

Formations Python & Machine Learning

Guide d'installation de l'environnement

ALIASE

Guide d'installation de l'environnement

- **Windows / MacOS : Aller à la page 3**
- **Linux : Aller à la page 34**

Windows / MacOS

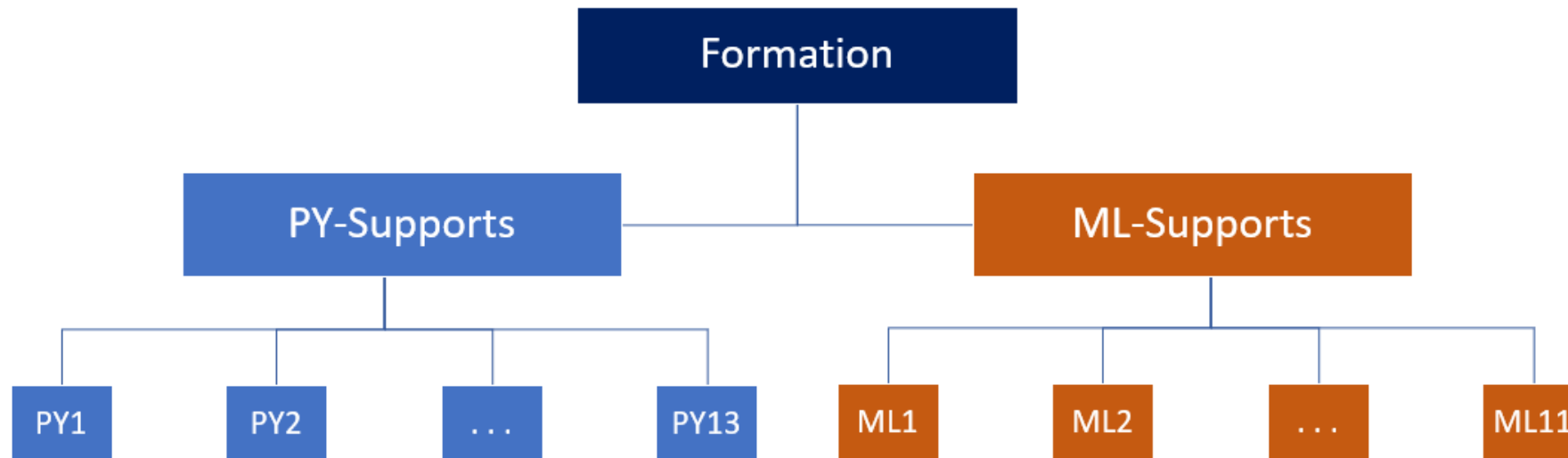
Guide d'installation de l'environnement

Contenu

- Installation de **Python**
- Installation de l'éditeur **Visual Studio Code**
- Installation de l'environnement **Anaconda** (Spyder, ...)
- Installation de la base de données **PostgreSQL**
- Installation des Frameworks **Flask** et **FastAPI**
- Installation de **TENSORFLOW** (**Uniquement en cas de formation IA / Machine Learning**)
- Installation de **Docker**
- Test de l'environnement

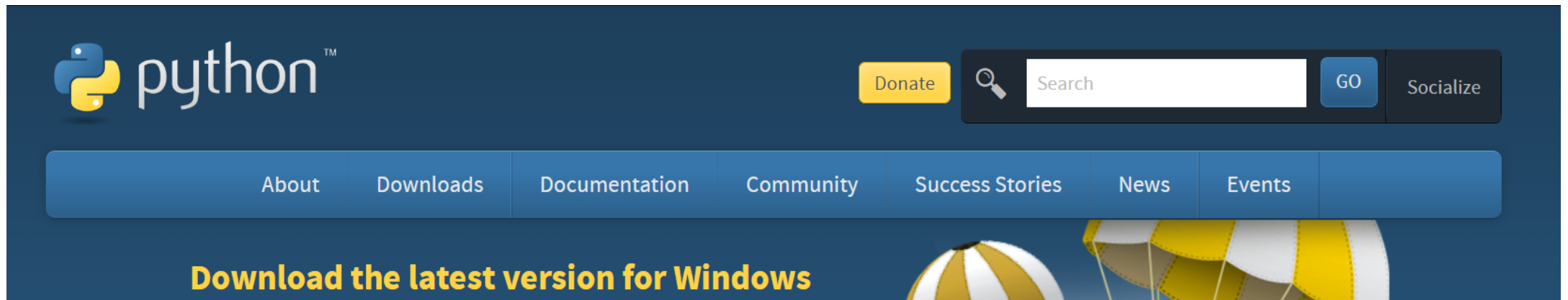
Organisation des dossiers sur votre disque

- . Créer un dossier **Formation** sur votre machine : C:\Users\user-name**Formation**
- . Créer 2 sous-dossiers sous le dossier Formation:
 - **PY-Supports** : dossier pour la partie *Python* (supports + corrigés des exercices)
Créer des sous-dossiers pour les chapitres : **PY1 ... PY5, PY7, PY10, PY13**
 - **ML-Supports** : dossier pour la partie *Machine Learning* (supports + corrigés des exercices)
Créer 11 sous-dossiers pour les chapitres : **ML1 ... ML11**



Python: Installation

1) Installer Python **version 3.12** (plus stable) partir de <https://www.python.org/downloads/>



- . Download
- . Run
- . **Ne pas oublier de cocher** "ADD TO PATH"

Editeurs

Installer Anaconda depuis : <https://www.anaconda.com/download>

Anaconda contient tous les outils et librairies utiles pour faire du Machine Learning :

- **Numpy,**
- **Matplotlib,**
- **Pandas,**
- **Sklearn,**
- **etc.**

Créer un environnement

Installation

1) Ouvrir le Terminal d'anaconda :

→ Barre de recherche en bas de votre écran

→ Chercher *Anaconda*



Anaconda Powershell Prompt (Anaco...

