

Chapitre 1 : Introduction à PYTHON

1. Introduction

Python est un langage de programmation simple, lisible et largement utilisé dans les sciences et les mathématiques. Il est devenu un outil incontournable dans l'enseignement supérieur pour expérimenter, simuler et visualiser des concepts mathématiques.

2. Qu'est-ce que Python ?

- Python est un langage de programmation interprété à usage général. Cela signifie que le code est exécuté ligne par ligne sans nécessiter de compilation préalable, ce qui facilite les tests et le débogage.
- Python prend en charge plusieurs approches de conception logicielle. Il supporte notamment la programmation procédurale, orientée objet et fonctionnelle.
- Python assure une gestion automatique de la mémoire et du ramasse-miettes (garbage collection) : le programmeur n'a pas besoin de gérer manuellement la mémoire, ce qui réduit les erreurs.
- Python est extensible : Il est possible d'intégrer des modules écrits dans d'autres langages comme C ou C++.
- Python est à typage dynamique : Les variables n'ont pas besoin d'être déclarées avec un type spécifique, ce qui simplifie l'écriture du code.

Objectifs :

- Un langage simple et intuitif, aussi puissant que ses principaux concurrents.
- Open source, permettant à chacun de contribuer à son développement.
- Un code aussi lisible que l'anglais courant.
- Adapté aux tâches quotidiennes, avec des temps de développement courts.

3. Pourquoi utiliser Python ?

Python est utilisé dans plusieurs domaines :

- développement de jeux
- création d'applications web
- résolution de problèmes métiers
- développement d'outils internes en entreprise
- recherche scientifique et applications académiques

L'une des principales raisons de son succès est sa **communauté**, très diversifiée et accueillante.

4. Installation de Python

Avant d'écrire le moindre programme, il faut installer Python sur son ordinateur. Deux approches principales existent :

- **Installation classique**
 - Télécharger Python depuis le site officiel python.org.

- Installer les bibliothèques manuellement à l'aide de pip

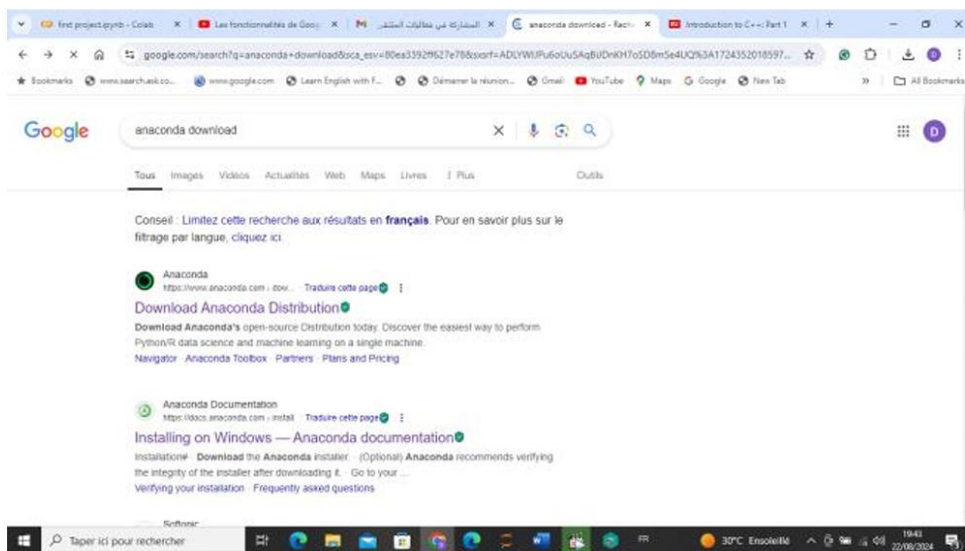
- **Installation via Anaconda**

La manière la plus simple de commencer est de télécharger la distribution gratuite **Anaconda**, fournie par Continuum Analytics, disponible sur les principales plateformes.

