Téléchargement : <https://git-scm.com/>

Documentation officielle : <https://git-scm.com/docs>

|  |
| --- |
| git (Local uniquement)- Basics |

Initialiser git dans un repos :

* git init => Va initialiser git dans le répertoire (.git)

Configurer le nom d’utilisateur :

* git config user.name nom\_a\_utiliser

Créer une branche :

* git branch nom\_de\_la\_branche

Se déplacer dans les branches :

* git checkout nom\_de\_la\_branche

Supprimer une branche :

* git checkout -d nom\_de\_la\_branche

Renommer une branche :

* git checkout -m nom\_ancienne\_branche nouvelle\_branche

Afficher les fichiers indexés ou non :

* git status

Add des fichiers à l’index :

* git add nom\_duçfichier
* OU git add .
* OU git add -i (intéractif)

|  |
| --- |
| Versionner le travail |

Versionner :

* git commit -m « message du commit »
* OU git commit -am « message du commit »

|  |
| --- |
| Annuler les changement |

Créer une nouveau commit depuis un ancien :

* git revert id\_du\_commit

Reset le travail (/ !\ uniquement si vous êtes sûre de vous) :

* git reset –hard id\_du\_commit

Retirer les fichiers inutiles du dossier de travail (dossier de travail .git, non pas dossier réel) :

* git clean -n (vérifier les documents)
* git clean -f (les effacer)

Update le dernier commit :

* git commit --amend

|  |
| --- |
| Branches (avancé) |

Créer une branche et s’y déplacer :

* git checkout -b nom\_de\_la\_branche

Fusionner deux branches :

* git checkout nom\_de\_la\_branche\_destinatrice
* git merge nom\_de\_la\_brancheçrappatriée

|  |
| --- |
| Lier à github/gitlab |

Récupérer un repository existant sur github :

* git clone lien\_du\_github

Créer une repository avec un travail existant en local :

* Créer une repository VIDE sur github (sans readme !)
* Copier coller les commandes données dans «…or create a new repository on the command line »

Push (bonne pratique) :

* git pull
* git add .
* git commit -m « message »
* git push

Récupérer le travail d’une branche non existante en local :

* git pull origin non\_de\_la\_branche\_distante

Récupérer le travail lié local/distant :

* git pull

Récupérer toutes les branches distantes :

* git fetch --all

Retour à une version précédente (classique) :

* git log
* Récupérer l’id de la version voulue
* git revert ID
* git add .
* git commit -m « message »
* git push

Retour à une version précédente (Si revert ne marche pas) :

* git log
* Récupérer l’id de la version voulue
* git reset –hard ID
* git add .
* git commit -m « message »
* git push – force

Merge 2 branche :

* git checkout branche\_vers\_la\_queulle\_merge
* git merge branche\_a\_arécuperer

Remiser des modifs :

* git stash

Utiliser des couleurs dans la sortie de git :

* git config color.ui true