

# Introduction à Git pour le Mini-Projet

## 1. Git

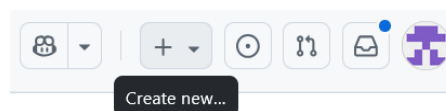
**Git** est un logiciel de gestion de versions. Il permet de travailler en groupe sur des fichiers qui sont mis en communs et actualisés au fur et à mesure.

Commencer par télécharger ce logiciel en allant ici : <https://git-scm.com/downloads> et en sélectionnant votre système d'exploitation.

## 2. Github

**Github** est un service web permettant de stocker des répertoires de fichiers sur des serveurs distants, et de synchroniser l'accès à ces fichiers à plusieurs en utilisant **Git**.

1. Créez-vous un compte sur <https://github.com> (si vous n'en avez pas). C'est un compte pour vos études/peut-être ensuite votre travail, utilisez donc vos nom et prénom dans votre pseudonyme.
2. Ensuite créer un nouveau répertoire :



puis "New repository".

3. Dans les informations demandées, vous n'avez besoin que d'ajouter le nom. Utilisez la forme suivante : **ProjetDB\_Nom1\_Nom2** avec Nom1 et Nom2 les noms des deux membres du binome (et avec un 3ème si c'est le cas).
4. Invitez ensuite en collaborateur le 2ème membre de votre binome, et **votre enseignant de TP** :




**Add collaborators to this repository**

Search for people using their GitHub username or email address.

Invite collaborators

5. Initialiser le dépôt avec un fichier (le sujet du projet, ou bien des fichiers demandés si vous avez déjà commencé) :

### Quick setup — if you've done this kind of thing before

 Set up in Desktop

or

HTTPS

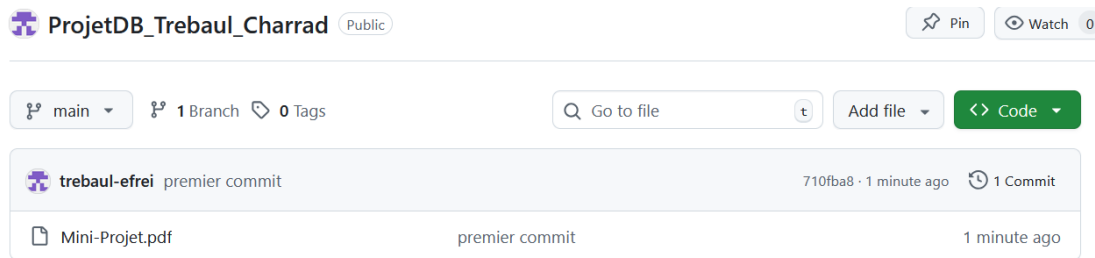
SSH

[https://github.com/trebaul-efrei/ProjetDB\\_Trebaul\\_Charrad.git](https://github.com/trebaul-efrei/ProjetDB_Trebaul_Charrad.git)

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

en cliquant sur "Uploading an existing file".

6. A cette étape, vous avez configuré votre **répertoire distant**, vous devez avoir un affichage du type :

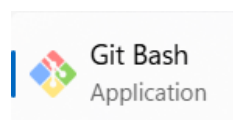


### 3. Copie locale

Le répertoire que vous avez créé est stocké sur un serveur distant, chacun des membres de votre groupe doit en avoir une copie sur sa machine en local. Pour cela vous pouvez utiliser **un des deux outils** que vous avez téléchargé avec Git : Git Bash (terminal) ou Git GUI (interface graphique). **Git Bash** est recommandé car il aide à comprendre les différentes étapes. Dans la suite, utiliser donc soit l'option 1 (recommandée), soit l'option 2 :

#### Option 1 : utiliser Git Bash (recommandée)

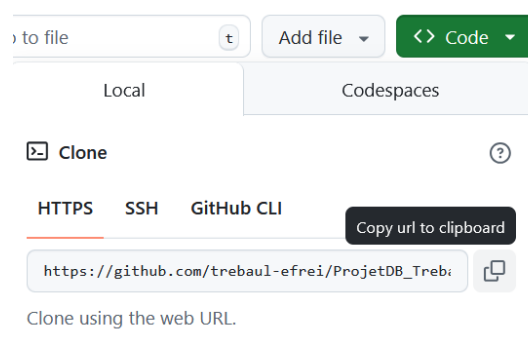
1. Ouvrir un terminal git avec **Git Bash** :