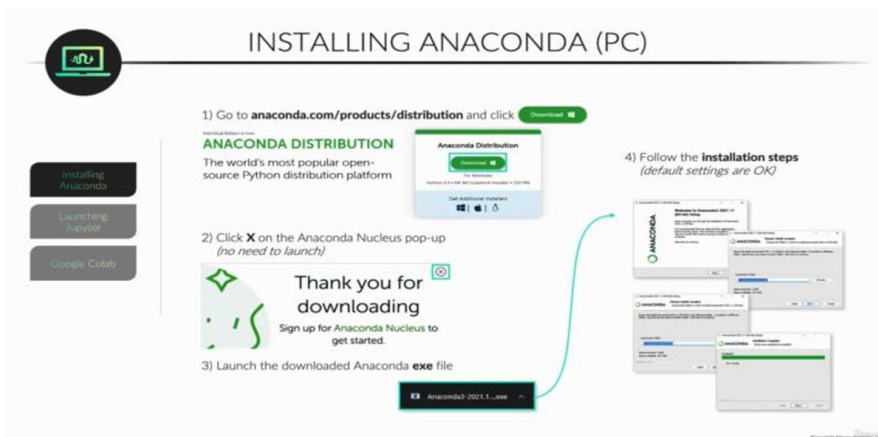
Il est préférable d'installer Anaconda car elle inclut un **gestionnaire de paquets puissant (conda)** permettant de gérer les dépendances.

Installation

1. Télécharger le programme d'installation Anaconda



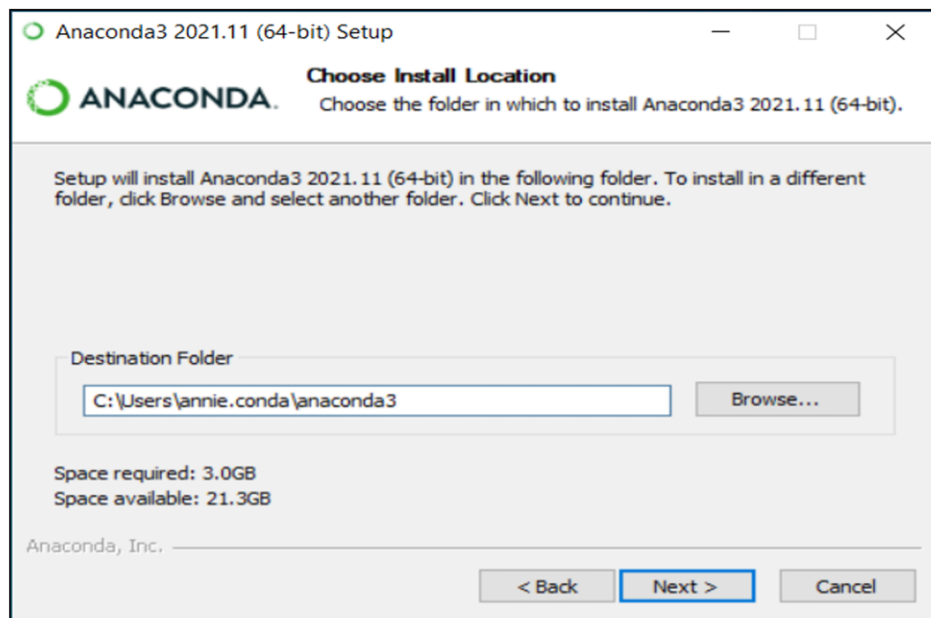
2. (Optionnel) Vérifier l'intégrité du fichier téléchargé



3. Aller dans le dossier *Téléchargements* et double-cliquer sur l'installateur
⚠ Ne pas lancer depuis *Favoris* pour éviter les erreurs de permission/
4. Cliquer sur **Next**

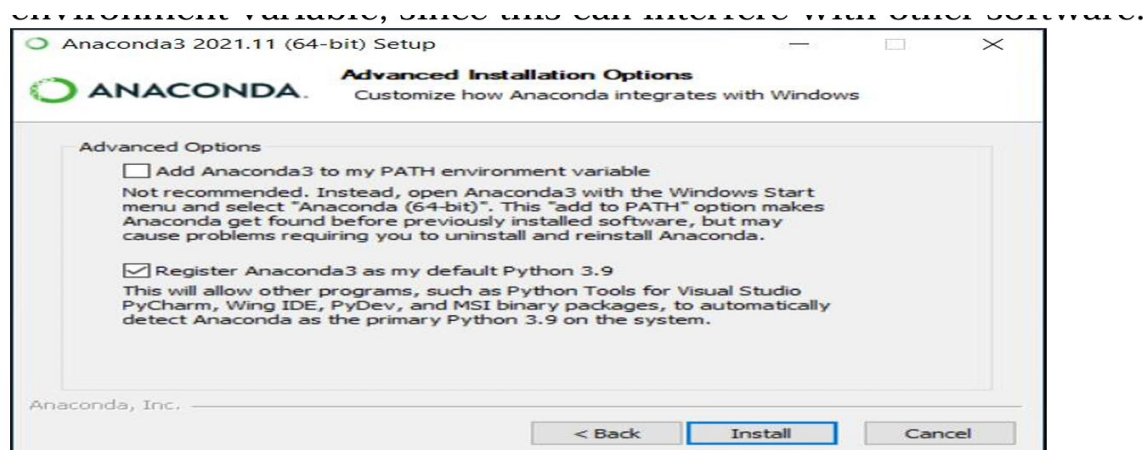
5. Lire les conditions et cliquer sur **I Agree**
6. Choisir **Just Me** (recommandé)
 - Installer uniquement pour l'utilisateur courant
 - Choisir **All Users** seulement si nécessaire (droits administrateur requis)
7. Cliquer sur **Next**
8. Choisir le dossier d'installation
 - ⚠ Éviter les chemins contenant des espaces ou caractères Unicode

