

1. Configuration locale de GIT

Installation de GIT sous linux :

sudo **apt-get install git-core**

Configuration locale:

```
git config --global user.name "Nom Prénom"
git config --global user.email "poweruser@ici_ou_la.fr"
#...
```

pour voir ce qui est configuré :
git config --list

2. Principales commandes de GIT (en mode local)

Commandes GIT (locales)	Utilités
git init	Initialise un référentiel local git (sous répertoire caché « .git ») au sein d'un projet neuf/originel.
git clone <i>url_referentiel_git</i>	Récupère une copie locale (sous le contrôle de GIT et avec toutes les versions des fichiers) d'un référentiel git existant (souvent distant)
git status git diff <i>fichier</i>	Affiche la liste des fichiers avec des changements (pas encore enregistrés par un commit) et git diff affiche les détails (lignes en + ou -) dans un certain fichier.
git add <i>liste_de_fichiers</i>	Ajoute un répertoire ou un fichier dans la liste des éléments qui seront pris en charge par git (lors du prochain commit).
git commit -m <i>message</i> [-a] ou [<i>liste_fichiers</i>]	Enregistre les derniers fichiers modifiés dans le référentiel git local (ceux précisés ou tous ceux ajoutés par add et affichés par status si -a)
git checkout <i>liste_de_fichiers</i>	Récupère les dernières versions depuis le référentiel local (sorte d'équivalent local du update de SVN, utile après un pull distant)
git --help git <i>cmde</i> --help	Obtention d'une aide (liste des commandes ou bien aide précise sur une commande)
git log --stat ou git log -p	Affiche l'historique des mises à jour -p : avec détails, --stat : résumé
git branch , git checkout <i>nomBranche</i> , git merge	Travailler (localement et ...) sur des branches
git grep <i>texte_a_rechercher</i>	Recherche la liste des fichiers contenant un texte
git tag <i>NomTag IdCommit</i>	Associer un tag parlant(ex: v1.3) à un id de commit.

git tag -l	Visualiser la liste des tags existants
git checkout tags/NomTag	Récupère la version identifiée par un tag

Exemples :

#initialisation

```
cd p1; git init
```

#affichage des éléments non enregistrés

```
cd p1; git status
```

→ affiche:

```
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#       modified:   src/f1.txt
#       modified:   src/f3-renamed.txt
no changes added to commit (use "git add" and/or "git commit -a")
```

commit all already tracked/added :

```
cd p1
# -a pour tous les fichiers listés dans git status
git commit -a -m "my commit message"
```

#commit all (with all new and deleted) :

```
cd p1
git add pom.xml.txt src/*
git status
# git commit gère tous les fichiers ajoutés (et supprimera de l'index ceux qui
# n'existent plus si option -a)
git commit -m "my commit message" -a
```

#checkout like local update

```
cd p1
git status
git checkout *
```

#historique des dernières mises à jour :

```
cd p1; git log --stat
```

---> affiche:

```
commit 93446a0f2194089d83c941a63768f212eb96e0f8
Author: developpeur fou <moi@ici_ou_la.everywhere>
Date:   Wed Dec 12 18:36:40 2012 +0100

    my commit message
pom.xml.txt      | 2 ++
src/f1.txt       | 1 +
src/f3-renamed.txt | 1 +
src/f4-renamed.txt | 1 +
```

<i>src/p/pf2-renamed.txt</i> 2 ++ 5 files changed, 7 insertions(+)
