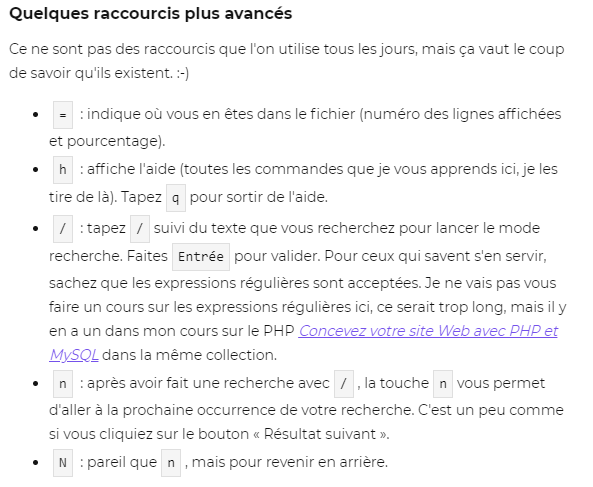
cat syslog

cat -n syslog

less : afficher le fichier page par page

less syslog





head & tail : afficher le début et la fin d'un fichier

head : afficher le début du fichier

head -n nbre : -n, suivi d'un nombre. Il permet d'afficher le nombre de lignes

head -n 3 syslog

head syslog

tail : afficher la fin du fichier

tail -nsuivi d'un nombre pour afficher les $x$ dernières lignes

tail -f : de « suivre » la fin du fichier au fur et à mesure de son évolution.

Faites Ctrl + C(Ctrl et C en même temps) pour arrêter la commande tail.

tail -f recherche les nouveaux changements dans le fichier toutes les secondes.

Si vous voulez, vous pouvez rajouter le paramètre-s suivi d'un nombre.

Par exemple, tail -f -s 3 syslog recherchera les changements toutes les trois secondes (plutôt que toutes les secondes).

tail syslog

tail -f syslog

tail -n 3 syslog

touch & mkdir : créer des fichiers et dossiers

touch : créer un fichier

espace dans le nom : entre ‘’ ou \

touch fichierbidon

touch fichierbidon autrefichierbidon.txt

touch "Fichier bidon"

mkdir : créer un dossier

mkdir mondossier

mkdir mondossier autredossier

mkdir :-p. Il sert à créer tous les dossiers intermédiaires.

mkdir -p animaux/vertebres/chat

cp & mv : copier et déplacer un fichier

cp : copier un fichier

cp fichierbidon fichiercopie

cp fichierbidon mondossier/ copierfichierbidondans le sous-dossiermondossier

cp fichierbidon mondossier/fichiercopiecopierfichierbidondansmondossiersous un autre nom

p fichierbidon /var/log/ un répertoire en absolu

Copier des dossiers

cp -R animaux autresanimaux

Utiliser le joker\*

cp \*.jpg mondossier/

cp so\* mondossier/

mv : déplacer un fichier

mv fichierbidon mondossier/  déplacer des dossiers

mv fichierbidon mondossier/  déplacera le dossieranimaux(et tous ses sous-dossiers) dansmondossier

mv \*.jpg mondossier/utiliser les jokers

Renommer un fichier

mv fichierbidon superfichier renommerafichierbidonensuperfichier. Après cette commande,fichierbidonn'existe plus, il a été renommé.

Déplacer et renommer un fichier à la fois

mv fichierbidon mondossier/superfichier déplacerfichierbidondansmondossiertout en lui affectant un nouveau nom

rm : supprimer des fichiers et dossiers

rm : supprimer un fichier

rm fichierbidonsupprimons cefichierbidon

rm fichierbidon fichiercopiesupprimer plusieurs fichiers en séparant leurs noms par des espaces

-i : demander confirmation

rm -i fichierbidon

rm: détruire fichier régulier vide `fichierbidon'?

**o** : signifie « Oui ». Sur certains systèmes anglais, il faudra peut-être utiliseryde Yes ;

**n** : signifie « Non ».

-f : forcer la suppression, quoi qu'il arrive

-f, c'est un peu le contraire de-i : c'est le mode des gros bourrins.

