GIT PENSE BÊTE

CONFIGURATION DU GIT
git init > création du dossier caché git
git config --global user.name "votrenom"
git config --global user.email "votreemail"
git config --global --list

GESTION DE COMMIT
git status > vérification du status du commit actuel
git add nomduficher > indexer un fichier
git reset nomdufichier > desindexer un fichier
git add . > tout indexer
git commit -m"message" > faire un commit dans le depot local
git commit -a -m "message" > indexer et commit en meme temps
git diff [fichier ou rien] > permet de voir les différences effectué avant indexation
git revert [nom du commit] > permet d'annuler un commit et de garder l'historique d'annulation
git reset [nom du commit] > permet d'annuler un commit mais ne garde pas l'historique

GESTION DE L'HISTORIQUE
git log >affiche l'historique des commits
git log -n 2 > afficher l'historique des 2 derniers commits
git log [nom de branche] > afficher les commits de la branche choisie
git show [sha-1-tag]
git checkout [sha-1-tag-master] deplacer le pointer head sur un ancien commit
git tag [nomdutag]
git tag --delete [nomdutag]

CONNEXION AU GITHUB
git remote add origin [chemin https du repository]
git remote -v
git remote set-url origin [chemin https du repository] > changer le repository

CLONER
git clone [chemin https du repository] [nomdudossier] > copie le repository distant en local

PUSH
git push -u origin master > push le projet en distant
git push --tags > push les tags en distant

PULL
git pull origin master > pull le projet en local

CREER UN FICHIER .GITIGNORE
touch .gitignore

GESTION DE CONFLIT DE COMMIT

RESOLUTION EN MERGE:

faire un git pull et gérer les modification manuellement suivis d'un git add et commit
git merge --abort > Si on veut annuler le merge en cours

RESOLUTION EN REBASE:

git pull --rebase

GESTION MULTIUTILISATEUR

git blame [nomdufichier] > regarde les lignes modifiés selon utilisateur
git blame -L 10,20 [nomdufichier] > regarde juste les ligne 10 a 20

GESTION DES BRANCH

git branch [nomdebranch] > creer une nouvelle branche
git branch > voir les branches existante
git checkout [nomdebranch] > se deplacer sur une branche
git branch -a > voir les branches sur le serveur
git branch -d [nom de la branche a supprimer] (ne pas etre sur la branch a supprimer)
git push origin --delete [nom de la branche a supprimer sur serveur distant]
git merge [nomdebranch a merge] > merger la branche choisi en étant sur le master

FAIRE UN CHERRY PICK

git cherry-pick [sha1 du commit voulu] > selectionne le commit pour le placer en haut du master