

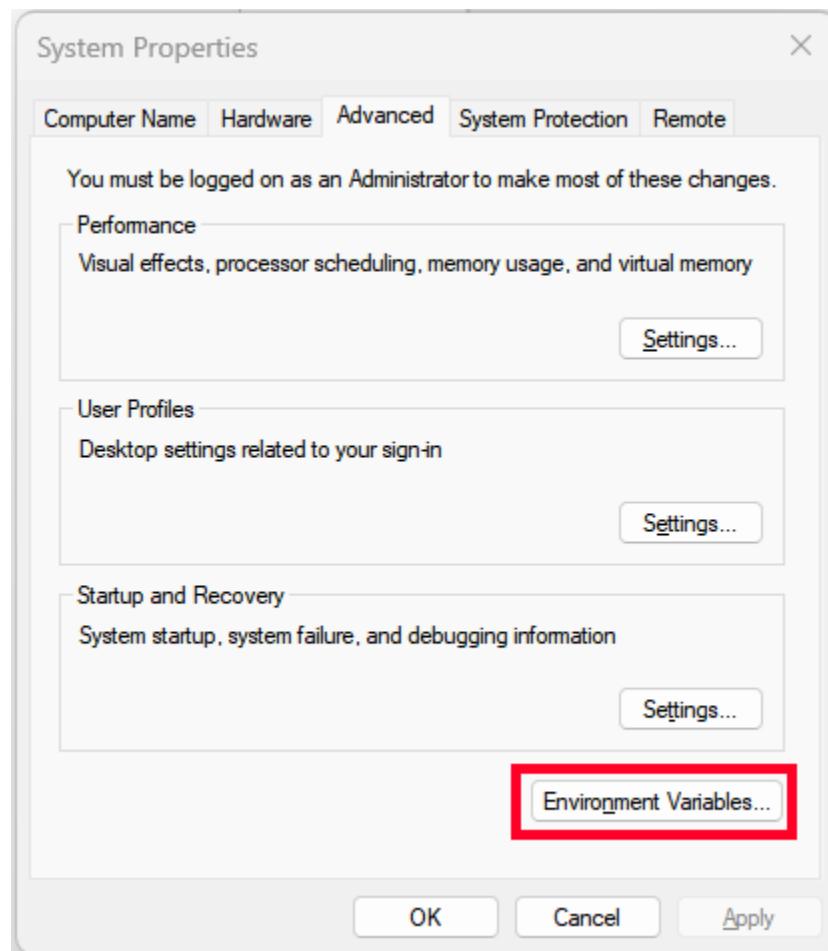
# TP1

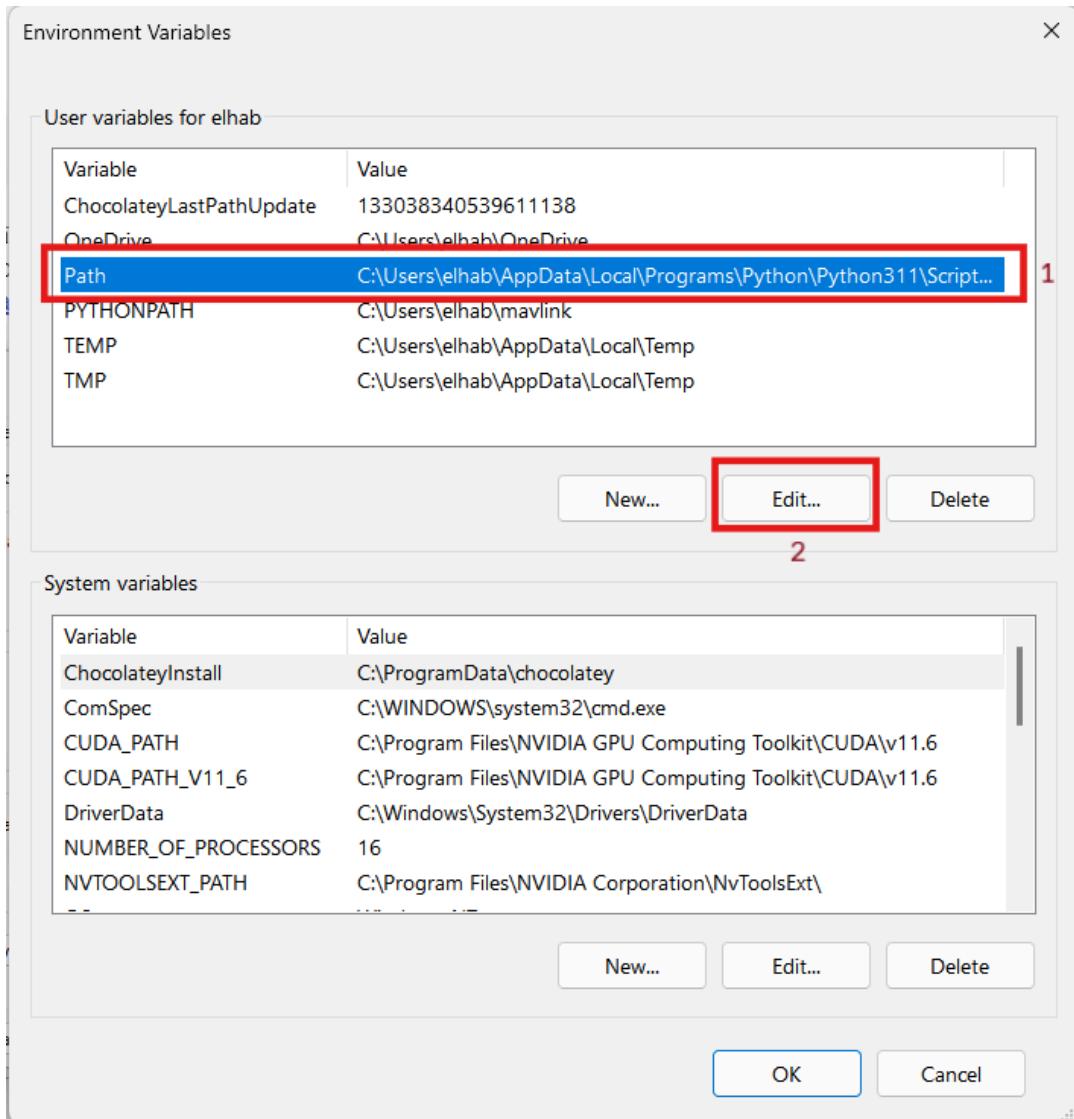
