

# Installer un environnement virtuel Python 3



Jean-Luc.Charles@ENSAM.EU  
Eric.Ducasse@ENSAM.EU

L'équipe enseignante de l'ENSAM Bordeaux recommande :

- L'utilisation de votre ordinateur portable personnel pour toutes les séances de Math-Info.
- L'utilisation de **Miniconda3** pour l'installation d'un **environnement virtuel Python 3** dédié sur vos portables GNU/Linux, Mac OS X & Windows.
- Les EDI (*Environnement de Développement Intégré*) **Idlex**, comme outil de développement simple et efficace, et **Jupyter Notebook**, comme outil d'apprentissage, de calcul formel et de pilotage, utilisables sur toutes les machines.

## Installation de Miniconda3 ☒

- ☒ Si **Anaconda3** ou **Miniconda3** est déjà installé, passer directement à la diapositive 5
- Sur le site [docs.conda.io/en/latest/miniconda.html](https://docs.conda.io/en/latest/miniconda.html) télécharger la dernière version de **Miniconda3** correspondant à votre OS.
- Lancer l'installateur... **vous devrez noter le chemin choisi pour le dossier d'installation Miniconda3** ~> cela vous servira par la suite...[Linux, taper : `bash ../miniconda3-latest-Linux-x86_64.sh`]

**Attention** : le chemin **Miniconda3** ne doit contenir ni espace, ni caractère accentué !



Windows :

`C:\Miniconda3` ou `C:\Users\Marie\Miniconda3` ~> conviennent

`C:\Yoann\Mes install\Miniconda3` ~> ne convient pas (espace)

`C:\Users\Léon\Miniconda3` ou `C:\Users\François\Miniconda3`

~> ne conviennent pas (e accent aigu, cédille)

MacOSX & GNU/Linux :

`/home/moi/miniconda3` ou `/Users/moi/opt/miniconda3` ~> conviennent

`/home/moi/Mes install/Miniconda3` ~> ne convient pas (espace)

`/Users/Léon/Miniconda3` ou `/home/François/Miniconda3`

~> ne conviennent pas (e accent aigu, cédille)



## Installation de **Miniconda3**

Dans un NOUVEAU **terminal** (Mac, Linux) ou la fenêtre « **Anaconda Prompt (Miniconda3)** » (Windows), taper (ou plutôt copier-coller) successivement les 4 commandes :

```
(base) ...> conda install --name base conda-anaconda-tos -y
```

```
(base) ...> conda tos accept --override-channels --channel https://repo.anaconda.com/pkgs/main
```

```
(base) ...> conda tos accept --override-channels --channel https://repo.anaconda.com/pkgs/r
```

```
(base) ...> conda tos accept --override-channels --channel https://repo.anaconda.com/pkgs/msys2
```

## Création d'un environnement virtuel Python

Dans un NOUVEAU **terminal** (Mac, Linux) ou la fenêtre « **Anaconda Prompt (Miniconda3)** » (Windows), taper la commande :

```
(base) ...> conda create -n minfo python=3.10 -y
```

## Activation de l'environnement virtuel

Dans le terminal, ou la fenêtre « Anaconda Prompt (Miniconda3) », taper :

```
(base) ...> conda activate minfo
```

Votre *prompt* (invite de commande) est maintenant préfixé par **(minfo)** :

```
(minfo) C:\Users\moi> pour Windows,
```

```
(minfo) /Users/moi> pour Mac, (minfo) /home/moi> pour Linux
```

## Installation des modules Python [1/2]

Dans le terminal, ou la fenêtre « Anaconda Prompt (Miniconda3) », avec l'environnement **minfo activé**, pour installer le module [numpy](#), taper :

```
(minfo) ...> conda install numpy=2 -y
```

Remarque : avec une bonne connexion Internet, on pourra installer plusieurs modules à la fois.

## Installation des modules Python [2/2]

De même installer le module `scipy` :

```
(minfo) ...> conda install scipy -y
```

le module `matplotlib` (avec `PyQt5` et non pas `PyQt6`) :

```
(minfo) ...> conda install pyqt=5 matplotlib -y
```

l'EDI **Jupyter** :

```
(minfo) ...> conda install jupyter -y ⊗
```

le module de calcul formel `sympy` :

```
(minfo) ...> conda install sympy -y
```

les modules `imageio` et `openpyxl` :

```
(minfo) ...> conda install imageio openpyxl -y
```

et enfin, l'EDI **Idlex** :

```
(minfo) ...> pip install idlex
```

⊗ En cas d'échec de l'installation de Jupyter sous Windows,

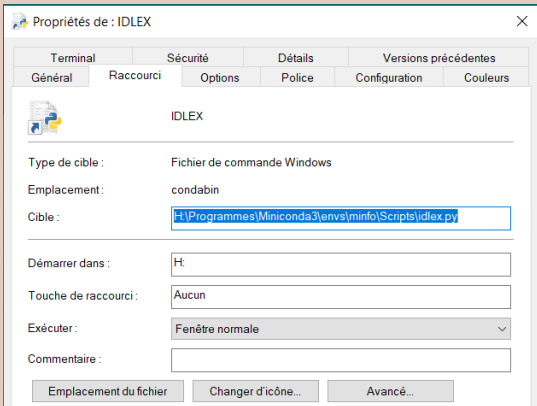
lancer **Microsoft PowerShell** en tant qu'administrateur et copier-coller l'instruction :

```
New-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\FileSystem" -Name "LongPathsEnabled" -Value 1 -PropertyType DWORD -Force
```

