

BM751

Outils Informatiques  
**Bibliothèques et Python**

# La bibliothèque : Aforge.NET

---

**AForge.NET** est une bibliothèque open source développée en C # conçue pour les développeurs et les chercheurs dans les domaines de la vision par ordinateur, Intelligence Artificielle, traitement de l'image, réseaux de neurones, algorithmes génétiques, la logique floue, l'apprentissage machine, robotique, etc.

**AForge.Imaging** : bibliothèque routines de traitement d'image et des filtres;

---

AForge.Vision : Bibliothèque de vision par ordinateur;

AForge.Video : ensemble de bibliothèques pour le traitement vidéo;

AForge.Neuro : réseaux de neurones de la bibliothèque de calcul;

AForge.Genetic : Bibliothèque de programmation de l'évolution;

AForge.Fuzzy : bibliothèque de calculs floue;

AForge.Robotics : bibliothèque soutien de certains kits robotiques;

AForge.MachineLearning : bibliothèque d'apprentissage machine;

# Aforge.Imaging,

---

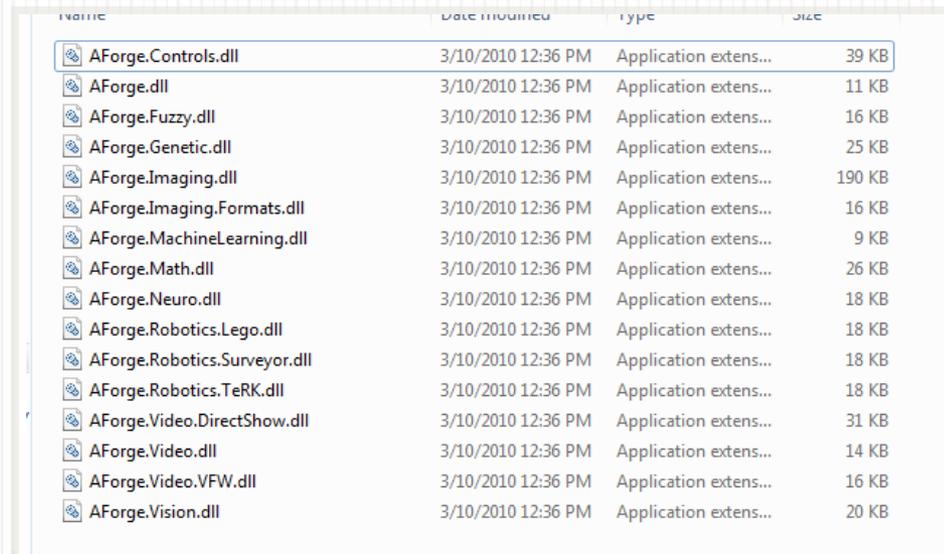
La plus part du travail que nous avons réalisé dans ce projet est basé sur la bibliothèque **Aforge.Imaging**, qui fournit différents filtres de traitement d'image, méthodes de segmentation, analyses.

# Installation : Aforge.NET

Nous devons d'abord télécharger AForge.NET

En supposant que nous utilisons un système d'exploitation Windows, téléchargez :

<http://www.aforgenet.com/framework/downloads.html>



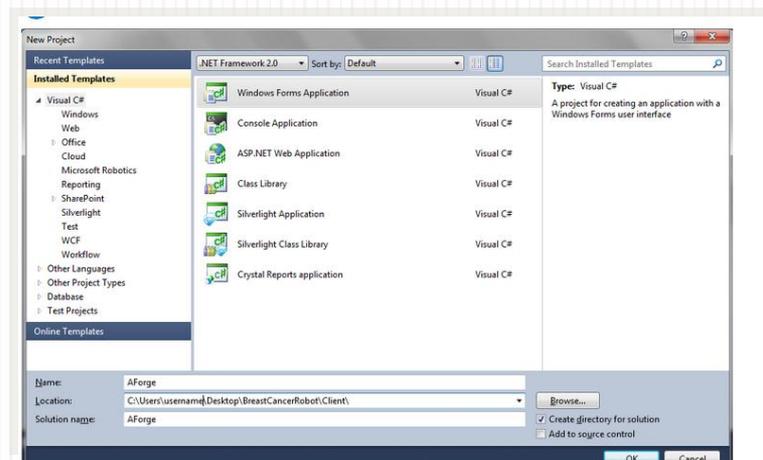
A screenshot of a Windows file explorer window displaying a list of files. The files are all application extensions (DLLs) and are sorted by name. The columns are 'Name', 'Date modifié', 'Type', and 'Size'. The first file, 'AForge.Controls.dll', is selected.

nom	Date modifié	Type	Size
AForge.Controls.dll	3/10/2010 12:36 PM	Application extens...	39 KB
AForge.dll	3/10/2010 12:36 PM	Application extens...	11 KB
AForge.Fuzzy.dll	3/10/2010 12:36 PM	Application extens...	16 KB
AForge.Genetic.dll	3/10/2010 12:36 PM	Application extens...	25 KB
AForge.Imaging.dll	3/10/2010 12:36 PM	Application extens...	190 KB
AForge.Imaging.Formats.dll	3/10/2010 12:36 PM	Application extens...	16 KB
AForge.MachineLearning.dll	3/10/2010 12:36 PM	Application extens...	9 KB
AForge.Math.dll	3/10/2010 12:36 PM	Application extens...	26 KB
AForge.Neuro.dll	3/10/2010 12:36 PM	Application extens...	18 KB
AForge.Robotics.Lego.dll	3/10/2010 12:36 PM	Application extens...	18 KB
AForge.Robotics.Surveyor.dll	3/10/2010 12:36 PM	Application extens...	18 KB
AForge.Robotics.TeRK.dll	3/10/2010 12:36 PM	Application extens...	18 KB
AForge.Video.DirectShow.dll	3/10/2010 12:36 PM	Application extens...	31 KB
AForge.Video.dll	3/10/2010 12:36 PM	Application extens...	14 KB
AForge.Video.VFW.dll	3/10/2010 12:36 PM	Application extens...	16 KB
AForge.Vision.dll	3/10/2010 12:36 PM	Application extens...	20 KB

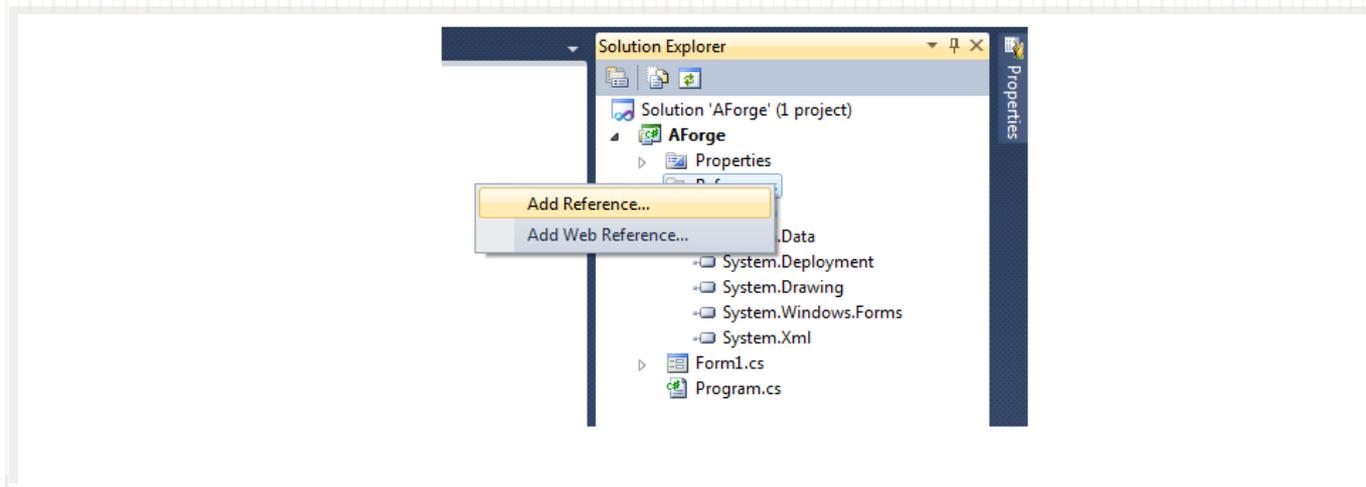
Maintenant, je suppose que Visual Studio est déjà installé sur votre ordinateur. Je vais faire une démonstration avec VS2010 (même 2008 est compatible).

Allez vers : Fichier -> Nouveau -> Projet

Sélectionnez «Application Windows Forms»



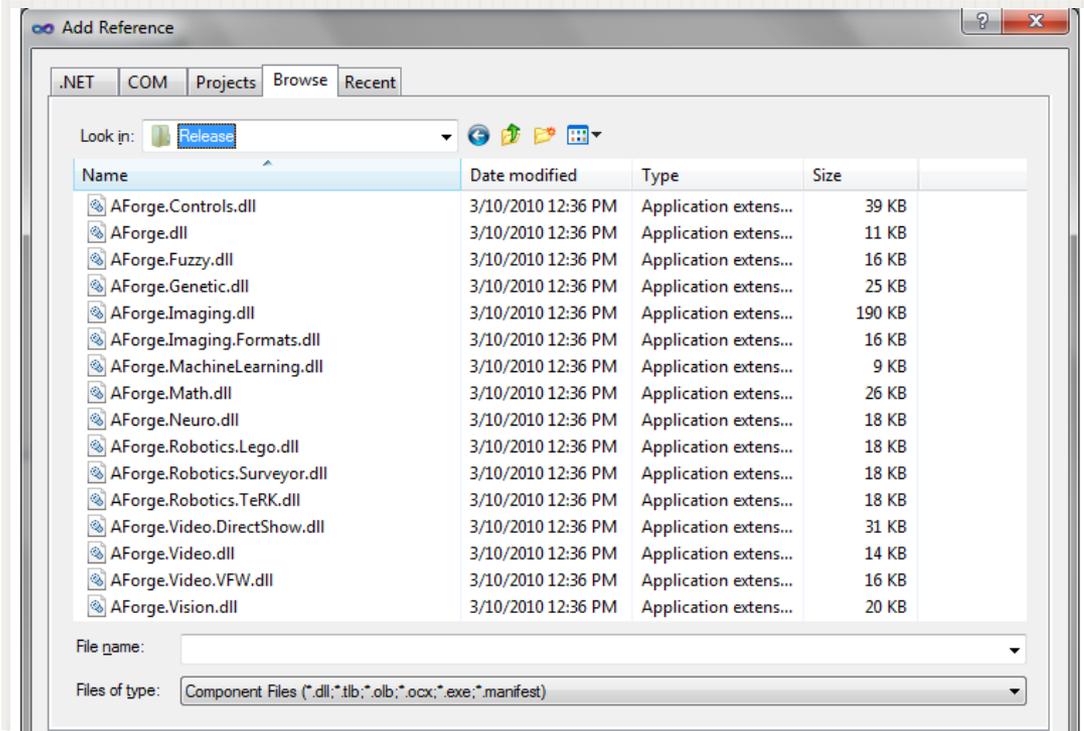
Ajoutez les bibliothèques AForge.NET comme référence. Pour ce faire, cliquez avec le bouton droit :



Ajoutez les bibliothèques nécessaires:

C: \ Program Files (x86) \ AForge.NET \ Framework \ Release - Pour OS 64 bits

C: \ Program Files \ AForge.NET \ Framework \ Release - Pour les systèmes d'exploitation 32 bits



---

Puisque notre prochaine application portera sur l'utilisation d'AForge.NET et **AForge.Imaging**, nous allons ajouter uniquement les binaires requis (.dll) à savoir : AForge.Imaging et AForge.Imaging.Filters;

En cliquant sur "OK", nous devrions être en mesure de voir les binaires (.dll) répertoriés dans l'explorateur de solutions.