## Configuration d'environnement:

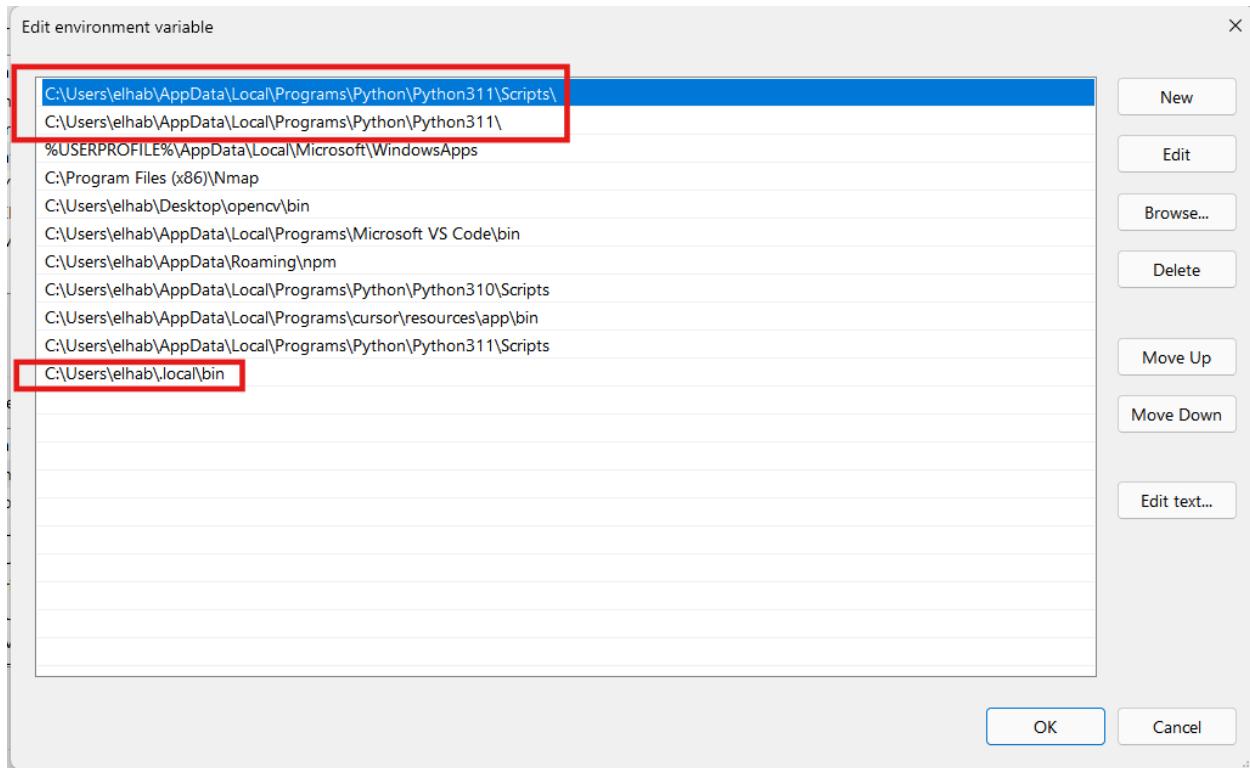
Install Python 3.11:

