Chapter I

INTRODUCTION TO PYTHON

1. What is Python?

Python is a high-level programming language that is widely used for its simplicity, readability, and versatility. So, we can say that Python is a language that allows you to tell a computer what to do using code that is easy to read and write, even for beginners.

Key characteristics

- **Easy to learn**: Python's syntax is close to plain English.
- **Interpreted**: You can run Python code without compiling it first.
- **Cross-platform**: Works on Windows, macOS, and Linux.
- **Open-source**: Free to use and supported by a large community.

2. How to install Python?

Before starting the Python installation, it is important to first verify whether Python is already available on the computer.

2.1 Checking whether Python is installed: There are many ways to check if Python is installed on your computer, depending on your operating system.

On windows:

1. Press **Win + R**, type **cmd**, then press **Enter**.
2. In the Command Prompt, type: **python –version** **(**or **python -V)** **(**or **py -V)**
3. If Python is installed, you will see something like: Python 3.9.6 (It is the version of your python)
4. ❌ If not installed, you may see: *"python is not recognized as an internal or external command"*

2.2 Installing Python

Follow these steps to install the official and the latest version of Python:

🔷 Step 1: Download Python

- ✓ Open your browser (Google Chrome or Firefox or any other browser) and go to the official website for Python: 👉 https://www.python.org/downloads/
- ✓ Click **Download Python 3.x.x** (the latest stable version).

🔷 Step 2: Install Python on Windows

- ✓ Double-click the downloaded .exe file.
- ✓ **IMPORTANT**: ✓ Check the box **"Add Python to PATH"**
- ✓ Click **Install Now**.
- ✓ Wait until the installation finishes.

🔷 Step 3: Verify the Installation (Windows)

- ✓ Open **Command Prompt (cmd)**.
- ✓ Type: **python –version or py --version**
- ✓ If installed correctly, you will see the Python version.

2.3 Installing pip:

After installing python, you are able to install pip: the Python package manager.

As usual, you can try to check if pip is installed by this command line: **pip –version** or **python -m pip --version**

If installed correctly, you will see the pip version.

If not installed: use this command line: **python -m ensurepip –upgrade**

After installation, we can install packages using pip:

Examples :

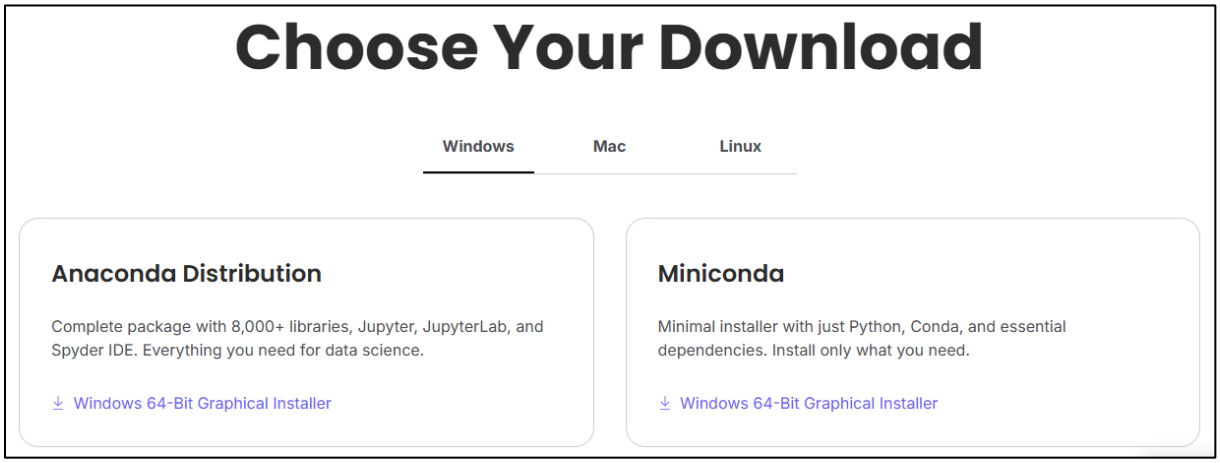
- ✓ pip install numpy
- ✓ pip install matplotlib