# Déclaration

```
using AForge;  
using AForge.Imaging;  
using AForge.Imaging.Filters;  
using AForge.Math;
```

---

# Initiation

```
private Bitmap image;  
    private Bitmap image_copie;  
    private Bitmap image_gray;  
    private Bitmap image_BW;  
    private Bitmap image_edge;
```

# Ouvrir une image

---

```
if (openFileDialog1.ShowDialog() == DialogResult.OK)
{
    image = new Bitmap(openFileDialog1.FileName);
    pictureBox1.Image = image;
    //pictureBox2.Image = image;
    //image_copie = null;
}
```

# Image en NG

---

```
Grayscale filter = new Grayscale(0.2125, 0.7154, 0.0721);  
    image_gray = filter.Apply(image);  
    pictureBox2.Image = image_gray;
```

## Segmentation

```
int a = Convert.ToInt16(textBox1.Text);  
Threshold filter = new Threshold(a);  
image_BW = filter.Apply(image_gray);  
    pictureBox2.Image = image_BW;
```

```
int count = 0;
    // Parcourir tous les pixels
    for (int y = 0; y < image_BW.Height; y++)
    {
        for (int x = 0; x < image_BW.Width; x++)
        {
            // Vérifier si le pixel est blanc
            if (image_BW.GetPixel(x, y).R == 255)
            {
                count++;
            }
        }
    }
    textBox2.Text = Convert.ToString(count);
```

# Contour

---

```
image_copie = (Bitmap)pictureBox2.Image;  
    // create filter  
Edges filter = new Edges();  
    // apply the filter  
image_edge = filter.Apply(image_BW);  
pictureBox2.Image = image_edge
```

# Opérateurs Morphologiques

## Erosion

---

```
image_copie = (Bitmap)pictureBox2.Image;  
    Erosion er = new Erosion();  
    image_BW = er.Apply(image_copie);  
    pictureBox2.Image = image_BW;
```

## Dilatation

```
image_copie = (Bitmap)pictureBox2.Image;  
    Dilatation dil = new Dilatation();  
    image_BW = dil.Apply(image_copie);  
    pictureBox2.Image = image_BW;
```

# La bibliothèque EMGU

## Installation

1. Allez vers : Fichier -> Nouveau -> Projet

C:...../Test

2. Sélectionnez «Application Windows Forms»

3. Copier tous les fichiers dll (nécessaire pour votre application)

C:...../Test

4. Ajouter les références nécessaires?

---

5. Aller à l'endroit outils puis vous cherchez -> NET frameworks  
composantes

Sélectionnez la bibliothèque Emgu.CV.dll comme un **HistogrammBox**

6. Copiez tous les fichiers OpenCV dans le répertoires :

C:...../Test/bin/Debug

# Initiation

```
using Emgu.CV.Structure;  
using Emgu.CV;  
using System.Runtime.InteropServices;  
using Emgu.CV.Util;  
using Emgu.CV.UI;
```

# Declaration

```
public partial class Form1 : Form  
{  
    Image<Bgr, byte> My_image;  
    Image<Bgr, byte> My_imagecopy;  
    Image<Gray, byte> gray_image;
```

## Ouvrir

```
OpenFileDialog openFile = new OpenFileDialog();  
if (openFile.ShowDialog() == DialogResult.OK)  
{  
    My_image = new Image<Bgr, byte>(openFile.FileName);  
    My_imagecopy = My_image.Copy();  
    pictureBox1.Image = My_image.ToBitmap();  
}
```

## GrayImage

```
My_imagecopy = My_image.Copy();  
Image<Gray, byte> gray_image = My_image.Convert<Gray,  
byte>();  
pictureBox2.Image = gray_image.ToBitmap();
```

## Segmentation

```
My_imagecopy = My_image.Copy();  
Image<Gray, byte> gray_image = My_image.Convert<Gray, byte>();  
Image<Gray, Byte> binai_image = gray_image.ThresholdBinary(new  
Gray(60), new Gray(255));  
Image<Gray, Byte> binai_image = gray_image.ThresholdBinaryInv(new  
Gray(60), new Gray(255));  
Image<Gray, Byte> binai_morph = binai_image.Dilate(2);  
pictureBox2.Image = binai_morph.ToBitmap();
```

---

# Bibliothèque ITK/VTK

# ITK

ITK (**Insight Toolkit**) ou « National Library Of Medicine Insight Segmentation and Registration Toolkit » est une bibliothèque open-source destinée pour la segmentation et le recalage. Elle est implémentée en C++ elle utilise un environnement de construction connu sous le nom de **Cmake** pour contrôler le processus de compilation selon les plates formes.

ITK a été développée et testée sur différents systèmes d'exploitation, et de plates-formes, notamment Windows, Linux sur diverses architectures.

---

La boîte ITK comprend de nombreuses classes offrant la possibilité de lire et écrire plusieurs formats de données comme DICOM, Analyse, Raw... ainsi que des formats d'images : PNG, TIFF, JPEG...

Pour traiter ces données, il existe plusieurs filtres : filtres de morphologie mathématique, filtres de dérivées, ...

On retrouve également de nombreux algorithmes destinés spécialement pour la segmentation et le recalage.

---

En 1999, l'institut national de la santé a attribué six contrats de trois ans pour développer un outil *open-source*, pour la segmentation et le recalage, connue sous le nom de ITK (Insight Toolkit), le chef de projet du d'ITK était **Dr. Terry Yoo.**

En 2002, la première version publique officielle d'ITK était disponible. Par ailleurs, la bibliothèque a attribuée treize contrats à plusieurs organismes pour élargir la bibliothèque ITK.

---

ITK fournit un grand ensemble de bibliothèques qui peuvent être utilisées pour créer vos propres applications en imagerie médicale

ITK contient un ensemble complet d'exemples

Elle montre comment utiliser ITK pour réaliser vos propres projets.

# Installation ITK

---

Nous allons expliquer comment installer la bibliothèque ITK sur Visual studio (2009/2010).

Nous supposons que votre IDE est déjà installé sur votre PC.

---

**Etape 1:** Télécharger ITK à partir du site Web :

<https://itk.org/download/>

Choisissez la version qui correspond mieux à votre système (Microsoft-Windows, LUNIX). Puis décompressez votre dossier dans un répertoire appelé **InsightToolkit-5.0.0**

## Etape 2: Préparation du Cmake

CMake est un système de construction. C'est un logiciel libre (licence BSD) et multiplate-forme ;

---

**Cmake** est utilisé pour configurer et construire ITK.

CMake contrôle le processus de compilation du logiciel avec une plateforme simple et des fichiers de configuration indépendants du compilateur.

CMake génère des Makefiles ou des espaces de travail natifs à utiliser avec l'environnement de développement correspondant à votre choix.

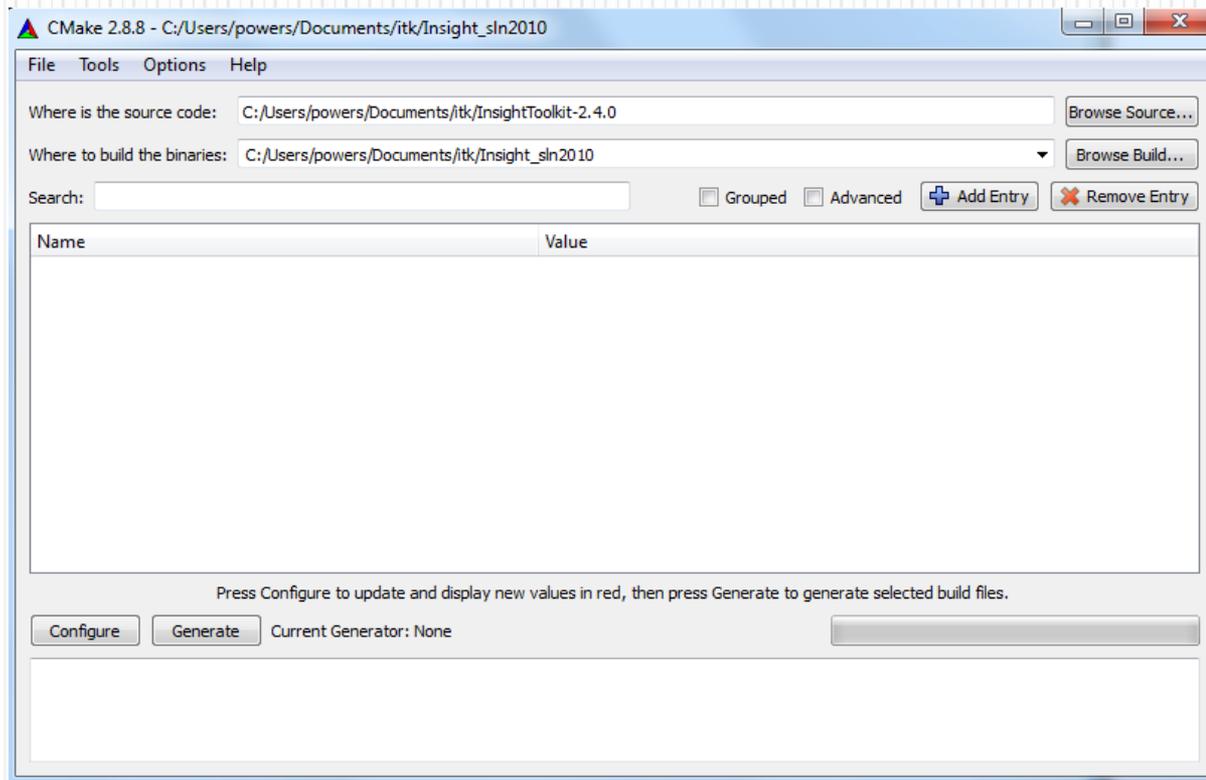
---

Les informations utilisées par CMake sont fournies dans les fichiers CMakeLists.txt qui sont présents dans chaque répertoire des sources ITK.

La configuration spécifiées par l'utilisateur incluent les chemins d'accès et la sélection des fonctionnalités logicielles.

**Télécharger et installer Cmake à partir de**  
**<https://cmake.org/download/>**

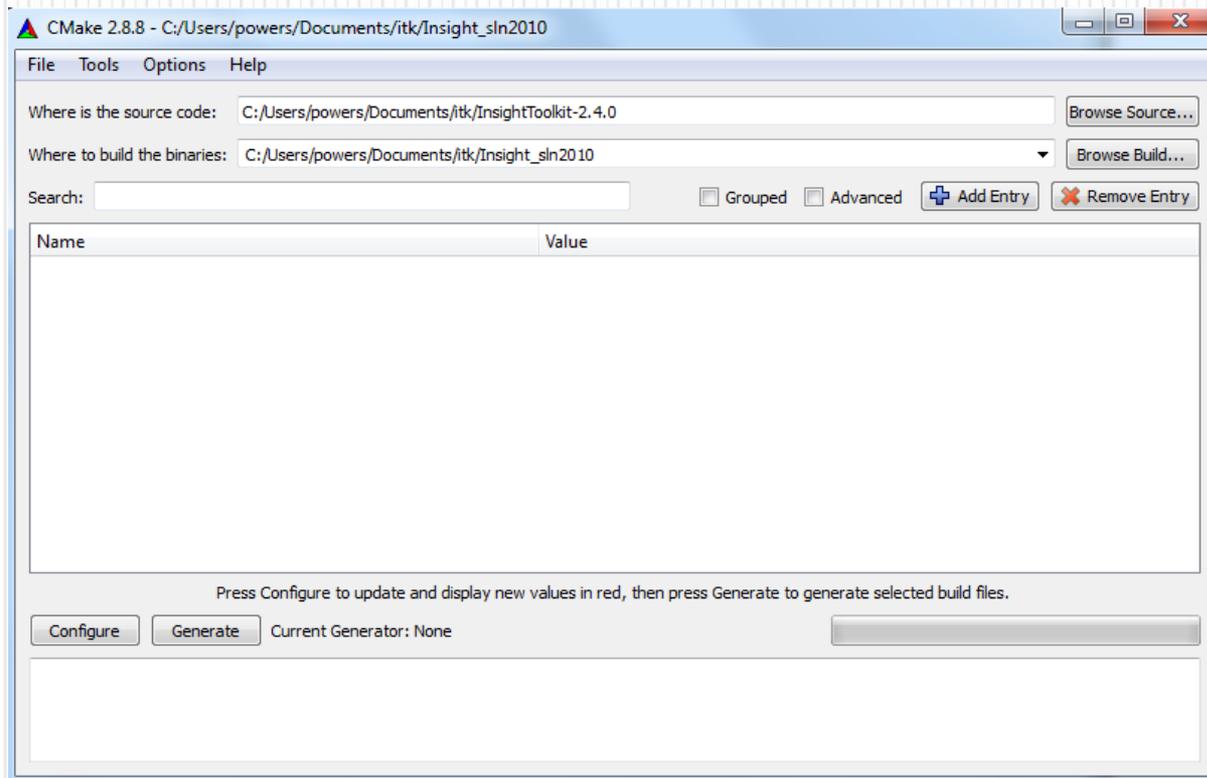
Lancer le CMake GUI. L'interface suivante doit vous apparaître.

