

## Bibliographie

- [1] Git Official Website, "Git Documentation", <https://git-scm.com/doc>
- [2] Academind, "Git MERGE vs REBASE", <https://youtu.be/CRIGDDprdOQ>
- [3] Academind, "Git Tutorial for Beginners - Crash Course", [https://www.youtube.com/watch?v=\\_OZVJpLHUaI](https://www.youtube.com/watch?v=_OZVJpLHUaI)
- [4] GitHub, "GitHub Guides", <https://guides.github.com/>
- [5] Academind, "GitHub Tutorial for Beginners | The Basics", <https://www.youtube.com/watch?v=mVnZVw4KJnc>
- [6] GitHub, "Connecting to GitHub with SSH", <https://help.github.com/en/github/authenticating-to-github/connecting-to-github-with-ssh>
- [7] Git Official Website, "GUI Clients", <https://git-scm.com/download/gui/linux>

## Configuration de Git

### ATTENTION :

- ⇒ Nécessaire d'adapter votre configuration de GIT à la configuration à respecter pour le projet
- ⇒ avant d'effectuer des modifications sur les fichiers !

Fichiers de configuration	Commandes	Interface graphique des GUI Git
---------------------------	-----------	---------------------------------

## Repository

Le chemin d'accès vers le « serveur » et le « disque dur » où est enregistré le « code history » Git. Il contient tous les fichiers et toute la structure des historiques des fichiers.

## Branch

On peut l'imaginer comme un « dossier » à l'intérieur du « repository ». On, y ajoute les modifications des fichiers ainsi appelés « commits ».

## Commits

Ce sont les différentes étapes/versions du code source à l'intérieur d'une branche.

## Merge

Le merge va permettre de fusionner une certaine branche dans une autre.

Exemple : Un développeur a terminé le codage d'une nouvelle fonctionnalité sur son ordinateur. L'opération de « fusion-merge » va permettre de fusionner son travail dans le code source collectif du projet ou de l'équipe.

## Conflits

Lorsque l'on essaye de fusionner deux branches dont une ligne a été modifiés dans les deux branches, un conflit se crée. Il n'est pas possible dans ce cas de réaliser une « fusion automatique » et il est nécessaire de résoudre le conflit manuellement, en acceptant l'une ou l'autre des modifications. Il est également possible de réécrire le code manuellement de façon à inclure les deux lignes.

```
$ git merge new-feature
Auto-merging test.txt
CONFLICT (content): Merge conflict in test.txt
Automatic merge failed; fix conflicts and then commit the result.
```

## Commandes GIT

Commande	Description
<b>git init</b>	Initialiser un nouveau dépôt Git
<b>git status</b>	Afficher les informations sur l'état actuel de la branche et des fichiers
<b>git add .</b>	Ajouter tous les fichiers afin qu'ils soient « traqués » par Git
<b>git commit -m "message"</b>	Mémoriser la modification des fichiers (Le message est une description succincte des changements effectués sur les fichiers)
<b>git branch</b>	Affiche les branches disponibles
<b>git log</b>	Affiche les logs sur une branche
<b>git reset --hard id-du-commit</b>	Permet de réinitialiser l'état du dépôt local vers un certain commit et d'effacer tous les commits intermédiaires
<b>git checkout --</b>	Permet de réinitialiser l'état du dépôt et revenir au dernier commit
<b>git checkout -b nom-de-la-branche</b>	Permet de créer et de basculer vers une nouvelle branche
<b>git checkout nom-de-la-branche</b>	Permet de basculer vers une certaine branche
<b>git checkout -f nom-branch</b>	Permet d'effacer les changements locaux et de se positionner sur une certaine branche
<b>git merge nom-de-la-branche</b>	Permet de fusionner une certaine branche dans la branche courante
<b>git branch -D nom-de-la-branche</b>	Permet de supprimer une branche
<b>git remote add origin url-repository</b>	Permet d'ajouter une origine au repository local. C'est une connexion entre le repository local et le repository « origin » distant
<b>git remote</b>	Permet d'afficher la liste des connexions remote pour le repository local
<b>git remote -v</b>	Idem à remote mais avec le détail des urls des connexions
<b>git remote rm nom-connexion</b>	Supprimer une connexion
<b>git push origin master</b>	Permet d'envoyer les données de la branche « master » vers la branche automatiquement sur le serveur « origin » de cette branche
<b>git branch -r</b>	Permet d'afficher la liste des branches sur le serveur
<b>git fetch</b>	Vérifier si les fichiers ont évolué sur le serveur git distant
<b>git pull</b>	Télécharger les dernières versions disponibles des fichiers d'une branche et les fusionner automatiquement avec le repository local
<b>git config --list</b>	Afficher la configuration (Voir aussi le fichier "~/.gitconfig" qui se trouve dans le dossier personnel de l'utilisateur)

## Github

Un service qui propose d'héberger votre serveur Git. En créant un compte sur Github vous allez pouvoir facilement utiliser le système Git. Mais c'est optionnel. Vous n'avez pas besoin de Github pour bénéficier du système Git.

Nous allons pouvoir enregistrer notre repository « sur le cloud » grâce à Github.

## IHM pour GIT

Git peut être utilisé en ligne de commande ou bien avec une interface Graphique.

Exemples d'interfaces : <https://git-scm.com/download/gui/linux>

On peut citer par exemple Git Extensions qui fonctionne sur Windows

## Comparateur fichier

Un comparateur de fichier va permettre de mettre en évidence les différences entre plusieurs fichiers ou dossiers.

Exemple : « diff », « Winmerge », « Beyond and Compare » ...

## Restituer un GIT

Toujours penser aux personnes novices qui ne sont pas experts en informatique. Il est nécessaire de savoir retranscrire les informations à quelqu'un de novice extérieur au projet.

Fournir des URLs accompagnés d'une description ainsi qu'une explication.