-v : dis-moi ce que tu fais, petit cachotier

rm -v fichierbidon fichiercopie

détruit `fichierbidon'

détruit `fichiercopie'

-r : supprimer un dossier et son contenu

rm -r animaux/

NON NON NON NE FAITES JAMAIS CA !!! => rm -rf /\*

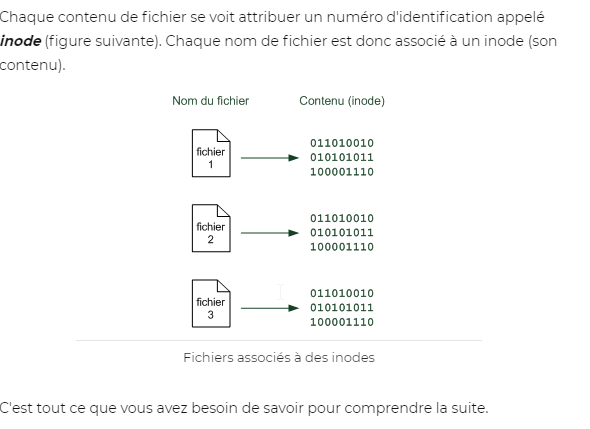
rm -rf \*supprime tous les fichiers et sous-dossiers du dossier dans lequel je me trouve.

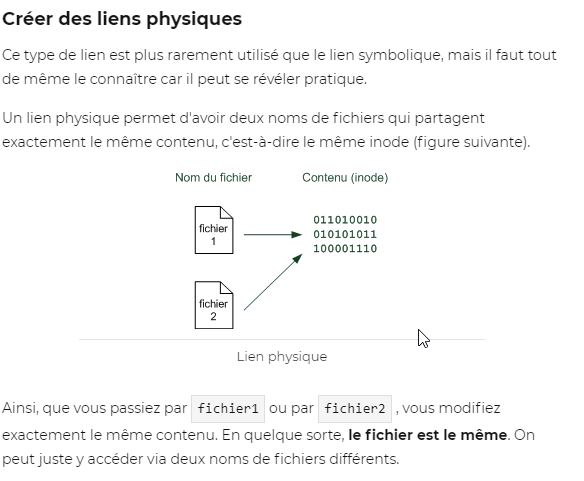
ln : créer des liens entre fichiers

Elle permet de créer des liens entre des fichiers, c'est-à-dire (pour employer des mots que vous connaissez) qu'elle permet de **créer des raccourcis**.

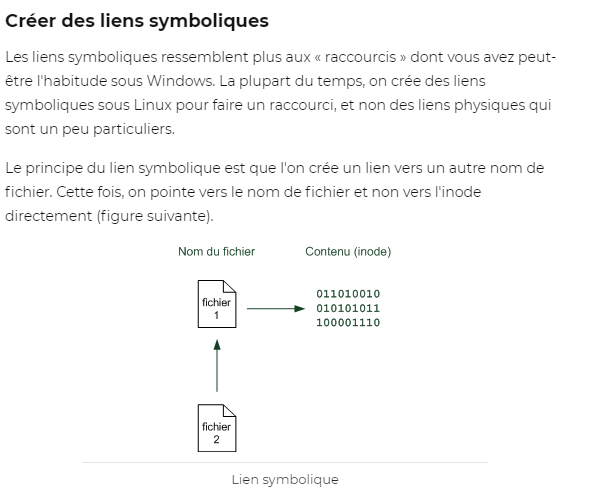
des liens **physiques** ;

des liens **symboliques**.





ln fichier1 fichier2créer un lien physique : nous allons créer unfichier2qui partagera le même inode (le même contenu) quefichier1.



ln -s fichier1 fichier2

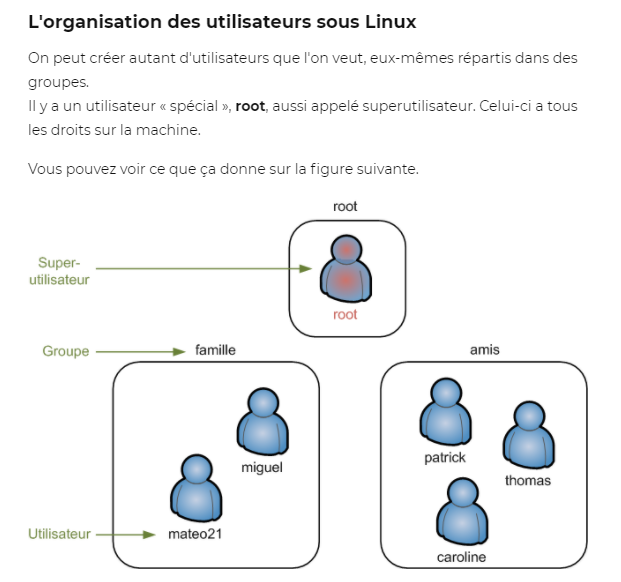
ls -l

total 0

-rw-r--r-- 1 mateo21 mateo21 0 2008-07-31 13:55 fichier1

lrwxrwxrwx 1 mateo21 mateo21 8 2008-07-31 14:15 fichier2 -> fichier1

Les utilisateurs et les droits



sudo: exécuter une commande en root

sudo : devenir root un instant

sudo commande

sudo su : devenir root et le rester

Le symbole # à la fin de l'invite de commandes vous indique que vous êtes devenus superutilisateur.  
Vous pouvez alors exécuter autant de commandes en root que vous le voulez.