2) Créer un environnement avec la version de python 3.12:

```
(base) PS conda create --name myenv312 python=3.12
```

3) Activer l'environnement avec python 3.12:

```
(base) PS conda activate myenv312
```

4) Installer spyder :

```
(myenv312) PS conda install -c anaconda spyder
```

Installation

5) Installer la librairie numpy :

(myenv312) PS conda install numpy

6) Installer la librairie scikit-learn :

(myenv312) PS conda install scikit-learn

7) Lancer spyder à partir du terminal :

(myenv312) PS spyder

Installation

8) **Installer les librairies** : Commandes à saisir dans la **console de spyder**

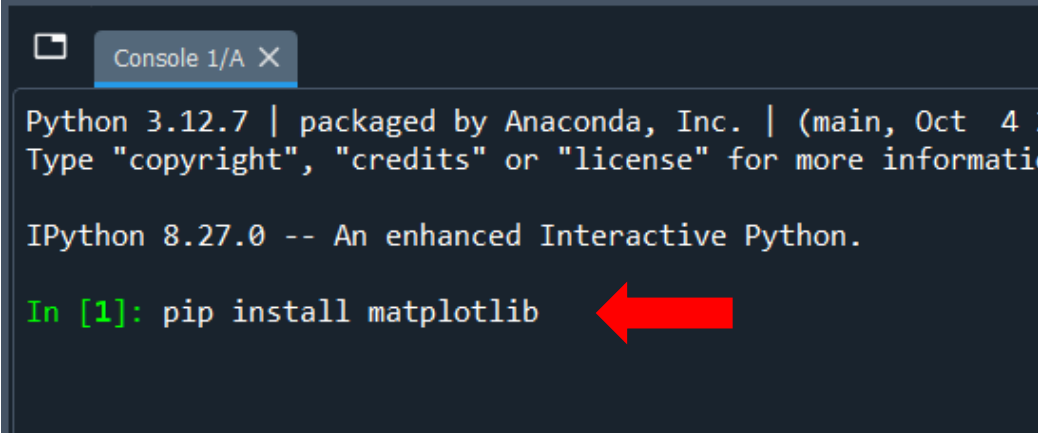
- Installer **matplotlib** : `pip install matplotlib`
- Installer **seaborn** : `pip install seaborn`
- Installer **pandas** :

`pip install pandas`

`pip install xlrd`

`pip install openpyxl`

`pip install xlswriter`



```
Console 1/A X
Python 3.12.7 | packaged by Anaconda, Inc. | (main, Oct 4 2023)
Type "copyright", "credits" or "license()" for more information.

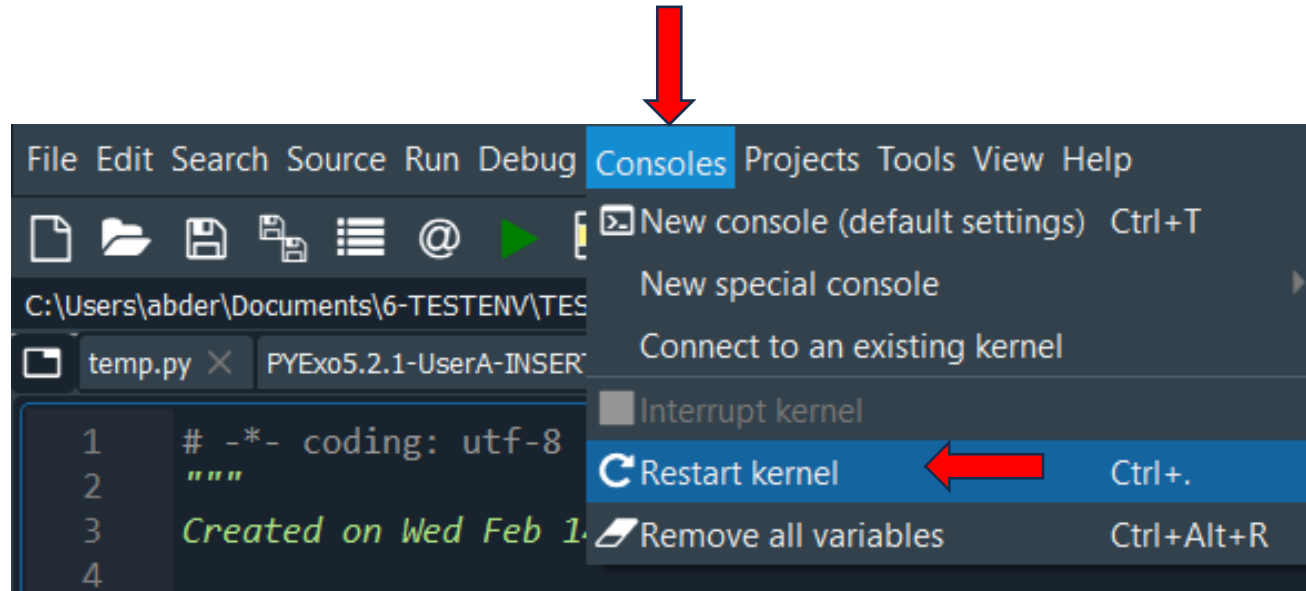
IPython 8.27.0 -- An enhanced Interactive Python.

In [1]: pip install matplotlib
```

- Installer **tensorflow** : `pip install --user tensorflow` *(Formation Machine Learning uniquement)*

