

# 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 35**

# Windows / MacOS

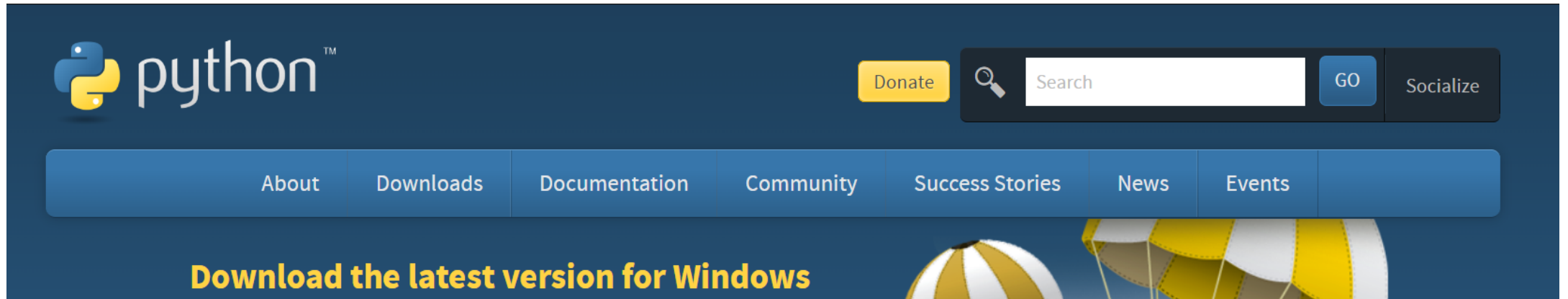
# Guide d'installation de l'environnement

## Contenu

- Installation de **Python**
- Installation de l'éditeur **Visual Studio Code**
- Installation de l'environnement **Anaconda** (Spyder, ...)
- Installation de **TENSORFLOW** ( **Uniquement en cas de formation Machine Learning** )
- Installation de la base de données **PostgreSQL**

# Python: Installation

1) Installer Python **3.12** à partir de <https://www.python.org/downloads/>



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

# Prévoir un dossier Formation

- . 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 + Exercices)  
Créer 4 sous-dossiers : **Module-PY1** . . . **Module-PY4**
  
  - **ML-Supports** : dossier pour la partie Machine Learning (supports + Exercices)  
Créer 8 sous-dossiers : **Module-ML1** . . . **Module-ML8**

# Editeurs

# Visual Code Editor: Installation

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

The screenshot shows the Visual Studio Code download page with the following structure:

- Navigation bar: Visual Studio Code, Docs, Updates, Blog, API, Extensions, FAQ, Download
- Platform selection buttons:
  - Windows (Windows 7, 8, 10)
  - Linux: .deb (Debian, Ubuntu) and .rpm (Red Hat, Fedora, SUSE)
  - Mac (macOS 10.10+)
- Download options table:
 

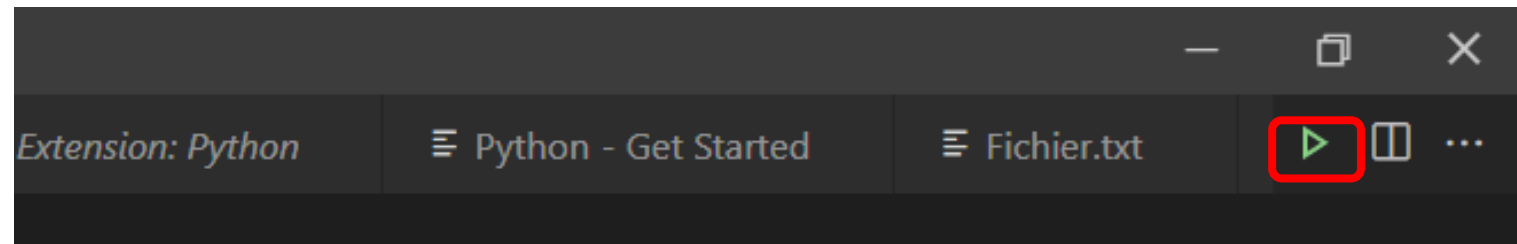
User Installer	64 bit	32 bit	ARM
<b>System Installer</b>	64 bit	32 bit	ARM
.zip	64 bit	32 bit	ARM



