**Git et GitHub (RESUME)**

**Git**

Git est un système de contrôle de version qui permet de :

* **Suivre les modifications** dans le code source.
* **Travailler en équipe** sur le même projet sans conflits.
* **Revenir à une version précédente** si nécessaire.

**GitHub**

GitHub est une plateforme en ligne qui permet de :

* **Héberger des projets** utilisant Git.
* **Collaborer avec d'autres développeurs** grâce à des outils comme les pull requests et les issues.
* **Partager du code** publiquement ou en privé.

**Configuration de Git**

Pour configurer Git, utilisez les commandes suivantes :

git config --global user.name "Votre Nom"

git config --global user.email "votre.email@example.com"

Vérifiez la configuration avec :

git config --list

**Commandes de Base Git**

Pour initialiser un repository local :

1. Créez un dossier, allez dans ce dossier, initialisez Git et créez un fichier README.md :

mkdir mon-projet

cd mon-projet

git init

echo "Description du projet" > README.md

Pour ajouter et valider des fichiers :

1. Ajoutez des fichiers avec git add et validez avec git commit :

git add README.md

git commit -m "Premier commit"

Pour vérifier l'état et l'historique :

1. Utilisez git status et git log :

git status

git log

**Lier et Mettre à Jour le Repository**

Liez votre repository local à GitHub avec :

git remote add origin https://github.com/votre-nom-utilisateur/mon-projet.git

git push -u origin main

Pour mettre à jour le repository, utilisez :

git add .

git commit -m "Ajout d'un nouveau fichier"

git push origin main

**Cloner et Créer une Branche**

Pour cloner un repository existant :

git clone https://github.com/devBonkey/atelier-git-github-dclic-togo.git

Pour créer une nouvelle branche :

git branch nouvelle-fonctionnalite

git checkout nouvelle-fonctionnalite

**Faire un Pull Request**

Poussez votre branche sur GitHub :

git push origin nouvelle-fonctionnalite