

Commandes importantes Git:

by Thibaut ☺

- Avant de commencer :

Git est souvent installé par défaut, mais il faut quand même enregistrer ses informations avant de pouvoir l'utiliser totalement.

- `git config --global user.name 'nom'`
- `git config --global user.email 'email'`

- Commencer un nouveau repo:


- > Créer le dossier du repo
- > Ouvrir un terminal avec comme current dir le dossier en question et faire un `git init`

- Sauvegarder des choses avec Git:

- > Il faut ajouter les fichiers que l'on veut save avec Git, ce n'est pas automatique !
A chaque fois qu'un nouveau fichier est crée, il faut l'ajouter.

- `git add <fichier>`

- > Ajouter tous les fichier d'un répertoire : (la + grande partie du temps)

- `git add .`  point

- > Enregistrer les modifications en mémoire : (commit)

- `git commit -m 'message'`
 ↑ ajout d'un message
 ↑ message entre ' ' ou " "

- ↳ Le message doit décrire les modifications qui viennent d'être faites.

⚠ Le commit ne prend en compte seulement les fichiers qui ont déjà été add! Pratique courante:

1. `git add .`
2. `git commit -m "message"`

- Voir les logs des modifications apportées au repo (voir qui a commit quoi):
 - `git log`

- > Pull de Github vers le local : `git pull origin main`

> Retourner à une ancienne version du repo (ancien commit).

- `git checkout <n° du commit>`

↑ se trouve dans les log

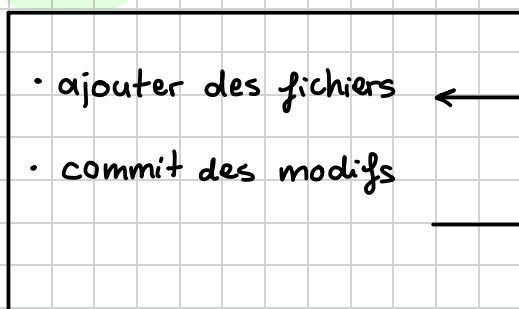
↳ En faisant ça, on va sortir de la branche en cours puisque l'on retourne dans le passé

- Envoyer ses fichiers vers Github:

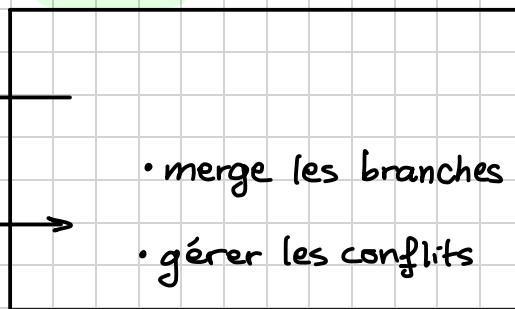
- 1
- > Si le repo ne vient pas d'un clonage (= crée pour la 1ère fois):
 - * Ajouter le lien (https) du projet Github à notre version locale Git
 - `git remote add origin <lien Github.git>`
 - > Si le repo vient d'un clonage, le lien d'origine est déjà enregistré -
- 2
- > Push notre local vers Github:
 - `git push -u origin <branche (souvent main)>`

- RÉSUMÉ -

Local



Github



Pull

Push

- Les branches:

> Les branches s'obtiennent:

- En retournant vers un ancien commit (voir checkout)

- En les créant volontairement - `git branch <nom>`

Suppression: `git branch -D <nom>`

- Lorsqu'il y a conflit entre le local et le Github.
(à gérer manuellement)



- Fetch :

Fetch sert à récupérer les modifications des fichiers qui ont été modifiés après notre pull. Cela crée une nouvelle branche.

Pour push correctement :

- `git fetch origin`
- `git merge origin/main`

Pour voir les modifications entre deux branches :

- `git diff main origin/main`