

## PRACTICAL-1

**AIM: (A) Introduction to Python Programming. Installation & Configuration of Python. Along with its all-major editors, IDLE, Pycharm, Anaconda, Jupyter, Interpreter etc.**

### Introduction to Python:

#### THEORY:

Python programming is like having a friendly conversation with your computer. It was created by a cool person named Guido van Rossum who wanted coding to be easy and enjoyable. Think of Python as a language that your computer easily understands because it keeps things simple and clear. Whether you want to build a website, play with data, or teach your computer to be smart, Python is your go-to buddy. It's not just about writing code; it's about making it fun and readable. Python comes with lots of built-in tools, like a super cool toolbox, and there's a big group of friends (a community) who help each other out. It doesn't matter if you use a Windows computer or a Mac – Python is like a universal translator, helping your code work smoothly on different systems.

#### Advantages of Python:

- 1) **Clear Syntax:** Python's concise and readable syntax promotes code clarity and reduces development time.
- 2) **Versatility:** Python is versatile, serving well in web development, data science, AI, and automation.
- 3) **Extensive Libraries:** The language comes with a rich standard library and a vast ecosystem of third-party modules.
- 4) **Community Support:** Python has a large and active community, offering support, resources, and frequent updates.
- 5) **Cross-Platform Compatibility:** Python code runs seamlessly on various operating systems, ensuring portability.

#### Disadvantages of Python:

- 1) **Execution Speed:** Python is an interpreted language, which can result in slower execution speeds compared to languages like C or C++. This can be a drawback for performance-critical applications or large-scale data processing.
- 2) **Global Interpreter Lock (GIL):** The Global Interpreter Lock in CPython, the default Python interpreter, can limit the execution of multiple threads at once. This can hinder the performance of multi-threaded Python programs, particularly in scenarios where parallel execution is essential.
- 3) **Mobile Development Limitations:** Python is not as widely used in mobile app development as some other languages. While frameworks like Kivy and BeeWare exist, they may not offer the same level of support and community as those associated with languages like Swift (for iOS) or Kotlin (for Android).

- 4) **Memory Consumption:** Python's dynamic typing and high-level abstractions can lead to higher memory consumption compared to languages with more static typing. This can be a concern in resource-constrained environments or when working with large datasets.
- 5) **Community Fragmentation:** Python has two major versions, Python 2 and Python 3, and while efforts have been made to transition to Python 3, there are still projects and libraries that use Python 2. This version split can lead to compatibility issues and requires careful consideration when selecting or maintaining projects.

### Applications of Python:

- 1) **Web Development:** Python is widely used for building dynamic websites and web applications. Frameworks like Django and Flask provide developers with powerful tools to create robust and scalable web solutions, making Python a popular choice in the web development community.
- 2) **Data Science and Machine Learning:** Python is a go-to language for data scientists and machine learning engineers. Libraries such as NumPy, Pandas, and Scikit-learn simplify data manipulation, analysis, and machine learning model development, contributing to Python's dominance in the field of data science.
- 3) **Automation and Scripting:** Python's simplicity and readability make it an excellent choice for automating repetitive tasks and writing scripts. Whether it's automating system administration, handling file operations, or creating small utility programs, Python's versatility shines in scripting applications.
- 4) **Artificial Intelligence (AI) and Natural Language Processing (NLP):** Python is extensively used in AI development, thanks to libraries like TensorFlow and PyTorch. Natural Language Processing tasks, such as language translation and sentiment analysis, also benefit from Python's rich ecosystem, including the NLTK and spaCy libraries.
- 5) **Desktop GUI Applications:** Python supports the development of desktop graphical user interface (GUI) applications using libraries like Tkinter, PyQt, and Kivy. These libraries enable developers to create cross-platform applications with user-friendly interfaces for tasks ranging from simple utilities to complex desktop applications.

### Python Installation on Windows

The installation requires downloading the official Python *.exe* installer and running it on your system. The sections below will explain several options and details during the installation process.