2. Se déplacer dans le répertoire dans lequel vous voulez stocker le répertoire du projet avec la commande `cd` suivie du chemin du répertoire, par exemple :

```
cd "C:\Users\Lena TREBAUL\OneDrive - Efrei\Documents\Cours\BDD\TI503N-2026"
```

3. Sur Github, copier l'URL du répertoire distant en cliquant sur "Code", elle doit s'afficher en-dessous :



4. Créer un clone du répertoire distant avec la commande `git clone` suivie de l'URL, par exemple :

```
Lena TREBAUL@DESKTOP-4Q1JJNC MINGW64 ~/OneDrive - Efrei/Documents/Cours/BDD/TI503N-2026
$ git clone https://github.com/trebaul-efrei/ProjetDB_Trebaul_Charrad.git
Cloning into 'ProjetDB_Trebaul_Charrad'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), 51.70 KiB | 6.46 MiB/s, done.
```

5. Vérifier que le répertoire a été créé avec la commande `ls` : vous devez voir le nom du répertoire dans la liste.
6. En vous déplaçant dans le nouveau répertoire avec `cd` , vous devez retrouver le fichier ajouté dans la partie précédente en utilisant `ls` :

```
Lena TREBAUL@DESKTOP-4Q1JJNC MINGW64 ~/OneDrive - Efrei/Documents/Cours/BDD/TI503N-2026
$ cd ProjetDB_Trebaul_Charrad/

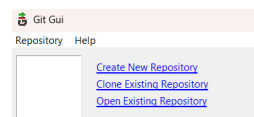
Lena TREBAUL@DESKTOP-4Q1JJNC MINGW64 ~/OneDrive - Efrei/Documents/Cours/BDD/TI503N-2026/ProjetDB_Trebaul_Charrad (main)
$ ls
Mini-Projet.pdf
```

## Option 2 : utiliser Git GUI

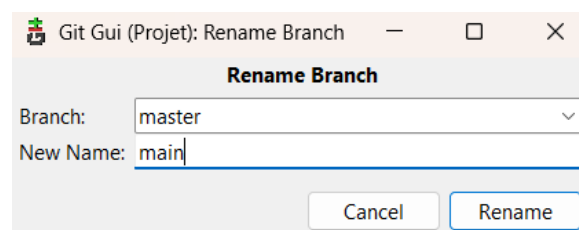
1. Lancer **Git GUI** :



2. Cliquer sur “Clone existing repository” :



3. Indiquer l'URL de votre répertoire distant, et celle du répertoire local (un répertoire sur votre ordinateur qui ne doit pas encore exister), et cliquer sur “Clone”.
4. Pour être en adéquation avec Github, il faut utiliser le même nom de branche que celui que vous aviez en créant le répertoire qui est `main` . Renommer votre branche si ce n'est pas le cas, en cliquant sur “Branch > Rename” :



## 4. Git add, commit et push

Une fois que vous avez commencé le mini-projet, quand vous voulez partager votre travail avec votre groupe, il faut :

### Option 1 : avec Git Bash (recommandée)

1. Ajouter les fichiers qui ont été modifiés, dont vous voulez partager la modification avec la commande `git add` :

```
$ git add prompt_conception.docx
```

2. Enregistrer les modifications apportées à ces fichiers, avec `git commit` suivi d'un message qui explique ce que vous avez modifié entre guillemets, par exemple :

```
$ git commit -m "ajout du prompt pour la partie conception"
[main b7eabfe] ajout du prompt pour la partie conception
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 prompt_conception.docx
```

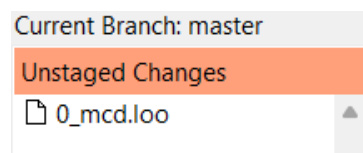
3. Envoyer les modifications sur le répertoire distant avec `git push` :

```
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 12.27 KiB | 12.27 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/trebaul-efrei/ProjetDB_Trebaul_Charrad.git
710fba8..b7eabfe main -> main
```

