

commit :	Un ensemble de changements sur le projet (ajout-retrait-modification de fichiers) par rapport à la version précédente.
ammend* :	Permet de fusionner les changements avec le dernier commit au lieu d'en créer un nouveau.
push :	Envoyer ses derniers commits au serveur pour les partager.
fetch :	Récupérer les derniers changements et commits depuis le serveur.
pull :	Équivalent de fetch suivit de merge(ou de rebase avec pull --rebase)
.gitignore :	Fichier spécial listant les fichiers et dossiers à ignorer par git (cache, fichiers temporaires...)
branch :	Suite de commits indépendants d'une autre branche. Souvent utilisé pour travailler sur des fonctionnalités instables, incomplètes, ou expérimentales sans modifier le projet principal.
master :	Nom par défaut pour la branche principale.
origin :	Nom par défaut représentant le serveur. Sa branche principale est donc origin/master par défaut.
merge :	Action de fusionner deux branches, pour ramener les changements de l'une dans l'autre. Souvent utilisé lorsqu'une branche fonctionnalité est terminée, et peut être intégrée dans master.
rebase* :	Action de décaler la base d'une branche ou d'un groupe de commits sur un autre commit. Surtout utilisé en rebasant une branche sur une autre afin de les avoir à la suite et non côte à côte, et ainsi éviter un merge dans l'historique. Le mode interactif permet aussi de nettoyer l'historique des commits avant un push.
conflit :	Lors d'un merge ou d'un rebase, les modifications des deux côtés sont mises en commun. Si un fichier est modifié dans les deux versions, le système ne peut choisir quelle version garder : il y a conflit. Il faut alors fusionner les modifications manuellement.

*Modifie l'historique. À ne JAMAIS utiliser sur des commits qui ont déjà été push.