Pour quitter le « mode root », tapez exit (ou faites la combinaison Ctrl + D).

root@mateo21-desktop:/home/mateo21# exit

exit

mateo21@mateo21-desktop:~$

adduser : ajouter un utilisateur

passwd : changer le mot de passe

deluser : supprimer un compte

addgroup : gestion des groupes

addgroup : créer un groupe

usermod : modifier un utilisateur

usermod -G amis,paris,collegues patrick.

delgroup : supprimer un groupe

chown :  : gestion des propriétaires d'un fichier

chown : changer le propriétaire d'un fichier

chown patrick rapport.txt

chgrp : changer le groupe propriétaire d'un fichier

chgrp amis rapport.txt

chown peut aussi changer le groupe propriétaire d'un fichier !

chown patrick:amis rapport.txt

-R : affecter récursivement les sous-dossiers

chown -R mateo21:mateo21 /home/patrick/

chmod : modifier les droits d'accès

Vous voyez tous ces d, r, w et x au début ? Ce sont ce qu'on appelle les droits d'accès du fichier ou dossier.

On peut voir cinq lettres différentes. Voici leur signification :

**d** (Directory) : indique si l'élément est un dossier ;

**l** (Link) : indique si l'élément est un lien (raccourci) ;

**r** (Read) : indique si on peut lire l'élément ;

**w** (Write) : indique si on peut modifier l'élément ;

**x** (eXecute) : si c'est un fichier, « x » indique qu'on peut l'exécuter. Ce n'est utile que pour les fichiers exécutables (programmes et scripts).

Si c'est un dossier, « x » indique qu'on peut le « traverser », c'est-à-dire qu'on peut voir les sous-dossiers qu'il contient si on a le droit de lecture dessus.

Si la lettre apparaît, c'est que le droit existe. S'il y a un tiret à la place, c'est qu'il n'y a aucun droit.

Pourquoi est-ce qu'on voit parfois r, w et x à plusieurs reprises ?

Les droits sont découpés en fonction des utilisateurs (figure suivante).



Le premier élément d mis à part, on constate que r, w et x sont répétés trois fois en fonction des utilisateurs :

le premier triplet rwx indique les droits que possède le **propriétaire** du fichier sur ce dernier ;

le second triplet rwx indique les droits que possèdent les autres membres du **groupe** sur ce fichier ;

enfin, le dernier triplet rwx indique les droits que possèdent tous les **autres** utilisateurs de la machine sur le fichier.

Ses droits sont : -rw-r--r--

**-** : le premier tiret indique qu'il ne s'agit pas d'un dossier. S'il y avait eu un d à la place, cela aurait indiqué qu'il s'agissait d'un dossier.

**rw-** : indique que le propriétaire du fichier, mateo21 en l'occurrence, peut lire et modifier (et donc supprimer) le fichier. En revanche, il ne peut pas l'exécuter car il n'a pas de x à la fin. Je rappelle que quiconque peut modifier un fichier a aussi le droit de le supprimer.

**r--** : tous les utilisateurs qui font partie du groupe mateo21 mais qui ne sont pas mateo21 peuvent seulement lire le fichier. Ils ne peuvent ni le modifier, ni l'exécuter. Je reconnais qu'avoir un nom de groupe identique au nom d'utilisateur peut embrouiller : si vous êtes aussi bien organisés que sur mon premier schéma, on parlera plutôt du groupe famille.

