

TP Git & GitHub – Niveau Débutant

Objectifs pédagogiques

- Comprendre la différence entre Git (local) et GitHub (distant)
- Créer et synchroniser un dépôt GitHub
- Utiliser les commandes Git de base dans un workflow simple

Astuce : Gardez toujours votre terminal ouvert et testez vos commandes étape par étape.

Contexte

Vous êtes développeur débutant. Vous devez créer un projet local, le sauvegarder sur GitHub et synchroniser vos modifications.

1 Création du dépôt GitHub

Sur GitHub, créez un nouveau repository nommé `exo-git-github` sans initialiser de README.

Explication : Cette étape prépare un espace distant où votre projet sera stocké et partagé avec d'autres collaborateurs.

2 Initialisation locale

Dans votre terminal, exécutez :

```
mkdir exo-git-github  
cd exo-git-github  
git init
```

Explication : La commande `git init` crée un dépôt Git local. Git pourra ainsi suivre les modifications de vos fichiers.

3 Premier commit

Créez un fichier README et effectuez votre premier commit :

```
echo "# Exercice GitHub" > README.md  
git add README.md  
git commit -m "Initialisation du projet"
```

Explication : `git add` sélectionne les fichiers à inclure dans le commit. `git commit` enregistre ces modifications dans l'historique local.

4 Lier le dépôt à GitHub

```
git branch -M main  
git remote add origin https://github.com/TON_PSEUDO/exo-git-github.git  
git push -u origin main
```

Explication :

- `git branch -M main` : renomme la branche principale en `main`.
- `git remote add origin ...` : relie le dépôt local au dépôt distant sur GitHub.
- `git push -u origin main` : envoie les commits sur GitHub et définit `origin` comme dépôt par défaut.

5 Synchronisation des modifications

Modifiez le fichier README.md directement sur GitHub, puis récupérez les changements en local :

```
git pull origin main
```

Explication : `git pull` récupère les dernières modifications du dépôt distant et les fusionne avec votre dépôt local pour rester synchronisé.

Conclusion

Ce TP vous a permis de découvrir le workflow Git + GitHub utilisé dans la majorité des projets professionnels. Vous savez maintenant :

- Créer un dépôt local et distant
- Ajouter et committer des fichiers
- Envoyer vos modifications sur GitHub
- Récupérer les modifications effectuées sur GitHub

Conseil : Habituez-vous à vérifier l'état de votre dépôt régulièrement avec `git status` et à consulter l'historique avec `git log --oneline`.