## Post-installation Windows : raccourci clavier pour le lancement de **Idlex** [1/3]

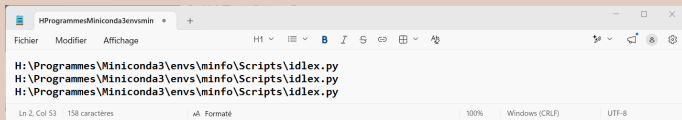
<Miniconda3> désignant le chemin d'accès du dossier Miniconda3 (H:\Programmes\Miniconda3 sur l'exemple) :

- Dans le dossier <Miniconda3>\envs\minfo\Scripts faire clic-droit sur le fichier **idlex.py** : « Afficher d'autres options ~ Envoyer vers... ~ Bureau ».
- Dans le bureau, renommer le raccourci **idlex.py** en **IDLEX**, puis faire un clic-droit « Propriétés » dessus pour faire afficher :



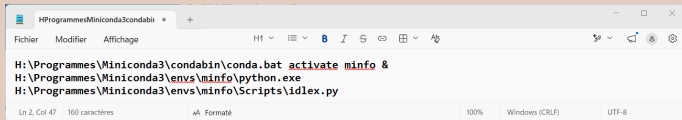
## Post-installation Windows : raccourci clavier pour le lancement de **Idlex** [2/3]

- Ouvrir le Bloc-notes et y copier 3 fois le champ « Cible » du raccourci :



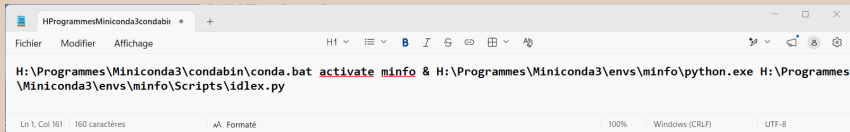
```
H:\Programmes\Miniconda3\envs\minfo\Scripts\idlex.py
H:\Programmes\Miniconda3\envs\minfo\Scripts\idlex.py
H:\Programmes\Miniconda3\envs\minfo\Scripts\idlex.py
```

- Remplacer dans la première ligne « `envs\minfo\Scripts\idlex.py` » par « `condabin\conda.bat activate minfo &` », dans la deuxième ligne « `Scripts\idlex.py` » par « `python.exe` » :



```
H:\Programmes\Miniconda3\condabin\conda.bat activate minfo &
H:\Programmes\Miniconda3\envs\minfo\python.exe
H:\Programmes\Miniconda3\envs\minfo\Scripts\idlex.py
```

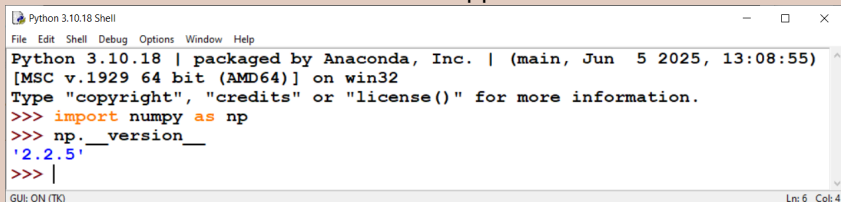
- Remettre les 3 lignes sur la même ligne, qu'on sélectionne par `ctrl+A` :



```
H:\Programmes\Miniconda3\condabin\conda.bat activate minfo & H:\Programmes\Miniconda3\envs\minfo\python.exe H:\Programmes\Miniconda3\envs\minfo\Scripts\idlex.py
```

## Post-installation Windows : raccourci clavier pour le lancement de **Idlex** [3/3]

- Copier-coller cette ligne dans le champ « Cible » du raccourci.
- Chemin du dossier de travail principal : dans le champ « Démarrer dans », mettre le chemin d'accès de son dossier de travail principal ou, à défaut, le chemin de son dossier utilisateur.
- Icône du raccourci  : en cliquant sur « Changer d'icône ~> Parcourir », installer l'icône <Miniconda3>\Lib\idlelib\Icons\idle.ico.
- Fermer le raccourci par « OK » puis double-cliquer dessus. Une fenêtre ressemblant à la fenêtre suivante doit apparaître :



```
Python 3.10.18 Shell
File Edit Shell Debug Options Window Help
Python 3.10.18 | packaged by Anaconda, Inc. | (main, Jun 5 2025, 13:08:55)
[MSC v.1929 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> import numpy as np
>>> np.__version__
'2.2.5'
>>> |
GUI: ON (TK) Ln: 6 Col: 4
```