⚠ Ne pas installer en mode administrateur sauf si nécessaire



9. Choisir :
 - Ajouter Anaconda au PATH
 - ou le définir comme Python par défaut

⚠ Il est déconseillé de l'ajouter au PATH (risque de conflits logiciels)



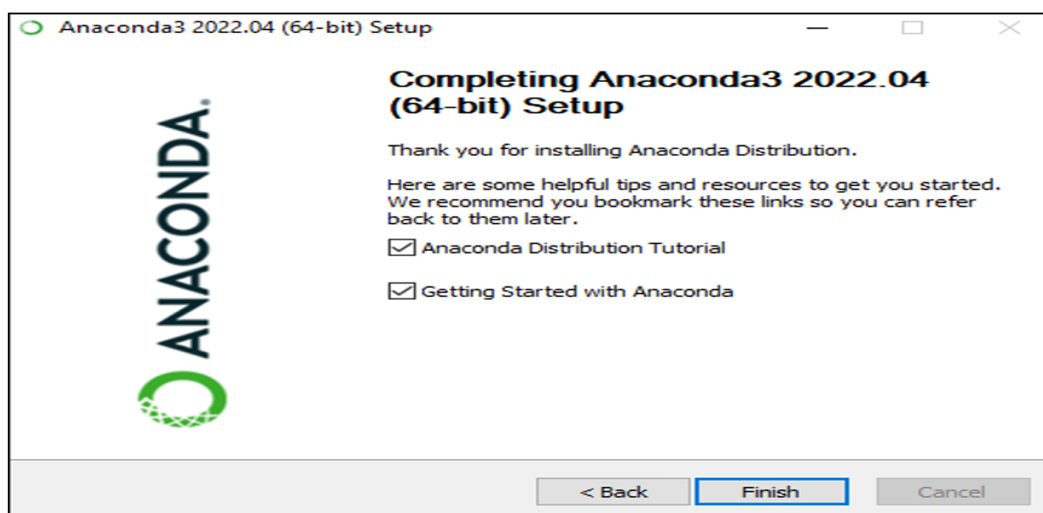
10. Cliquer sur **Install**

11. Cliquer sur **Next**

12. (Optionnel) pour en savoir plus sur le service de notebooks cloud d'Anaconda, rendez-vous sur : <https://www.anaconda.com/code-in-the-cloud>.