<https://www.python.org/downloads/release/python-3119/>

Configuration des env Path:







Ajouter les chemins suivants:

- C:\Users\\$USER\AppData\Local\Programs\Python\Python311\
- C:\Users\\$USER\AppData\Local\Programs\Python\Python311\Scripts\

Remplacer \$USER par votre nom d'utilisateur.

## Installation de Poetry:

Insaller poetry en utilisant les commandes :

```
py -m pip install --upgrade pip pipx  
pipx install poetry
```

Vérifiez si Poetry est installé :

```
poetry --version
```

Lancer la commande suivante pour configurer poetry à créer le venv sur notre répertoire :

```
poetry config virtualenvs.in-project true
```

## Installation de Git:

### Installation sur Windows:

L'application git est disponible pour télécharger sur le site web de Git

<https://git-scm.com/download/win>

### Installation sur Linux:

Installation en terminal avec commande apt

`sudo apt install git`

Ou aussi sur le site web de Git:

<https://git-scm.com/download/linux>

## Initialisation d'un dépôt Git

1. Créez un dossier nommé `tp1_git_intro`.
2. Ouvrez-le dans VS Code ou un terminal.
3. Exécutez les commandes suivantes :

`git init`

`git status`

4. Créez un fichier `README.md` avec une brève description du projet.
5. Ajoutez et validez les modifications :

`git add README.md`

`git commit -m "Premier commit : ajout du fichier README"`

1. Affichez le log :

`git log`

## Suivi des modifications

1. Créez un nouveau fichier `main.py` :

```
print("Bonjour DevOps & MLOps!")
```

2. Ajoutez-le et validez les modifications.
3. Modifiez le fichier pour afficher un message différent, puis validez à nouveau :

```
print("Bienvenue au TP Git.")
```

4. Comparez les versions :

```
git diff  
git log
```

## Branches et fusion

1. Créez une nouvelle branche :

```
git branch feature-readme  
git checkout feature-readme
```

2. Modifiez **README.md** pour ajouter une section “Auteur : Votre nom”.
3. Commitez les changements puis revenez sur la branche principale :

```
git checkout main
```

4. Fusionnez :

```
git merge feature-readme
```

5. Supprimez la branche :

```
git branch -d feature-readme
```

## Connexion à GitHub

1. Créez un dépôt vide privé sur GitHub appelé **tp1\_git\_intro**.
2. Liez votre dépôt local :

```
git remote add origin https://github.com/<votre-utilisateur>/tp1_git_intro.git  
git push -u origin main
```

3. Vérifiez sur GitHub que vos fichiers apparaissent bien.

## Résolution de conflit

1. Créez deux branches : `main` et `feature-message`.
2. Modifiez `main.py` dans les deux branches avec des lignes différentes et commitez.
3. Tentez une fusion :  
`git merge feature-message`
4. Résolvez le conflit manuellement et faites un commit.