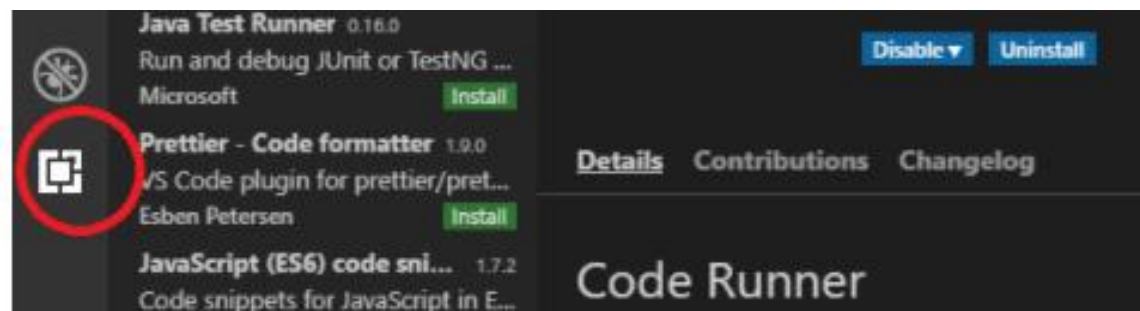
# Visual Code Editor: Installation (Cont)

## 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*'



Anaconda contient tous les outils et bibliothèques dont nous avons besoin pour faire du Machine Learning :  
**Numpy, Matplotlib, Pandas, Sklearn, etc.**

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

Une fois Anaconda installé, lancer l'application **Spyder**, l'application Web qui permet de créer des codes Python.



# Tensorflow

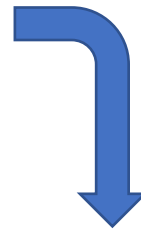
**Uniquement en cas de formation Machine Learning**

# TENSORFLOW : Installation

3 méthodes pour installer TENSORFLOW :

- **Méthode-1** : à partir de Spyder
- **Méthode-2** : à partir de Anaconda
- **Méthode-3** : à partir de la fenêtre "Anaconda Prompt"

Voir les 3 démarches dans les pages suivantes



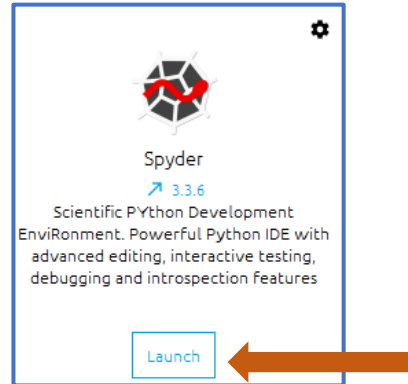
# TENSORFLOW

**Méthode-1 :**  
**à partir de Spyder**

# TENSORFLOW : Installation

**Méthode-1 :** à partir de Spyder

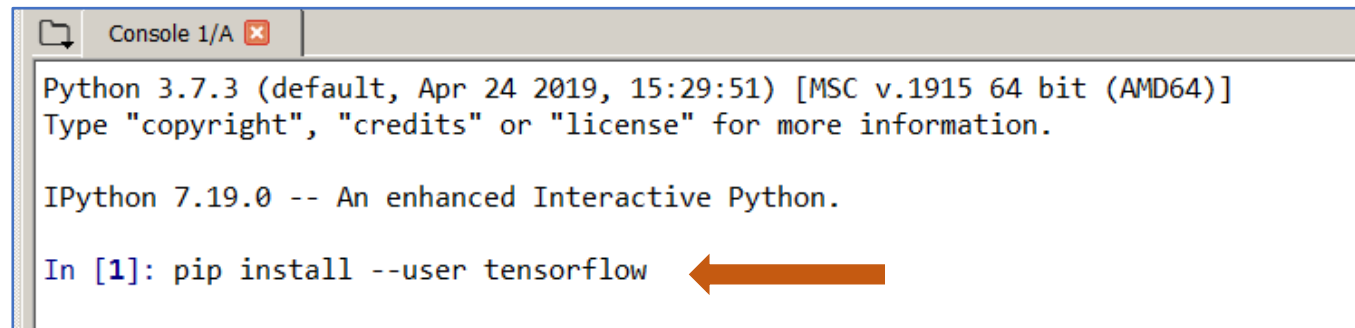
1) Aller sur Anaconda → Ouvrir 'Spyder'



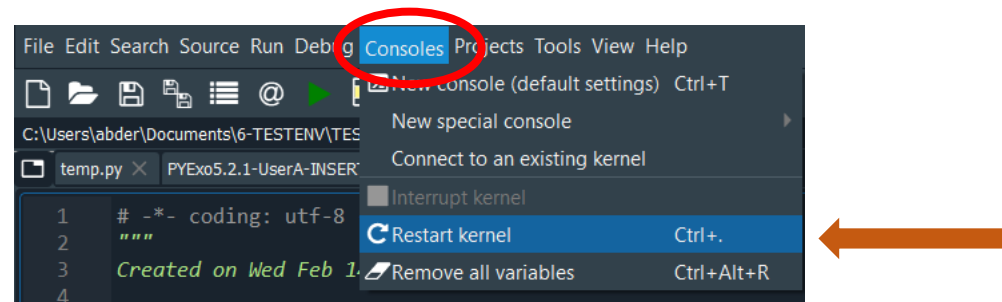
2) Taper : `pip install --user tensorflow`

dans la fenêtre console ( à droite ) .

