

Guide pour commencer à utiliser Python

Nous supposons que vous savez installer **python** et utiliser les commandes de base.

Si ce n'est pas le cas, voici un bref récapitulatif des éléments importants. Nous vous invitons à consulter les liens suivants en fonction de vos besoins.

Pour en apprendre plus sur le langage python :

<https://docs.python.org/fr/3/tutorial/>

Pour installer python :

<https://www.python.org/downloads/>

Python peut être utilisé à travers un simple éditeur de texte comme bloc note mais on utilise généralement des **interfaces de développement intégré (IDE)** qui présentent un grand nombre de fonctionnalités améliorant l'expérience utilisateur.

Nous vous conseillons d'utiliser l'une des IDE suivantes:

1. **Spyder** : très utilisé dans la communauté scientifique. <https://www.spyder-ide.org/>
2. **VScode** : le plus largement utilisé et le plus polyvalent, il permet de coder avec n'importe quel langage de programmation. <https://code.visualstudio.com/>
3. **Jupyter notebook** : le plus facile à utiliser et à lire, permet de mélanger des blocs de texte (markdown) et des blocs de script. <https://jupyter.org/>

Mentions spéciales si cela vous intéresse:

1. **Anaconda** : ce n'est pas une IDE mais une plateforme qui propose et installe un grand nombre d'IDE et qui utilise les commandes conda. Attention, le logiciel prend une grande quantité de stockage à l'installation. <https://www.anaconda.com/download>
2. **Antigravity** : VScode like, développé par Google qui propose des agents IA plus performants pour aider à coder. <https://antigravity.google/>

Il existe un grand nombre de vidéo, forums, etc. qui permettent de commencer à coder sur ces logiciels.

Environnement virtuel

Qu'est-ce qu'un environnement virtuel ?

Un **environnement virtuel** est un dossier isolé dans lequel vous installez une version spécifique de Python et un ensemble de bibliothèques (packages) spécifiques à un projet.

Imaginez que chaque projet Python est une petite pièce fermée. Dans cette pièce, vous mettez uniquement les outils (bibliothèques/packages) dont vous avez besoin pour ce projet précis. Ce qui se passe dans cette pièce n'affecte pas le reste de votre ordinateur.

Est-ce que c'est obligatoire ?

Techniquement, non. Vous pouvez installer toutes vos bibliothèques directement sur votre système (ce qu'on appelle l'installation "globale").

Pratiquement, oui. Dès que vous dépassez le stade de l'apprentissage des bases (variables, boucles) pour commencer à créer de vrais projets, cela devient indispensable. Dans le milieu professionnel, ne pas utiliser d'environnement virtuel est considéré comme une **très mauvaise pratique**.

Pourquoi on l'utilise ?

Éviter les conflits de versions (Le problème principal)

Imaginez deux projets :

- Projet A a besoin de la bibliothèque Pandas version 1.0.
- Projet B a besoin de Pandas version 2.0 (car il utilise de nouvelles fonctions).

Si vous installez tout au même endroit sur votre ordinateur, vous ne pouvez avoir qu'une seule version. En installant la version 2.0 pour le Projet B, vous risquez de "casser" le Projet A qui ne supporte pas les changements de la nouvelle version. L'environnement virtuel règle ce problème en créant des espaces séparés.

Garder votre système "propre"

Sur macOS et Linux, le système d'exploitation utilise lui-même Python pour fonctionner. Si vous installez, modifiez ou supprimez des bibliothèques globales de manière désordonnée, vous risquez de casser des outils système essentiels. L'environnement virtuel protège votre ordinateur. Même sur Windows, il est toujours préférable d'utiliser un environnement virtuel.

Faciliter la collaboration

Avec un l'environnement virtuel, vous pouvez générer un fichier (souvent nommé requirements.txt) qui liste exactement les versions des outils que vous utilisez. Quand vous donnez votre code à un collègue, il peut recréer exactement le même environnement que vous en une seule commande. Cela évite le célèbre : "Mais ça marchait sur mon ordi !".

Comment créer et activer un environnement virtuel ?

Nous allons créer un environnement virtuel nommé **mon_projet** mais vous pouvez lui donner le nom qui vous désirez.

1. Ouvrez votre terminal (ou "Anaconda Prompt" sur Windows)

Option A : Avec Conda (Recommandé si vous avez installé Anaconda/Miniconda)

2. Créer l'environnement nommé 'mon_projet'

commande : conda create --name mon_projet python=3.13

3. activer l'environnement

commande : conda activate mon_projet

Option B : Avec venv (Standard Python)

2. Créer l'environnement nommé 'mon_projet' *commande : python -m venv mon_projet*

3. L'activer

- Sur Windows :

commande : mon_projet\Scripts\activate

- Sur Mac/Linux :

commande : source mon_projet/bin/activate

Et voilà ! Vous avez un environnement virtuel python prêt pour votre projet.

Vous pouvez désormais installer des libraires pythons en faisant :

- *commande : pip install nom_librairie*

commande : `pip install openpyxl` pour les lire des fichiers excel
ou encore à l'aide de conda :

- *commande : `conda install openpyxl`*