

Une version de fichier suivi par git est dans 3 états, ou alors il n'y a pas encore de suivi de version sur le fichier (untracked)

modifiée(modified), indexée (staged) et validée (committed).

Modifiée (modified) : Fichier qui a subi des modifications, mais qui n'a pas été validé dans la base de donnée de suivi des versions du projet.

Indexée (staged) : Le fichier a été marqué dans sa version courante pour être validé (committed) dans le prochain appel à git commit.

Validée (committed) : Toutes les versions du fichier sont stockées dans la base de données de suivi des version du projet.

L'aide en ligne de toute commande git peut être consultée avec la commande suivante :

```
git help <nom_de_commande>
```

Configuration de git :

Les valeurs passées avec l'option --global sont stockées dans le fichier ~/.gitconfig et peuvent aussi y être modifiées.

```
git config --global user.name "Bob Eponge"
```

Mettre la valeur du nom de l'auteur des modifications à « Bob Eponge ».

```
git config --global user.email bob@png.com
```

Mettre la valeur de l'e-mail de l'auteur des modifications à bob@eponge.com

```
git config --list
```

Afficher toutes des valeurs de configuration existantes

Exemple

```
$ git config --global user.name "Bob eponge"
$ git config --get user.name
Bob eponge
$ git config --global user.email "bob@png.com"
$ cat ~/.gitconfig
[user]
    email = bob@png.com
    name = Bob eponge
```

Commande de création ou clonage de dépôt git

Toutes les méta-données ainsi que le contenu de chaque version du dépôt sont stockée dans le répertoire .git de la racine du dépôt. Les autres fichiers ne sont qu'une version possible parmi d'autres du dépôt.

git init Initialise un nouveau dépôt git dans le répertoire courant

git clone Clone un dépôt git existant depuis l'adresse fournie

Exemple

```
$ git init
Initialized empty Git repository in ...

$ git clone /mon_super_depote/projet_morpheus
$ git clone hoerdtm@A433-01:projet_morpheus
```

Ajouter des fichiers dans l'état Indexé ou Validé

Les fichiers qui sont effacés d'une version doivent aussi être commités : ils n'apparaîtront plus dans les versions suivantes, à moins qu'on les rajoute à nouveau.

git add <fichier1> ... <fichier2> <repertoire> Place un ensemble de fichiers ou du contenu d'un répertoire dans l'état « staged »

git commit Place un ensemble de fichiers « staged » dans l'état « committed ».

Après cette commande les fichiers font partie d'une version déterminée du dépôt : il y a une nouvelle version après chaque appel à commit.

```
git commit -a
```

Équivalent à :

```
git add <fichier> ; git commit <fichier>
```

```
git commit -m « message »
```

Inscrit directement le message de log de modification sans passer par l'éditeur texte.

Exemple

```
$ git add *.c
$ git add README
$ git commit -m "Version initiale du projet"
```

Etats des fichiers et log de suivi

```
git status
```

Affiche l'état de chaque fichiers. Si affiche «nothing to commit », le dépôt est à jour.

```
git log
```

Affiche pour chaque commit, son auteur, sa date et un hash qui représente l'état de l'index des version au moment du commit. Ce hash peut-être utilisé pour basculer sur une version précédente.

```
git diff
```

Affiche les différences entre les fichiers modifiés et la version actuelle du dépôt au format *patch*

Exemple

```
$ git status
# On branch master
nothing to commit (working directory clean)

$ git log
commit b92cf85f3071d88d04bd516c5197c0d4cd6c09ef
Author: Bob eponge <bob@png.com>
Date: Tue Oct 7 14:54:23 2014 +0200

    version initiale

$ git diff
diff --git a/README b/README
index ce88c3b..e54a5b3 100644
--- a/README
+++ b/README
@@ -1,2 @@
+Une petite modification
texte initial
```

Branches git :

Plusieurs versions évoluant différemment peuvent être maintenues par git. On les appelle des branches. La branche racine de toutes les autres est appelée « master »

```
git branch
```

Affiche la liste des branches du projet ainsi que la branche courante. Le signe * indique la branche actuelle.

```
git checkout <nom_branch>
```

Passe l'ensemble des fichiers suivis sur la branche appelée nom_branch

```
git checkout -b <new_branch>
```

Crée une nouvelle branche appelée new_branch

Exemple
<pre>\$ ls README \$ git branch master * maverion \$ git checkout master Switched to branch 'master' #le fichier README n'est pas dans la branch master du dépôt. \$ ls \$</pre>

Git et partage des données via un serveur :

Un dépôt git peut très bien fonctionner sans serveur. Utiliser un serveur pour stocker son/ses dépôts n'est utile que si on veut partager les différentes version de son/ses dépôts avec d'autres ou alors qu'il soit accessible via un réseau.

Afin de partager un dépôt avec d'autres, il faut exporter un dépôt existant dans un dépôt nu, c'est à dire un dépôt qui ne contient pas de répertoire de travail. Par convention le nom de ce dépôt fini par .git.

```
git pull
```

Télécharge les dernières modifications du dépôt présent sur le serveur

```
git push
```

Partage les modifications du dépôt local sur le serveur

```
git remote add origin ssh...
```

Définition du serveur par défaut pour les commandes distantes

```
git remote -v
```

Affiche le/les serveurs utilisée pour les commandes distantes.

```
git clone --bare monprojet monprojet.git
```

Clône un dépôt existant monprojet et crée un nouveau dépôt nu, c'est à dire qui ne contient pas de copie du répertoire de travail.

Annuler des modifications dans l'état indexé

De manière générale seules les modifications de fichiers dans l'état « indexé (staged) » peuvent être facilement annulées/corrigées, une fois validés/commités, les changements sont conservés dans la base de données du dépôt. Aussi, avant de faire un commit, il vaut mieux vérifier si tout est correct plutôt que revenir en arrière par la suite.

```
git reset HEAD <fichier>
```

pour enlever un fichier qui est de l'état staged pour lequel on voudrait rajouter une modification.

```
git commit --amend
```

Annule dernier commit

Exemple
<pre>\$ git remote -v origin hoerdtm@A433-01:testgit.git (fetch) origin hoerdtm@A433-01:testgit.git (push)</pre>