L'installation prends qq minutes (5-10 mins)



3) Spyder → Onglet Console → Restart Kernel



# VALIDATION DE L'INSTALLATION

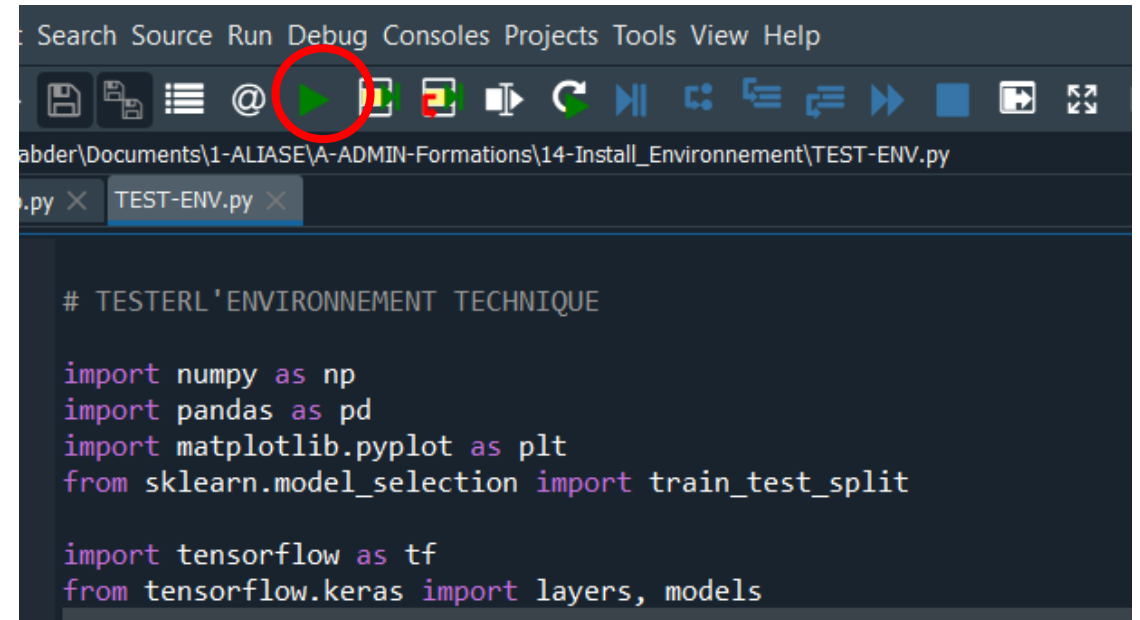
**S'assurer que l'installation est complète, et que l'environnement est prêt**

- 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



```
# 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
```

# 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]:
```



# TENSORFLOW


**Méthode-2 :**

**à partir du terminal "Anaconda Prompt"**

# TENSORFLOW : Installation

**Méthode-2 :** à partir du terminal "Anaconda Prompt"

1) Ouvrir le terminal : Search → Anaconda Powershell



Anaconda Powershell Prompt (Anaco...

2) Créer un environnement avec un nom, par exemple **myenv312** :

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

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

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

```
(myenv312) PS conda install tensorflow
```

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

```
(myenv312) PS conda install scikit-learn
```

**Méthode-2 :** à partir du terminal "Anaconda Prompt"

4) Lancer **spyder** à partir du terminal: (myenv312) PS **spyder**

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

- Installer **matplotlib** :

**pip install matplotlib**

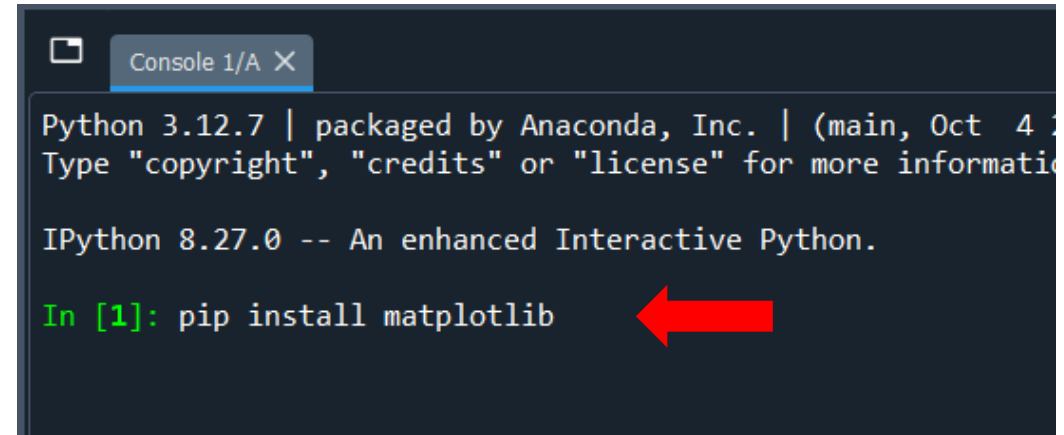
- Installer **pandas** :

**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, 11:59:07) [AMD64]  
Type "copyright", "credits" or "license()" for more (ipython>)  
IPython 8.27.0 -- An enhanced Interactive Python.  
In [1]: pip install matplotlib
```