13. Après installation, une fenêtre « Thanks for installing Anaconda » apparaît



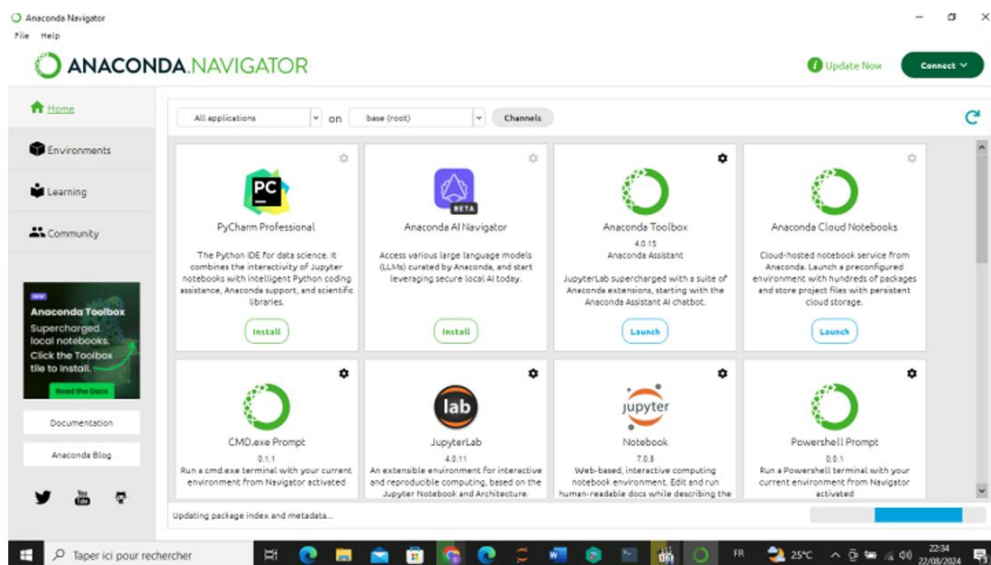
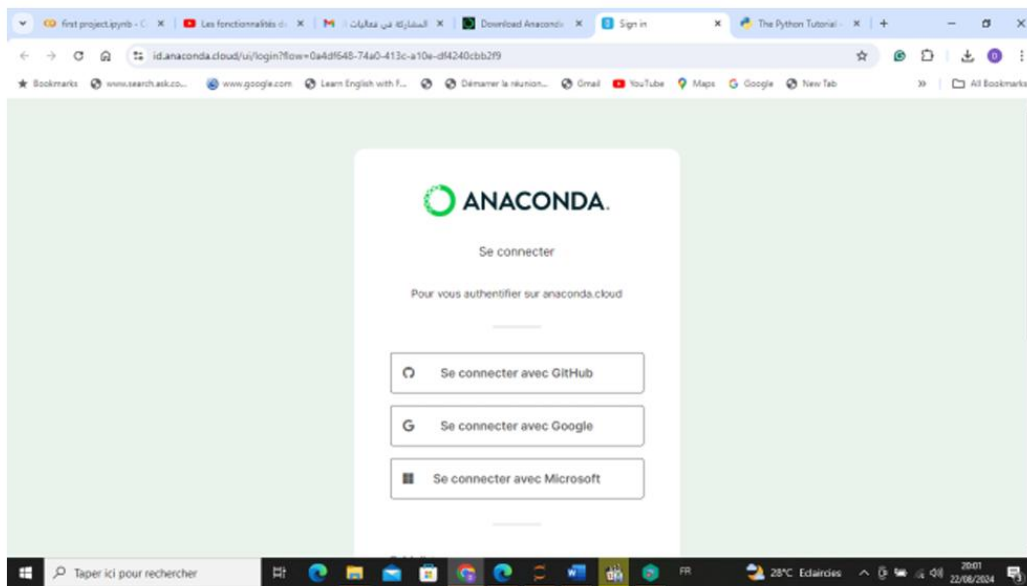
14. Optionnel :

- cocher le tutoriel Anaconda

- en apprendre plus sur Anaconda

Puis cliquer sur **Finish**

15. Vérifier l'installation



5. Configuration de l'environnement

Une fois Python installé, il faut choisir un outil pour écrire et exécuter le code :

- VS Code : léger, pratique pour le développement général.
- PyCharm : puissant, spécialement conçu pour Python.

- Jupyter Notebook : idéal pour les cours et les expériences scientifiques, car il permet de mélanger texte, équations et code.


YOUR FIRST JUPYTER NOTEBOOK

Installing Anaconda

Launching Jupyter

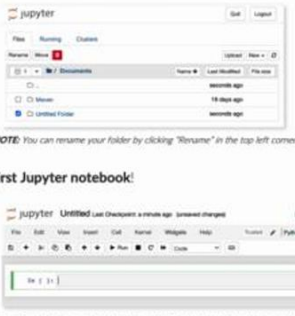
Google Colab

1) Once inside the Jupyter interface, **create a folder** to store your notebooks for the course



NOTE: You can rename your folder by clicking "Rename" in the top left corner

2) Open your new coursework folder and **launch your first Jupyter notebook!**



NOTE: You can rename your notebook by clicking on the title at the top of the screen

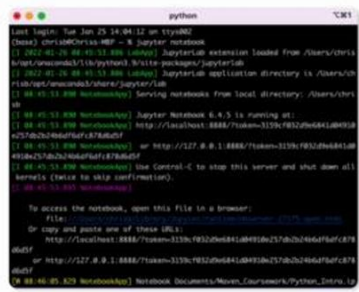
THE NOTEBOOK SERVER


Installing Anaconda

Launching Jupyter

Google Colab

NOTE: When you launch a Jupyter notebook, a terminal window may pop up as well; this is called a **notebook server**, and it powers the notebook interface





If you close the server window, your notebooks will not run!

