

GIT TUTOR

Serveurs Git

- <https://github.com> (dépôt privé payant)
- <https://bitbucket.org> (dépôt privé gratuit)
- <https://gitlab.org> (open source)

Configuration

Aide

```
git config
```

Projet | Utilisateur | Système

```
git config [ -- [ local | global | system ] ] -l
```

Emplacements

- local : `.git/config`
- global : `~/.gitconfig`
- system : `/etc/gitconfig`

Lecture | Ecriture

nécessaire à la première utilisation:

```
git config --get user.name
git config --get user.email

git config --add user.name inuitLarson
git config --add user.email francois@larsonneur.fr
```

Alias

```
git config --global alias.st status
git config --global alias.co checkout
```

permet le raccourci :

```
git st[atus]
```

directement dans ~/.gitconfig

```
[alias]
  st = status
  co = checkout
  l = log --pretty=oneline --abbrev-commit --graph --decorate --all
  b = branch
  cg = config --global
  a = add
  c = commit
  cm = commit
```

```
cgl = config --global -l
```

Repository

Nouveau

```
git init
```

Projet existant

```
git clone [url]
```

Historique

Par défaut HEAD :

```
git log [REF]  
git show [ REF ]
```

où REF = HEAD | 40 caractères (chawan!)

Modification

De la copie locale:

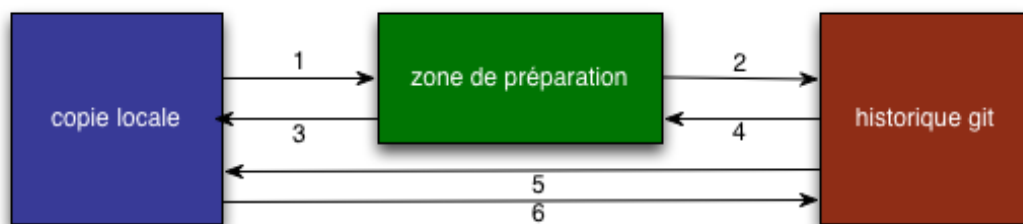
```
git diff
```

De la copie indexée non commitée

```
git diff --staged
```

***Changements d'états d'un fichier:**

non indexé (untracked) | modifié (modified [not staged]) | indexé (staged) | commité



1 (ajouter | supprimer | déplacer un fichier dans la zone de préparation)

```
git add [file]  
git add -u ( fichiers connus par git )  
git rm file ( suppression )  
git mv file ( déplacement )
```

2 (commiter les modifications [d'un fichier])

```
git commit [file]
git commit -m "message" [file]
```

3 (ramener la zone de préparation dans la copie locale)

```
git checkout -- <file>
```

4 (ramener HEAD dans la zone de préparation)

```
git reset HEAD <file>
```

5 = 4 & 3 (ramener HEAD dans la copie locale)

```
git checkout HEAD <file>
```

6 = 1 & 2 (comité directement une modification)

git commit -a[m "message"] [file]

Générer une nouvelle clé SSH

lister mes clés SSH

```
ls -al ~/.ssh
```

nouvelle clé SSH

```
ssh-keygen -t rsa -b 4096 -C "inuit.larson@orange.fr"
```

puis les réponses par défaut.

copier la clé public vers Github → settings → SSH keys)

```
cat ~/.ssh/id_rsa.pub
```

cloner en utilisant le protocole SSH (+sûr+rapide)

git clone "git@github.com:InuitLarson/Realtime-Markdown-Viewer.git"

Exporter un repository (dans un répertoire existant)

```
git archive master | tar -x -C ~/export
git archive master | gzip > export.tar.gz
```

Tagger un commit

créer

tag léger (local)

```
git tag TAG [REF]
```

tag lourd (lourd)

```
git tag -a [-m MESSAGE] TAG [REF]
```

supprimer

```
git tag -d TAG  
git push --delete REMOTE TAG
```

publier

```
git push --tag
```

git push REMOTE TAG

Gestion des branches

lister

Par défaut, les branches locales :

```
git branch [ -r(emote) | -a(ll) ]
```

créer

Sans changer de branche :

```
git branch BRANCH [REF]
```

En changeant de branche :

```
git checkout -b BRANCH [REF]
```

équivalent à :

```
git branch BRANCH [REF]  
git checkout BRANCH
```

Attention : Modification du dépôt local

supprimer

La branche doit être fusionner :

```
git branch -d BRANCH
```

sinon :

```
git branch -D BRANCH
```

Attention : Les commits de la branche sont orphelins. Ils seront supprimer sous 90 jours (valeur par défaut du délai d'optimisation du dépôt Git).

fusionner

```
git merge BRANCH
```

résoudre des conflits

```
git diff  
git mergetool
```

Modifier l'outil de résolution de conflit :

```
git config --global merge.tool kdiff3
```

Sous Mac, avec l'*alias.cg* = "*config --global*" :

```
git cg mergetool.kdiff3.path "/Applications/kdiff3.app/Contents/MacOS/kdiff3"
```

annuler la résolution d'un conflit

Revenir avant le merge:

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

Publier