Synchronisation de répertoire

Principe

On liste les fichiers présents dans les deux répertoires et on copie vers la destination uniquement ceux qui sont plus récents

Un essai en local

```
mkdir -p ~/rsync/SRC ~/rsync/DEST
cd ~/rsync
touch ~/rsync/SRC/titi.txt
rsync -avvP ~/rsync/SRC/ ~/rsync/DEST
touch ~/rsync/SRC/toto.txt
```

Utilisation en réseau

```
touch ~/rsync/SRC/tata.txt
rsync -avvP ~/rsync/SRC/ ssh.ens-lyon.fr:~/rsync/DEST
```

Très utile pour les sauvegardes ...

Accéder en ssh à l'ENS Lyon

le serveur : ssh.ens-lyon.fr

Clients en mode texte

- putty (Windows)
- terminal Mac
- terminal Linux

En mode graphique

- Winscp (Windows)
- Fugu (Mac)
- gFTP

Travailler en ligne de commande : Bash

Visualiser le contenu d'un fichier

Parcourir le fichier

```
cd Matos
more hexa_aniso_stability.txt
cat hexa_aniso_stability.txt
```

Voir les extrémités du fichier

```
head hexa_aniso_stability.txt
tail hexa_aniso_stability.txt
tail -f ~/.bash_history
```

Ouvrez un autre terminal Ctrl + Shift ⇑ + n , tapez des commandes et quittez le nouveau terminal avec exit ou Ctrl + d

Sélectionner des élements dans un fichier

Par lignes : grep

```
grep 38 hexa_aniso_stability.txt
grep \.38 hexa_aniso_stability.txt
ls -l ~ | grep rwx
ls -al ~ | grep '\.g'
```

Par sélection précise : awk

```
awk 'NR == 5 , NR == 10 {print  $0 }' hexa_aniso_stability.txt
awk 'NR<=200 {print NR" : "$1,$2}'  hexa_aniso_stability.txt
awk 'NR<=200 {print NR" : "$1,$2}'  hexa_aniso_stability.txt > test
```

Faire des substitutions dans un fichier

On va utiliser sed.

Remplacer des . par des ,

```
sed s/+/-/ hexa_aniso_stability.txt
sed s/+/-/g hexa_aniso_stability.txt
sed s/"\."/,/g hexa_aniso_stability.txt
sed s/"\."/,/g hexa_aniso_stability.txt > new_file
cat newfile
```

Enchainer le tout

```
grep 38 hexa_aniso_stability.txt | awk 'NR<=50 {print $1,$4}'
| sed s/"\."/,/g > toto.csv && localc toto.csv
```

Travailler en ligne de commande : Bash

Créer un script Bash

Ouvrez votre éditeur favori (nano, gedit, vi) et créer un nouveau fichier nommé premier_script.sh

premier_script.sh

```
#!/bin/bash
echo 'ceci est mon premier script'
ls -l ~/var/log
```

Rendre le script executable et le lancer

```
chmod u+x premier_script.sh
./premier_script.sh
```

Les variables

Variables simples

```
var1=coucou
var2="a tout le monde"
echo $var1 $var2
var1=2
var2=3
var3=$(( $var1 + $var2 ))
echo $var3
```

Créer des variables à partir du résultat d'une commande

```
var=‘ls‘
echo $var
```

Variables d'environnement et tableaux

Variables d'environnement

```
env
echo $USERNAME
echo $SHELL
```

Tableaux

```
tab[0]=toto
tab[1]=titi
echo $tab[0]
echo ${tab[0]}
```

Arguments du shell

```
$0 nom du script
$1 $2 parametre donner au script
$#   nombre de parametres
```

Les structures conditionnelles

Structure de base

```
if <condition>; then
<commands>
fi
```

Quelques exemples

```
if [ "$stringvar" == "tux" ];
then echo 'la mascotte de linux';
fi
if [ ! -f nom_fichier ];
then echo 'no file';
else echo 'fichier present';
fi;
case
```

Conditions multiples

```
space=‘df -h | awk ’{print $5}’ | grep % | sort -n |
tail -1 | cut -d "%" -f1 -‘
case $space in
[1-6]*)
  Message="All is quiet."
  ;;
[7-8]*)
  Message="Start thinking about cleaning out some stuff.
  is $space % full."
  ;;
9[1-8])
  Message="Better hurry with that new disk...  One partition
  is $space % full."
  ;;
*)
  Message="I seem to be running with an
nonexistent amount of disk space..."
  ;;
esac
echo $Message
```