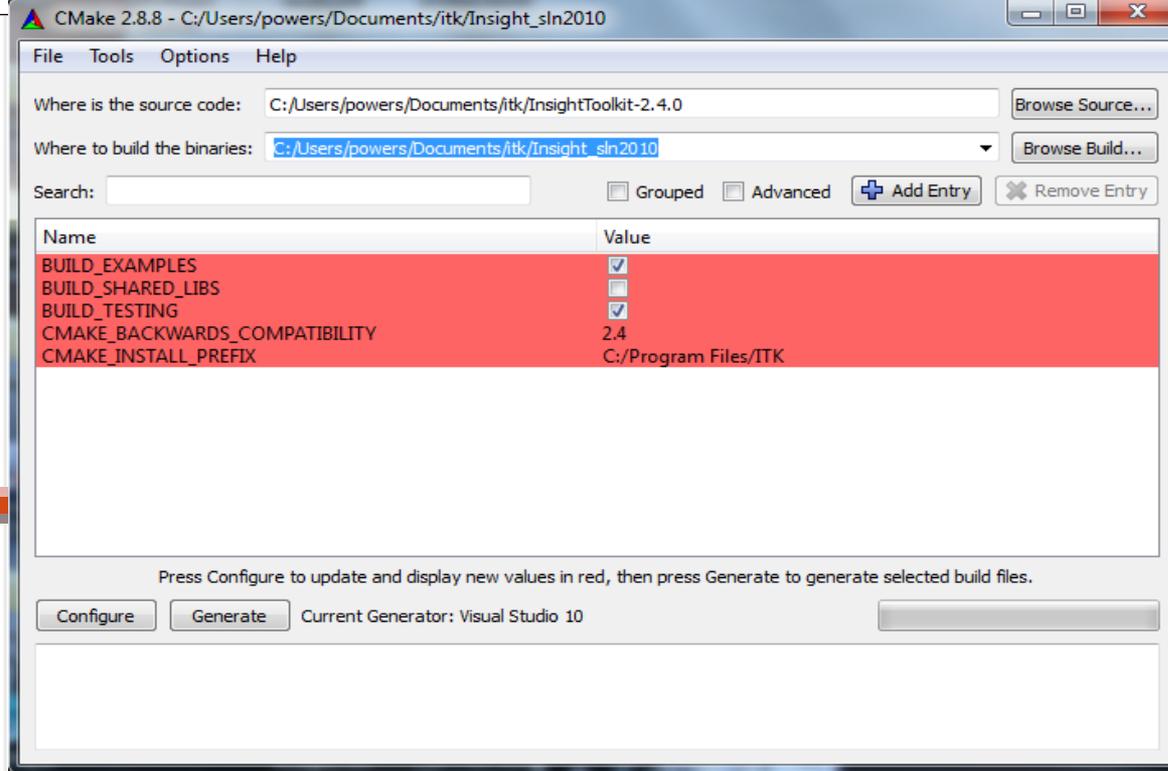


Dans notre cas le dossier du code source se trouve dans  
C:/Users/powers/Documents/itk/InsightToolkit-2.4.0

et le dossier binaire dans C:/Users/powers/Documents/itk/Insight\_sln2010.

Ensuite cliquer sur le bouton « Configure ». Lorsque la configuration est finie,  
cliquer sur Ok pour générer le projet Visual Studio.





Une fois la génération finie, aller dans le répertoire insight\_bin2010 et double cliquer sur le fichier **ITK.sln** afin d'ouvrir le projet ITK avec Visual Studio.

Générer tous les projets existant en cliquant sur « Build All ». Cela prendra 1 heure ou plus selon la puissance de votre ordinateur.

# VTK

VTK est un logiciel libre, portable et orienté objet destiné à l'infographie en 3D , au traitement et à la visualisation des images, en général et en particulier des images médicales.

VTK fournit une variété de représentation de données y compris des ensembles de points organisés ou non, des polygones des images, et des volumes,.

# Installation de la bibliothèque VTK (même chose ITK)

---

Une fois la génération finie, aller dans le répertoire `insight_bin2010` et double cliquer sur le fichier **VTK.sln** afin d'ouvrir le projet ITK avec Visual Studio.

Générer tous les projets existant en cliquant sur « Build All ». Cela prendra 1 heure ou plus selon la puissance de votre ordinateur.

# Installer anaconda(python)

Python est souvent un langage de premier choix pour des développeurs qui veulent créer des applications, leurs versions mobiles, des services web, réaliser des analyses complexes de données ou déployer des algorithmes de machine learning.

La base d'utilisateurs de Python est particulièrement importante et en croissance. À chaque mise à jour, la Python Software Foundation ajoute des fonctionnalités qui répondent aux exigences actuelles telles que l'informatique distribuée et le machine learning.

# Avantages de Python

## 1. Bibliothèques étendues

Téléchargements Python avec une bibliothèque complète et *contiennent du code à des fins diverses telles que les expressions régulières, la génération de documentation, les tests unitaires, les navigateurs Web,, la manipulation d'images, etc.* Nous n'avons donc pas à écrire manuellement le code complet pour cela.

## 2. Extensible

Python peut être **étendu à d'autres langues**. Vous pouvez écrire une partie de votre code dans des langages comme C ++ ou C. Cela est pratique, en particulier dans les projets.

## 3. Intégrable

Complémentaire à l'extensibilité, Python est également intégrable. Vous pouvez mettre votre code Python dans votre code source d'un langage différent, comme C ++. Cela nous permet d'ajouter **capacités de script** à notre code dans l'autre langue.

## 4. Amélioration de la productivité

La simplicité du langage et les bibliothèques étendues rendent les programmeurs **plus productif** que les langages comme Java et C ++. En outre, le fait que vous ayez besoin d'écrire moins et de faire plus de choses.

---

## 5. Opportunités IOT

Étant donné que Python constitue la base de nouvelles plates-formes comme Raspberry, il trouve l'avenir brillant pour l'Internet des objets. C'est un moyen de connecter la langue avec le monde réel.

## 6. Simple et facile

Lorsque vous travaillez avec Java, vous devrez peut-être créer une classe à imprimer '**Bonjour le monde**'. Mais en Python, juste une instruction d'impression fera l'affaire. C'est aussi assez **Facile à apprendre, et comprendre, et code.**

## 7. Orienté objet

Les fonctions nous aident à réutiliser le code, les classes et les objets nous permettent de modéliser le monde réel. Une classe permet **encapsulation de données** et fonctionne en un seul.

## 8. Libre et open-source

Comme nous l'avons dit plus tôt, Python est **disponible gratuitement**. Mais vous pouvez non seulement *télécharger Python* gratuitement, mais vous pouvez également télécharger son code source, y apporter des modifications et même le distribuer. Il se télécharge avec une vaste collection de bibliothèques pour vous aider dans vos tâches.

## 9. Portable

Lorsque vous codez votre projet dans un langage comme C ++, vous devrez peut-être y apporter des modifications si vous souhaitez l'exécuter sur une autre plate-forme. Mais ce n'est pas la même chose avec Python. Ici, vous devez **coder une seule fois** et vous pouvez l'exécuter n'importe où.

# Avantages de Python par rapport à d'autres langues

---

## 1. Moins de codage

Presque toutes les tâches effectuées en Python nécessitent moins de codage lorsque la même tâche est effectuée dans d'autres langues. Python dispose également d'un support de bibliothèque standard génial, vous n'avez donc pas à rechercher de bibliothèques tierces pour faire votre travail. C'est la raison pour laquelle de nombreuses personnes suggèrent d'apprendre le Python aux débutants.

## 2. Abordable

Python est gratuit, donc les particuliers, les petites entreprises ou les grandes organisations peuvent tirer parti des ressources disponibles gratuitement pour créer des applications. Python est populaire et largement utilisé, il vous offre donc un meilleur support communautaire.

## 3. Python est pour tout le monde

Le code Python peut s'exécuter sur n'importe quelle machine, que ce soit Linux, Mac ou Windows. Les programmeurs doivent apprendre différents langages pour différents travaux, mais avec Python, vous pouvez créer des applications Web de manière professionnelle, effectuer une analyse des données et *apprentissage automatique*, et également créer des jeux et des visualisations puissantes. Il s'agit d'un langage de programmation polyvalent.

# Inconvénients de Python

## 1. Limitations de vitesse

Nous avons vu que le code Python est exécuté ligne par ligne (**exécution lente**).

---

## 2. Restrictions de conception

Python est **typé dynamiquement**. Cela signifie que vous n'avez pas besoin de déclarer le type de variable lors de l'écriture du code.

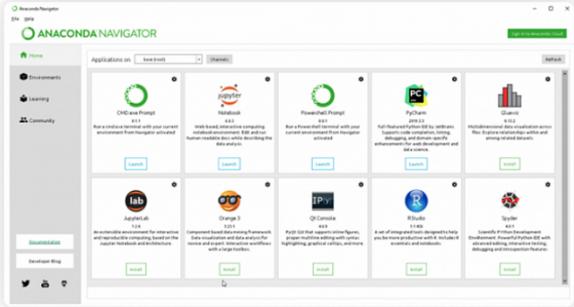
## 3. Couches d'accès aux bases de données sous-développées

Par rapport aux technologies plus largement utilisées comme **JDBC (connectivité Java DataBase)** et **ODBC (connectivité Open DataBase)**, Les couches d'accès aux bases de données de Python sont un peu sous-développées. Par conséquent, il est moins souvent appliqué dans les grandes entreprises.

# Installer anaconda(python)

Téléchargez anaconda à travers ce lien :

<https://www.anaconda.com/products/individual>



The screenshot shows the Anaconda Navigator desktop application. The interface is clean and modern, with a sidebar on the left containing navigation options like 'Home', 'Environments', 'Learning', and 'Community'. The main area displays a grid of application tiles, each with an icon and a brief description. The tiles include 'Data Panel', 'Notebook', 'PowerLab Jupyter', 'Python', 'Classical', 'AppLab', 'Orange 3', 'QI Console', 'RStudio', and 'Tiger'. Each tile has a 'Launch' button. The top of the window shows the title 'ANACONDA NAVIGATOR' and a search bar.

User interface makes learning easier

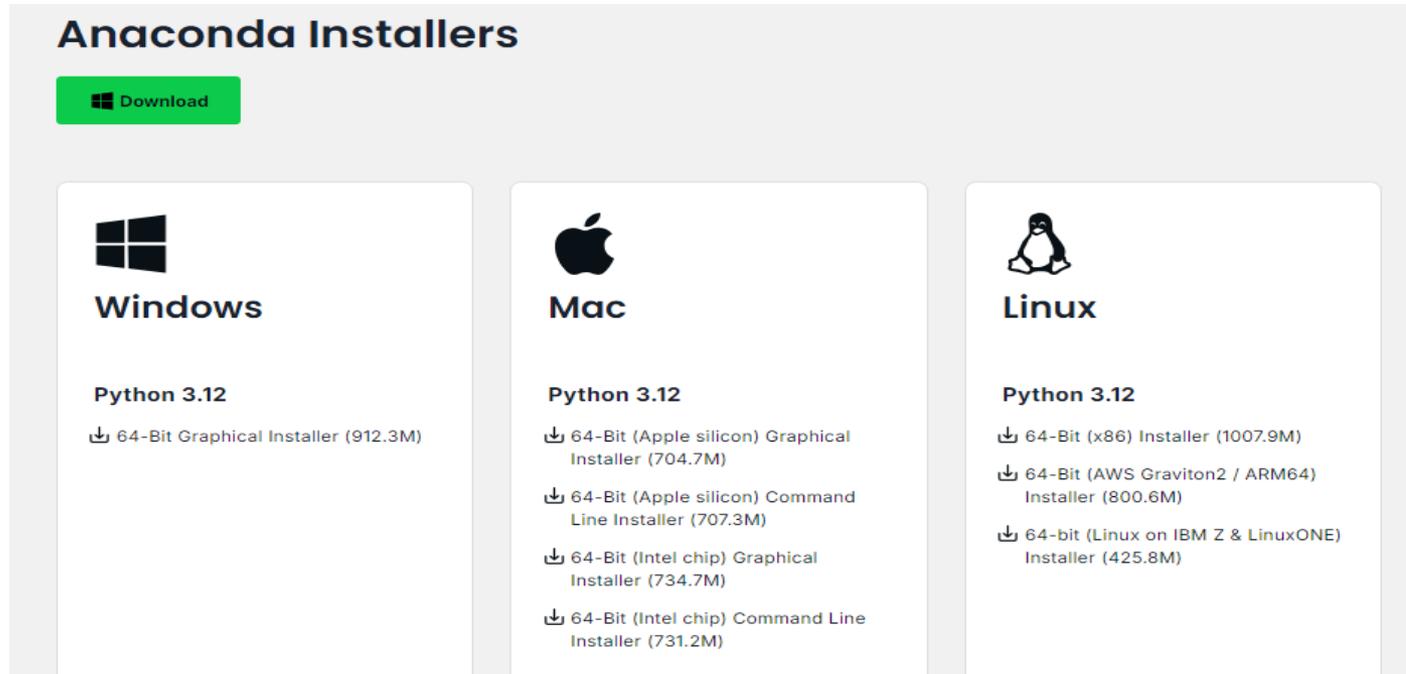
Anaconda Navigator is a desktop GUI that comes with Anaconda Individual Edition. It makes it easy to launch applications and manage packages and environments without using command-line commands.

