

# Installation de Python, VS Code et Jupyter (Windows, macOS, Linux)

28 février 2025

## 1 Vérifier si Python est installé

Avant d'installer Python, il est utile de vérifier s'il est déjà présent sur votre système. Pour cela, ouvrez un terminal et exécutez la commande suivante selon votre système d'exploitation.

— **Windows** : Ouvrez PowerShell et tapez :

```
1     python --version
2     python3 --version
3
```

— **macOS / Linux** : Ouvrez un terminal et tapez :

```
1     python3 --version
2
```

Si Python est installé, la version actuelle sera affichée.

## 2 Installer Python

Si Python n'est pas installé ou si la version est trop ancienne, suivez les instructions ci-dessous pour l'installer.

### 2.1 Windows

1. Rendez-vous sur [Python.org](https://python.org) et téléchargez l'installateur.
2. Exécutez l'installateur et cochez **"Add Python to PATH"** avant d'installer.
3. Vérifiez l'installation avec :

```
1 python --version
2 python3 --version
3
```

### 2.2 macOS

Sur macOS, il est recommandé d'utiliser Homebrew pour installer Python :

```
1 /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/
   Homebrew/install/HEAD/install.sh)"
2 brew install python
3 python3 --version
```

### 2.3 Linux (Ubuntu/Debian)

Sous Linux, utilisez les commandes suivantes :

```
1 sudo apt update
2 sudo apt install python3 python3-pip python3-venv
3 python3 --version
4 pip3 --version
```

### 3 Installer Visual Studio Code

VS Code est un éditeur de code puissant qui facilite l'écriture et l'exécution de programmes Python.

- **Windows et macOS** : Téléchargez et installez VS Code.
- **Linux** : Installez-le en ligne de commande :

```
1 sudo apt update
2 sudo apt install code
3 # ou
4 sudo snap install --classic code
```

### 4 Configurer un environnement virtuel

Un environnement virtuel permet d'isoler les bibliothèques installées afin d'éviter les conflits entre projets. Ainsi, chaque projet peut disposer de ses propres dépendances sans impacter les autres.

Dans un terminal, exécutez :

```
1 # Remplacez "mon_env" par le nom de votre choix,
2 # typiquement "ICS".
3 # Les packages que vous installerez par la suite
4 # ne seront accessibles que dans cet environnement.
5     python3 -m venv mon_env
6
7 # macOS/Linux
8     source mon_env/bin/activate
9
10 # Windows (PowerShell)
11     mon_env\Scripts\activate
```

Lorsque l'environnement virtuel est activé, l'invite de commande affichera (**mon\_env**) au début de la ligne. Toutes les installations de paquets (via pip) ne concerneront que cet environnement spécifique.

Pour désactiver l'environnement virtuel, utilisez la commande :

```
1 deactivate
```

### 5 Installer Jupyter, NumPy et Matplotlib

Installez ces paquets uniquement dans l'environnement virtuel activé (vous devez voir **mon\_env** en début de ligne de commande) :

```
1 pip install jupyter numpy matplotlib
```

## 6 Lancer un Notebook Jupyter dans VS Code

1. Ouvrir VS Code et activer l'environnement virtuel en appuyant sur `Ctrl+Shift+P`,
2. Tapez `Python: Select Interpreter`, et choisissez `mon_env/bin/python`, ou `mon_env\Scripts\python.exe` sous Windows.
3. Créer un fichier avec l'extension `.ipynb`.
4. Exécuter une cellule Jupyter via l'extension VS Code.
5. Ou bien, lancer Jupyter manuellement avec :

```
1     jupyter notebook
2
```

Cela ouvrira Jupyter dans votre navigateur, vous permettant d'écrire et d'exécuter du code interactivement.

## 7 Tester NumPy et Matplotlib

Essayez ce code dans un notebook Jupyter ou un script Python :

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 x = np.linspace(0, 10, 100)
5 y = np.sin(x)
6
7 plt.plot(x, y)
8 plt.xlabel("x")
9 plt.ylabel("sin(x)")
10 plt.title("Exemple Matplotlib")
11 plt.show()
```

Si un graphique s'affiche, l'installation est réussie !