

# Procédure d'utilisation de Git et GitHub

---

## 1. Introduction

Git est un système de gestion de versions distribué, permettant de suivre les modifications apportées à des fichiers et de coordonner le travail entre plusieurs développeurs. GitHub est une plateforme en ligne permettant d'héberger des dépôts Git, de collaborer et de partager du code facilement.

## 2. Cas d'utilisation simple

Voici les étapes typiques pour démarrer un projet avec Git et le publier sur GitHub :

1. Initialiser un dépôt local :

```
git init
```

2. Ajouter un dépôt distant :

```
git remote add origin https://github.com/your-username/nom-du-depot.git
```

3. Ajouter les fichiers :

```
git add .
```

4. Faire un premier commit :

```
git commit -m "Initial commit"
```

5. Pousser sur GitHub :

```
git branch -M main  
git push -u origin main
```

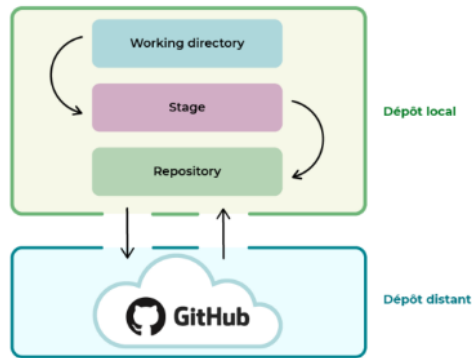
## 3. Glossaire des commandes Git

Commande	Description
git init	Initialise un dépôt Git local.
git clone URL	Clone un dépôt distant sur votre machine locale.
git status	Affiche l'état des fichiers (modifiés, non suivis, etc.).
git add <fichier>	Ajoute un fichier à l'index pour le prochain commit.
git commit -m "message"	Crée un snapshot des fichiers ajoutés.

git push	Envoie les modifications vers le dépôt distant.
git pull	Récupère et fusionne les modifications depuis le dépôt distant.
git branch	Liste les branches locales.
git branch <nom>	Crée une nouvelle branche.
git checkout <nom>	Bascule vers une autre branche.
git merge <nom>	Fusionne une branche dans la branche actuelle.
git log	Affiche l'historique des commits.
git stash	Met de côté les modifications non committées.
git stash apply	Restaure les modifications mises de côté.
git remote	Affiche les dépôts distants configurés.
git remote -v	Affiche les URL des dépôts distants.
git remote add <name> <url>	Ajoute un dépôt distant.
git remote remove <name>	Supprime un dépôt distant.
git fetch	Récupère les branches et les commits du dépôt distant sans fusionner.
git reset --hard HEAD^	Annule le dernier commit localement.
git commit --amend	Modifie le dernier commit.
git revert HEAD	Crée un commit inverse du dernier commit (sans le supprimer).

## Comparaison

Critère	HTTPS	SSH
Sécurité	Chiffrement en transit, mais les mots de passe peuvent être vulnérables	Chiffrement en transit et authentification par clés, plus sécurisé
Authentification	Utilise un nom d'utilisateur et un mot de passe/PAT	Utilise des paires de clés publiques et privées
Configuration	Simple à configurer initialement	Nécessite une configuration initiale plus complexe
Utilisation	Nécessite une saisie fréquente des identifiants	Aucune saisie nécessaire après la configuration initiale
Idéal pour	Utilisateurs occasionnels ou débutants	Utilisateurs réguliers et développeurs



#### Configuration et initialisation

**\$ cd Documents/PremierProjet**  
Pour vous positionner dans le dossier PremierProjet

**\$ git init**  
Pour initialiser un nouveau dépôt Git

**\$ git clone URL\_DU\_REPO**  
Pour cloner un dépôt existant à partir de l'URL fournie

#### Travailler avec des repos distant

**\$ git push**  
Pour envoyer la nouvelle version sur le dépôt distant

**\$ git pull**  
Pour récupérer les dernières modifications du dépôt distant

#### Gestion des fichiers et des commits

**\$ git status**  
Pour montrer l'état des fichiers

**\$ git add fichier.html**  
Pour ajouter des fichiers à l'index pour le prochain commit

**\$ git commit -m "Message de commit"**  
Pour créer un nouveau commit avec les fichiers ajoutés à l'index

#### Gestion des branches

**\$ git branch**  
Pour lister toutes les branches

**\$ git branch NOM\_DE\_LA\_BRANCHE**  
Pour créer une nouvelle branche

**\$ git checkout NOM\_DE\_LA\_BRANCHE**  
Pour changer de branche

**\$ git merge NOM\_DE\_LA\_BRANCHE**  
Pour fusionner la branche spécifiée dans la branche actuelle

#### Historique et inspection

**\$ git log**  
Pour voir l'historique des commits

**\$ git stash**  
Pour enregistrer temporairement des modifications non indexées

**\$ git stash apply**  
Pour appliquer les modifications enregistrées avec stash