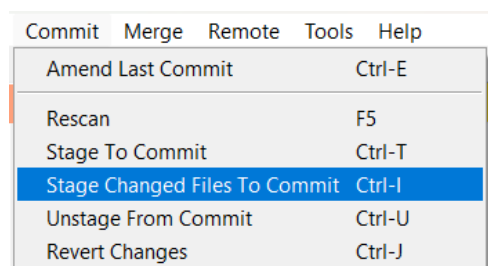
Elles doivent apparaître sur votre répertoire Github.

## Option 2 : avec Git GUI

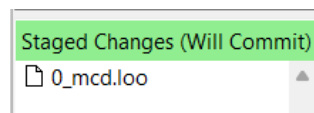
1. Si vous avez des modifications en local qui n'apparaissent pas sur le répertoire distant, elles sont détectées dans la partie "Unstaged changes" de la fenêtre :



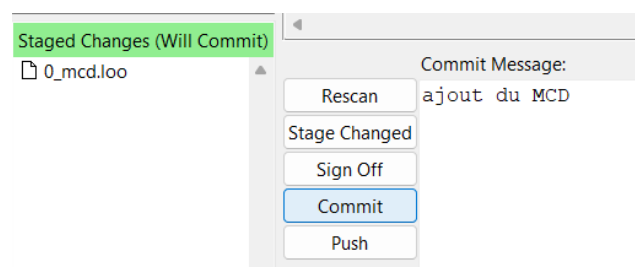
2. Cliquer sur Commit en haut de la fenetre puis sur "Stage changed files to commit" :



Les fichiers ajoutés apparaissent alors ici :



3. Ajouter un message pour indiquer ce qui a été modifié, puis cliquer sur "Commit" :



4. Envoyer les modifications sur le répertoire Github en cliquant sur "Push" (*ne pas changer le nom des branches*). Vérifier qu'elles y apparaissent bien.

## 5. Git pull

Dès que vous reprenez votre travail, vérifiez qu'un autre membre de votre groupe n'a pas apporté de changement au répertoire.

### Option 1 : avec Git Bash (recommandée)

Pour cela, faites un `git pull` :

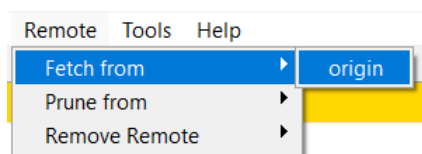
```
$ git pull
Already up to date.
```

“Already up to date” indique qu'il n'y a pas eu de changements. Si c'est le cas, les fichiers sont récupérés sur votre machine locale depuis le répertoire Github, comme ici :

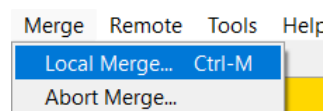
```
$ git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 3.73 KiB | 212.00 KiB/s, done.
From https://github.com/trebaul-efrei/ProjetDB_Trebaul_Charrad
   b7eabfe..4d72fae  main      -> origin/main
  * [new branch]    master    -> origin/master
Updating b7eabfe..4d72fae
Fast-forward
 0_mcd.loo | Bin 0 -> 82852 bytes
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 0_mcd.loo
```

### Option 2 : avec Git GUI

1. Commencer par chercher les modifications avec “fetch from” dans l'onglet “Remote” :



2. Puis fusionner les modifications avec “Merge > Local merge” :



### Bonnes pratiques

- Se diviser les tâches entre membres du groupes pour ne pas travailler sur les mêmes fichiers en même temps,
- Dès que vous reprenez le travail dans votre dossier, chercher les modifications sur le répertoire Github avec 'git pull',
- Quand vous avez terminé de travailler sur une partie, faites un commit ('git add' puis 'git commit') avec un message indiquant ce que vous avez terminé,
- A la fin d'une session de travail (ou avant), faites un 'git push',
- Indiquer aux membres de votre groupe que vous avez apporté une modification pour qu'ils fassent un 'git pull' sur leur répertoire local.