# VALIDATION DE L'INSTALLATION

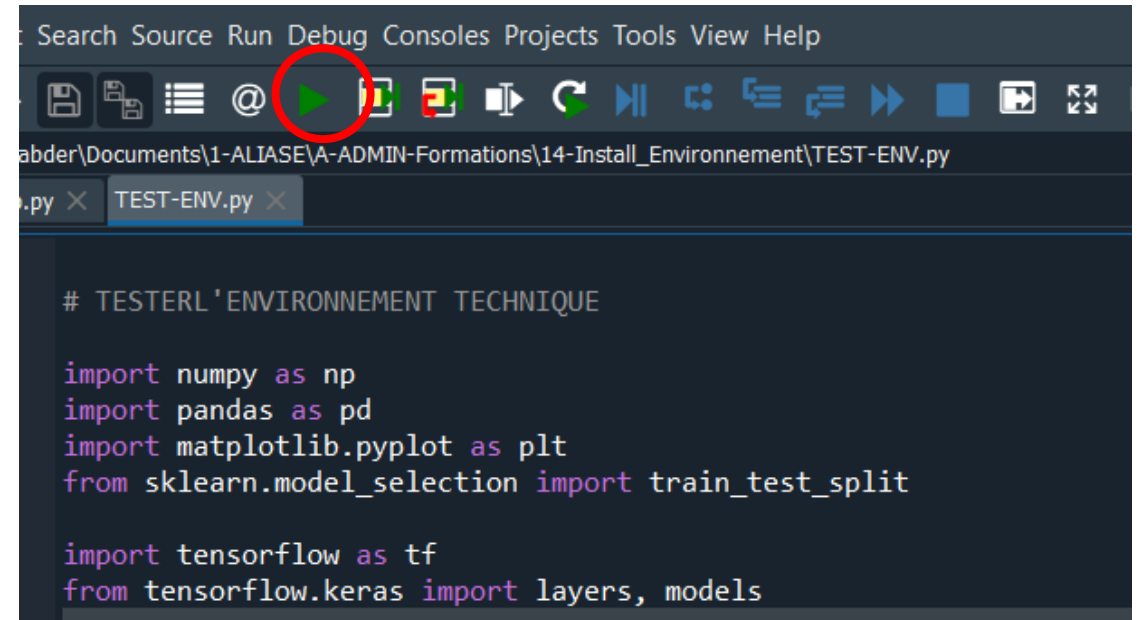
**S'assurer que l'installation est complète, et que l'environnement est prêt**

- 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



```
# 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
```

