

# Guide Git Projet THERMO (Windows + VS Code)

## Introduction

Ce guide présente les étapes essentielles pour utiliser Git sur Windows avec VS Code. Il est structuré pour distinguer ce qui doit être fait **une seule fois** et ce qui doit être fait **à chaque projet ou session de travail**.

## 1 À faire la première fois

### 1.1 Installer Git et VS Code

- Télécharger **Git** : <https://git-scm.com/downloads>
- Télécharger **VS Code** : <https://code.visualstudio.com/download>
- Optionnel : installer l'extension **GitLens** dans VS Code pour visualiser l'historique des commits et des branches.

### 1.2 Configurer Git

Ouvrir PowerShell ou Git Bash :

```
git config --global user.name "Votre Nom"  
git config --global user.email "votre.email@example.com"  
git config --list
```

### 1.3 Se connecter à GitHub

**Option A – HTTPS :**

- GitHub demande maintenant un **token personnel** à utiliser comme mot de passe :  
<https://github.com/settings/tokens>

**Option B – SSH (recommandé) :**

```
ssh-keygen -t ed25519 -C "votre.email@example.com"  
# Copier le contenu de C:\Users\VotreNom\.ssh\id_ed25519.pub  
# Ajouter sur GitHub > Paramètres > SSH et GPG Keys > Nouvelle cl SSH  
ssh -T git@github.com
```

### 1.4 Tests pour vérifier la configuration

```
git --version # Vérifie que Git est installé  
git config --list # Vérifie la configuration utilisateur  
ssh -T git@github.com # Vérifie la connexion SSH GitHub
```

## 2 À faire à chaque projet ou session

### 2.1 Cloner un dépôt

```
git clone https://github.com/yanismedouni/PI4-THERMO.git
```

```
cd PI4-THERMO  
code .
```

### 2.2 Vérifier l'état des fichiers

```
git status
```

### 2.3 Ajouter et committer des changements

```
git add fichier.txt # Ajouter un fichier spécifique  
git add . # Ajouter tous les fichiers  
git commit -m "Message clair et descriptif"
```

### 2.4 Pousser les changements sur GitHub

```
git push origin main # Remplacer 'main' par la branche appropriée
```

### 2.5 Récupérer les changements de GitHub

```
git pull origin main
```

### 2.6 Travailler avec les branches

```
git checkout -b nom_de_branche # Crer et basculer sur une nouvelle branche  
git checkout main # Revenir sur la branche principale  
git checkout nom_de_branche # Passer une autre branche  
git push -u origin nom_de_branche # Pousser la branche sur GitHub  
git merge nom_de_branche # Fusionner une branche dans la branche courante
```

### 2.7 Résoudre les conflits

- Git affichera les conflits comme suit :

```
<<<<< HEAD  
vos modifications  
=====  
modifications de l'autre  
>>>> nom_de_branche
```

- Choisir la version à conserver, supprimer les marqueurs, sauvegarder :

```
git add fichier.txt  
git commit -m "Rsolution du conflit"
```

## 2.8 Commandes Git utiles

Commande	Description
git status	Vérifier l'état des fichiers
git log	Historique des commits
git diff	Voir les changements
git branch	Lister les branches
git checkout branche	Changer de branche
git add .	Ajouter tous les fichiers
git commit -m "msg"	Committer
git push	Envoyer les commits sur GitHub
git pull	Récupérer les commits de GitHub
git merge branche	Fusionner une branche dans la branche courante

## 2.9 Bonnes pratiques pour débutants

- Toujours **pull avant push**.
- Faire des commits **petits et clairs**.
- Travailler **sur des branches** pour chaque nouvelle fonctionnalité ou correction.
- Utiliser l'extension **GitLens** dans VS Code pour visualiser l'historique et les différences.