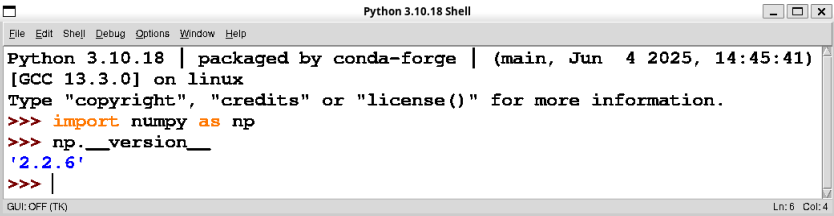
- Tester la console **Idlex** par les deux commandes apparaissant sur la figure ci-dessus (noter les doubles « \_ » dans « `__version__` »).

## Post-installation Windows : dysfonctionnement créé par l'antivirus **Avast**

- Si vous avez l'antivirus **Avast** installé sur votre ordinateur, il arrive fréquemment que soudainement, les lancements de **Idlex** et **Jupyter** ne fonctionnent plus. Ne vous affolez pas !
- Explication : c'est juste que le fichier `python.exe` se trouvant dans le dossier `<Miniconda3>\envs\minfo` a été « *mis en quarantaine* » par l'antivirus.
- Remède : Il faut donc remettre le fichier `python.exe` à sa place. Pour ce faire, lancer Avast Antivirus. Dans l'onglet **Protection**, double-cliquer sur la zone **Quarantaine**.  
Y rechercher le fichier ↪ clic droit dessus ↪ Restaurer (voir par exemple [ici](#)).

## Mac & GNU/Linux : lancement de **Idlex**

- Dans un terminal avec l'environnement **minfo activé**, taper la commande :  
`(minfo) ...> idlex`
- Une fenêtre ressemblant à la fenêtre suivante doit apparaître :



```
Python 3.10.18 Shell
File Edit Shell Debug Options Window Help
Python 3.10.18 | packaged by conda-forge | (main, Jun  4 2025, 14:45:41)
[GCC 13.3.0] on linux
Type "copyright", "credits" or "license()" for more information.
>>> import numpy as np
>>> np.__version__
'2.2.6'
>>> |
GUI: OFF (TK) Ln: 6 Col: 4
```

- Tester la console **Idlex** par les deux commandes apparaissant sur la figure ci-dessus (noter les doubles « \_ » dans « `__version__` »).

## Désactiver l'environnement virtuel (**minfo**)

Avec certains Mac notamment, si l'activation de l'environnement virtuel Python (**minfo**) crée des problèmes pour certaines activités dans le terminal, vous pouvez :

- revenir à l'environnement virtuel (**base**) en tapant :  
`(minfo) ...> conda deactivate`
- sortir complètement de tout environnement virtuel en tapant à nouveau :  
`(base) ...> conda deactivate`

## Lancement de Jupyter Notebook

- Lancer « Anaconda Prompt (Miniconda3) » (Windows) ou un terminal (Mac, Linux).
- Se positionner dans l'environnement `minfo` :  
`(base) ...> conda activate minfo`
- Lancer **Jupyter Notebook** par :  
`(minfo) ...> jupyter notebook <chemin du dossier-racine>`

**Attention** : sous Windows, une fois **Jupyter Notebook** lancé, on ne peut pas changer d'unité logique (disque, clef USB, etc.). La spécification du chemin du « dossier-racine » qui contiendra les notebooks est donc impérative si l'on veut travailler sur une clef USB ou un disque spécifique.

Exemple : si l'on veut travailler sur le « Disque amovible » nommé « D: », le lancement se fera par :

```
(minfo) ...> jupyter notebook D:
```

ou, en spécifiant un dossier :

```
(minfo) ...> jupyter notebook "D:\mon_dossier"
```

## Postinstallation de Jupyter Notebook

La première fois que l'on lance `jupyter notebook` :

- Dans le menu `Settings`, sélectionner `Settings Editor`.
- Dans la rubrique `Notebook` mettre à `none` le paramètre `Windowing mode` (le rechercher par `ctrl+F`), à la place de la valeur `full` initialisée par l'installation et qui crée des bogues d'affichage.

### Vérification :

- dans le fichier texte `tracker.jupyterlab-settings` se trouvant dans le dossier `<.jupyter path>/lab/user-settings/@jupyterlab/notebook-extension`, doit apparaître la ligne :  
    « `"windowingMode": "none",` ».
- Le chemin `<.jupyter path>` est celui qui se termine par `.jupyter` dans la liste affichée par la commande « `jupyter --paths` », rubrique `config`.