# 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]:
```

# TENSORFLOW

**Méthode-3 :**  
**à partir d'Anaconda**

## Méthode-3 : à partir de Anaconda

1) Créer un nouvel environnement

→ Environments

→ Create (en bas de page)

2) . Saisir un nom : myenv1

. Packages : choisir python version

3) Cliquer sur Create (à droite de la page)

The screenshot shows the Anaconda Navigator interface. The 'Environments' tab is selected in the left sidebar, indicated by a red circle and the number '1'. The main panel displays a list of installed packages. A 'Create new environment' dialog box is open, showing the name 'myenv1' in the 'Name' field (indicated by a red arrow and the number '2'), the location 'C:\Users\labder\Anaconda3\envs\myenv1', and the 'Python 3.8' package selected (indicated by a red arrow). The 'Create' button in the dialog is circled in red and labeled with the number '3'. At the bottom of the main panel, the 'Create' button is also circled in red.

Name	T	Description
✓ _ipyw_jlab_nb_ex...	○	
✓ alabaster	○	
✓ anaconda	○	
✓ anaconda-client	○	
✓ anaconda-project	○	
✓ anyio	○	
✓ appdirs	○	
✓ argh	○	
✓ argon2-cffi	○	
✓ asn1crypto	○	
✓ astroid	○	
✓ astropy	○	
✓ async-generator	○	
✓ async_generator	○	
✓ atomicwrites	○	
✓ attrs	○	
✓ autopep8	○	
✓ babel	○	
✓ backcall	○	

## Méthode-3 : (Suite)

4) Cliquer sur **Update index**

5) Select **Not installed**

6) Cocher **tensorflow**

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

puis une 2<sup>nd</sup> fois **APPLY** lorsque  
le popup sera affiché

ANACONDA.NAVIGATOR

Home

Environments

Learning

Community

Search Environments

base (root)

myenv1

Channels: Not installed (5) Update index... (4) tensor

Name	T	Description
<input type="checkbox"/> keras	🟢	Deep learning library for theano and tensorflow
<input type="checkbox"/> keras-gpu	🟢	Deep learning library for theano and tensorflow