**r--** : tous les autres (ceux qui ne font pas partie du groupe mateo21) peuvent seulement lire le fichier.

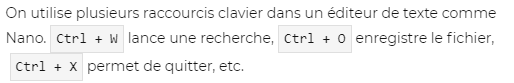
chmod : modifier les droits d'accès

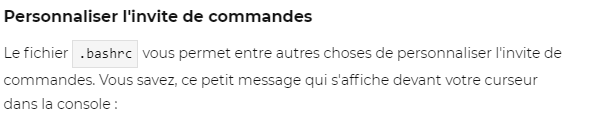
Alias : raccourci d’1 commande et de ces paramètres

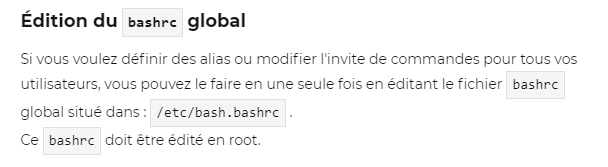
alias ls='ls --color=auto'

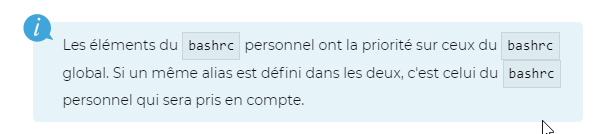
alias ll='ls -lArth'

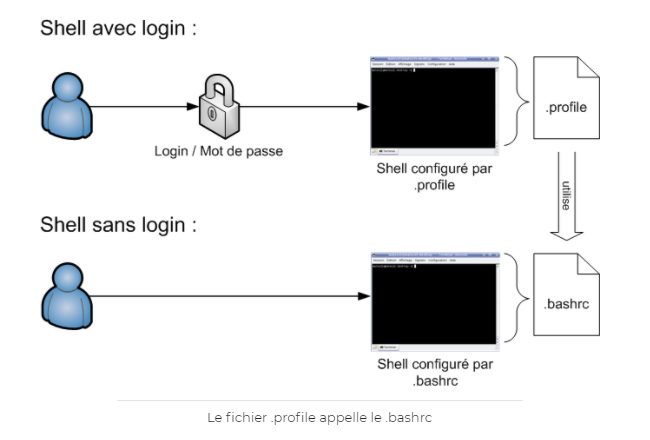
Fichier Vim éditeur :



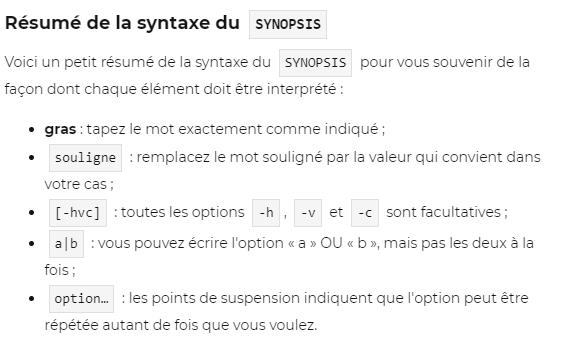


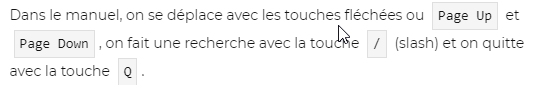






man : afficher le manuel d'une commande





apropos  mot\_clé : trouver une commande

commande – h avoir de l’aide sur la commande

La commande whatis commande : man allégé

locate : une recherche rapide pour trouver le chemin des fichiers

find ou type quoi : une recherche approfondie du répertoire des fichiers

find -name "logo.png"

find /var/log/ -name "syslog"

find / -name "syslog" recherche sur tout le disque

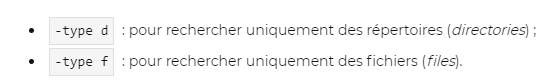
Recherche à partir de la taille

find  ~ -size +10M

Recherche à partir de la date de dernier accès

find -name "\*.odt" -atime 6

Rechercher uniquement des répertoires ou des fichiers



find /var/log -name "syslog" -type d

Utilisation avancée avec manipulation des résultats

find   -name "\*.jpg" -print

Afficher les fichiers de façon formatée

find . -name "\*.jpg" -printf "%p - %u\n"

Supprimer les fichiers trouvés

find   -name "\*.jpg" -delete

Appeler une commande

find   -name "\*.jpg" -exec chmod 600 {} \;

Extraire, trier et filtrer des données

grep : filtrer des données

Son rôle est de rechercher un mot dans un fichier et d'afficher les lignes dans lesquelles ce mot a été trouvé.