Les boucles for

Syntaxe générale

```
for variable in liste_valeurs
    do instruction(s)
done
```

Un exemple avec une variable

```
var='A B C'
for i in $var
do echo $i
done
```

Un exemple quand on connait le nombre d'itérations

```
for i in {0..10..2}
do
echo "Welcome $i times"
done
```

Les boucles while et until

Syntaxe générale

```
COMPTEUR=0
while [  $COMPTEUR -lt 10 ]; do
    echo The COMPTEUR is $COMPTEUR
    let COMPTEUR=$COMPTEUR+1
done
```

Un exemple avec

```
COMPTEUR=20
until [  $COMPTEUR -lt 10 ]; do
    echo COMPTEUR $COMPTEUR
    let COMPTEUR=$COMPTEUR-1
done
```

Convertir une série d'images

Le commande convert

permet de transformer une image (rotation, format, decoupage, ...)

Aller dans Matos/images

Le script

```
#!/bin/bash
for f in $(find -type f -iname '*.png')
do
    dest=`echo ${f%.*}`
    echo "${f} to ${dest}.jpg"
    convert "${f}" "${dest}.jpg"
done
```

Travailler en ligne de commande : Bash

Les alias

Ouvrez le fichier .bashrc

Principe

une chaine de caractère est associé à une commande prédéfinie

Exemples

```
alias info='cd ~/InfoScientifique'
alias transfert_ssh_ens='rsync -avvP ~/InfoScientifique/
ssh.ens-lyon.fr:
```

Le PATH

Principe

le PATH est une variable d'environnement qui indique où aller chercher les executables

Valeur par defaut

```
echo $PATH
```

Ajoutons un répertoire où l'on met nos scripts

```
mkdir -p ~/bin/scripts
Dans le fichier .bashrc :
export $PATH=$PATH:~/bin/scripts
```

Mettons un peu de couleur

Le prompt

```
PS1='${debian_chroot:+($debian_chroot)}\[\033[01;33m\]\u
@\h\[\033[00m\]:\[\033[01;35m\]\w\[\033[00m\]\$'
```

Quelques commandes

```
alias ls='ls --color=auto'
alias grep='grep --color=auto'
```

Produire des documents scientifiques : LaTeX

LaTeX

On va :

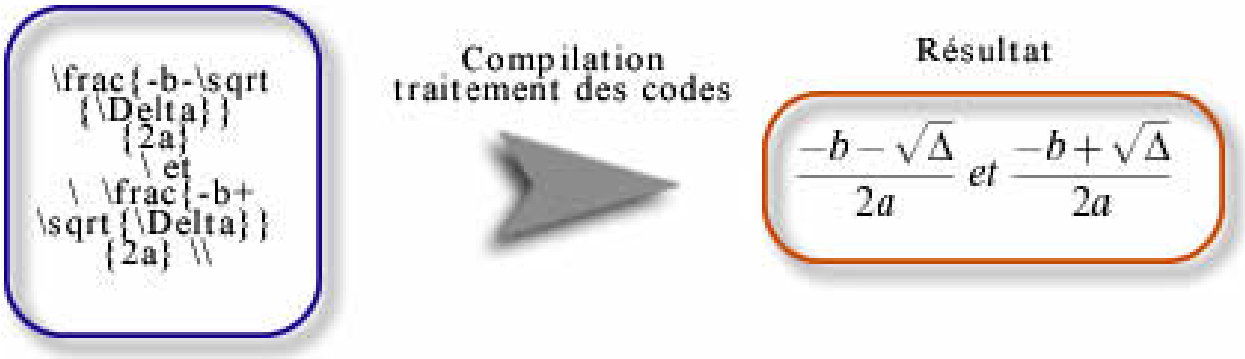
- introduire le principe du langage
- compiler un fichier LaTeXet visualiser le PDF
- organiser le document
- mettre en forme de belles équations
- gérer les images, les tables, etc ...
- découvrir la puissance du langage (table des matières, index, listes de tables et figures)
- gérer la bibliographie

On appliquera tout cela à un rapport, une présentation et un poster.

