

TP Git : Initiation au Contrôle de Version

Objectifs du TP :

- Comprendre l'intérêt de Git dans la gestion de projet.
 - Apprendre les commandes Git de base.
 - Manipuler un dépôt Git local.
 - Interagir avec un dépôt distant (GitHub).
-

Matériel requis :

- Ordinateur avec Git installé (<https://git-scm.com/>).
 - Un compte GitHub (<https://github.com/>).
 - Un éditeur de code (ex : NotPad++, VS Code).
-

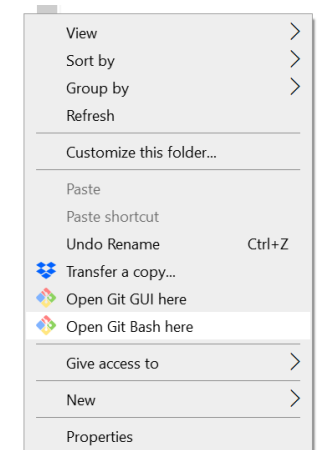
Status des fichiers :

| État du fichier | Message dans git status | Signification |
|------------------------|--|--|
| Nouveau fichier | Untracked file | Git ne le suit pas encore |
| Fichier modifié | modifié (unstaged) | Tu as changé un fichier, mais pas encore add |
| Fichier prêt à valider | Modifications à valider (staged) | Il sera inclus au prochain commit |
| Rien à valider | Rien à valider, la copie de travail est propre | Tout est à jour |

1. Créer un répertoire de travail local et le nommer « TP GIT »

2. Initialiser un dépôt Git local

- Ouvrir le terminal Git depuis le dossier local créé avec clique droit – Open Git Bash here here



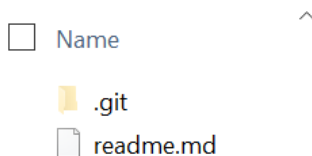
git init

```
salah bentalba@DESKTOP-P8K7TIM MINGW64 ~/Desktop/git tp/Projet Git
$ git init
Initialized empty Git repository in C:/Users/salah bentalba/Desktop/git tp/Proje
t Git/.git/
```

Explication : Initialise un dépôt Git vide dans le dossier courant. Cela crée un sous-dossier .git contenant tout l'historique des versions.

3. Créer un premier fichier et le suivre avec Git

- Créer un fichier de text avec un bout de code quelconque et le nommer : readme.md



git status

```
salah bentalba@DESKTOP-P8K7TIM MINGW64 ~/Desktop/git tp/Projet Git (main)
$ git status
On branch main

No commits yet

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

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

git status affiche l'état des fichiers : non suivis, modifiés, ou en attente de validation.

git add readme.md

```
salah bentalba@DESKTOP-P8K7TIM MINGW64 ~/Desktop/git tp/Projet Git (main)
$ git add readme.md
```

Git status

```
salah bentalba@DESKTOP-P8K7TIM MINGW64 ~/Desktop/git tp/Projet Git (main)
$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   readme.md
```

Explication :

- git add ajoute un fichier dans la zone de staging (prêt à être commité).

git commit -m "Ajout du fichier readme"

```
salah bentalba@DESKTOP-P8K7TIM MINGW64 ~/Desktop/git tp/Projet Git (main)
$ git commit -m "Ajout du fichier readme"
[main (root-commit) 11a07de] Ajout du fichier readme
 1 file changed, 1 insertion(+)
 create mode 100644 readme.md
```

Explication :

- git commit valide les modifications et les enregistre dans l'historique avec un message descriptif.

4. Afficher l'historique des versions

git log

```
salah bentalba@DESKTOP-P8K7TIM MINGW64 ~/Desktop/git tp/Projet Git (main)
$ git log
commit 11a07deec8712c03874ab72438316654e72eba32 (HEAD -> main)
Author: salahbentalba <salaheddine.bentalba@gmail.com>
Date:   Fri May 2 00:24:20 2025 +0100

    Ajout du fichier readme
```

Affiche la liste des commits effectués avec leurs identifiants, auteur, date et message.

5. Modifier un fichier et enregistrer une nouvelle version

- Modifier le fichier readme.md avec une phrase supplémentaire.

git status

```
salah bentalba@DESKTOP-P8K7TIM MINGW64 ~/Desktop/git tp/Projet Git (main)
$ git status
On branch main
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:   readme.md

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

```
git add readme.md
```

```
salah bentalba@DESKTOP-P8K7TIM MINGW64 ~/Desktop/git tp/Projet Git (main)
$ git add readme.md
```

```
git commit -m "Mise à jour du readme"
```

```
salah bentalba@DESKTOP-P8K7TIM MINGW64 ~/Desktop/git tp/Projet Git (main)
$ git commit -m "Mise à jour du readme"
[main 0bd0a1e] Mise à jour du readme
1 file changed, 2 insertions(+), 1 deletion(-)
```

6. Créer un compte GitHub et un dépôt distant

- Aller sur github.com > New Repository
- Nom : projet-git
- Ne pas cocher "Initialize with README"

7. Lier le dépôt local à GitHub

```
git remote add origin https://github.com/votre-compte/projet-git.git
```

```
salah bentalba@DESKTOP-P8K7TIM MINGW64 ~/Desktop/git tp/Projet Git (main)
$ git remote add origin https://github.com/Salahbentalba/Projet-GIT.git
```

```
git push -u origin main
```

```
salah bentalba@DESKTOP-P8K7TIM MINGW64 ~/Desktop/git tp/Projet Git (main)
$ git push -u origin main
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (6/6), 476 bytes | 476.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Salahbentalba/Projet-GIT.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Explication :

- git remote add déclare un lien avec un dépôt distant.
- git push envoie les commits vers GitHub.

9. TP à rendre :

- Capture de l'écran du terminal avec git log
- Lien vers le répô GitHub contenant un readme.md mis à jour

Conclusion

Ce TP vous a permis de prendre en main Git pas à pas, de comprendre l'intérêt du versionnage de code et d'intégrer GitHub comme outil collaboratif. C'est la base essentielle de toute pratique DevOps moderne.