

## **MÉMENTO GIT**

- git init
  - (sans option) : Crée un répertoire vierge pour un nouveau projet git
  - --bare : Créer un dépôt vierge de type Serveur
- git clone
  - (sans option) récupère un projet git existant depuis une URL
  - --bare : Créer un dépôt de type serveur a partir de la copie
- git status : Liste les fichiers modifiés et non *commité*
- git diff nomFichier : permet de voir les modifications effectué sur le fichier nommé et de le comparer à l'original.
- git add
  - . : ajoute tout les fichiers modifier à la liste des commits
  - nomFichier : ajout "nomFichier" à liste des fichiers à commiter
- git commit
  - -a : commit tous les fichiers modifié (-> qu'ils soient/ou pas dans la liste des fichiers à commiter.
  - -m (facultatif) : Ajouter un commentaire texte mis entre côte
    - nomFichier : commit "nomFichier" (si il fait parti de la liste des commits)
- git reset
  - --hard (facultatif) : Annuler le commit et le fichier local revient à la version antérieur précisé (cf : ci-dessous)
    - HEAD : revenir au dernier commit
    - HEAD^ : revenir à l'avant-dernier commit
    - HEAD^^ : revenir à l'avant avant-dernier commit
- git checkout
  - nomFichier: le fichier revient à la dernière version du commit. Toute les modifications sont supprimés.
  - nomBranche : Permet de switcher entre les branches. Toutes modifications non-commitées seront perdu.
- git pull : récupère les fichiers provenant du serveur
- git push : envoi les commits vers le serveur.
- git branch
  - (sans option) : liste les branches (\* : détermine la branche sur laquelle on se trouve)
  - (suivi du nomDeLaBranche) : créé une branche à partir de la branche sur laquelle on se trouve
  - -d nomDeLaBranche : supprimer la branche "nomDeLaBranche" SI elle a été mergé
  - -D nomDeLaBranche : Supprimer la branche peu importe son état.
- git merge nomDeLaBranche : fusionne les commits de la branche nomDeLaBranche sur la branche sur laquelle on se trouve.
- git stash

- (sans options) : Mettre en “pause” les modifications apporté sur la branche sans avoir à les commiter. Afin de ne pas les perdre.
  - apply : permet de revenir à l’état de modification lors du précédent “git stash”.
  - --track localBranch origine/serverBranch : copie la branche du serveur vers le dépôt git en local
  - pop : récupère les dernières modification qui ont été stash sur une branch quelconque, vers la branch sur laquelle on se trouve
- git grep
  - -n : permet de rajouter le numéros de ligne à la liste des résultats de la recherche.
    - “Mot à rechercher” : recherche “Mot à rechercher” dans tout les fichiers du dépôt git.
- git remote
  - -v : liste les serveurs liés au dépôts
  - add (origin ssh://.../.git) : permet de rajouter un nouveau point d'accès “origin” au dépôt
  - rm (origin) : supprime le point d'accès “origin”

## **INFORMATIONS**

1. Lorsque les fichiers sont “NOUVEAU” (n’existe pas sur le dépôt GIT), il faut obligatoirement faire un “git add .” pour l’ajouter et ensuite les commiter.
2. Pour ignorer un fichier et/ou un répertoire. On crée un fichier “.gitignore” à la racine du dépôt, dans lequel on y indique le nom de dit fichier et/ou répertoire à ignorer. 1 par ligne.

→ Déposé la clé ssh “id\_rsa.pub” dans le répertoire ~/.ssh/authorized\_keys/  
 → git clone ssh://u71831724@liwix.fr/~/DEPOTS/liwix.git

Source :

- <http://www.frenex.fr/Connexion-SSH-par-cle-privee-cle.html>
- [http://wiki.deimos.fr/index.php?title=Mise\\_en\\_place\\_d'un\\_serveur\\_et\\_client\\_Git](http://wiki.deimos.fr/index.php?title=Mise_en_place_d'un_serveur_et_client_Git)