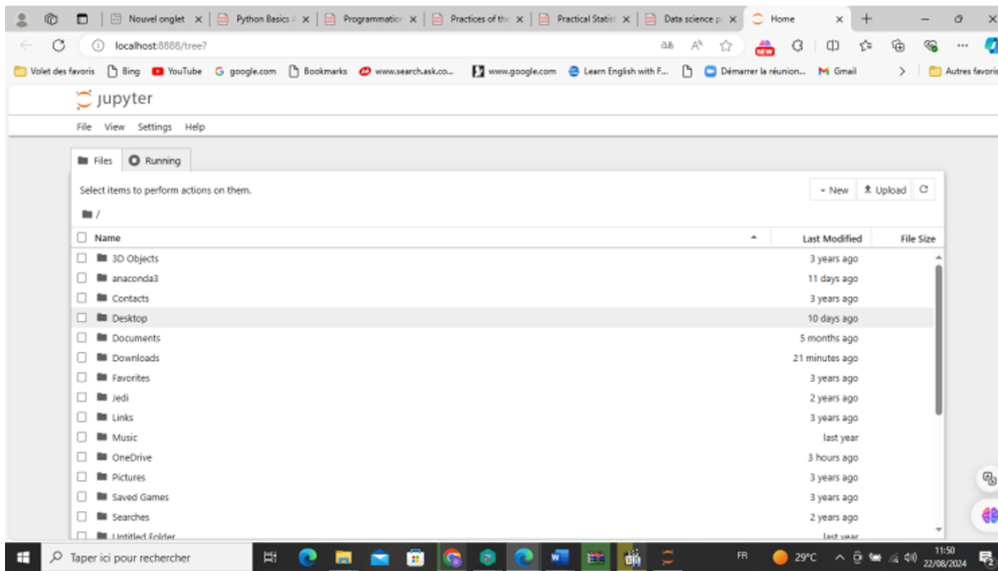
Depending on your OS, and method of launching Jupyter, one may not open. As long as you can run your notebooks, don't worry!

6. Exécution du code Python

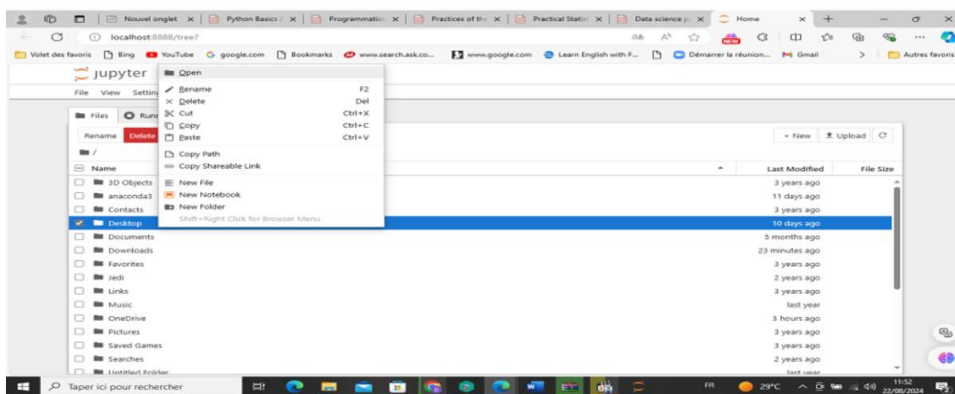
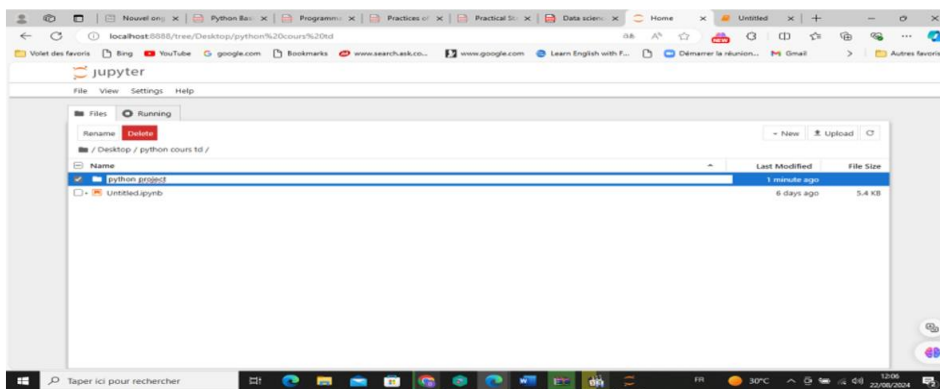
Pour exécuter un programme Python :

1. Créer un dossier de travail qui permet d'organiser les fichiers.
2. Créer un fichier .py qui Contient le code Python.
3. Écrire et exécuter le code à l'aide d'un IDE ou du terminal.