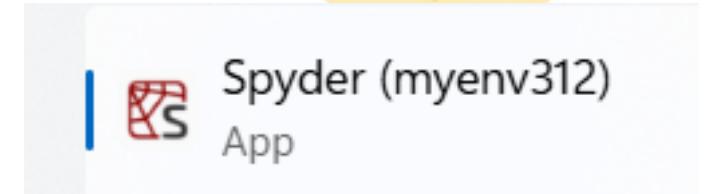
Installation

9) Redémarrer le noyau de spyder → Onglet Console → *Restart Kernel*

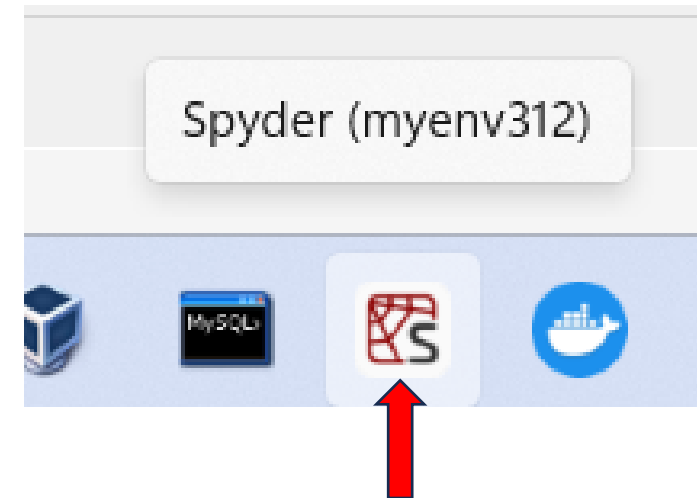


Installation

10) Barre de recherche en bas de votre écran → Chercher spyder, puis l'épingler à la barre de tâche de votre ordinateur



11) Lancer spyder en cliquant sur le bouton créé en bas de votre écran



Installation

12) Validation de l'installation : S'assurer que l'installation est complète, et que l'environnement est prêt

- Créer un fichier **test-env.py** :

File -> New File

File -> Save as **test-env.py**

- Saisir les lignes de code suivantes :



- Lancer le programme



```
# TESTER L'ENVIRONNEMENT TECHNIQUE

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split

import tensorflow as tf
from tensorflow.keras import layers, models
```

- S'assurer qu'aucun message d'erreur n'apparaît sur la console

Installation

Les messages sur la Console Spyder devraient ressembler à ceci :

Ce message d'information
n'est pas bloquant



```
Help Variable Explorer Plots Files
Console 1/A X
Bureau )
----- import Librairies -----

----- Fin import Librairies -----

2024-02-04 10:56:58.765923: I tensorflow/core/platform/
cpu_feature_guard.cc:182] This TensorFlow binary is optimized to us
available CPU instructions in performance-critical operations.
To enable the following instructions: SSE4.1 SSE4.2, in other
operations, rebuild TensorFlow with the appropriate compiler flags.

----- Fin import tensorflow -----

In [2]:
```

Tensorflow

Installation (Méthode-2)

(Formation Machine Learning uniquement)

Installation TENSORFLOW

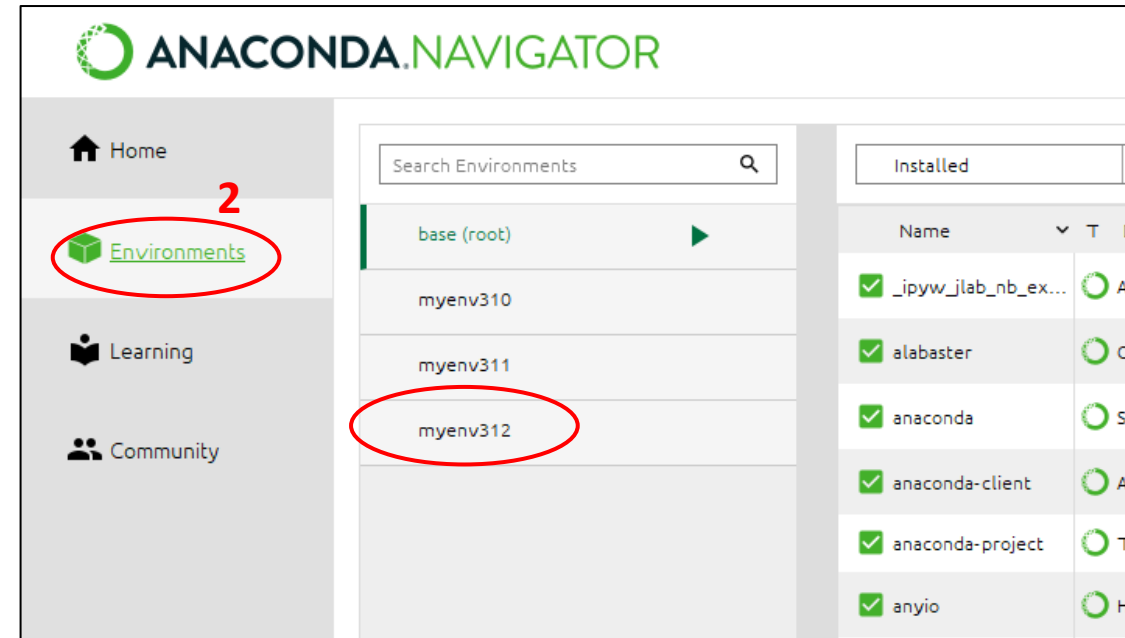
Méthode-2 → Plan B

*A faire uniquement si tensorflow installé
avec la méthode précédente ne fonctionne pas*

