

GIT

Commandes Git	Définitions	Syntaxe	Options	Autres options	Exemples
Git config	Configurer le nom utilisateur pour tout Git	<code>Git config --global user.name 'nom utilisateur'</code>	--global	User.name	<code>git config --global user.name "Rabah Daguelou"</code>
	Configurer l'adresse mail utilisateur pour tout Git	<code>Git config --global user.email adresse mail</code>	--global	User.email	<code>git config --global user.email lounislina@yahoo.fr</code>
Mkdir	Créer un repertoire	Mkdir nom-Repertoire			Mkdir projet1
Cd	Accéder à un repertoire	Cd nom-repertoire			Cd projet1
Touch	Créer un fichier	Touch nom-fichier.ext			Touch fichier1.txt
Ls	Lister les contenu d'un repertoire	Ls			Ls
Git init	Initialiser un dépôt Git	Git init (chemin) nom-repertoire			Git init projet-git1
Git status	Affiche l'état des fichiers dans un repertoire	Git status (chemin)			Git status
Git add	-Indexer un fichier suivi (avant la validation) -Placer sous suivi des fichiers non suivis en les indexant	Git add (chemin) nom-fichier	. Git add . (Mettre tous les fichiers dans la zone d'index)		Git add fichier.txt Git add *.txt Git add *.* Git add projet-git1
Git commit	Valider des fichiers et les ajouter en base	Git commit (chemin)(fichiers)	-m (ajouter directement le message)	-a Pour éviter de passer	Git commit Git commit -m 'nouveau fichier'

			sans ouvrir l'éditeur VIM)	par git add (mettre d'un coup des fichiers suivis modifiés dans la zone d'index et les valider.)	
Git log	Afficher l'historique des commits réalisés	Git log			Git log
Git clone	Cloner un projet	Git clone (chemin nom-projet à cloner) (chemin nom-projet cloné)	Cloner un dépôt en local		git clone https://github.com/rabah-daguelou/p3_daguelou_rabah.git
rm	Supprimer un fichier (Le fichier reste suivi par git)	Rm (chemin) repertoire/fichier			Rm fichier.txt
Git rm	Supprime un fichier et l'exclure du suivi de version	Git Rm (chemin) repertoire /fichier			Git rm fichier1.txt
Git rm – cached	Exclure un fichier du suivi sans le supprimer	Git rm –cached (chemin)			Git rm –cached fichier.txt
Echo	Ajout du texte dans un fichier	Echo 'texte à ajouter' >> (chemin) fichier			Echo 'je rajoute du texte dans ce fichier' >> fichier.txt
cat	Afficher le contenu d'un fichier	Cat (chemin) nom-fichier			Cat fichier1.txt
Git reset head	Retirer une version d'un fichier de la zone d'index				Git reset head fichier.txt

Git mv		git mv ancien-nom- fichier nouveau- nom-fichier.			Git mv fichier.txt fichier.doc
Git diff	Affiche les modifications apportées aux fichiers				
Git branch	1-Lister les branches 2-Créer une branche	Git branch Git branch (nom de la branche)			Git branch Git branch branche1
Git checkout	Changer de branche	Git checkout (nom de la branche)			Git checkout branche1
Git merge	Rapatrier les modifications d'une branche sur la branche principale Master	Aller sur Master (Git checkout master) Git merge (nom de la branche à fusionner)			Git merge branche1
Git branch -d	Supprimer une branche	Git branch -d (nom de la branche)			Git branch -d branche1
Git remote add	Créer un dépôt distant	Git remote add (nom du dépôt) (url du dépôt)	Pointer sur le dépôt distant	Git remote add rd https://github.com/rabah-daguelou/p3_daguelou_rabah.git	
Git push	Déposer un projet	Git push (label du dépôt) (nom de la branche à déposer)			Git push origin master
Git stash	Créer une remise (Annuler une modification sans la supprimer)	Nettoyer la branche master par exemple de modifications non voulues.			Git stash
Git stash list	Afficher la liste des remises				

Git stash apply	Récupérer les remises créés pour les commiter	Git stash apply (identifiant) Chercher dans la liste des remises			Git stash apply stash@{0}
-----------------	---	---	--	--	---------------------------

Vocabulaire :

Untracked : non versionné (n'est pas suivi)

Commit : validé en base

Fileglobe : tous les fichiers

Changes to be committed : modifications à valider

Master : (La branche)

Root-commit : Premier commit

Somme de contrôle : ex. c8cc4bc : Permet d'identifier de manière unique

Nothing to commit, working tree clean : Aucun fichier à valider (Tous les fichiers donc sont sous suivi de version et enregistrés en base)

Un commit est tout simplement un enregistrement de votre travail à un instant T sur la branche courante où vous êtes.

La commande **Git push** permet d'envoyer les modifications que l'on a réalisées en local sur le dépôt à distance, alors que la commande **Git pull** permet de récupérer en local le projet distant !

La suppression d'une branche entraînera la suppression de tous les fichiers et modifications que nous n'aurez pas commités sur cette branche.

La commande `git remote` vous permet de créer, d'afficher et de supprimer des connexions avec d'autres dépôts.

`git remote add <name> <url>` Permet de créer une connexion avec un dépôt distant.