grep texte nomfichier

grep alias .bashrc

-i : ne pas tenir compte de la casse (majuscules / minuscules)

grep -i alias .bashrc

-n : connaître les numéros des lignes

grep -n alias .bashrc

-v : inverser la recherche : ignorer un mot

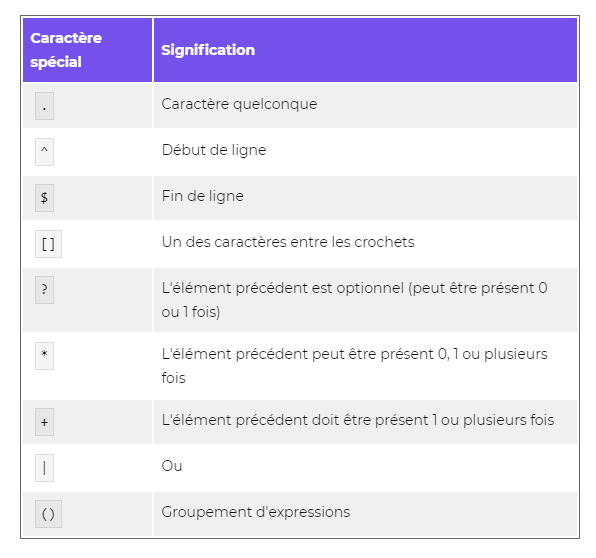
vous voulez connaître toutes les lignes qui **ne contiennent pas** un mot donné,

grep -v alias .bashrc

-r : rechercher dans tous les fichiers et sous-dossiers

grep -r "Site du Zéro" code/

Utilisergrepavec des expressions régulières



grep -E Alias .bashrc

grep -E ^Alias .bashrc

grep -E ^alias .bashrc

grep -E [Aa]lias .bashrc

grep -E [0-4] .bashrc

grep -E [a-zA-Z] .bashrc

sort : trier les lignes d’un fichier

sort noms.txt

-o : écrire le résultat dans un fichier

sort -o noms\_tries.txt noms.txt

-r : trier en ordre inverse

sort -r  noms.txt

-R : trier aléatoirement

sort -R noms.txt

-n : trier des nombres

sort nombres.txt

sort -n nombres.txt

wc : compter le nombre de lignes

wc noms.txt

-l : compter le nombre de lignes

wc -l noms.txt

-w : compter le nombre de mots

wc -w noms.txt

-c : compter le nombre d'octets

wc -c noms.txt

-m : compter le nombre de caractères

wc -m noms.txt

uniq : supprimer les doublons

 uniq doublons.txt

uniq doublons.txt sans\_doublons.txt Vous pouvez demander à ce que le résultat sans doublons soit écrit dans un autre fichier plutôt qu'affiché dans la console

-c : compter le nombre d'occurrences

uniq -c doublons.txt

-d : afficher uniquement les lignes présentes en double

uniq -d doublons.txt

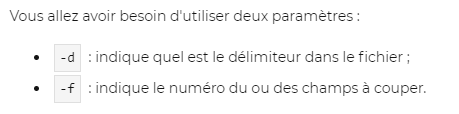
cut : couper une partie du fichier

 cut -c 2-5 noms.txt conserver uniquement les caractères 2 à 5 de chaque ligne du fichier