Méthode-2 :

1) Lancer Anaconda

2) Environments : Selectionner myenv312



Méthode-2 : (Suite)

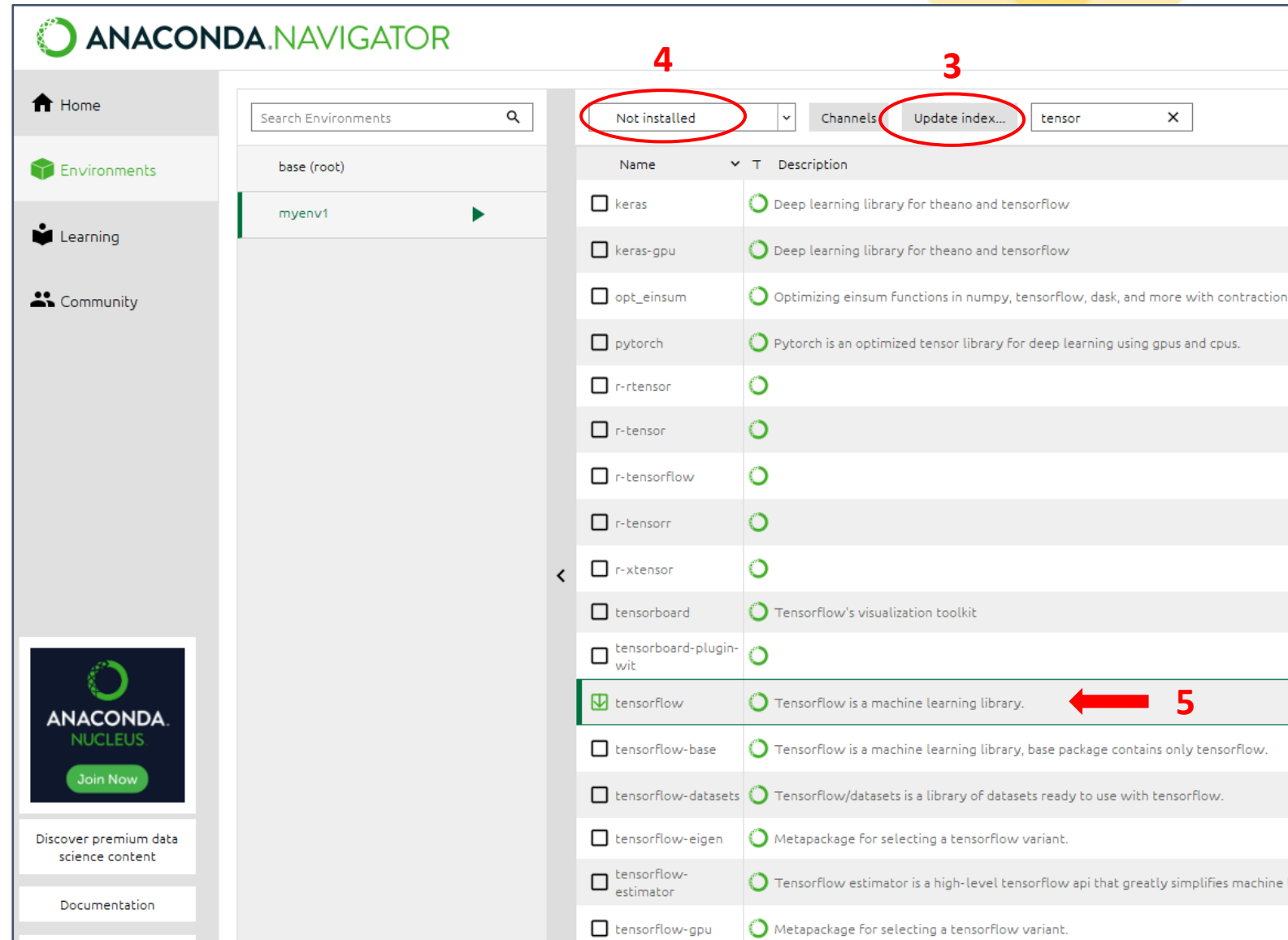
3) Cliquer sur **Update index**

4) Select **Not installed**

5) Cocher **tensorflow**

6) Cliquer sur **APPLY** (en bas à droite)

puis une 2nd fois **APPLY** lorsque
le popup sera affiché



Bases de données SQL

PostgreSQL : installation sur Windows

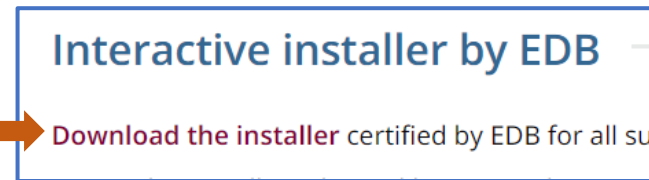
1) <https://www.postgresql.org>

2) Click sur Download

3) Click sur windows



4) click sur **download the installer**



5) choisir windows x86-64 --> lancer le .exe --> Cliquer sur NEXT

6) Choisir (et bien noter) un **mot de passe** du superuser **postgres**

7) Ne PAS modifier le no de port du serveur. Il doit rester à l'écoute sur le no de port : 5432