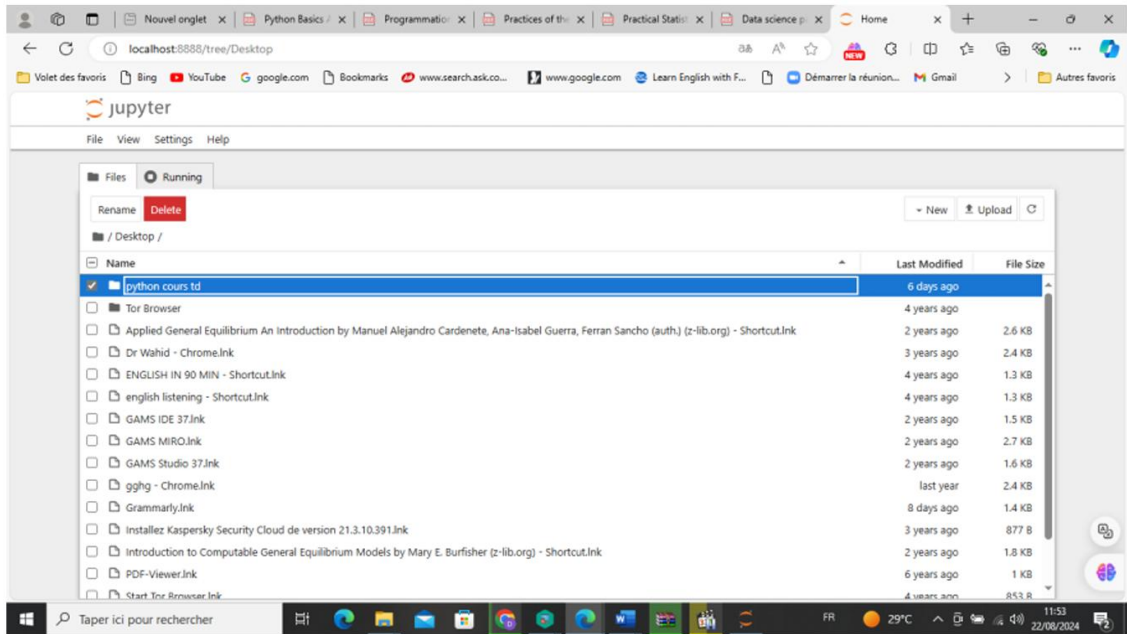
Il est aussi possible d'ajouter des notes en Markdown dans certains environnements comme Jupyter.



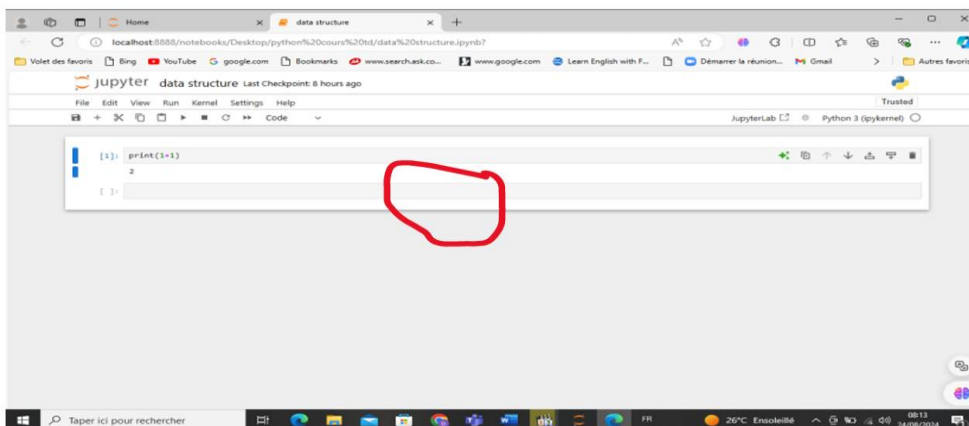
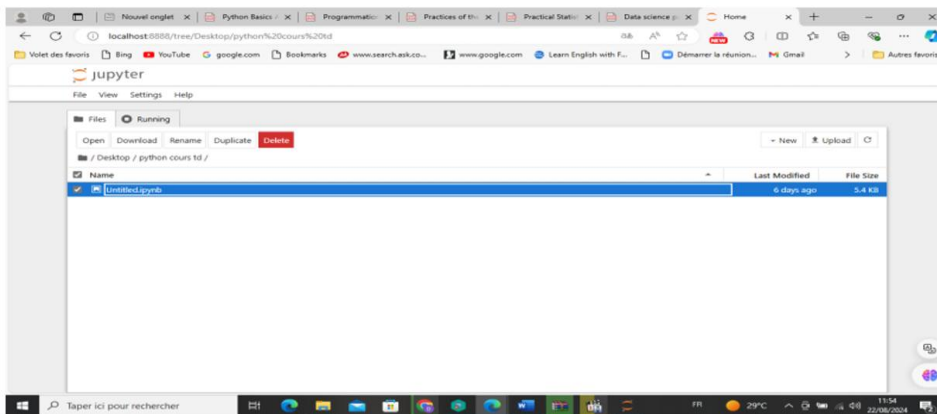
- Créer un nouveau dossier



