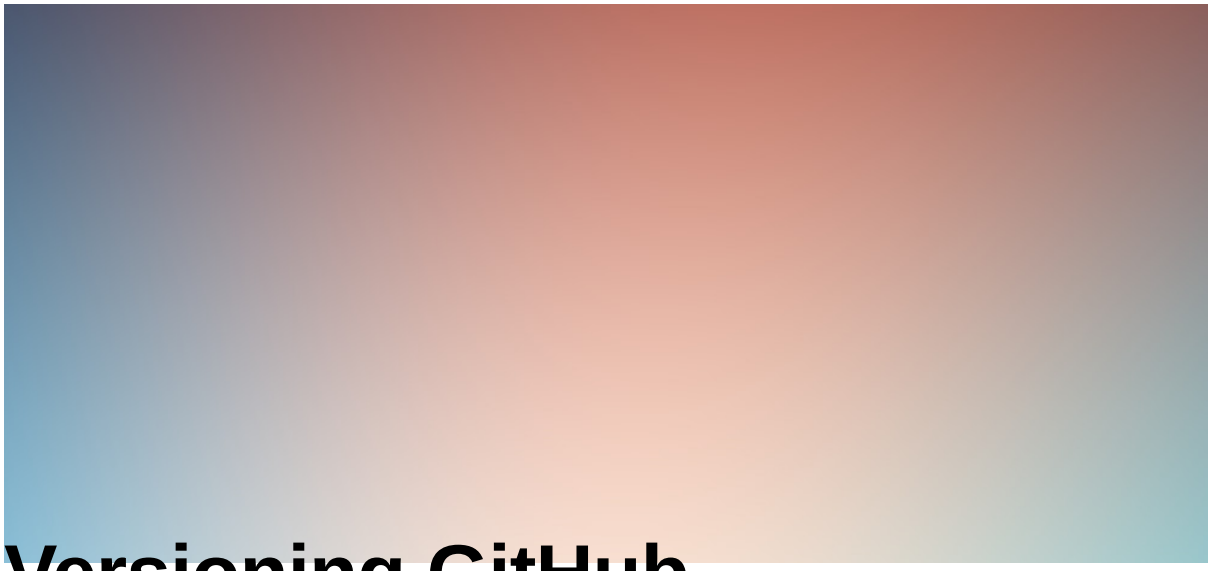


Versioning GitHub

Outils de versioning

SCM → intérêt ?

Travailler de manière collaborative

Ils permettent une maitrise totale du code source → Traçabilité, collaboration, peu de conflit, code fonctionnel et dev simultanée, création de workflow.

GIT → système décentralisé.

Installation et création de compte sur GitHub.

créer un repository sur GitHub.

fonctionnement = sauvegarde l'intégralité des versions a chaque fois en "instantané". → "Snapshots".

```
//1
git status
//2
git add * ou nom de fichier
//3
git commit -m 'Hellooo !'
//4
git log
//5
git push -u origin main
//6
```

1) Aucun commit effectuée, et suivis de fichier dans la repo Git. le faire a la racine du projet.

2) Mettre les fichiers en commit pour que ceux-là soit suivi par Git.

3) -m pour un message. demande de mail et de nom d'user pour mener a bien la requête. Essayer de commit le plus possible avec des phrases claire et pas des 'ok' ou 'bug résolu'. Bien expliquer. c'est un peu la description d'une "Snapshop".

4) Pour vérifier les logs de notre GitHub.

5) envoyer le code sur un remote et verifier si les fichier son tous synchronisé. ici synchronise la branche main local avec la branche main sur github.

Checkout → pour récupérer notre code en local sur GitHub.

Jour 2 :

mail : mnbdpro@gmail.com

mise en place de la connexion ssh avec l'aide de la Doc Git.

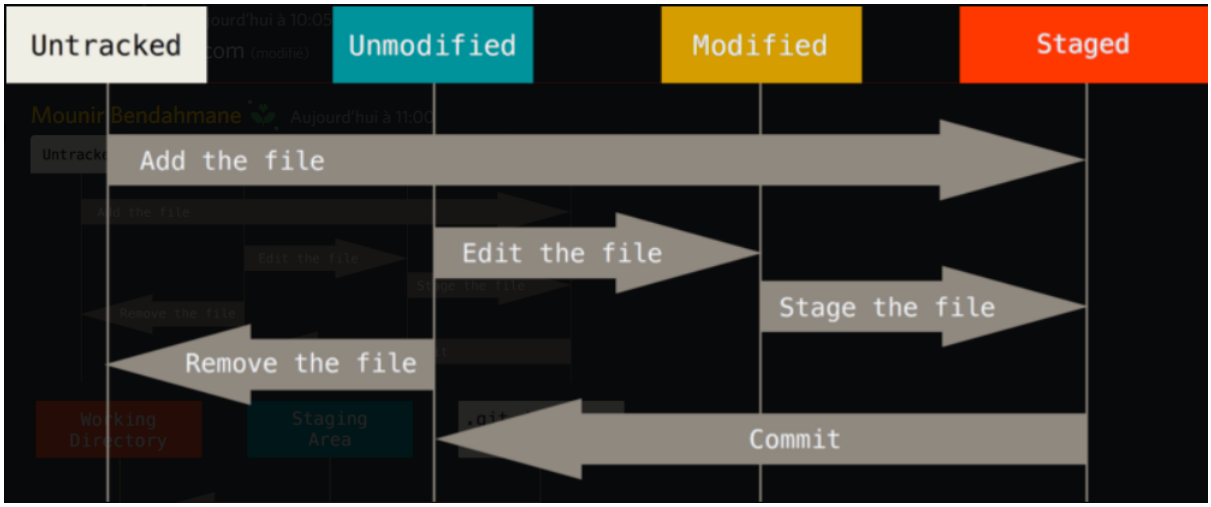
```
yoann@Nitro5deYo MINGW64 ~/documents/GitHub (main)
$ ssh -T git@github.com
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
Enter passphrase for key '/c/Users/yoann/.ssh/id_ed25519':
Hi YoYoDaa! You've successfully authenticated, but GitHub does not provide shell access.
```