Expedite your data science journey with easy access to training materials, documentation, and community resources including Anaconda.org.

[Install Anaconda](#)

# Telechargement de fichier

- Lien: <https://www.anaconda.com/download/success>

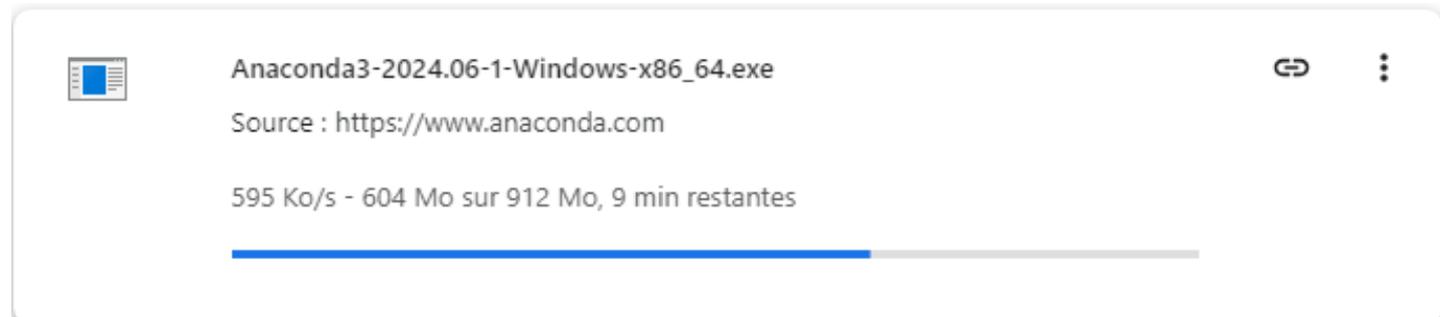


**Anaconda Installers**

[Download](#)

Operating System	Python Version	Installer Type	Size
Windows	Python 3.12	64-Bit Graphical Installer	912.3M
		64-Bit (Intel chip) Command Line Installer	731.2M
Mac	Python 3.12	64-Bit (Apple silicon) Graphical Installer	704.7M
		64-Bit (Apple silicon) Command Line Installer	707.3M
		64-Bit (Intel chip) Graphical Installer	734.7M
Linux	Python 3.12	64-Bit (x86) Installer	1007.9M
		64-Bit (AWS Graviton2 / ARM64) Installer	800.6M
		64-bit (Linux on IBM Z & LinuxONE) Installer	425.8M

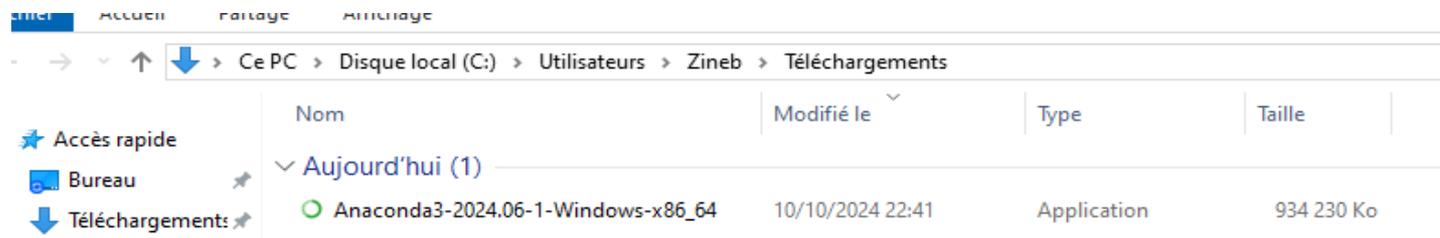
# Telechargement de fichier



Anaconda3-2024.06-1-Windows-x86\_64.exe

Source : <https://www.anaconda.com>

595 Ko/s - 604 Mo sur 912 Mo, 9 min restantes

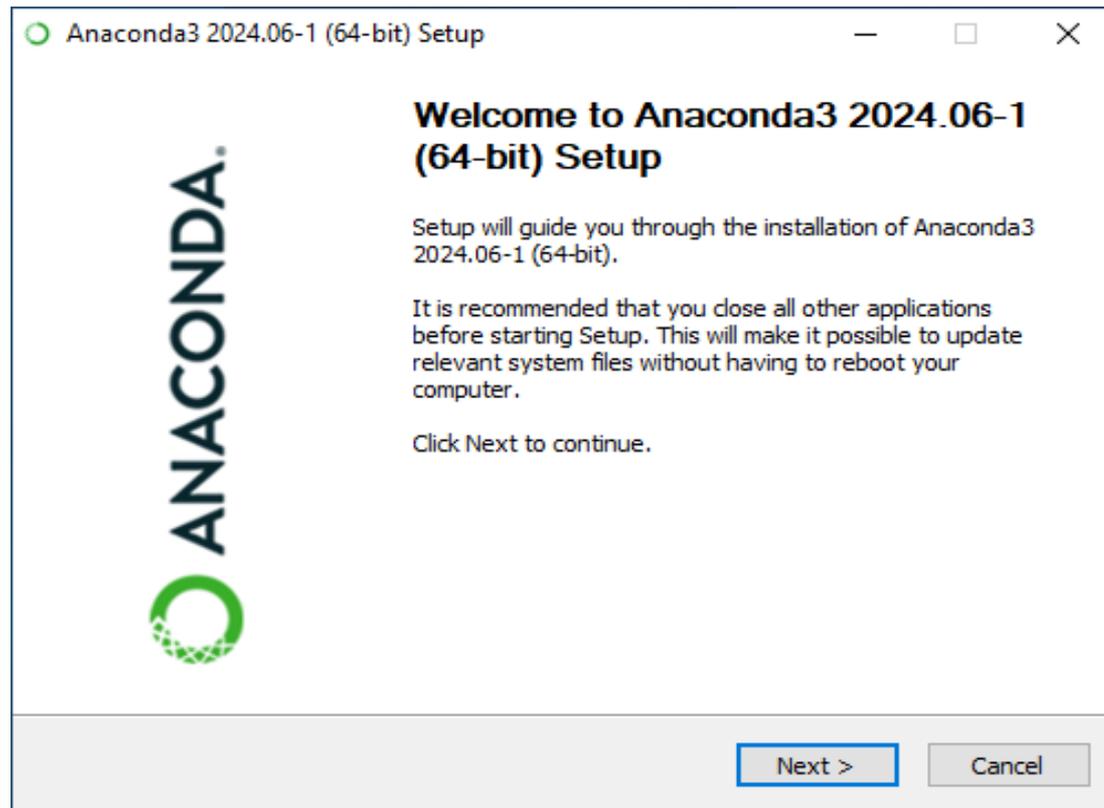


Accueil Partage Affichage

Ce PC > Disque local (C:) > Utilisateurs > Zineb > Téléchargements

Nom	Modifié le	Type	Taille
Aujourd'hui (1)			
Anaconda3-2024.06-1-Windows-x86_64	10/10/2024 22:41	Application	934 230 Ko

# Installation



# Installation

The screenshot shows a Windows-style window titled "Anaconda3 2024.06-1 (64-bit) Setup". The window contains the Anaconda logo and the text "License Agreement". Below this, it says "Please review the license terms before installing Anaconda3 2024.06-1 (64-bit)." and "Press Page Down to see the rest of the agreement." A scrollable text area contains the following text:

**ANACONDA TERMS OF SERVICE**  
Please read these Terms of Service carefully before purchasing, using, accessing, or downloading any Anaconda Offerings (the "Offerings"). These Anaconda Terms of Service ("TOS") are between Anaconda, Inc. ("Anaconda") and you ("You"), the individual or entity acquiring and/or providing access to the Offerings. These TOS govern Your access, download, installation, or use of the Anaconda Offerings, which are provided to You in combination with the terms set forth in the applicable Offering Description, and are hereby incorporated into these TOS. Except where indicated otherwise, references to "You" shall include Your Users. You hereby acknowledge that these TOS are binding, and You affirm and signify your consent to these TOS by registering to, using, installing, downloading, or accessing the Anaconda Offerings

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Anaconda3 2024.06-1 (64-bit).

Anaconda, Inc. \_\_\_\_\_

< Back   **I Agree**   Cancel

# Installation

Anaconda3 2024.06-1 (64-bit) Setup

 **ANACONDA.** **Choose Install Location**  
Choose the folder in which to install Anaconda3 2024.06-1 (64-bit).

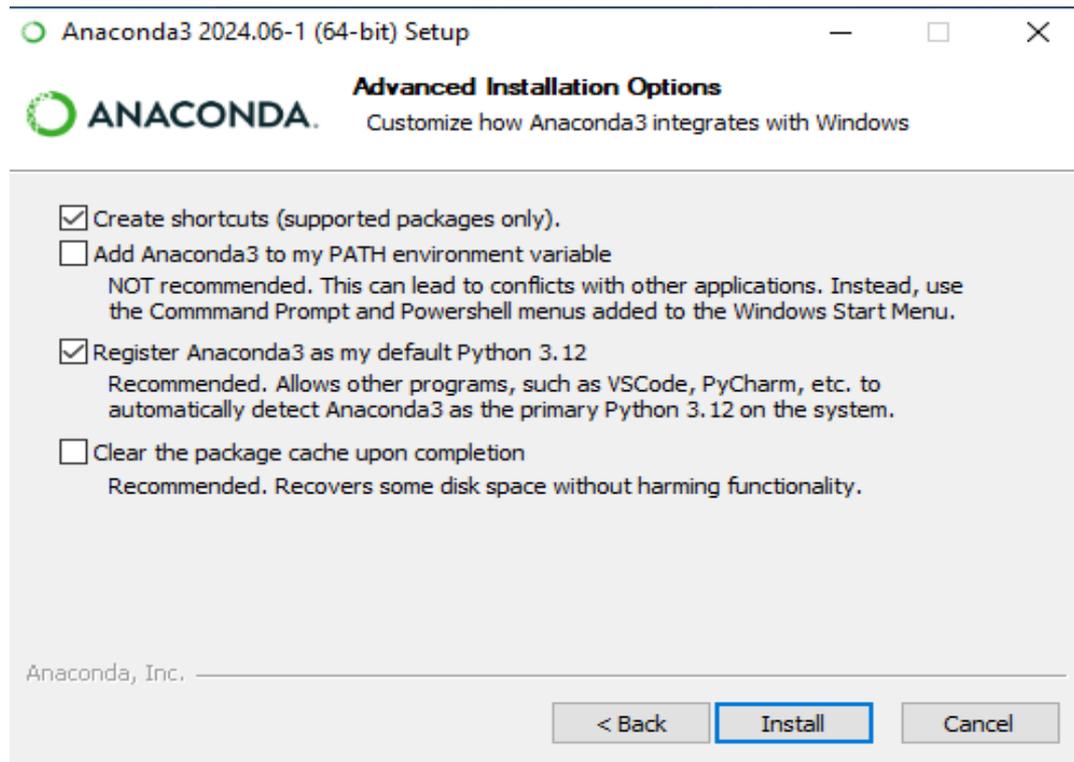
Setup will install Anaconda3 2024.06-1 (64-bit) in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder

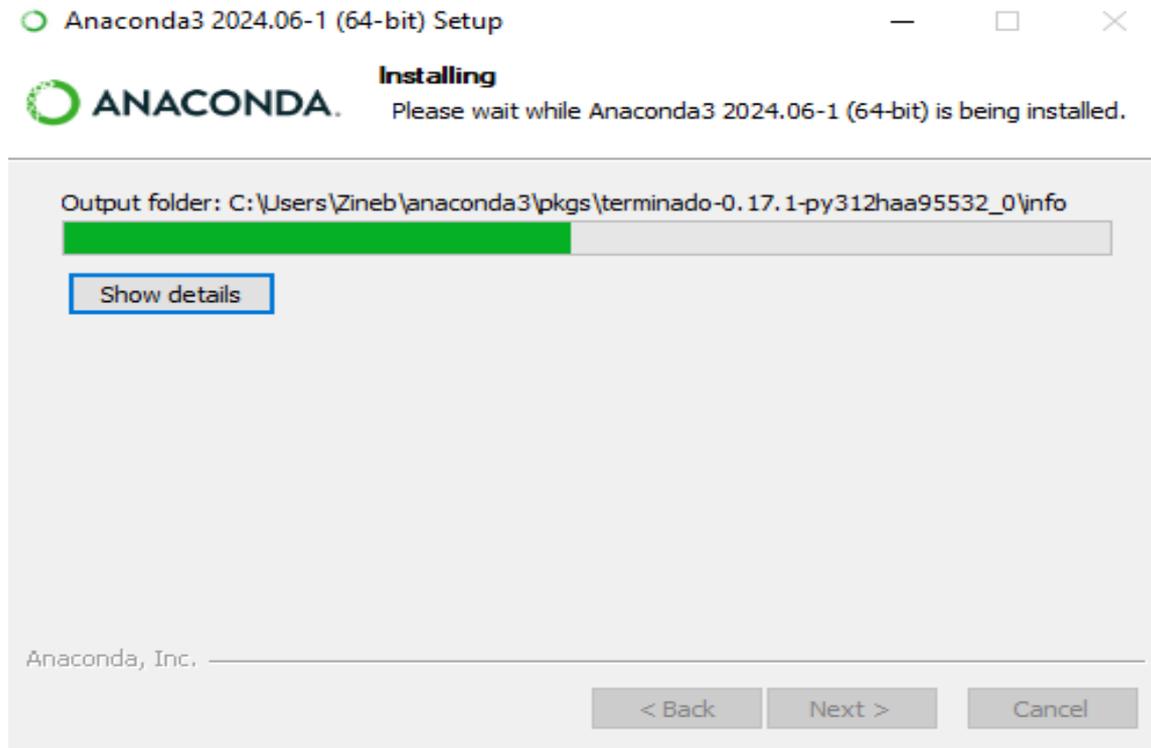
Space required: 5.0 GB  
Space available: 34.0 GB

Anaconda, Inc.