Produire des documents scientifiques : LaTeX

Principe général

Un fichier source transforé en PDF par la commande `pdflatex`



Séparation du contenu et de la mise en page

Produire des documents scientifiques : LaTeX

Compilons notre premier fichier

```
\documentclass[12pt]{article}
\usepackage[utf8]{inputenc}
\begin{document}
Bonjour tout le monde, ceci est mon premier document \LaTeX.

Comment ça marche ?
\end{document}
```

Ouvrir un terminal

Créer un répertoire `Latex`
Ouvrir un fichier `hello.tex` et y mettre le texte ci-dessus
Compiler le programme `pdflatex hello.tex`
Visualiser le document produit avec `evince`

Structure du fichier source

Analysons le code

```
\documentclass{article}
\usepackage[utf8]{inputenc}
\begin{document}
Bonjour tout le monde, ceci est
mon premier document \LaTeX.

Comment ça marche ?
\end{document}
```

- un entête ou *header* qui contient :
  - la classe (article, book, présentation, poster)
  - les packages utilisés (accentuation, formules mathématiques ...)
  - des informations (auteur, titre, ...)
- un corps de document qui contient tout ce qui sera affiché (textes, figures, tableaux)

Les classes et les options du documents

Quelques classes utiles

- article
- book
- beamer
- a0poster

Quelques options

- a4paper
- 10pt
- times, helvetica
- twocolumn

Quelques packages indispensables

Les accents

```
\usepackage[utf8]{inputenc}
\usepackage[latin1]{fontenc}
```

Les règles typographiques françaises

```
\usepackage[french]{babel}
```

Les mathématiques

```
\usepackage{amsmath}
```

Les figures

```
\usepackage{graphicx}
```

Produire des documents scientifiques : LaTeX

Taille de la police

\tiny Portez ce vieux whisky au juge blond qui fume

\scriptsize Portez ce vieux whisky au juge blond qui fume

\footnotesize Portez ce vieux whisky au juge blond qui fume

\small Portez ce vieux whisky au juge blond qui fume

\normalsize Portez ce vieux whisky au juge blond qui fume

\large Portez ce vieux whisky au juge blond qui fume

\Large Portez ce vieux whisky au juge blond qui fume

\LARGE Portez ce vieux whisky au juge blond qui fume

\huge Portez ce vieux whisky au juge blond qui fume

\Huge Portez ce vieux whisky au juge blond qui fume

Mettre en valeur le texte

Texte en gras

```
{\bf Portez ce vieux whisky au juge blond qui fume}
```

Portez ce vieux whisky au juge blond qui fume

Texte en italique

```
{\it Portez ce vieux whisky au juge blond qui fume}
```

Portez ce vieux whisky au juge blond qui fume

Texte souligné

```
\underline{Portez ce vieux whisky au juge blond qui fume}
```

Portez ce vieux whisky au juge blond qui fume

Texte en majuscules

```
{\sc Portez ce vieux Whisky au juge blond qui fume}
```

PORTEZ CE VIEUX WHISKY AU JUGE BLOND QUI FUME

Alignement du texte

Par défaut le texte est justifié et aligné sur la gauche.

Centrer le texte

```
\begin{center}
Texte centr\'e
\end{center}
```

Texte centré

Décaler le texte à droite

```
\begin{flushright}
Texte d\'ecal\'e \`a droite
\end{flushright}
```

Texte décalé à droite

les listes

Listes à puces

```
\begin{itemize}
\item un element
\item un autre
\item encore un autre
\end{itemize}
```

- un element
- un autre
- encore un autre

Les listes numérotées

```
\begin{enumerate}
\item un element
\item un autre
\item encore un autre
\end{enumerate}
```

1. un element
2. un autre
3. encore un autre

On peut bien évidemment les imbriquer ...

La structure du document

```
\section{Les \'el\'ements finis}
\subsection{Principe}
\subsubsection{Un peu d\'historique}
```

Mettez cela dans votre fichier et compilez ...

Les mathématiques

Introduire des maths au sein d'un texte

```
Consid\'erons $x\in
\mathbb{R}$ un
nombre al\'eatoire
```

Considérons $x \in \mathbb{R}$ un nombre aléatoire

En laissant un espace avant et apès

```
Consid\'erons $$x\in
\mathbb{R}$$ un
nombre al\'eatoire
```

Considérons

$$x \in \mathbb{R}$$

un nombre aléatoire

Quelques exemples

```
$\int_{x=0}^{\infty}\frac
{\partial^2 f}{\partial x^2}
-\nabla  \otimes f dx$
```

$$\int_{x=0}^{\infty} \frac{\partial^2 f}{\partial x^2} - \nabla \otimes f dx$$