

# Fiche procédure Git

## Table des matières

- Installation de Git
- Configuration initiale
- Utilisation de Git (commandes de base)
- Bonnes pratiques
- Exemple de workflow Git

## Installation de Git

Sous Windows :

- Télécharge le programme d'installation sur [git-scm.com](https://git-scm.com) et lance-le.
- Suis les instructions (les options par défaut conviennent généralement).

Sous macOS :

- Utilise Homebrew :
- `bash`

```
brew install git
```

- 

- Ou télécharge l'installateur sur [git-scm.com](https://git-scm.com).

Sous Linux :

- Utilise le gestionnaire de paquets :
- `bash`

```
sudo apt update  
sudo apt install git
```

- 

## Configuration initiale

Configure ton nom d'utilisateur et ton email (indispensable pour les commits) :

```
bash
```

```
git config --global user.name "Ton Nom"  
git config --global user.email "ton.email@example.com"
```

Vérifie la configuration :

```
bash
git config --list
```

## Utilisation de Git (commandes de base)

### 1. Initialiser un dépôt

```
bash
git init
```

### 2. Cloner un dépôt existant

```
bash
git clone https://github.com/utilisateur/projet.git
```

### 3. Vérifier l'état du dépôt

```
bash
git status
```

### 4. Ajouter des fichiers à l'index

```
bash
git add fichier.txt
# ou pour tout ajouter :
git add .
```

### 5. Enregistrer un commit

```
bash
git commit -m "Message explicite"
```

### 6. Voir l'historique des commits

```
bash
git log --oneline
```

## 7. Envoyer les modifications vers le dépôt distant

```
bash
git push origin main
```

## 8. Récupérer les modifications du dépôt distant

```
bash
git pull origin main
```

## Bonnes pratiques

- Commits fréquents et explicites : un message clair pour chaque modification.
- Utilise des branches pour les nouvelles fonctionnalités ou corrections.
- Tire (« pull ») avant de pousser (« push ») pour éviter les conflits.
- Ne commit jamais de mots de passe ou de données sensibles.
- Lis les messages d'erreur : ils sont souvent très explicites.

## Exemple de workflow Git

```
bash
# 1. Cloner le dépôt (si besoin)
git clone https://github.com/utilisateur/projet.git

# 2. Créer une nouvelle branche
git checkout -b nouvelle-fonctionnalite

# 3. Ajouter/modifier des fichiers
# ...

# 4. Ajouter les fichiers modifiés
git add .

# 5. Committer les modifications
git commit -m "Ajout de la nouvelle fonctionnalité"

# 6. Pousser la branche sur le dépôt distant
```

```
git push origin nouvelle-fonctionnalite
```

*# 7. Créer une Pull Request sur GitHub/GitLab*