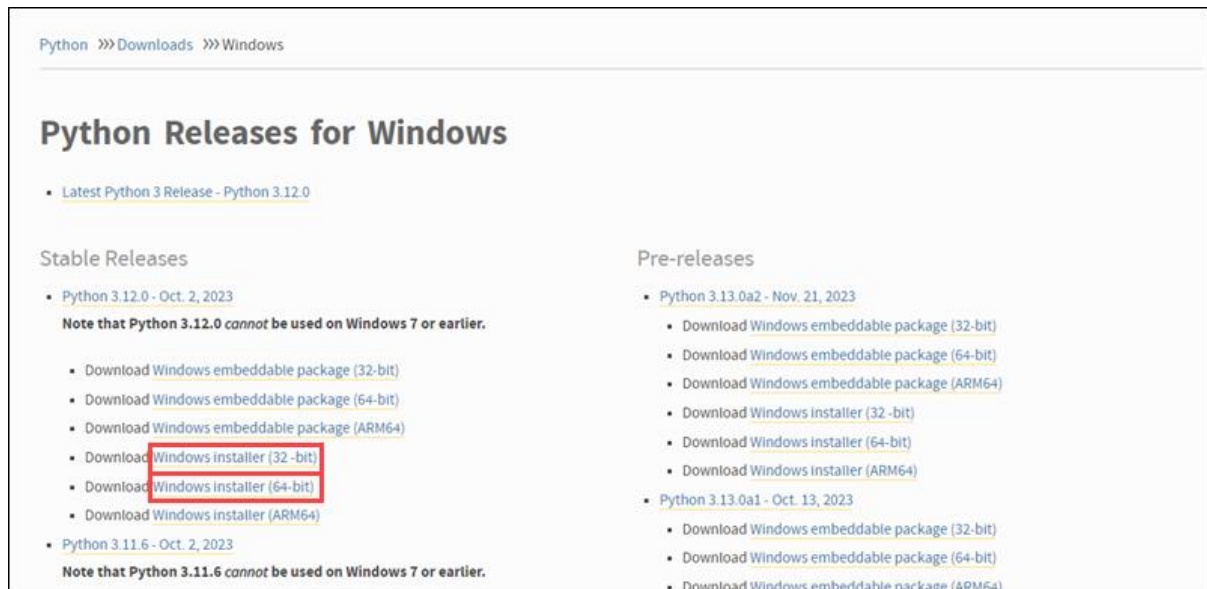
#### Step 1: Select Python Version

Deciding on a version depends on what you want to do in Python. The two major versions are Python 2 and Python 3.

## Step 2: Download Python Executable Installer

Start by downloading the Python executable installer for Windows:

1. Open a web browser and navigate to the Downloads for Windows section of the official Python website.
2. Locate the desired Python version.