- ✓ pip install pandas

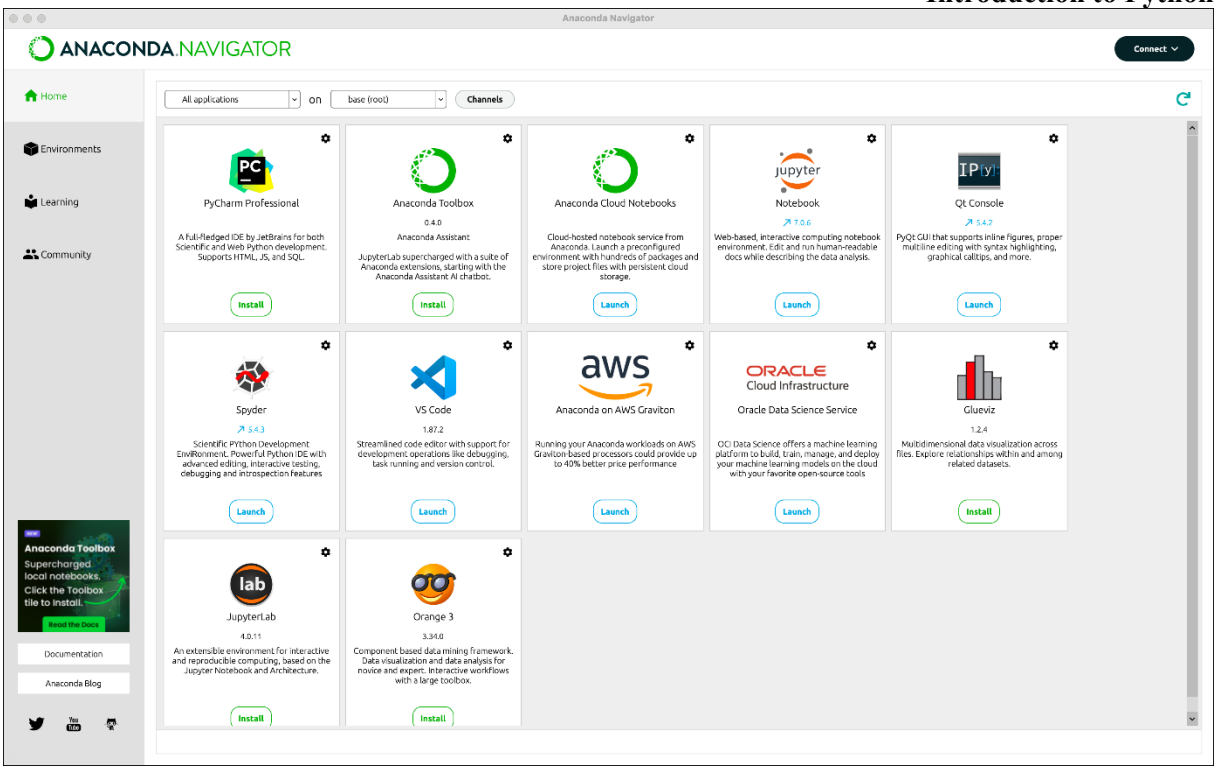
2.6 Installing Anaconda environment

Anaconda is a free and open-source distribution of the Python and R programming languages, specifically designed for data science, machine learning, and large-scale data processing. It provides an integrated environment that simplifies package installation, dependency management, and application deployment, which makes it particularly well suited for scientific and research-oriented computing.

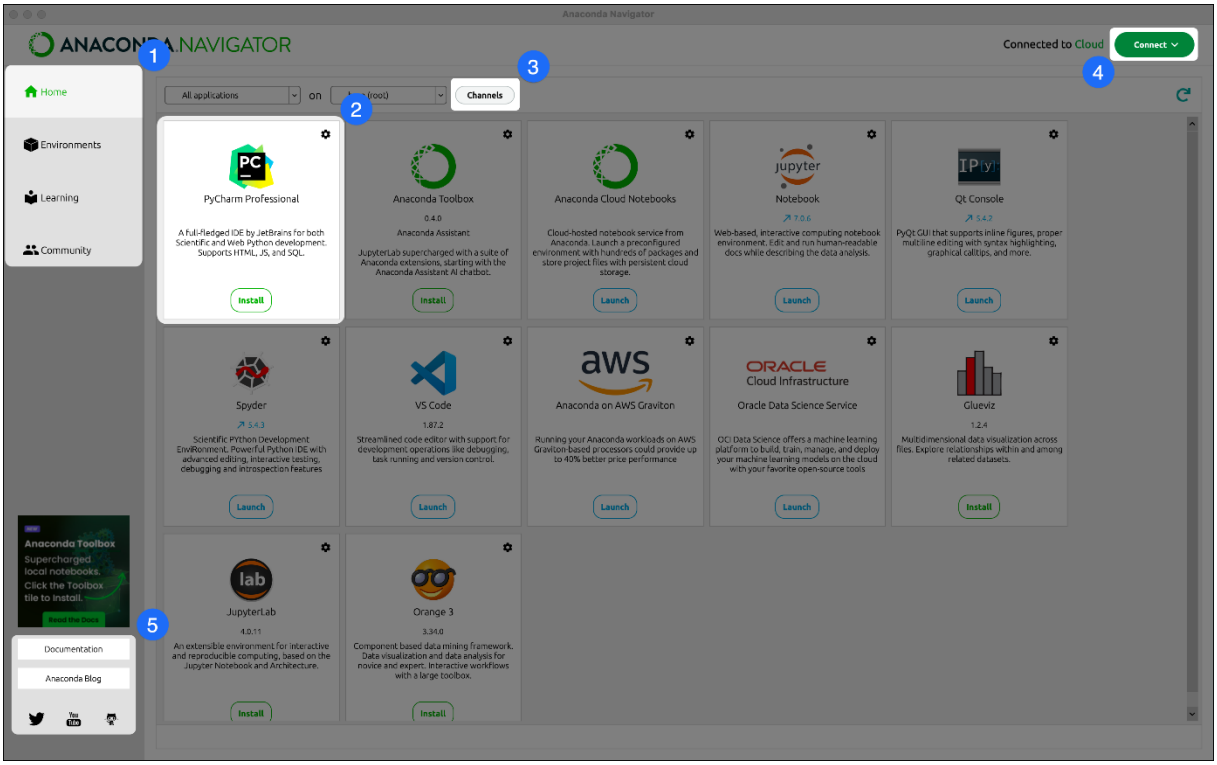
You can install Anaconda by downloading it from the official website (https://www.anaconda.com/products/distribution), choose the version that suits your operating system and following the installation instructions provided.