<input type="checkbox"/> opt_einsum	🟢	Optimizing einsum functions in numpy, tensorflow, dask, and more with contraction
<input type="checkbox"/> pytorch	🟢	Pytorch is an optimized tensor library for deep learning using gpus and cpus.
<input type="checkbox"/> r-tensor	🟢	
<input type="checkbox"/> r-tensor	🟢	
<input type="checkbox"/> r-tensorflow	🟢	
<input type="checkbox"/> r-tensorr	🟢	
<input type="checkbox"/> r-xtensor	🟢	
<input type="checkbox"/> tensorboard	🟢	Tensorflow's visualization toolkit
<input type="checkbox"/> tensorboard-plugin-wit	🟢	
<input checked="" type="checkbox"/> tensorflow	🟢	Tensorflow is a machine learning library. ← 6
<input type="checkbox"/> tensorflow-base	🟢	Tensorflow is a machine learning library, base package contains only tensorflow.
<input type="checkbox"/> tensorflow-datasets	🟢	Tensorflow/datasets is a library of datasets ready to use with tensorflow.
<input type="checkbox"/> tensorflow-eigen	🟢	Metapackage for selecting a tensorflow variant.
<input type="checkbox"/> tensorflow-estimator	🟢	Tensorflow estimator is a high-level tensorflow api that greatly simplifies machine l
<input type="checkbox"/> tensorflow-gpu	🟢	Metapackage for selecting a tensorflow variant.

ANACONDA NUCLEUS  
Join Now

Discover premium data science content

Documentation

APPLY



## Méthode-3 : (Suite)

8) Cliquer sur **Home**

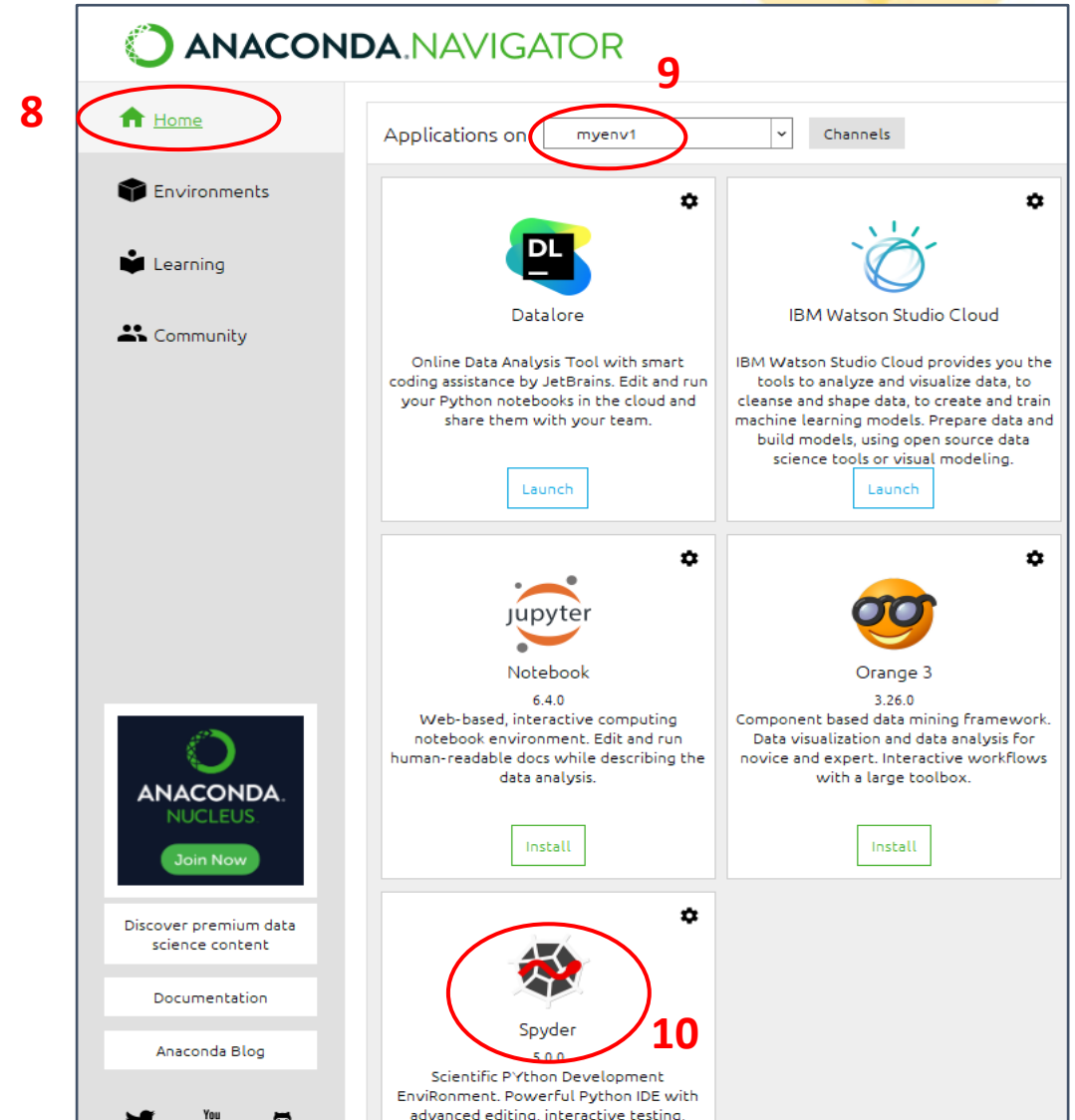
9) Select **myenv1** dans Applications

10) Cliquer sur **Install Spyder**

puis sur **Launch Spyder**

11) Dans la console de Spyder, installer :

- . **pip install matplotlib**
- . **pip install scikit-learn**
- . **pip install pandas**
- . **pip install xlrd**
- . **pip install openpyxl**
- . **pip install xlswriter**



# VALIDATION DE L'INSTALLATION

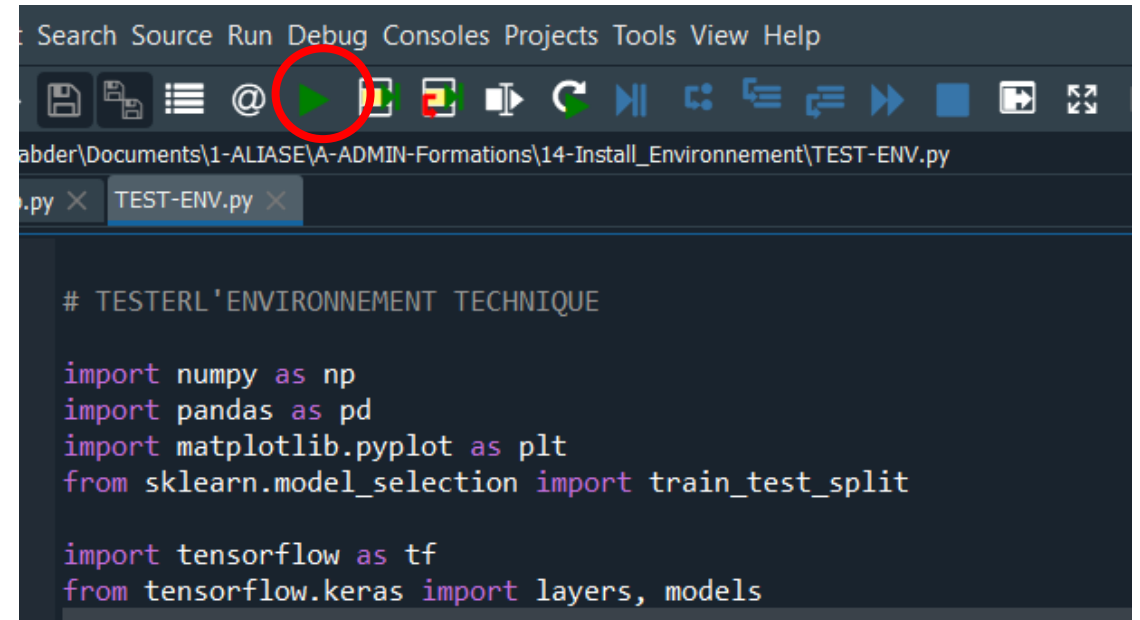
**S'assurer que l'installation est complète, et que l'environnement est prêt**

- 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