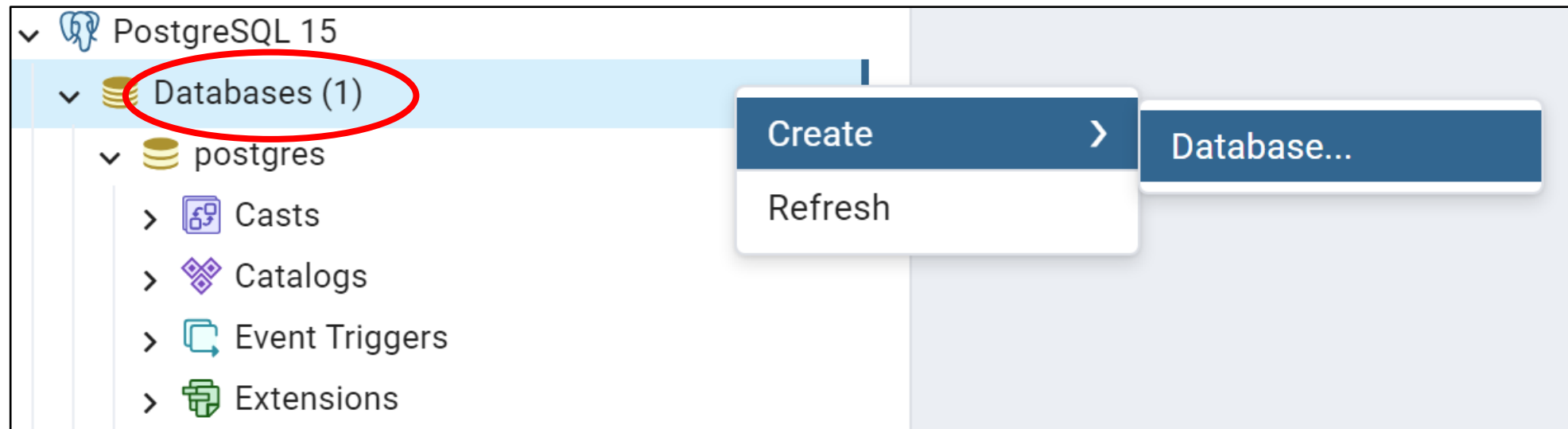
8) Sélectionner "**local server**" (choix par défaut)

9) A la question : **Applications you would like to install ?** , Cliquer sur **Cancel**

pgAdmin : Interface d'administration de PostgreSQL

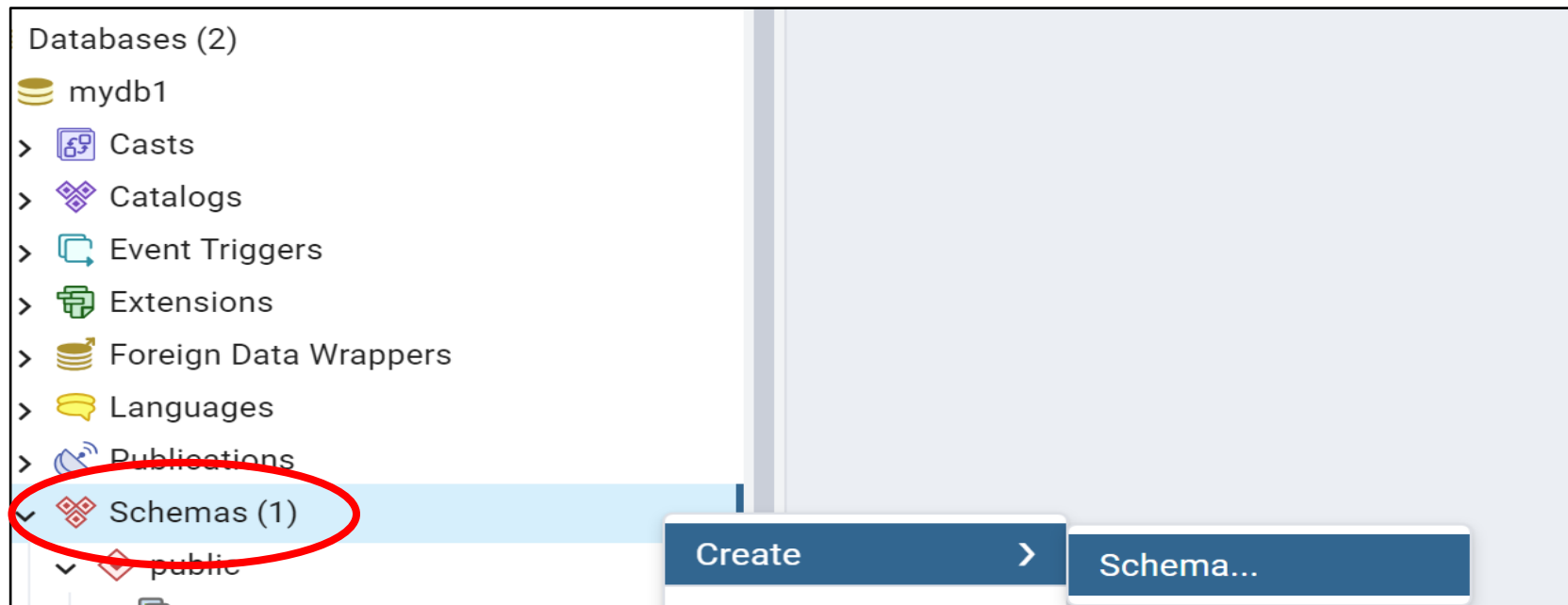
1) Créer une base de données

- Lancer **pgAdmin** → Entrer le password du superuser postgres
- Cliquer sur **servers** → PostgreSQL → databases → click droite → **create database** → **mydb1**



2) Créer un schema

- Lancer **pgAdmin** → Entrer le password du superuser postgres
- Cliquer sur databases → **mydb1** → Schémas → Create → Schema → **S1**



pgAdmin : Interface d'administration de PostgreSQL

3) psycopg2 : module d'interface Python - PostgreSQL

Installation :