After completing the installation, the Anaconda environment can be launched by selecting its desktop icon.



3. Environment configuration:



1. ***Navigator page:*** Access the main pages of the Navigator application. The Home page is open by default. For more information on the contents of each page, see **Navigator pages**.
2. ***Application/package tile:*** Install or launch popular graphical Python applications that work well with Navigator.
3. ***Channels:*** Manage the channels Navigator can use to source and install packages.
4. ***Connect:*** Connect to Anaconda.com or any of the free or subscription repositories Anaconda provides. Signing into a repository service will enable searching for packages within that repository.
5. ***Outside links:*** Visit Anaconda's documentation, blog, and social media.

Here is a **brief and clear description** of the main code editors and environments commonly used with **Anaconda**:

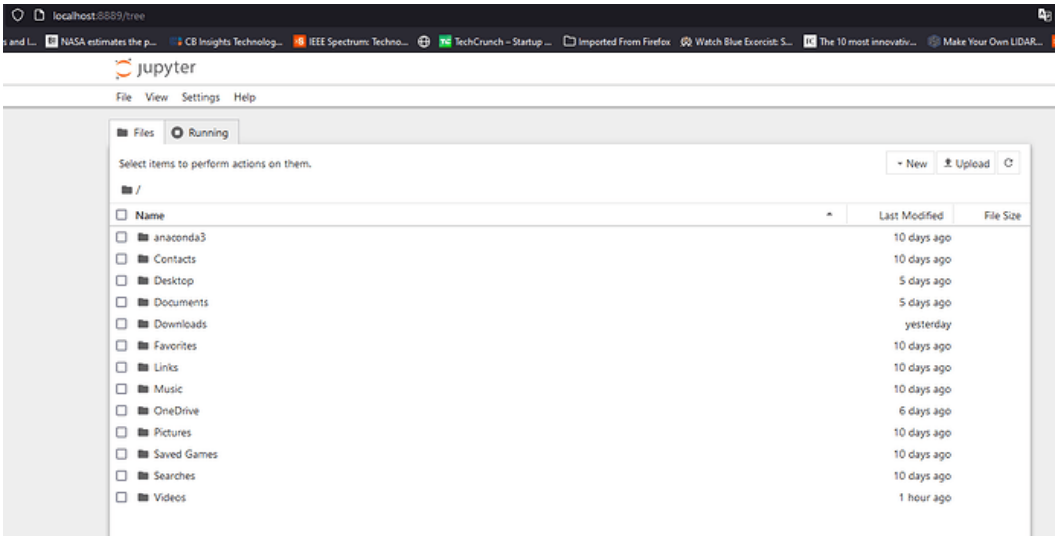
- **Visual Studio Code (VS Code):** VS Code is a lightweight and versatile code editor that supports Python through extensions. It offers features such as syntax highlighting, debugging, Git integration, and support for virtual environments, making it suitable for both general programming and data science workflows.
- **PyCharm:** PyCharm is a powerful integrated development environment (IDE) specifically designed for Python. It provides advanced tools for code analysis, debugging, project management, and testing. PyCharm is well suited for large or complex Python projects and professional software development.
- **Spyder:** Spyder is a scientific Python IDE included with Anaconda. It is designed for data science and scientific computing, offering an interactive console, variable explorer, built-in plotting, and an editor similar to MATLAB, making it ideal for analysis and experimentation.

- **Jupyter Notebook:** Jupyter Notebook is an interactive web-based environment that allows users to combine executable code, text, equations, and visualizations in a single document. It is particularly useful for data analysis, research, teaching, and sharing reproducible results.

4. Getting started with Jupyter Notebook:

Jupyter Notebook is an interactive environment that allows you to write and run code, visualize results, and add explanatory text in a single document. It is widely used in data science, machine learning, and scientific computing.

To get started, first launch **Anaconda**, then open **Jupyter Notebook** from the Anaconda Navigator. Your web browser will open automatically, displaying the Jupyter interface.



4.1 Create a new Notebook:

In the Jupyter home page, click on **New** (top-right corner) → **Notebook**.

A new notebook will open in a new tab and select **Python 3** as kernel from the list.

You can now write Python code in the cells and run it by pressing **Shift + Enter.**

Click on the notebook title (usually *Untitled*), and enter a new name and press **Enter.**

4.2 Add Code Cells / Markdown Cells:

In Jupyter Notebook, there are two main types of cells. **Code cells** are used to write and run Python code, while **Markdown cells** are used to add text, titles, and explanations. You can easily add new cells by clicking the **"+"** button.