```
# TESTERL '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
```

# 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]:
```

# Bases de données SQL

**OPTIONNEL** pour les formations IA, Machine Learning

Ce module est présenté à la demande.

Si le participant le demande, et si la durée de la formule de formation choisie le permet, nous l'insérons parmi les chapitres à présenter.

# PostgreSQL : installation sur Windows

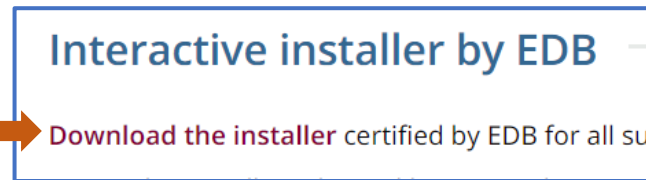
1) <https://www.postgresql.org>

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**

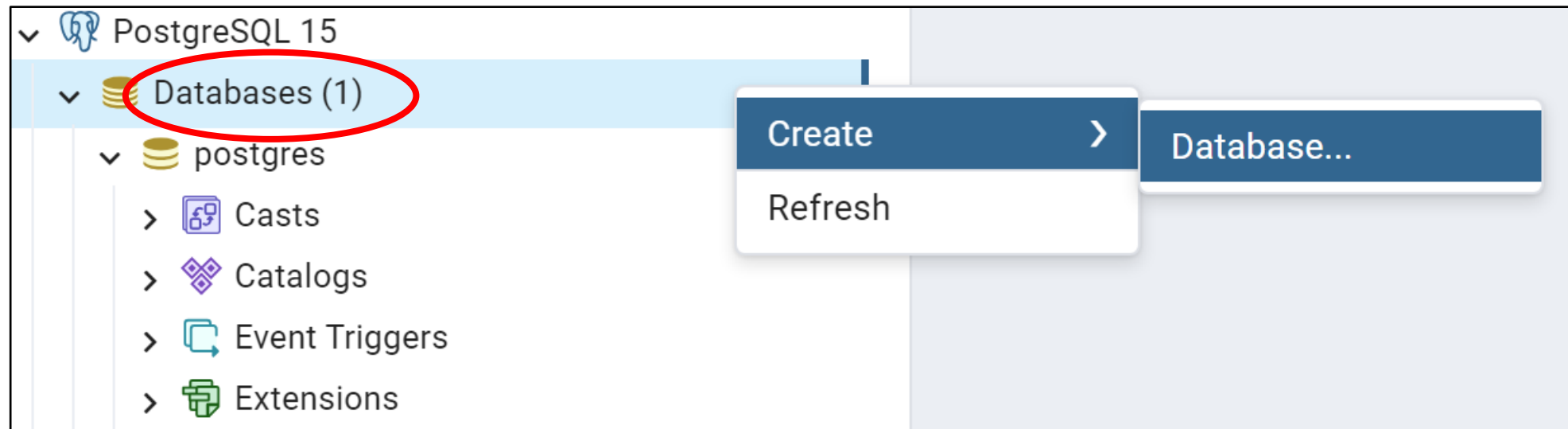
7) Ne PAS modifier le no de port du serveur. Il doit rester à l'écoute sur le no de port : 5432

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

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

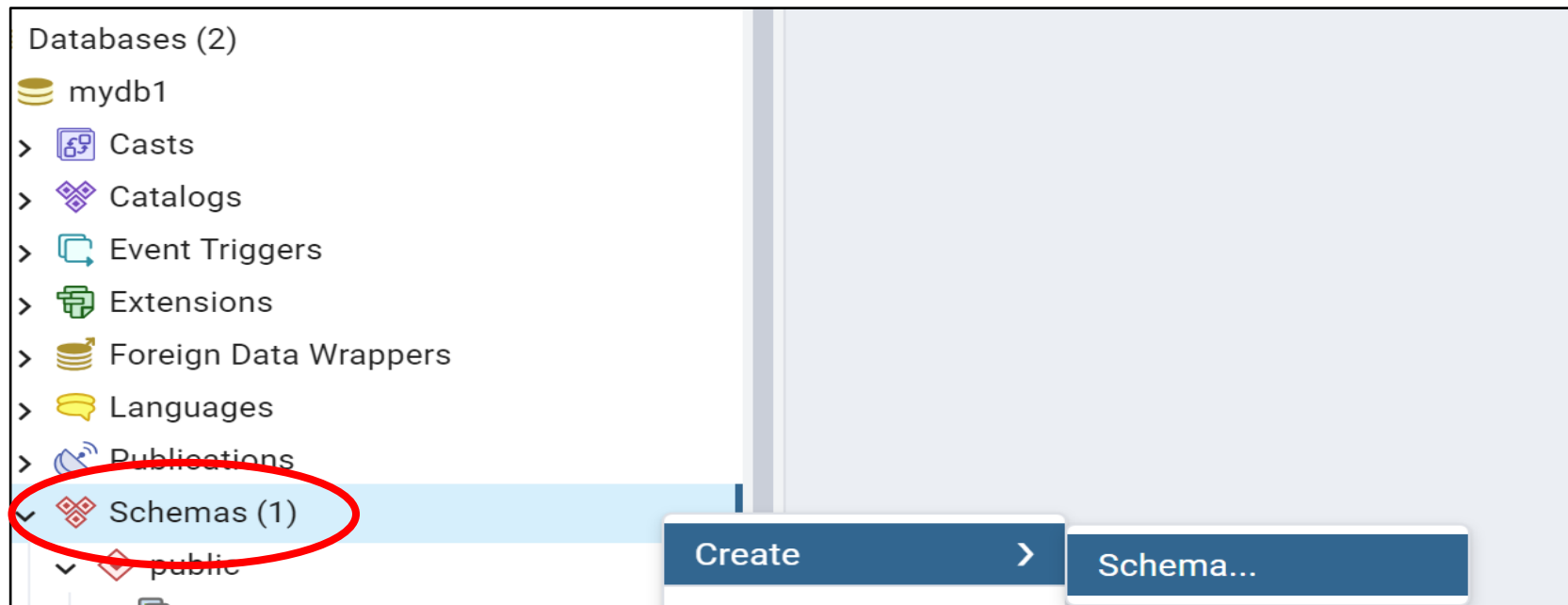
## 1) Créer une base de données

- 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 une fenêtre DOS
- **pip install psycopg2**