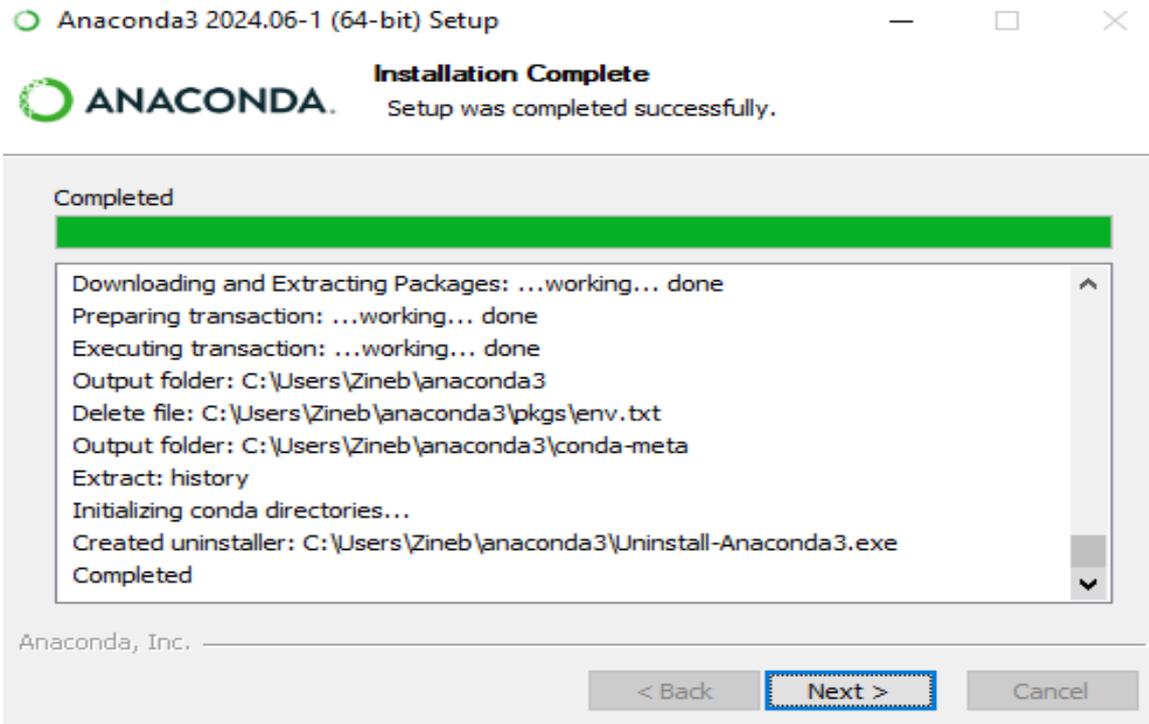
# Installation



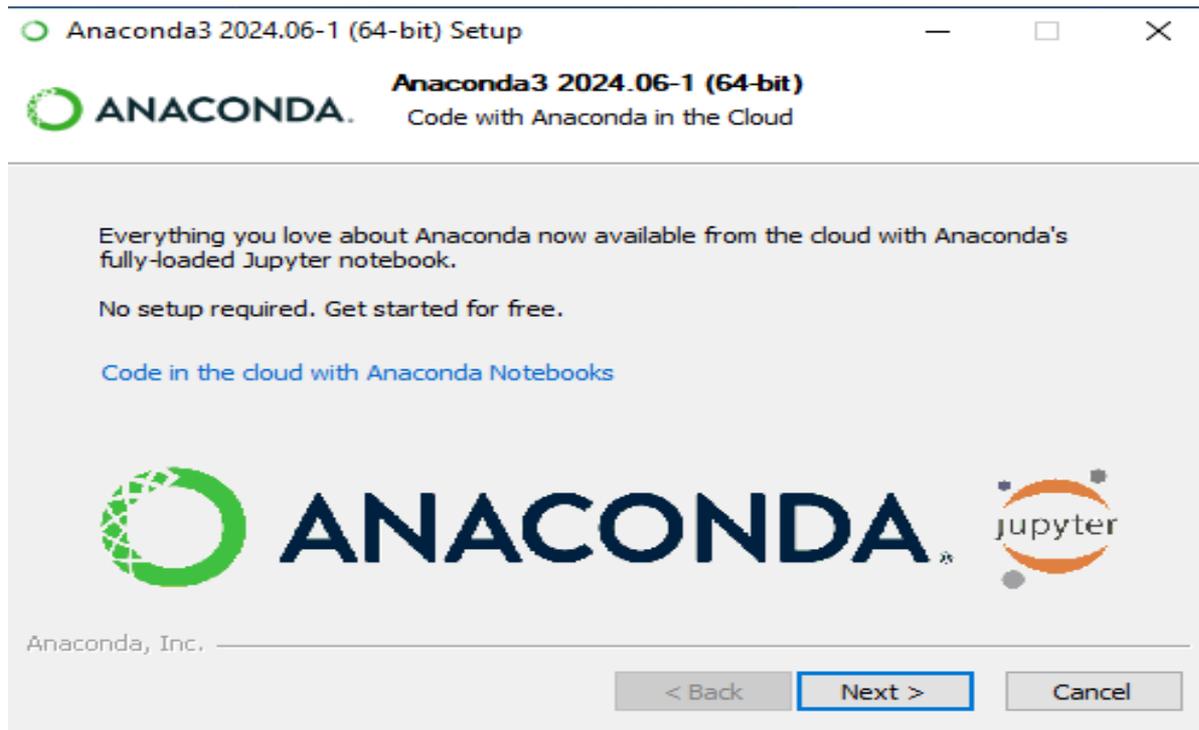
# Installation



# Installation



# Installation



# Installation

Anaconda3 2024.06-1 (64-bit) Setup



## Completing Anaconda3 2024.06-1 (64-bit) Setup

Thank you for installing Anaconda Distribution.

Here are some helpful resources to get you started. We recommend you bookmark the 'Getting Started with Anaconda Distribution' link so you can refer back to it later.

- Launch Anaconda Navigator
- Getting Started with Anaconda Distribution

< Back

Finish

Cancel

# Plateforme anaconda

The screenshot displays the Anaconda Navigator desktop application. The window title is "Anaconda Navigator" and the menu bar includes "File" and "Help". The main header features the "ANACONDA.NAVIGATOR" logo, an "Update Now" button, and a "Connect" button. A left sidebar contains navigation options: "Home", "Environments", "Learning", and "Community". Below the sidebar is a promotional banner for "Anaconda Toolbox" with a "Read the Docs" button, and links to "Documentation" and "Anaconda Blog".

The main content area shows a grid of installed applications under the "Channels" tab, filtered to "base (root)". Each application tile includes an icon, name, version, a brief description, and a "Launch" button. The applications listed are:

- Anaconda Toolbox** (4.0.15): Anaconda Assistant. JupyterLab supercharged with a suite of Anaconda extensions, starting with the Anaconda Assistant AI chatbot.
- Anaconda Cloud Notebooks**: Cloud-hosted notebook service from Anaconda. Launch a preconfigured environment with hundreds of packages and store project files with persistent cloud storage.
- CMD.exe Prompt** (0.1.1): Run a cmd.exe terminal with your current environment from Navigator activated.
- JupyterLab** (4.0.11): An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.
- Jupyter Notebook** (7.0.8): Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the
- Powershell Prompt** (0.0.1): Run a Powershell terminal with your current environment from Navigator activated.
- Qt Console** (5.5.1): PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
- Spyder** (5.5.1): Scientific PYTHON Development Environment. Powerful Python IDE with advanced editing, interactive testing,

# Plateforme anaconda

The screenshot displays the Anaconda Navigator application window. The title bar reads "Anaconda Navigator". The menu bar includes "File" and "Help". The main header features the "ANACONDA.NAVIGATOR" logo, an "Update Now" button with an information icon, and a "Connect" button with a dropdown arrow.

The left sidebar contains navigation options: "Home", "Environments", "Learning", and "Community". Below these is a promotional banner for "Anaconda Toolbox" with a "Read the Docs" button, followed by "Documentation" and "Anaconda Blog" buttons, and social media icons for Twitter, YouTube, and GitHub.

The main content area is split into two panes. The left pane, titled "Search Environments", shows a search bar and a list with one entry: "base (root)". The right pane, titled "Installed", shows a search bar, a "Channels" button, and an "Update index..." button. Below this is a table of installed packages:

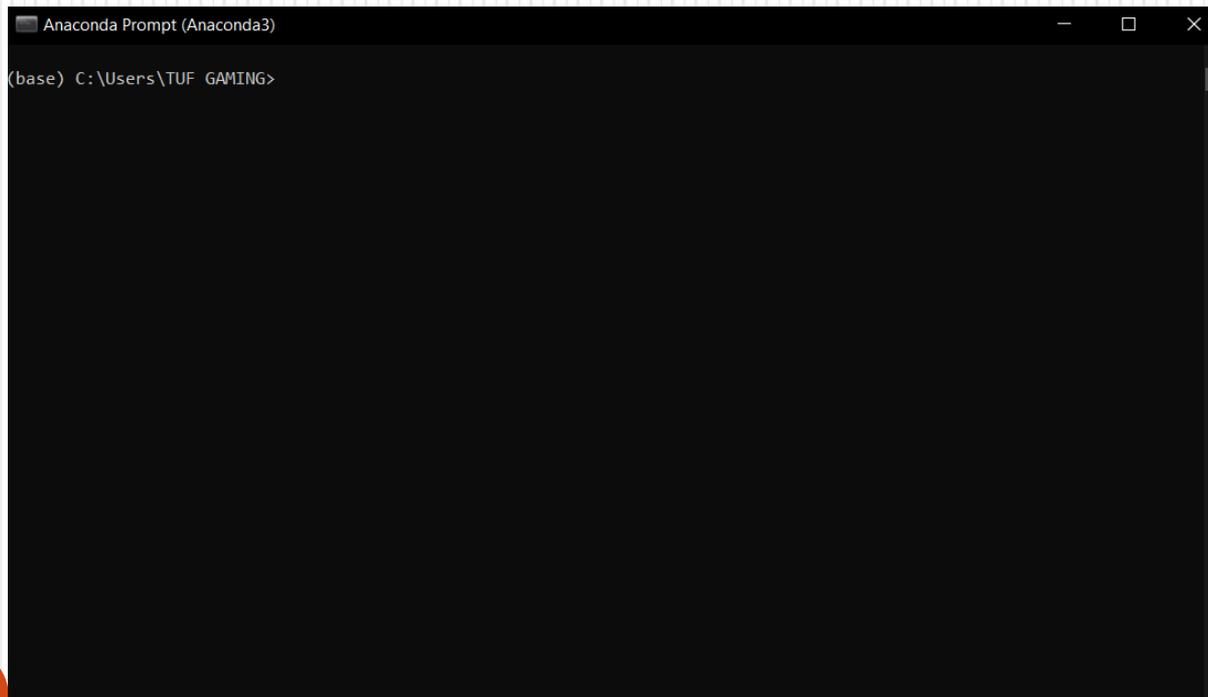
Name	Description	Version
<input checked="" type="checkbox"/> _anaconda_depends	Simplifies package management and deployment of anaconda	2024.06
<input checked="" type="checkbox"/> abseil-cpp	Abseil common libraries (c++)	2021110
<input checked="" type="checkbox"/> aiobotocore	Async client for aws services using botocore and aiohttp	2.12.3
<input checked="" type="checkbox"/> aiohttp	Async http client/server framework (asyncio)	3.9.5
<input checked="" type="checkbox"/> aiortools	Asyncio version of the standard multiprocessing module	0.7.1
<input checked="" type="checkbox"/> aiosignal	Aiosignal: a list of registered asynchronous callbacks	1.2.0
<input checked="" type="checkbox"/> alabaster	Configurable, python 2+3 compatible sphinx theme.	0.7.16
<input checked="" type="checkbox"/> altair	Altair: a declarative statistical visualization library for python	5.0.1
<input checked="" type="checkbox"/> anaconda-anon-usage		0.4.4
<input checked="" type="checkbox"/> anaconda-catalogs		0.2.0

At the bottom of the right pane, there is a status bar that says "Loading packages of C:\Users\Zineb\anaconda3..." and a "Cancel" button.