- Ouvrir le terminal cmd
- **pip install psycopg2**

```
C:\Users\abder>  
C:\Users\abder>  
C:\Users\abder>  
C:\Users\abder>  
C:\Users\abder>pip install psycopg2
```



Flask et FastAPI

Frameworks de développement web

- Installer les frameworks Flask et FastAPI
- Saisir les commandes suivantes dans la console de spyder :

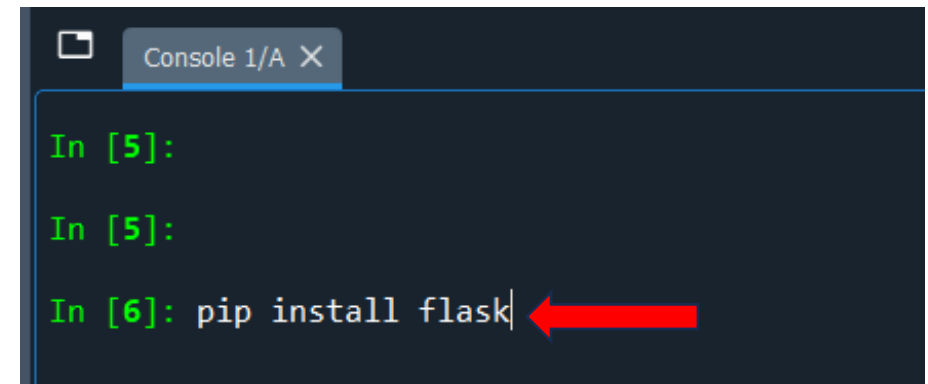
`pip install flask`

`pip install fastapi`

`pip install uvicorn`

`pip install bs4`

`pip install requests`



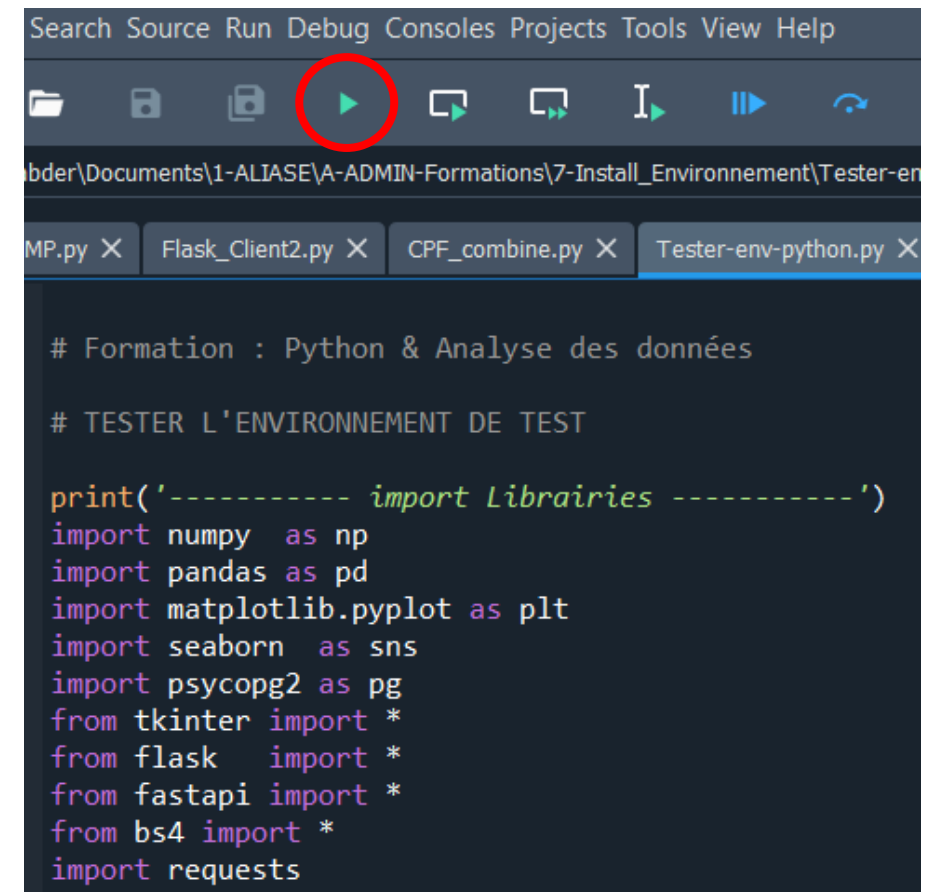
The screenshot shows a Spyder IDE console window titled 'Console 1/A'. It contains three lines of input, each starting with 'In [5]:'. The third line, 'In [6]: pip install flask', is currently being edited, with a red arrow pointing to the end of the command line.

```
Console 1/A X
In [5]:
In [5]:
In [6]: pip install flask
```

Installation : Flask et FastAPI

Validation de l'installation : S'assurer que l'installation est complète, et que l'environnement est prêt

- Ouvrir Spyder
- Créer un fichier `tester-env-python.py` :
 - File -> New File
 - File -> Save as `tester-env-python.py`
- Saisir les lignes de code suivantes : 
- Lancer le programme 
- S'assurer qu'aucun message d'erreur n'apparaît sur la console



```
# Formation : Python & Analyse des données

# TESTER L'ENVIRONNEMENT DE TEST

print('----- import Librairies -----')
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import psycpg2 as pg
from tkinter import *
from flask import *
from fastapi import *
from bs4 import *
import requests
```

Docker

Conteneuriser une application



1. Installer docker engine :

. <https://docs.docker.com/desktop>

. *Install Docker Desktop* (en bas de la page)