cut -c -3 noms.txt  conserver du 1er au 3ème caractère

cut -c 3- noms.txt conserver du 3ème au dernier caractère

Couper selon un délimiteur



cut -d , -f 1 notes.csv

cut -d , -f 3 notes.csv cut -d , -f 1,3 notes.csv

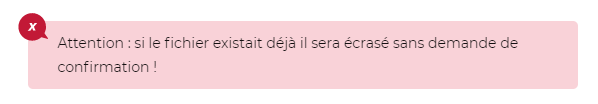
Les flux de redirection

> et >> : rediriger le résultat dans un fichier

cut -d , -f 1 notes.csv

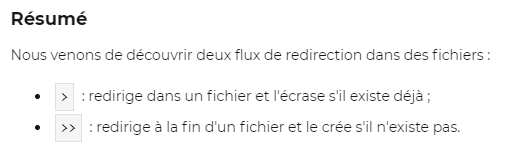
> : rediriger dans un nouveau fichier

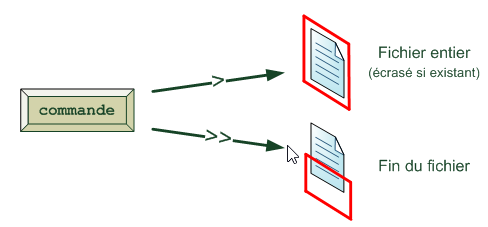
cut -d , -f 1 notes.csv > eleves.txt



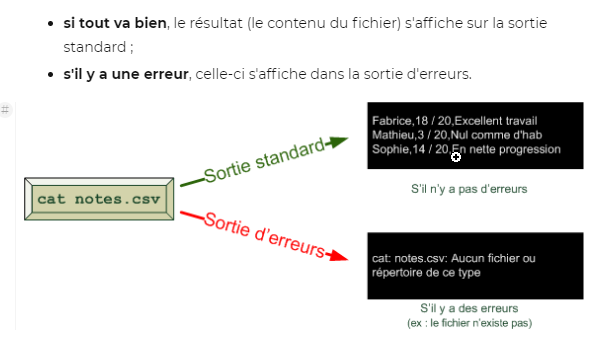
>> : rediriger à la fin d'un fichier

macommande >> resultats.log





2>, 2>> et 2>&1 : rediriger les erreurs



cut -d , -f 1 fichier\_inexistant.csv > eleves.txt 2> erreurs.log

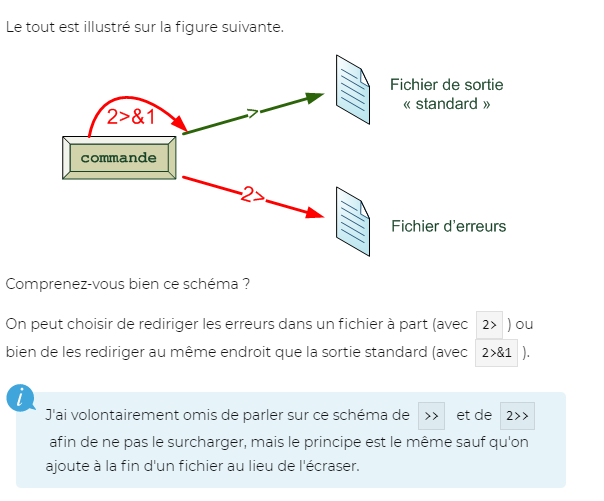
Notez qu'il est aussi possible d'utiliser 2>>

pour ajouter les erreurs à la fin du fichier.

Fusionner les sorties

cut -d , -f 1 fichier\_inexistant.csv > eleves.txt 2>&1

les erreurs seront **ajoutées** à la fin du fichier eleves.txt comme le reste des messages.



< et << : lire depuis un fichier ou le clavier

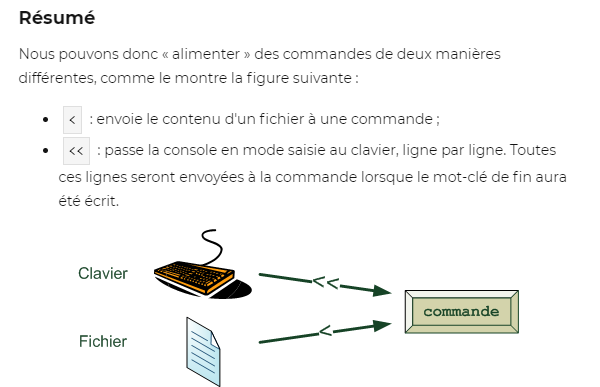
< : lire depuis un fichier

Le chevron ouvrant < (à ne pas confondre avec le chevron fermant que nous avons utilisé tout à l'heure) permet d'indiquer d'où vient l'entrée qu'on envoie à la commande.

cat < notes.csv

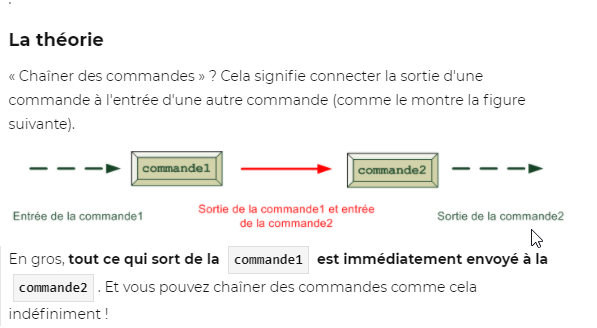
<< : lire depuis le clavier progressivement

 wc -m << FIN



sort -n << FIN > nombres\_tries.txt 2>&1Les nombres saisis au clavier seront envoyés à nombres\_tries.txt, de même que les erreurs éventuelles.

| : chaîner les commandes



##### Trier les élèves par nom

cut -d , -f 1 notes.csv | sort

cut -d , -f 1 notes.csv | sort > noms\_tries.txtOn peut même aller plus loin et écrire cette liste triée dans un fichier

##### Trier les répertoires par taille

du | sort -nr

du | sort -nr | head

du | sort -nr | less

##### Lister les fichiers contenant un mot

sudo grep log -Ir /var/log  | cut -d : -f 1  | sort | uniq