Avec l'installation d'anaconda, python va être installé automatiquement ainsi que plusieurs autres librairies. En plus un environnement de base va être aussi créé qui s'appelle « **base** ».

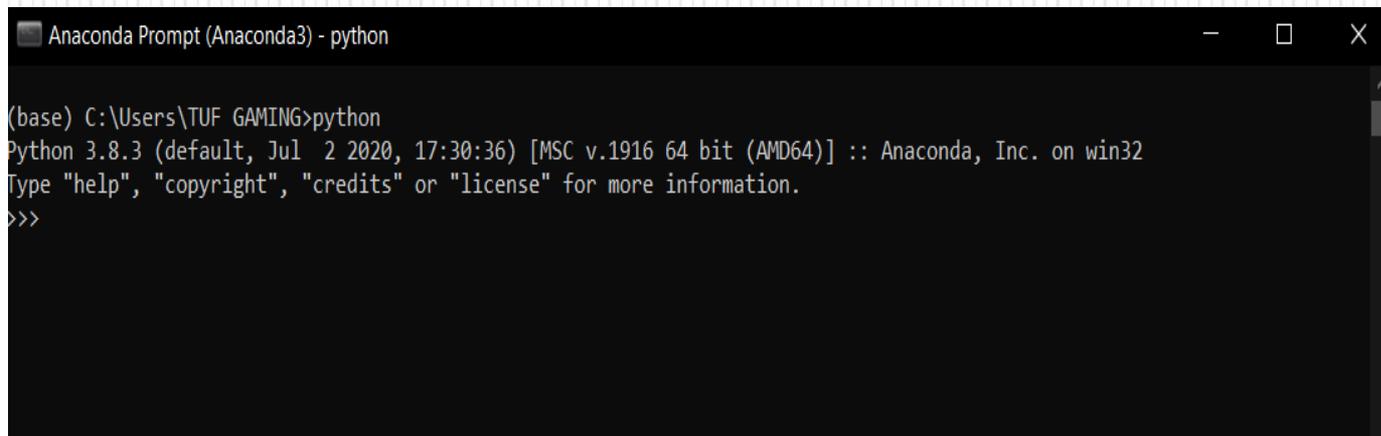
Pour vérifier tous ça, nous lançons l'application **Anaconda Prompt** : dans le menu démarrer nous cherchons « Anaconda Prompt (Anaconda3) »,

nous obtenons cette fenêtre :



Cmd (de command prompt en anglais) ou l'invite de commandes est un logiciel d'interprétation des commandes DOS

Pour vérifier par exemple la version de python, nous écrivons python comme ceci :

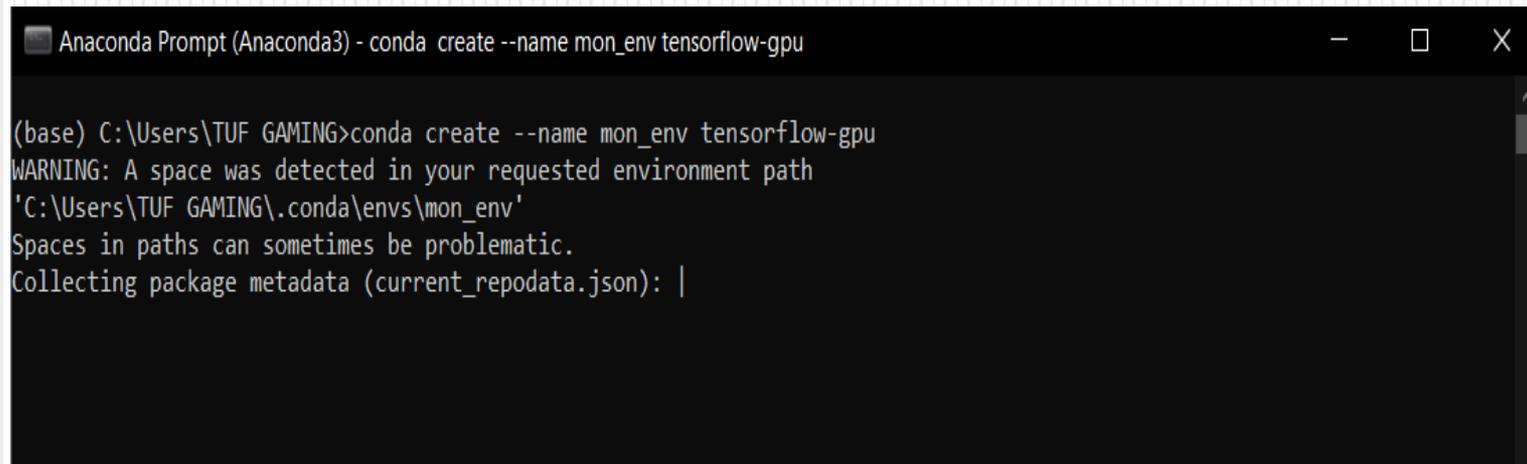


```
Anaconda Prompt (Anaconda3) - python
(base) C:\Users\TUF GAMING>python
Python 3.8.3 (default, Jul 2 2020, 17:30:36) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

# Créer un environnement virtuel

Dans le prompt, créez un nouvel environnement anaconda qui s'appelle par exemple (mon\_env) en tapant:

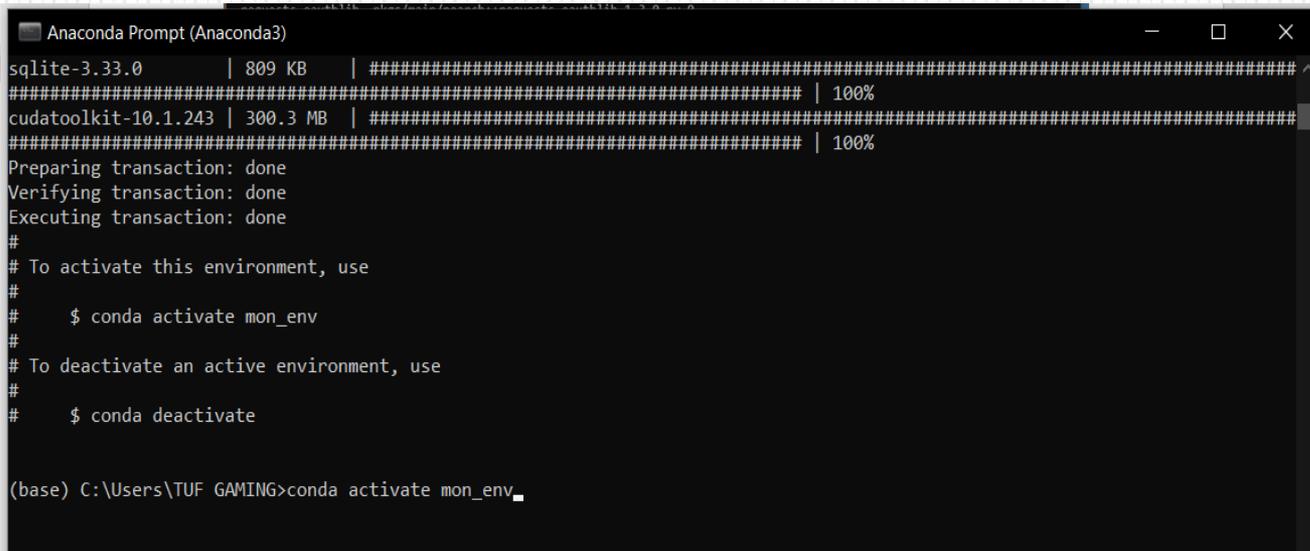
**conda create --name mon\_env tensorflow-gpu**



```
Anaconda Prompt (Anaconda3) - conda create --name mon_env tensorflow-gpu
(base) C:\Users\TUF GAMING>conda create --name mon_env tensorflow-gpu
WARNING: A space was detected in your requested environment path
'C:\Users\TUF GAMING\.conda\envs\mon_env'
Spaces in paths can sometimes be problematic.
Collecting package metadata (current_repodata.json): |
```

- Tensorflow est une bibliothèque de logiciels opensource pour le calcul numérique haute performance (par exemple le deep learning).
- pour que le téléchargement commencera (cette étape prend du temps aussi):

Quand le téléchargement se terminera, nous activons l'environnement avec :



```
Anaconda Prompt (Anaconda3)
sqlite-3.33.0 | 809 KB | ##### | 100%
cudatoolkit-10.1.243 | 300.3 MB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#   $ conda activate mon_env
#
# To deactivate an active environment, use
#
#   $ conda deactivate

(base) C:\Users\TUF GAMING>conda activate mon_env_
```

Maintenant nous avons quitté l'environnement de base et nous sommes dans notre environnement « mon\_env » :

Nous pouvons effectuer un calcul simple par exemple une opération de division comme suit :

```
Anaconda Prompt (Anaconda3) - python
# To deactivate an active environment, use
#
# $ conda deactivate

(base) C:\Users\TUF GAMING>conda activate mon_env

(mon_env) C:\Users\TUF GAMING>
(mon_env) C:\Users\TUF GAMING>python
Python 3.7.9 (default, Aug 31 2020, 17:10:11) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> a = 88
>>> b = 14
>>> c = a/b
>>> c
6.285714285714286
>>>
```

Si nous voulons quitter cet environnement nous tapons : `conda deactivate mon_env`

# Installer l'IDE (Integrated Development Environment)

Un IDE est une application logicielle utilisée par les développeurs pour créer des programmes. Les meilleurs IDE et éditeurs de code python sont : PyCharm, visual studio, sublime texte, **jupyter notebook**, .....

## Exemple :

Nous essayons d'installer **jupyter notebook** avec « conda install jupyter » dans Anaconda Prompt après avoir activé notre environnement « mon\_env »

```
Anaconda Prompt (Anaconda3) - conda install jupyter

(base) C:\Users\TUF GAMING>conda activate mon_env

(mon_env) C:\Users\TUF GAMING>conda install jupyter
Collecting package metadata (current_repodata.json): done
Solving environment: failed with initial frozen solve. Retrying with flexible solve.
Solving environment: failed with repodata from current_repodata.json, will retry with next repodata source.
Collecting package metadata (repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.8.3
  latest version: 4.9.1

Please update conda by running

  $ conda update -n base -c defaults conda

## Package Plan ##

environment location: C:\Users\TUF GAMING\.conda\envs\mon_env

added / updated specs:
- jupyter

The following packages will be downloaded:
```

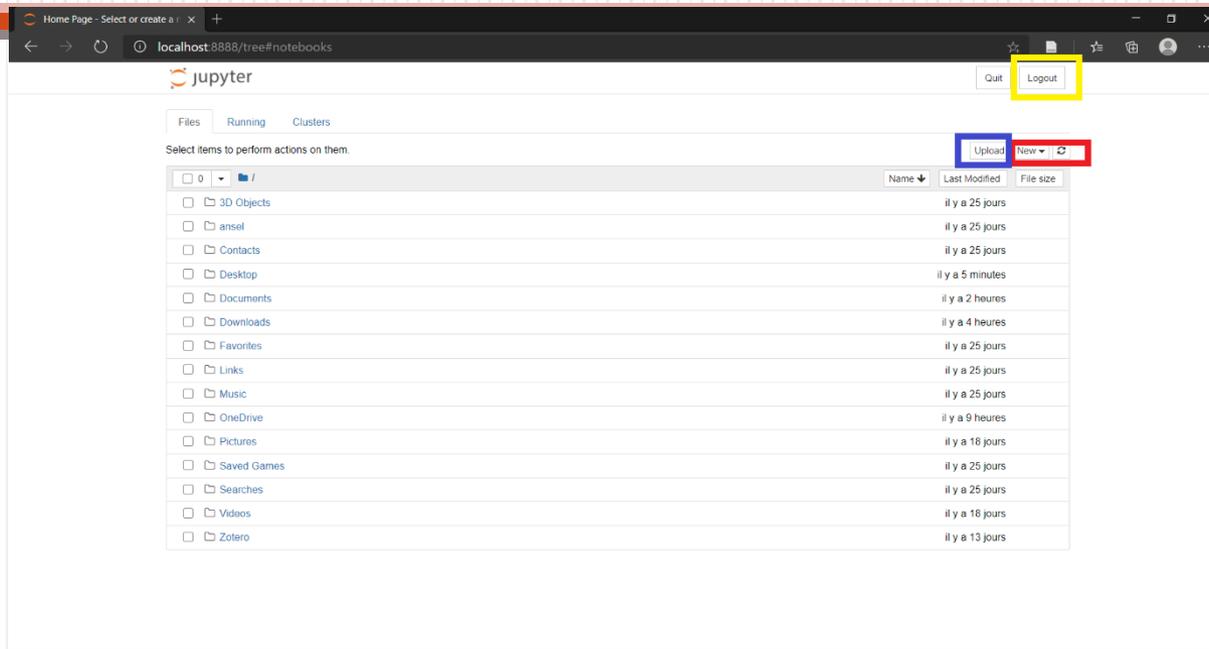
Pour lancer le jupyter nous tapons : jupyter notebook