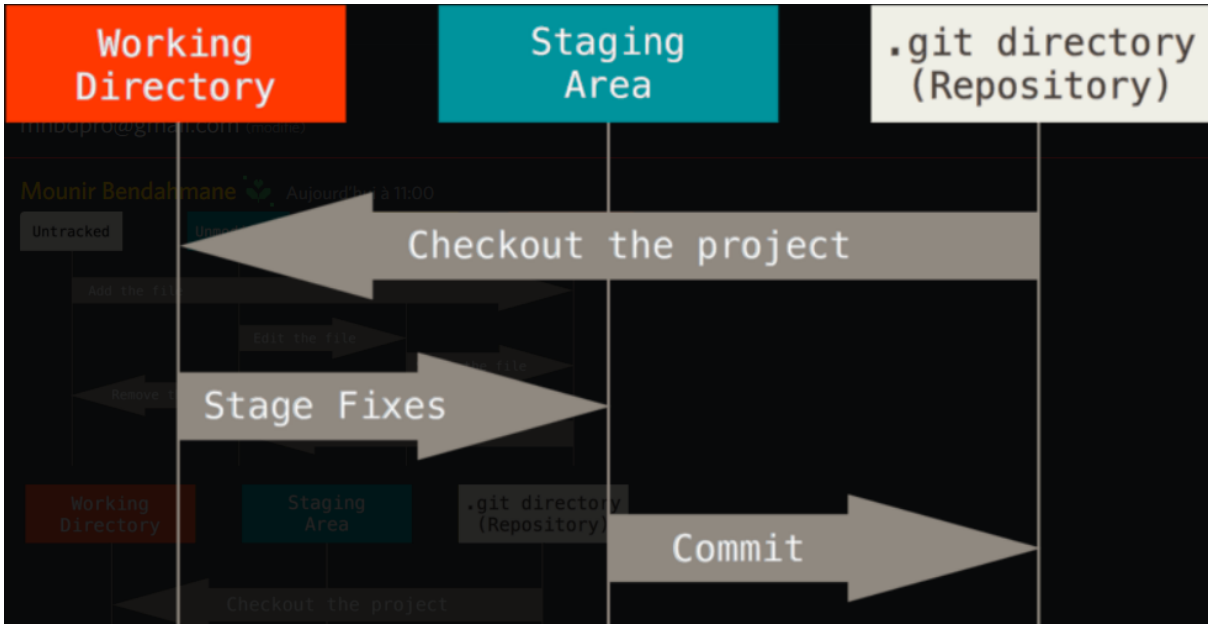
git commit -a permet de de add et de commit a la volée

-m pour ajouter un message

```
yoann@Nitro5deYo MINGW64 ~/Documents/GitHubIntro (main)
$ git add test.txt

yoann@Nitro5deYo MINGW64 ~/Documents/GitHubIntro (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   test.txt


yoann@Nitro5deYo MINGW64 ~/Documents/GitHubIntro (main)
$ git commit -m "addtest"
[main bed9882] addtest
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 test.txt

yoann@Nitro5deYo MINGW64 ~/Documents/GitHubIntro (main)
$ git push
Enter passphrase for key '/c/Users/yoann/.ssh/id_ed25519':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 317 bytes | 317.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:YoYoDaa/VersionningGit.git
   f2a600d..bed9882  main -> main

yoann@Nitro5deYo MINGW64 ~/Documents/GitHubIntro (main)
$ rm test.txt

yoann@Nitro5deYo MINGW64 ~/Documents/GitHubIntro (main)
$ git status
```

commit - pull - push





rm _rf .git/ pour supprimer le projet (pas *).

git rm —cached test.txt pour cacher un document

des repo sont déja préfait pour le gitignore, ils servent a ignorer certain fichier inutile a commit. il commence par un point ‘.’

```
yoann@Nitro5deYo MINGW64 ~/Documents/GitHubIntro (main)
$ git restore --staged test.txt

yoann@Nitro5deYo MINGW64 ~/Documents/GitHubIntro (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        test.txt

nothing added to commit but untracked files present (use "git add" to track)
```

Pour le retirer du suivis git.

autre méthode : git reset HEAD test.txt

créer une branche = créer une copi d'un projet a partie d'un commit donnée toute les modif de cette branche ne serons pas faut dans le main.

création de branche quasi instantanée. c'est un pointeur (renvoie vers l'adresse de qqch)

command git branch = créer une nouvelle branche

→ le résultat de la branch sans rien, montre les branch, et savoir sur quelle branch ont se situe.

comment fusionner les branches ? on verra…

pour fusionner 2 branch → git merge (branch)

pour supprimer une branch → git branch -d (nomBranch)

commenter toute les commandes dans le projets

git diff → voir si il y a des modifs sur un projets.

pour push → git add . ou git add (fichier) → git commit -m "(message)" → git push

projet pour les taches