

BRANCHES

`git branch`
liste les branches

`git branch -a`
liste toutes les branches (dont les distantes)

`git branch <new_branch>`
crée une nouvelle branche à partir de la branche courante

`git checkout -b <new_branch>`
crée une nouvelle branche à partir de la branche courante et se positionne dessus

`git checkout -b <new_branch> <other_branch>`
crée une nouvelle branche à partir d'une autre branche et se positionne dessus

`git checkout <branch>`
se positionne sur une branche

`git branch -d <old_branch>`
supprime une branche

`git push <origin> <branch>`
6 pousse une branche sur le dépôt distant

`git push -u <origin> <master>`
6 pousse une branche sur le dépôt distant et la traque (devient la branche par défaut pour les futures `git push/git pull` sans paramètre)

`git push <origin> :<branch>`
supprime une branche du dépôt distant

HISTORIQUE ET ÉTAT

`git status`
affiche l'état du dépôt local

`git log`
affiche l'historique des commits

`git log --oneline --decorate --all --graph`
affiche l'historique complet de toutes les branches sous forme d'arbre

`git show <refname>`
affiche les données d'un tag, d'un commit (commit, commentaires)

`git diff <oldref> <newref>`
compare les différences entre 2 commits

TAGS

`git tag`
liste les tags

`git tag -l "<pattern>"`
liste tous les tags correspondant à un pattern

`git tag -a <name> -m "<comment>"`
appose un tag sur un commit particulier

`git tag -d <tag>`
supprime un tag

`git checkout tags/<tag>`
récupère le contenu d'un tag (en detached HEAD)

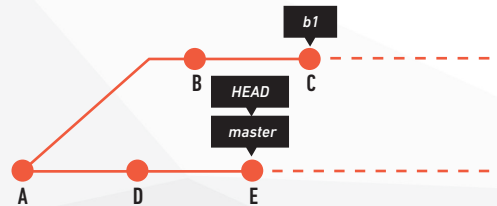
`git push <origin> <tag>`
pousse un tag sur le dépôt distant

`git push <origin> --tags`
pousse tous les tags sur le dépôt distant

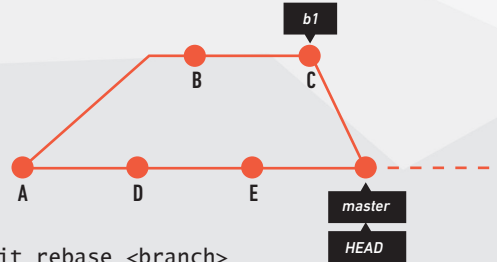
`git push <origin> :<tag>`
supprime le tag du dépôt distant

FUSIONS

AVANT :



APRÈS : `git merge <branch>`
fusionne une branche sur la branche courante



`git rebase <branch>`
fusionne une branche sur la branche courante en aplatissant l'historique



SQLI
ENTERPRISE

CONFIGURATION

Changer les options git config [--global] <varname> <value>
Valeur utiles :

user.name
nom du committer

user.email

adresse e-mail du committer

http.proxy, https.proxy

proxy HTTP/S de la forme : http://user:password

host:port

alias.<alias_name>
crée un nouvel alias.

Création de certificat

ssh-keygen -t rsa -C "<my@email.com>"

et laisser le certificat se générer dans le chemin habituel :

\$HOME/.ssh/id_rsa

Tester la connexion SSH à un dépôt (github par exemple)

ssh -T <git@github.com> -i <path/to/id_rsa>

CRÉATION DE DÉPÔTS

git init .

crée un nouveau dépôt dans le répertoire courant

git clone <url>

git clone <url> <my_repo>

clone un dépôt distant

git remote add <origin> <url>

ajoute un dépôt distant origin au dépôt local

git remote -v

liste les dépôts distants associés au dépôt local

MISES À JOUR

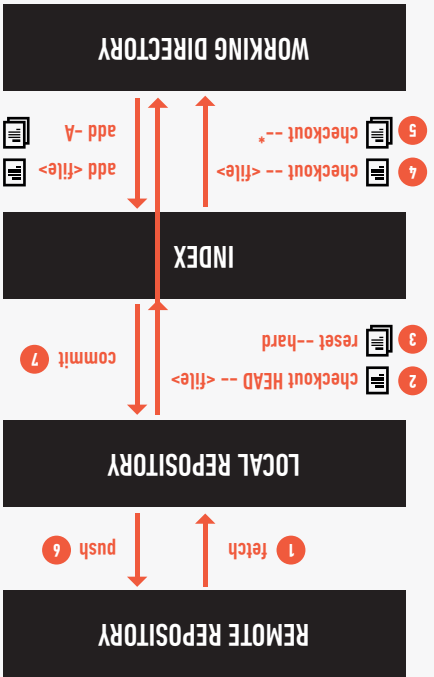
1 git fetch <origin> <branch>
récupère une branche distante, sans impacter le répertoire de travail

git pull <origin> <branch>
met à jour mon répertoire de travail avec le contenu distant de master

CHANGEMENTS

Commande pour un fichier
Commande pour plusieurs fichiers

git add <new_file>
8 ajoute un fichier modifié (ou non traqué) à l'index
git add -A
9 prend en compte les ajouts, modifications et suppressions de fichiers dans l'index
git add .
9 ajoute tous les fichiers modifiés (ou non traqués) à l'index
git mv <old_file> <new_file>
renomme/déplace un fichier
git rm <file>
supprime un fichier localement et dans l'index
git commit -m "<comment>"
7 commit les fichiers présents dans l'index
git rm --cached <file>
supprime un fichier de l'index exclusivement



ANNULATIONS

git reset <file>
annule l'ajout d'un fichier dans l'index
git reset
annule tout le contenu de l'index

Working directory
git checkout HEAD --<file>
2 annule la modification d'un fichier et récupère le contenu du dernier commit

git checkout <refname> <file>
récupère le contenu d'un fichier pour un commit particulier

git reset --hard
3 annule toutes les modifications en cours et récupères le contenu du dernier commit

git checkout --<file>
4 annule la modification locale d'un fichier et repositionne sur le contenu de l'index

git checkout --*
5 annule toutes les modifications en cours et récupère le contenu de l'index
git reset --hard HEAD
annule le dernier commit (et repositionne le contenu working directory avec le dernier commit)

git commit --amend -m "<my corrected comment>"
réécrit le commentaire du dernier commit

MISES DE CÔTÉ

git stash -u
met de côté son travail (espace de travail et index en prenant en compte les fichiers non traqués)

git stash list
liste les stashes disponibles

git stash pop
dépile le dernier stash ajouté