- Choisir un dossier



- Choisir un fichier



Vous pouvez écrire :

- du code Python
- des notes (*Markdown*)

PYCHARM : PyCharm est un environnement de développement (IDE) dédié à Python, développé par JetBrains. Il facilite l'écriture du code grâce à des fonctionnalités comme l'auto-complétion, la détection des erreurs et le débogage.

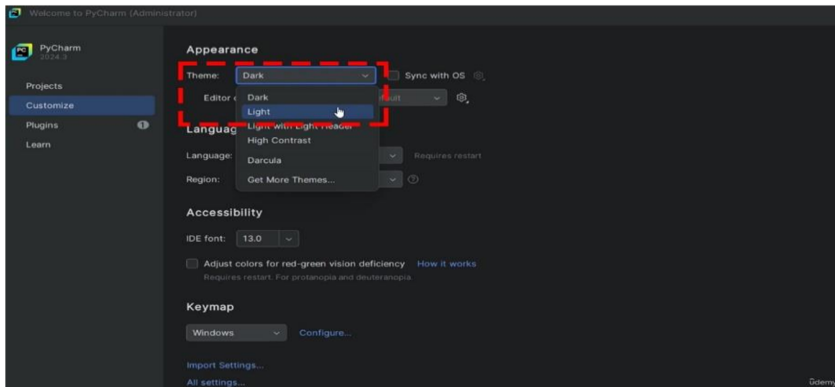
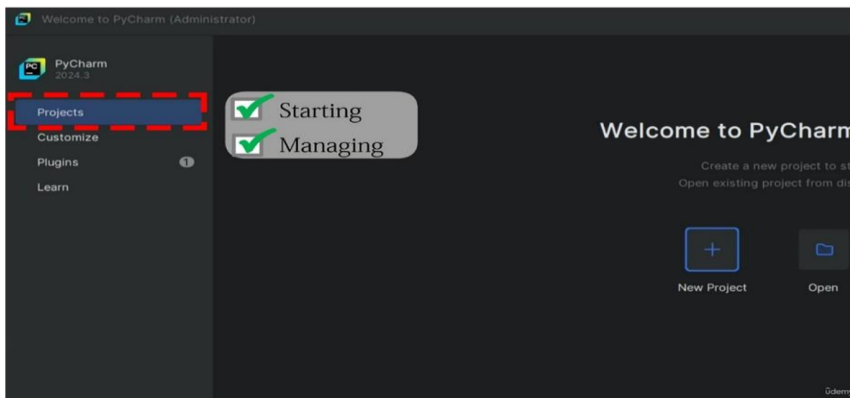
Il existe en deux versions : une version gratuite (Community) adaptée aux débutants, et une version payante (Professional) pour des projets plus avancés.

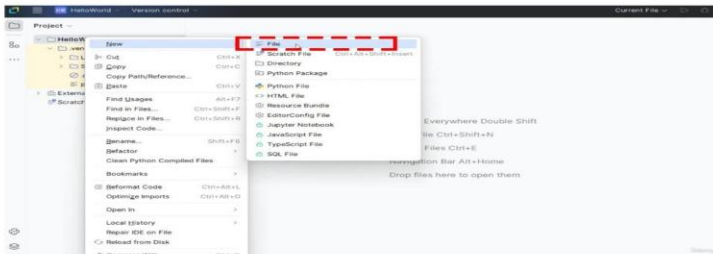
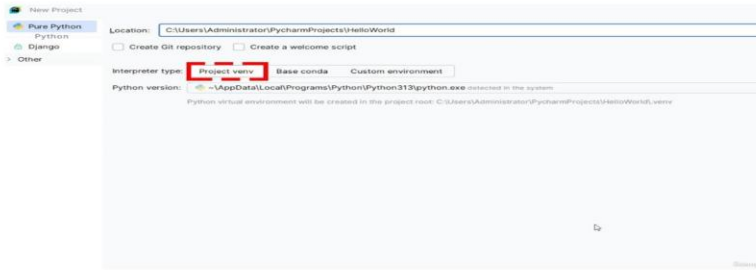
Intoduction to IDE



