* git init tp-git
* cd C:/Users/<<username>>/tp-git
* ls -a
* On branch master

No commits yet

nothing to commit (create/copy files and use “git add” to track)

No commits yet

Untracked files:

(use "git add <file>..." to include in what will be committed)

fichier1.md

fichier2.md

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

* git add fichier1.md
* On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: fichier1.md

Untracked files:

(use "git add <file>..." to include in what will be committed)

fichier2.md

* git commit -m “message”
* On branch master

Untracked files:

(use "git add <file>..." to include in what will be committed)

fichier2.md

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

* On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: fichier1.md

Untracked files:

(use "git add <file>..." to include in what will be committed)

fichier2.md

no changes added to commit (use "git add" and/or "git commit -a")

* git add fichier1.md
* git add fichier2.md
* git commit -m "Ajout fichier2.md et modification du fichier1.md"
* id 1 : 142890c6715427b59f246c9188709ee88ac764eb

id 2: e773c341148e96e3ae70236edf923f40760b1a67

* Cela permet de savoir s’il a déjà était commit ou non
* git branch fonctionnalite1
* git branch
* $ git switch fonctionnalite1
* L’étoile nous dit dans quelle branche nous sommes
* on peut utiliser un paramètre qui montre ce qu’il y a été changé
* git commit -m "Fonctionnalité 1 - première phase"
* git commit -m "Fonctionnalité 1 - terminé"
* les fichiers sont différents car on a créé / modifier des fichiers dans une branche différente
* git branch fonctionnalite2
* Car on aurait du faire à la suite de la branche fonctionnalite1
* git commit -m "Ajout fichier5.md"
* les point représentent les commits
* sur le graphe on peut savoir où on est grâce à un rond vide
* $ git merge fonctionnalite2 master
* git branch fonctionnalite3
* git commit -m “Modification fichier 1 pour fonctionnalite3”
* On peut cliquer sur le commit pour voir les fichiers modifiés et cliquer sur les fichiers modifiés pour voir ce qu’il y a été changé
* Git place que juste la branche actuelle sur la branche main
* La commande à fusionner plusieurs branches sur le main
* Il y a une erreur car les versions ont des différences pour le fichie1
* git branch -d

* git clone “liens du répertoire à cloner”
* Un fork est une copie indépendante du répertoire et le clone est une copie qui continue de synchroniser avec le répertoire
* On crée un fork pour repartir de 0 en changeant de répertoire et un clone pour continuer à modifier le répertoire actuelle
* git remote -v
* On ne voit qu’une branche
* On peut la supprimer
* Une nouvelle branche a été créer
* il a ajouter les modifications