# 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 navigateur à partir de l'interface graphique, puis aller sur [www.anaconda.com](http://www.anaconda.com)
- Copier le nom du fichier **Anaconda3-2023.9-0-Linux\_x86\_64.sh**
- Ouvrir le terminal
- Lancer l'installation comme suit : **bash ~/Téléchargements/Anaconda3-2023.9-0-Linux\_x86\_64.sh**  
Répondre **oui** aux questions

## 2) Initialiser anaconda :

- Arrêter/relancer la machine virtuelle
- Ouvrir le terminal et créer un environnement : **conda create --name myenv312 python=3.12**
- Activer l'environnement avec les commandes suivantes :  
\$ source **~/bashrc**  
(base) \$ **conda activate myenv312**

### 3) 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**

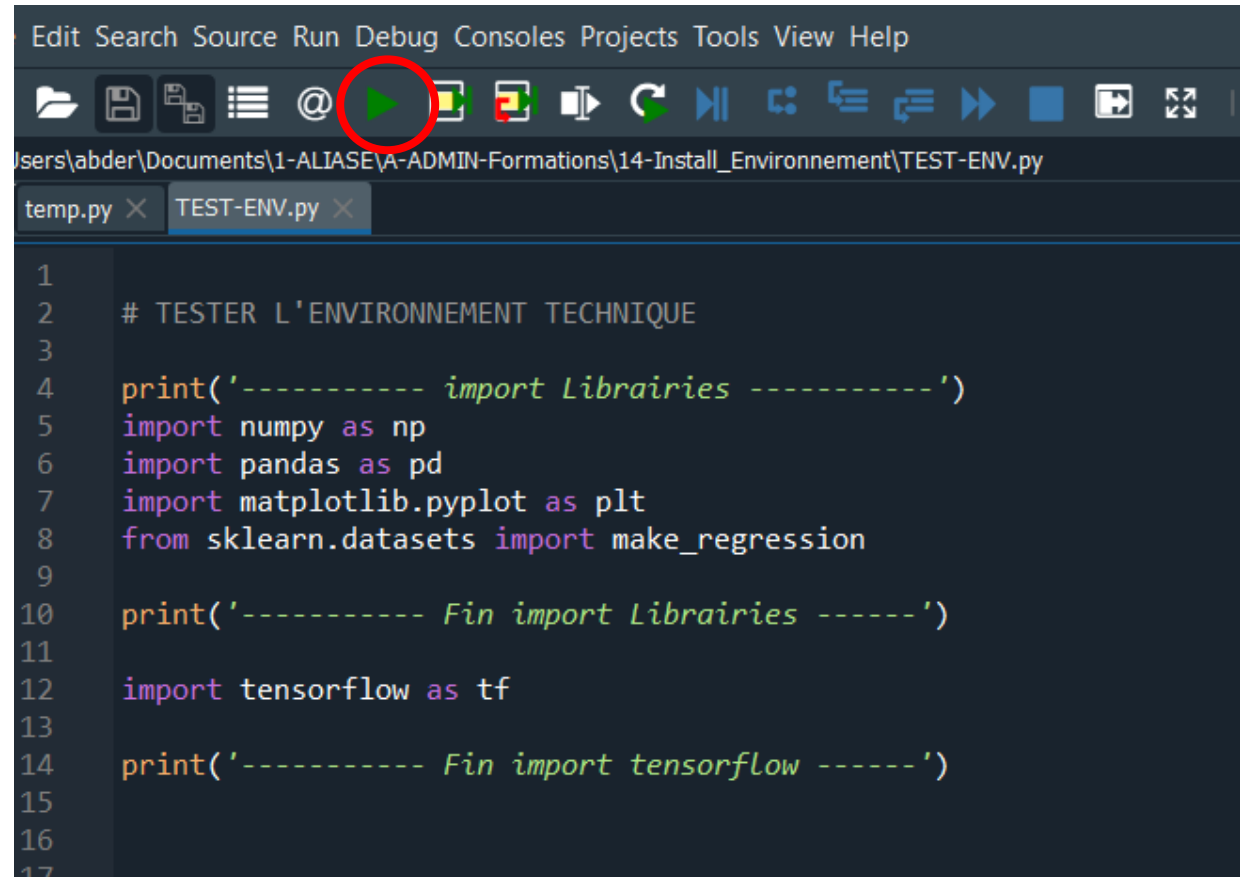
### 4) Installer les librairies : Commandes à saisir dans la **console de spyder**

- Installer **matplotlib** : **pip install matplotlib**
- 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]:
```