2. Lancer docker

3. Tester Docker : Terminal> `docker --version`

Visual Code Editor: Installation

Optionnel : serait utile en cas de besoin (pb technique avec spyder)

2) Installer l'éditeur VSCode à partir de <https://code.visualstudio.com/download#>

Visual Studio Code Docs Updates Blog API Extensions FAQ

Download

Windows
Windows 7, 8, 10

.deb
Debian, Ubuntu

.rpm
Red Hat, Fedora, SUSE

Mac
macOS 10.10+

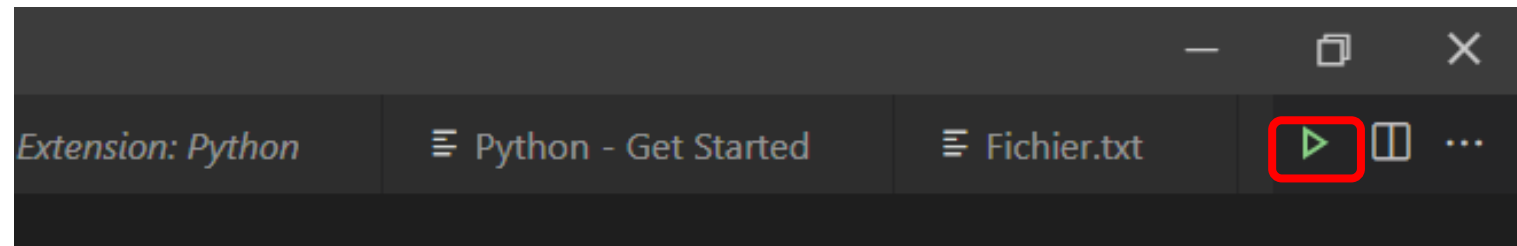
User Installer	64 bit	32 bit	ARM
System Installer	64 bit	32 bit	ARM
.zip	64 bit	32 bit	ARM

.deb 64 bit
.rpm 64 bit
.tar.gz 64 bit

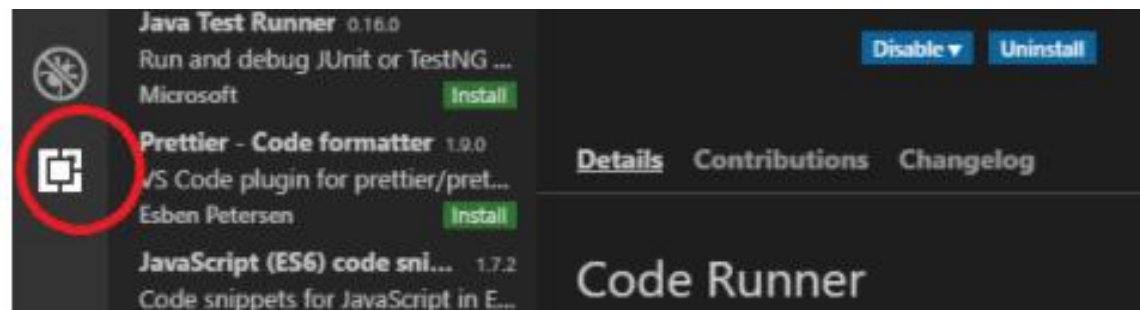
Visual Code Editor: Installation (Cont)

3) Lancer Visual Studio Code

Si vous ne voyez pas le bouton 'Run' (bouton vert en haut à droite)



Vous pouvez le faire apparaître en activant l'extension '**Code Runner**'



Navigateur

Navigateur

Pour rejoindre la visioconférence, il est recommandé d'utiliser le navigateur **Chrome** .

S'il n'est pas installé sur votre machine, veuillez l'installer à partir de ce site :

<https://www.google.com/chrome>

et le marquer comme navigateur par défaut.

Linux

1) Installer anaconda :

- Ouvrir le site www.anaconda.com
- Cliquer sur **Free download** (en haut à droite de la page)
- Cliquer sur "**Sign-In**" (en haut à droite de la page)
- Dans le cadre "**Download Now**", cliquer sur "**Get Started**"
- Cliquer sur "**Linux**"
- Dans le cadre "**Anaconda Distribution**", cliquer sur "**64-Bit(x86) Installer**"
- Télécharger le fichier : **Anaconda3-2025.xx-Linux-x86_64.sh** (script d'installation Linux)
- Ouvrir le terminal
- Aller dans le dossier où le fichier a été téléchargé . Ex : `cd ~/Téléchargements` ou `cd ~/Downloads`)
- Lancer l'installation comme suit : `bash Anaconda3-2025.xx-Linux_x86_64.sh`
 - . Appuyer sur **Enter** pour faire défiler la licence
 - . Taper : **yes** pour accepter
- Choisir le dossier d'installation par défaut : `/home/user/anaconda3` , Appuyer sur **Enter** pour accepter

2) Initialiser anaconda :

- À la question : ***Do you wish the installer to initialize Anaconda3 ?***

répondre : **yes**

Cela ajoute la configuration dans `~/.bashrc` afin que conda fonctionne automatiquement dans le terminal

- **Recharger le Terminal :**

Fermer et rouvrir le terminal , ou exécuter : `source ~/.bashrc`

- **Vérifier l'installation :**

Tester : `conda --version`

ou

`conda list`

Si tout est correct, la liste des packages Anaconda s'affiche.