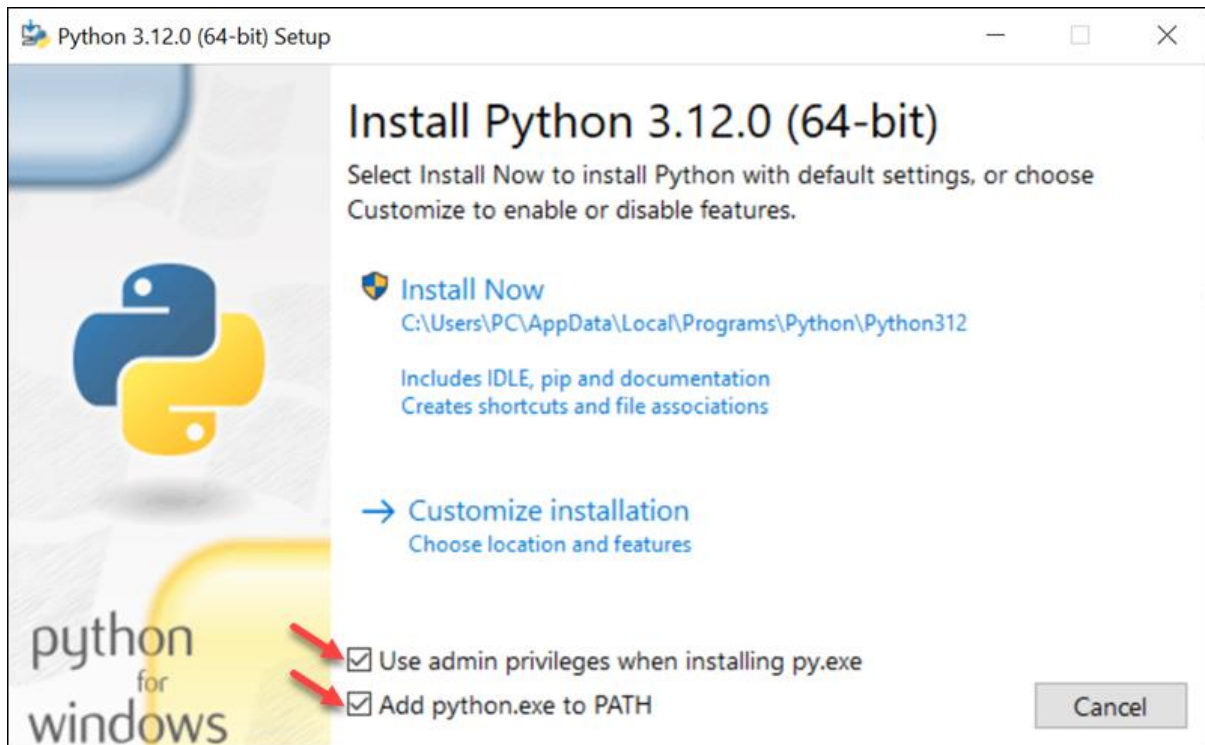
3. Click the link to download the file. Choose either the Windows 32-bit or 64-bit installer.

The download is approximately 25MB.

## Step 3: Run Executable Installer

The steps below guide you through the installation process:

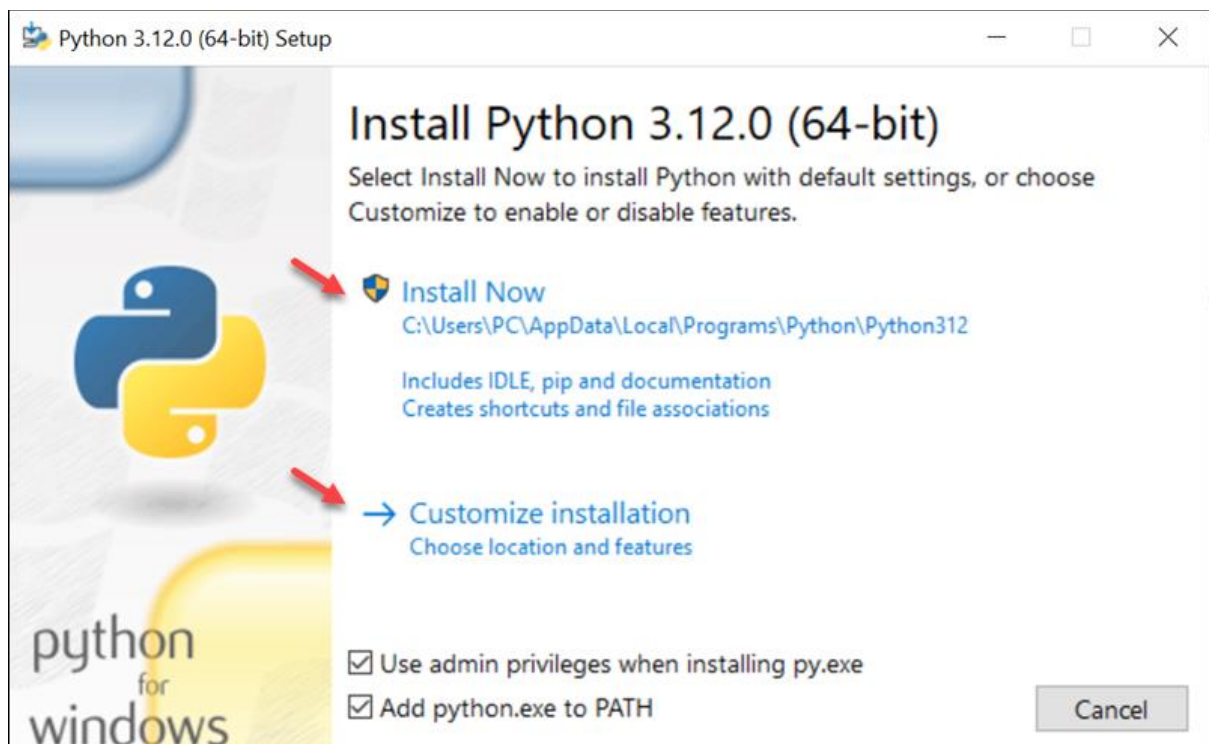
1. Run the downloaded **Python Installer**.
2. The installation window shows two checkboxes:
  - **Admin privileges.** The parameter controls whether to install Python for the current or all system users. This option allows you to change the installation folder for Python.
  - **Add Python to PATH.** The second option places the executable in the PATH variable after installation. You can also add Python to the PATH environment variable manually later.