```
Anaconda Prompt (Anaconda3) - conda install jupyter

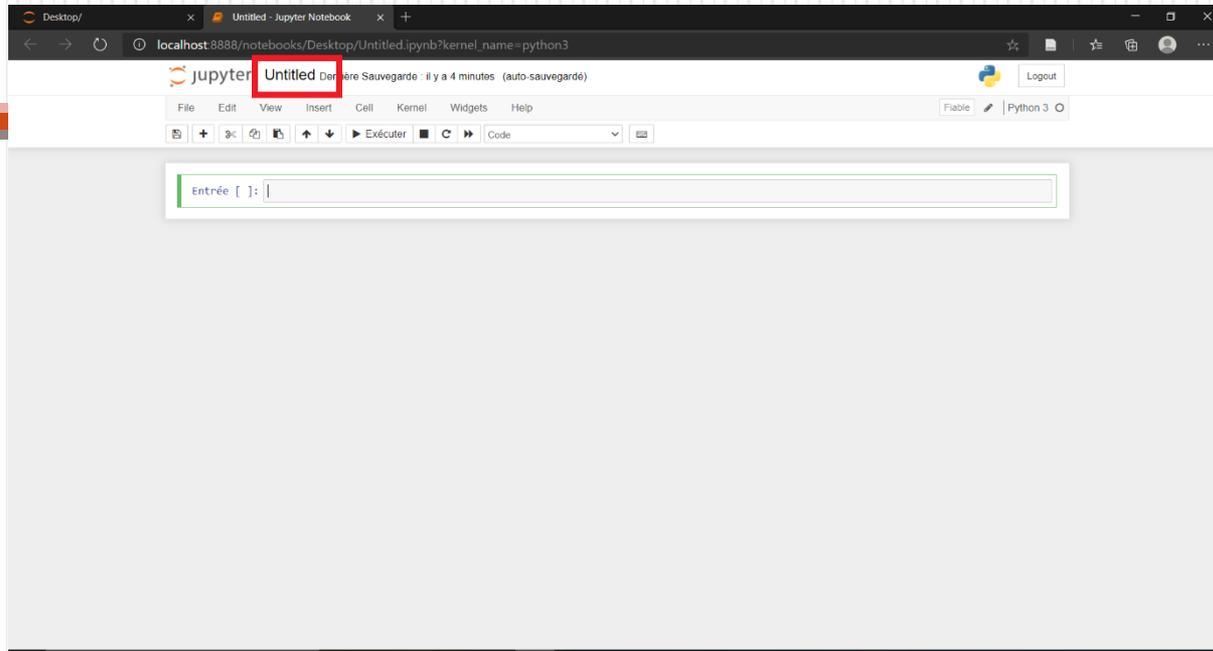
pyqt-5.9.2 | 3.3 MB | ##### | 100%
jsonschema-3.2.0 | 47 KB | ##### | 100%
qtconsole-4.7.7 | 96 KB | ##### | 100%
ipywidgets-7.5.1 | 98 KB | ##### | 100%
mistune-0.8.4 | 55 KB | ##### | 100%
ipython_genutils-0.2 | 39 KB | ##### | 100%
pandoc-2.11 | 14.4 MB | ##### | 100%
terminado-0.9.1 | 27 KB | ##### | 100%
prompt-toolkit-3.0.8 | 248 KB | ##### | 100%
bleach-3.2.1 | 112 KB | ##### | 100%
importlib_metadata-2 | 11 KB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(mon_env) C:\Users\TUF GAMING>jupyter notebook
```

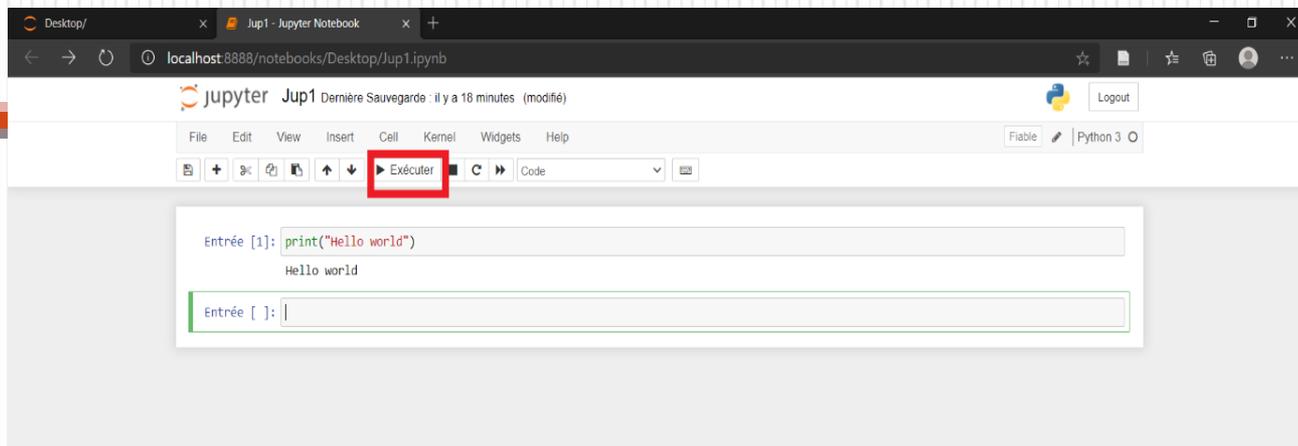
Jupyter s'ouvre avec un navigateur, nous cliquons par exemple sur Microsoft Edge et cette fenêtre s'affichera :



En rouge : pour créer un nouveau fichier, En bleu : pour ouvrir un fichier  
En jaune : pour quitter  
Nous cliquons sur « new » après sur « python3 », un nouveau navigateur jupyter s'ouvre.



Nous essayons maintenant notre premier code python avec Jupyter Notebook :



Pour installer d'autres bibliothèques dans notre environnement :

conda activate mon\_env puis :

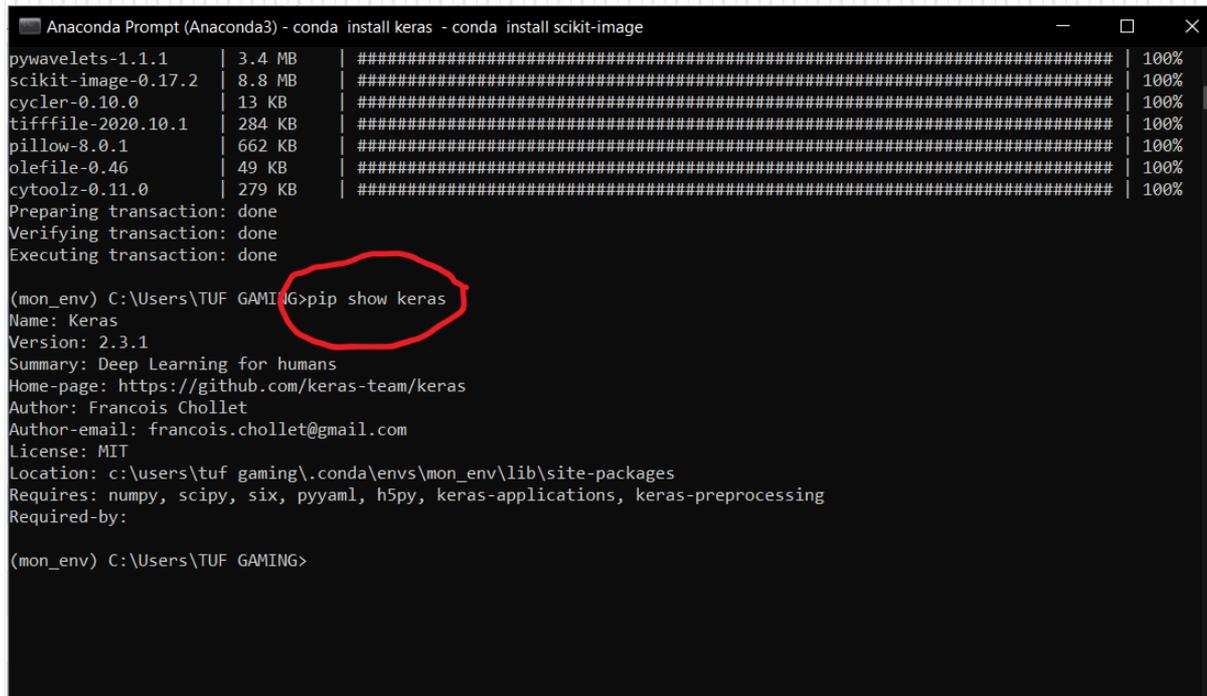
conda install keras

conda install matplotlib

**pip install opencv-python**

Pour afficher une bibliothèque dans anaconda prompt pour voir si elle est bien installée

**Exemple** : nous voulons afficher keras nous tapons « **pip show keras** »



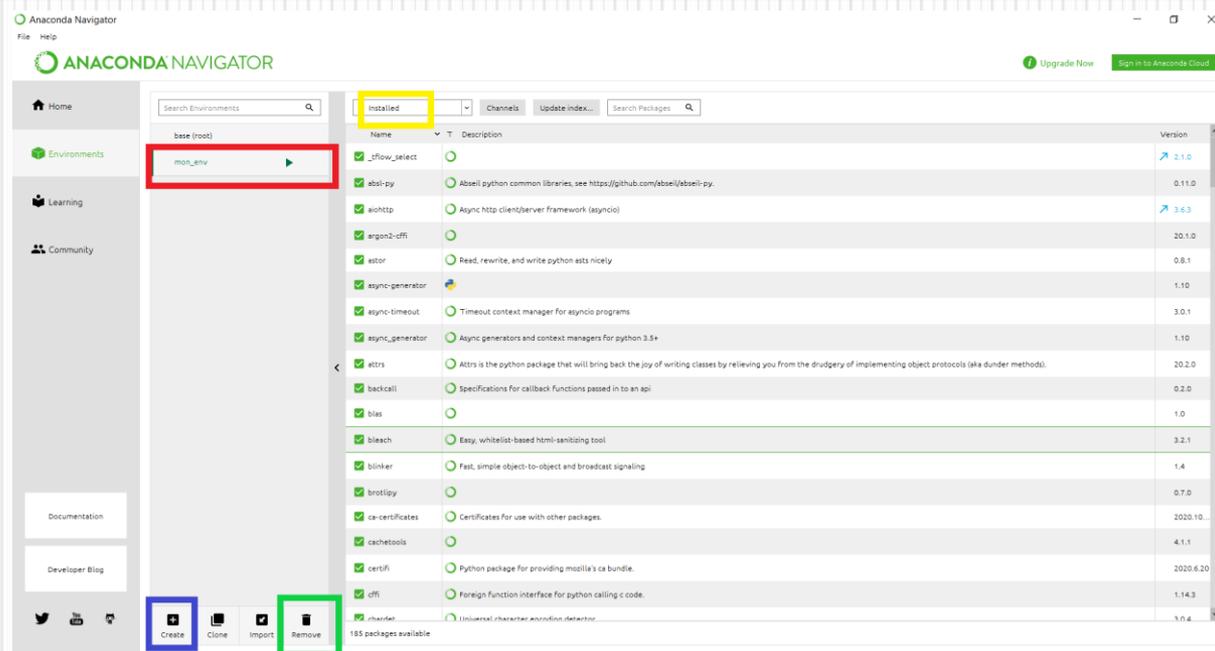
```
Anaconda Prompt (Anaconda3) - conda install keras - conda install scikit-image
pywavelets-1.1.1 | 3.4 MB | ##### | 100%
scikit-image-0.17.2 | 8.8 MB | ##### | 100%
cyclor-0.10.0 | 13 KB | ##### | 100%
tiffiffle-2020.10.1 | 284 KB | ##### | 100%
pillow-8.0.1 | 662 KB | ##### | 100%
olefile-0.46 | 49 KB | ##### | 100%
cytoolz-0.11.0 | 279 KB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(mon_env) C:\Users\TUF GAMING>pip show keras
Name: Keras
Version: 2.3.1
Summary: Deep Learning for humans
Home-page: https://github.com/keras-team/keras
Author: Francois Chollet
Author-email: francois.chollet@gmail.com
License: MIT
Location: c:\users\tuf gaming\.conda\envs\mon_env\lib\site-packages
Requires: numpy, scipy, six, pyyaml, h5py, keras-applications, keras-preprocessing
Required-by:

(mon_env) C:\Users\TUF GAMING>
```

Anaconda nous offre une application qui s'appelle Anaconda Navigator qui permet de voir nos environnements, créer un nouveau, installer directement des bibliothèques ...

Nous tapons sur le menu démarrer « Anaconda Navigator (Anaconda3) » :



# Lire et afficher une image

Lisons et affichons une image en couleur « 01\_test » avec la bibliothèque opencv :

```
import cv2
```

```
img = cv2.imread('C:/Users/TUF GAMING/Desktop/exo_opencv/01_test.tif')
```

```
cv2.imshow('Image originale', img)
```

```
cv2.waitKey(0) # attend jusqu'à ce qu'une touche soit enfoncée
```

```
cv2.destroyAllWindows() # quitte la fenêtre affichant l'image
```

Lisons deux images et les affichons dans une seule figure avec les bibliothèques opencv et matplotlib :

```
import cv2
```

```
import matplotlib.pyplot as plt
```

```
img1 = cv2.imread('C:/Users/TUF GAMING/Desktop/exo_opencv/01_test.tif')
```

```
img2 = cv2.imread('C:/Users/TUF GAMING/Desktop/exo_opencv/boule.JPG')
```

Dans l'exemple suivant, on va transformer l'image en couleur « 01\_test » en image en niveaux de gris et l'enregistrer dans le même dossier contenant l'image originale.

```
import cv2  
# read image as grey scale  
grey_img = cv2.imread('C:/Users/TUF GAMING/Desktop/exo_  
opencv/01_test.tif', cv2.IMREAD_GRAYSCALE)  
# save image  
cv2.imwrite('C:/Users/TUF GAMING/Desktop/exo_opencv/01  
_test_grey.tif', grey_img)
```

Nous pouvons accéder à la hauteur, la largeur et le nombre de canaux à partir de `img.shape`: la hauteur est à l'index 0, la largeur est à l'index 1; et nombre de canaux à l'index 2.

```
import cv2  
img = cv2.imread('C:/Users/TUF GAMING/Desktop/exo_openc  
v/01_test.tif')  
# obtenir les dimensions de l'image  
dimensions = img.shape  
# hauteur, largeur, nombre de canaux dans l'image  
height = img.shape[0]  
width = img.shape[1]  
channels = img.shape[2]
```

# Redimensionner une image :

```
import cv2
```

```
img = cv2.imread('C:/Users/T  
v/01_test.tif')
```

```
width = 300
```

```
height = 250
```

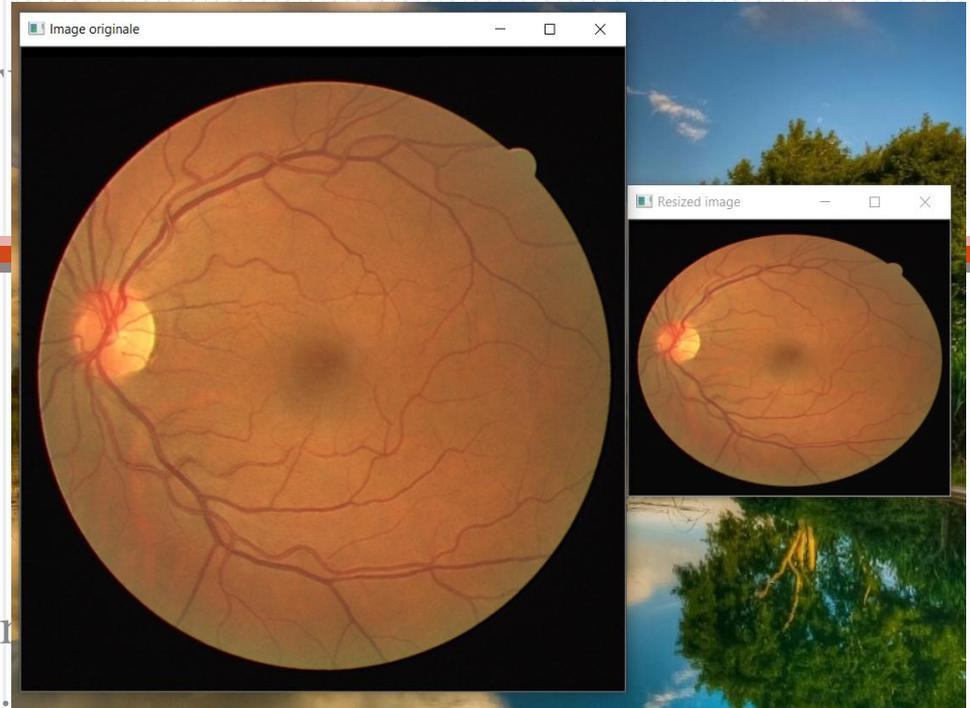
```
dim = (width, height)
```

```
resized = cv2.resize(img, dim, in
```

```
cv2.imshow('Resized image', resized)
```

```
cv2.waitKey(0)
```

```
cv2.destroyAllWindows()
```

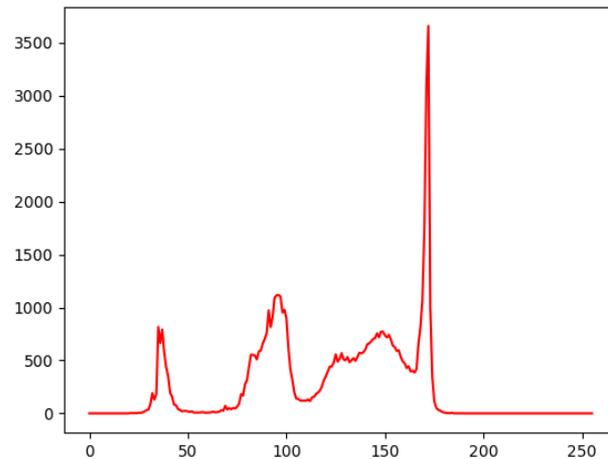


## Calcul d'histogramme :

Pour trouver l'histogramme d'une image on peut utiliser la fonction `calcHist` :

```
import cv2
import matplotlib.pyplot as plt
img = cv2.imread('C:/Users/TU...
v/boule.JPG')
hist = cv2.calcHist([img],[0],None)

plt.plot(hist)
plt.show()
```



enc

## Egalisation d'histogramme :

```
import cv2  
equ = cv2.equalizeHist(img)
```

## Seuillage binaire :

---

```
import cv2  
img = cv2.imread('C:/Users/TUF GAMING/Desktop/exo_opencv  
/boule.JPG')  
  
# perform binary thresholding on the image with s = 60  
r, threshold = cv2.threshold(img, 60, 255, cv2.THRESH_BINARY)
```

## Filtre de Gauss :

```
import cv2  
  
img = cv2.imread('C:/Users/TUF GAMING/Desktop/exo_opencv/  
01_test.tif')  
  
img_gauss = cv2.GaussianBlur(img, (7,7), 0)
```

---

## Filtre médian :

```
import cv2  
  
img = cv2.imread('C:/Users/TUF GAMING/Desktop/exo_opencv/  
foie_segment.JPG')  
  
img_med = cv2.medianBlur(img, 5)
```

# le détecteur de Canny :

```
import cv2
```

```
img = cv2.imread('C:/Users/TUF GAMING/Desktop/exo_openc  
v/foie_segment.JPG')
```

```
edge_img = cv2.Canny(img, 100, 200)
```

---

## .Erosion et dilatation :

```
img = cv2.imread('C:/Users/TUF GAMING/Desktop/exo_openc  
v/foie_segment.JPG')
```

```
mask = img.copy()
```

```
mask1 = cv2.erode(mask, None, iterations=5)
```

```
mask2 = cv2.dilate(mask, None, iterations=5)
```

# Google Colab

---

Google Colab, ou Google Collaboratory, est un environnement en ligne gratuit fourni par Google qui permet d'exécuter du code Python directement depuis votre navigateur, sans configuration locale nécessaire. Il est principalement utilisé pour les projets d'apprentissage automatique, de traitement de données, de deep learning et d'autres domaines liés à l'intelligence artificielle (IA).

# Fonctionnalités principales

- **Exécution sur GPU et TPU** : Google Colab offre l'accès gratuit à des unités de traitement graphique (GPU), ce qui permet d'accélérer le traitement des tâches complexes comme le deep learning.
- **Facilité de collaboration** : Plusieurs utilisateurs peuvent collaborer en temps réel sur un même notebook, à la manière de Google Docs.
- **Accès aux bibliothèques populaires** : Colab intègre des bibliothèques Python préinstallées telles que TensorFlow, PyTorch, Keras, OpenCV, NumPy, Pandas, et bien d'autres, facilitant ainsi les projets de machine learning et de data science.
- **Intégration avec Google Drive** : Il est possible de stocker et accéder facilement aux fichiers via Google Drive, permettant un partage de données rapide et une sauvegarde en ligne.

# Pourquoi utiliser Google Colab ?

- **Gratuit** : Contrairement à d'autres plateformes de calcul intensif, Colab est gratuit, même pour l'accès au GPU. Il existe cependant une option payante (Colab Pro) offrant de meilleures ressources (plus de mémoire, GPU de plus longue durée).
- **Facilité d'utilisation** : Vous pouvez commencer à coder sans aucune configuration particulière.
- **Partage et collaboration** : Vous pouvez facilement partager des notebooks avec d'autres personnes, qui peuvent alors exécuter le code, faire des modifications et les sauvegarder.

# Cas d'utilisation

- **Projets d'apprentissage automatique**: En particulier ceux qui nécessitent une puissance de calcul supplémentaire via le GPU.
- **Visualisation des données** : Grâce aux outils comme Matplotlib ou Seaborn, vous pouvez créer des visualisations directement dans le notebook.
- **Enseignement et formation** : De nombreux éducateurs et chercheurs utilisent Colab pour enseigner des concepts d'IA et de science des données, en partageant des notebooks interactifs.

# Comment commencer ?

- **Créer un nouveau notebook** : Depuis Google Drive ou directement depuis la page Colab, cliquez sur "Nouveau notebook".
- **Écrire du code Python** : L'interface est similaire à Jupyter Notebook. Vous pouvez ajouter des cellules de code ou des cellules de texte (Markdown).
- **Utiliser des ressources supplémentaires** : Activez un GPU ou un TPU en allant dans Exécution > Modifier le type d'exécution et sélectionnez le matériel approprié.

# Avantages et limitations

- **Avantages :**
  - ✓ Accès à des ressources de calcul puissantes sans installation.
  - ✓ Intégration facile avec Google Drive pour la gestion de fichiers.
  - ✓ Facilité de partage et collaboration en temps réel.
- **Limitations :**
  - ✓ Ressources limitées pour la version gratuite (durée limitée d'utilisation du GPU/TPU).
  - ✓ Moins flexible que les environnements locaux pour les configurations très spécifiques.

# Exemple d'application : Projet de Deep Learning avec Google Colab

Voici un exemple d'application pour entraîner un modèle de deep learning :

**1. Importer les bibliothèques :**

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

**2. Charger un dataset depuis Google Drive ou une source publique :**

```
from google.colab import drive  
drive.mount('/content/drive')
```

### 3. Prétraitement des données, construction du modèle et entraînement :

```
model = models.Sequential([
    layers.Conv2D(32, (3, 3), activation='relu', input_shape=(64, 64, 3)),
    layers.MaxPooling2D((2, 2)), layers.Flatten(),
    layers.Dense(64, activation='relu'),
    layers.Dense(10, activation='softmax')])
model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])
model.fit(train_images, train_labels, epochs=10,
validation_data=(test_images, test_labels))
```

# Conclusion :

---

Google Colab est un outil puissant et accessible pour les développeurs, les chercheurs, les étudiants et les professionnels du domaine de l'IA et du data science. Il permet de lancer rapidement des projets complexes sans configuration matérielle coûteuse.

**Example:**

---

**Merci pour votre attention**