3) Créer un environnement :

- Ouvrir le terminal
- Créer un environnement : `conda create --name myenv312 python=3.12`
- Activer l'environnement avec les commandes suivantes :
 - \$ source `~/bashrc`
 - (base) \$ `conda activate myenv312`

4) Installer les librairies : dans l'environnement **myenv312**

- Installer **tensorflow** : (myenv312) \$ **conda install tensorflow**
- Installer **spyder** : (myenv312) \$ **conda install -c anaconda spyder**
- Installer **sklearn** : (myenv312) \$ **conda install scikit-learn**
- Ouvrir **spyder** : (myenv312) \$ **spyder**

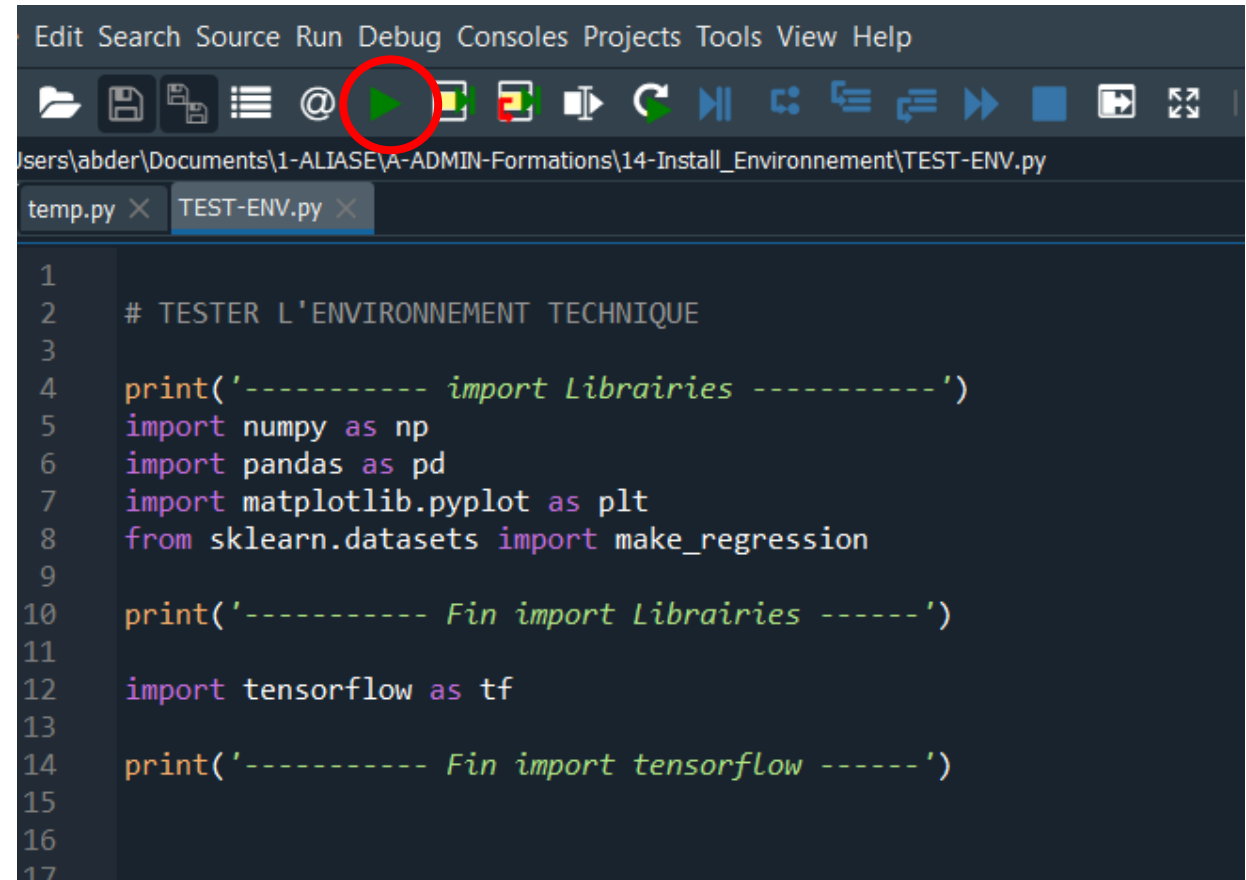
5) Installer les librairies : Commandes à saisir dans la **console de spyder**

- Installer **matplotlib** : **pip install matplotlib**
- Installer **seaborn** : **pip install seaborn**
- Installer **pandas** :
pip install pandas
pip install xlrd
pip install openpyxl
pip install xlswriter

VALIDATION DE L'INSTALLATION

S'assurer que l'installation est complète, et que l'environnement est prêt

- Ouvrir Spyder
- Créer un fichier **test-env.py** :
 - File -> New File
 - File -> Save as **test-env.py**
- Saisir les lignes de code suivantes : 
- Lancer le programme 
- S'assurer qu'aucun message d'erreur n'apparaît sur la console



```
1
2 # TESTER L'ENVIRONNEMENT TECHNIQUE
3
4 print('----- import Librairies -----')
5 import numpy as np
6 import pandas as pd
7 import matplotlib.pyplot as plt
8 from sklearn.datasets import make_regression
9
10 print('----- Fin import Librairies -----')
11
12 import tensorflow as tf
13
14 print('----- Fin import tensorflow -----')
15
16
17
```

VALIDATION DE L'INSTALLATION

Les messages sur la Console Spyder devraient ressembler à ceci :

Ce message d'information
n'est pas bloquant



```
Help Variable Explorer Plots Files
Console 1/A X
Bureau )
----- import Librairies -----

----- Fin import Librairies -----

2024-02-04 10:56:58.765923: I tensorflow/core/platform/
cpu_feature_guard.cc:182] This TensorFlow binary is optimized to us
available CPU instructions in performance-critical operations.
To enable the following instructions: SSE4.1 SSE4.2, in other
operations, rebuild TensorFlow with the appropriate compiler flags.

----- Fin import tensorflow -----

In [2]:
```