- ✓ Automates Tasks
- ✓ Offers Smart Suggestions
- ✓ Organizes Projects

Prepared by Imran Afzal
www.utclisolutions.com





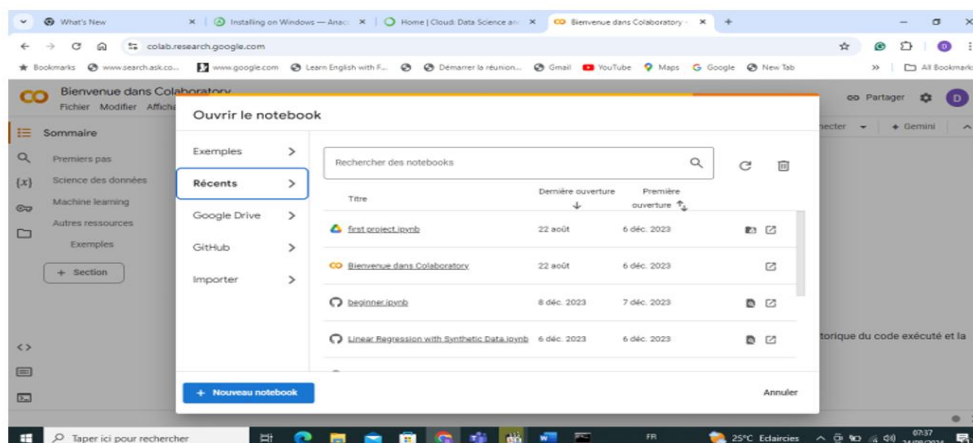
7. Outils en ligne pour pratiquer Python : Ces outils permettent de programmer sans installation :

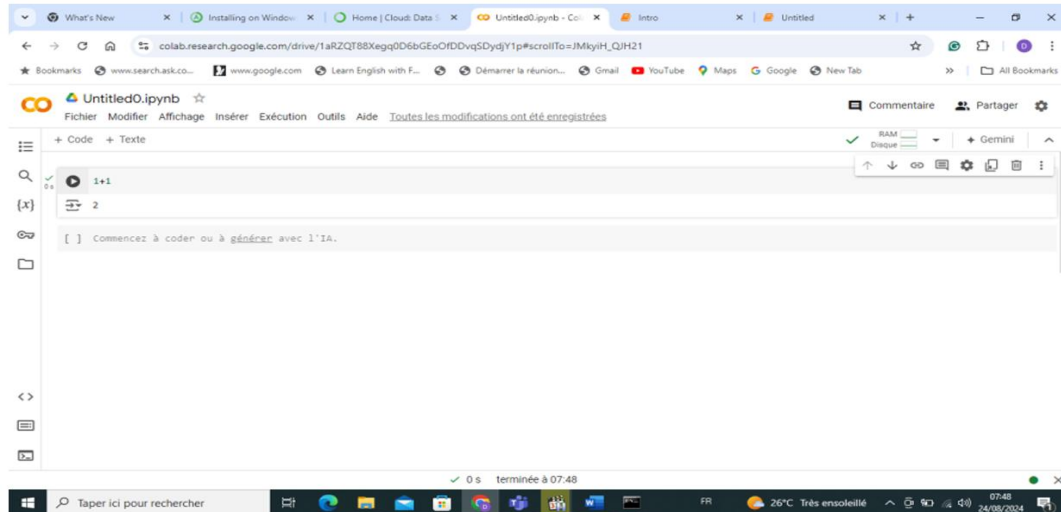
- **Google Colab**
Exécution de code Python en ligne avec sauvegarde automatique.
- **Jupyter Notebook (en ligne)**
Permet de tester rapidement du code.

Ils sont particulièrement utiles pour les débutants ou pour travailler sur différents ordinateurs.

a. Google Colab (notebooks en ligne)

- Créer un nouveau notebook





b. Jupyter Notebook

- Accéder à : [Jupyter.org/try](https://jupyter.org/try)
- Choisir **Jupyter Notebook**

8. Conclusion

Ce chapitre a permis d'introduire le langage Python en mettant en évidence ses caractéristiques principales, ses domaines d'application ainsi que les différentes étapes nécessaires à son installation et à sa prise en main. Grâce à sa simplicité, sa lisibilité et sa richesse fonctionnelle, Python constitue un outil essentiel pour l'apprentissage de la programmation et la résolution de problèmes dans divers domaines, notamment scientifiques et académiques.

Ainsi, la compréhension de ces notions de base représente une étape fondamentale avant d'aborder les concepts plus avancés du langage, qui seront développés dans les chapitres suivants.