For the most straightforward installation, we recommend ticking both checkboxes.

3. Select the **Install Now** option for the recommended installation (in that case, skip the next two steps).

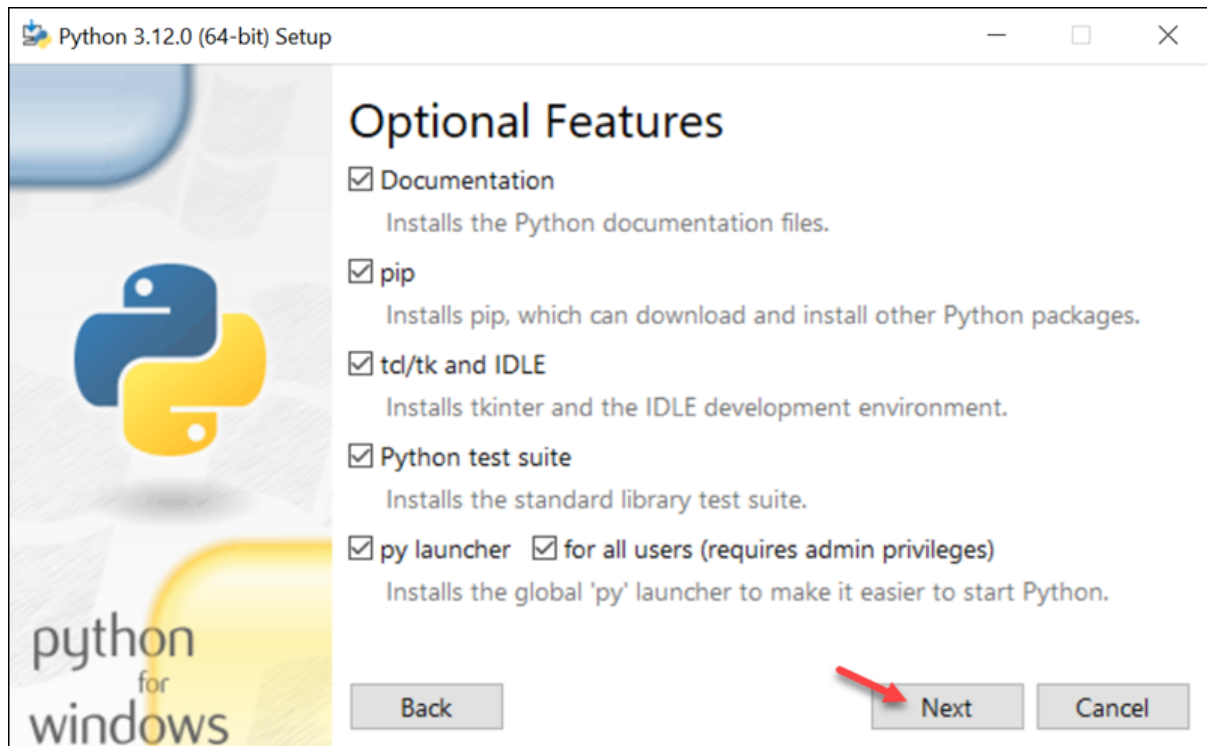
To adjust the default installation options, choose **Customize installation** instead and proceed to the following step.



