

CONFIGURATION BASIQUE

L'auteur

```
git config --global user.name "XXX"
git config --global user.email "XXX"
```

L'éditeur par défaut de git

```
git config --global core.editor nano
```

Afficher la configuration

```
git config --list
```

CRÉATION

Cloner un dépôt existant

```
git clone
```

Créer un nouveau dépôt local

```
git init
```

Créer un nouveau sur serveur

```
git init --bare
```

MODIFICATIONS LOCALES

Etat de votre dépôt

```
git status
```

Afficher les modifications

```
git diff
```

Ajouter les fichiers au commit

```
git add .
```

Ajouter un fichier au commit

```
git add <fichier>
```

Publier le commit

```
git commit -m "<description>"
```

Modifier le dernier commit

```
git --amend
```

AI DE MÉMOIRE git

Toute remarque ou suggestion pour améliorer cet aide mémoire Git est la bienvenue.

Fabrice DEJAIGHER - Chtiland.fr

Sites Git

<http://git-scm.com>

<https://github.com>

MAJ & PUBLISH

Liste des dépôts git distants

```
git remote -v
```

Publier sa branche sur le dépôt origin

```
git push origin master
```

Récupérer tous les changements du dépôt distant SANS LES INTÉGRER

```
git fetch origin
```

Fusionner les changements d'une branche dans le HEAD actuel

```
git merge <branche>
```

Récupérer tous les changements du dépôt distant ET LES INTÉGRER

```
git pull origin master
```

pull est équivalent à fetch+merge

HISTORIQUE

De tous les commits depuis le début

```
git log
```

D'un fichier en particulier

```
git log -p <fichier>
```

Qui et quand a modifié un fichier ?

```
git blame <fichier>
```

BRANCHES & TAGS

Lister toutes les branches existantes

```
git branch -a
```

Déplacer le HEAD sur une branche

```
git checkout <branche>
```

Créer une branche basée sur le HEAD

```
git branch <nouveau_nom>
```

Supprimer une branche locale

```
git branch -d <branche>
```

TAGS (VERSIONING)

Liste des tags

```
git tag -l
```

Ajouter un tag à la branche courante

```
git tag v1.0
```

Pousser le(s) tag(s)

```
git push origin --tags
```

ANNULER

Annuler tous les changements locaux de votre dossier de travail

```
git reset --hard HEAD
```

Annuler les changements sur un fichier en particulier

```
git checkout HEAD <fichier>
```

Redéfinir votre HEAD sur un ancien commit et ...

...annuler toutes les modifications depuis

```
git reset --hard <commit>
```

...préserver toutes les modifications comme non indexées

```
git reset <commit>
```

...préserver les modifications locales non committées

```
git reset --keep <commit>
```

Pour annuler les changements committés

```
git revert <commit>
```