The default installation installs Python to `C:\Users\[user]\AppData\Local\Programs\Python\Python[version]` for the current user. It includes IDLE (the default Python editor), the PIP package manager, and additional documentation. The installer also creates necessary shortcuts and file associations.

Customizing the installation allows changing these installation options and parameters.

4. Choose the optional installation features. Python works without these features, but adding them improves the program's usability.



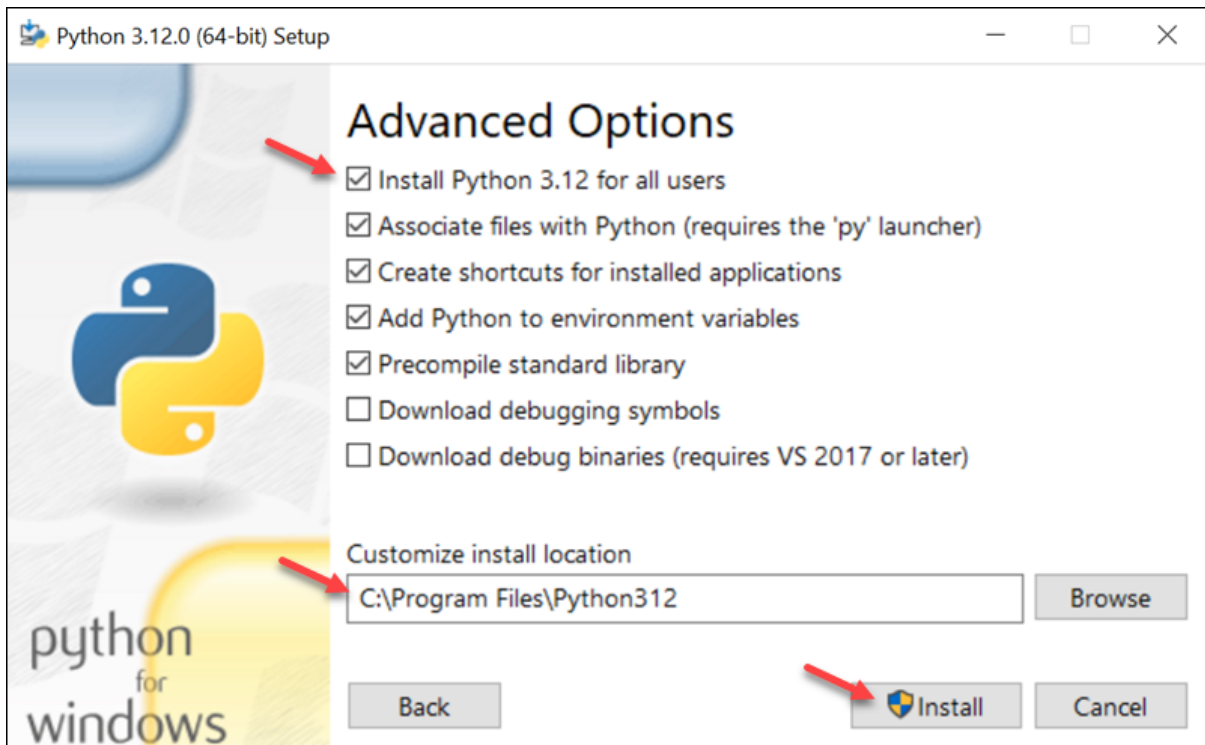
Click **Next** to proceed to the Advanced Options screen.

5. The second part of customizing the installation includes advanced options.

Choose whether to install Python for all users. The option changes the install location to `C:\Program Files\Python[version]`. If selecting the location manually, a common choice is `C:\Python[version]` because it avoids spaces in the path, and all users can access it. Due to administrative rights, both paths may cause issues during package installation.

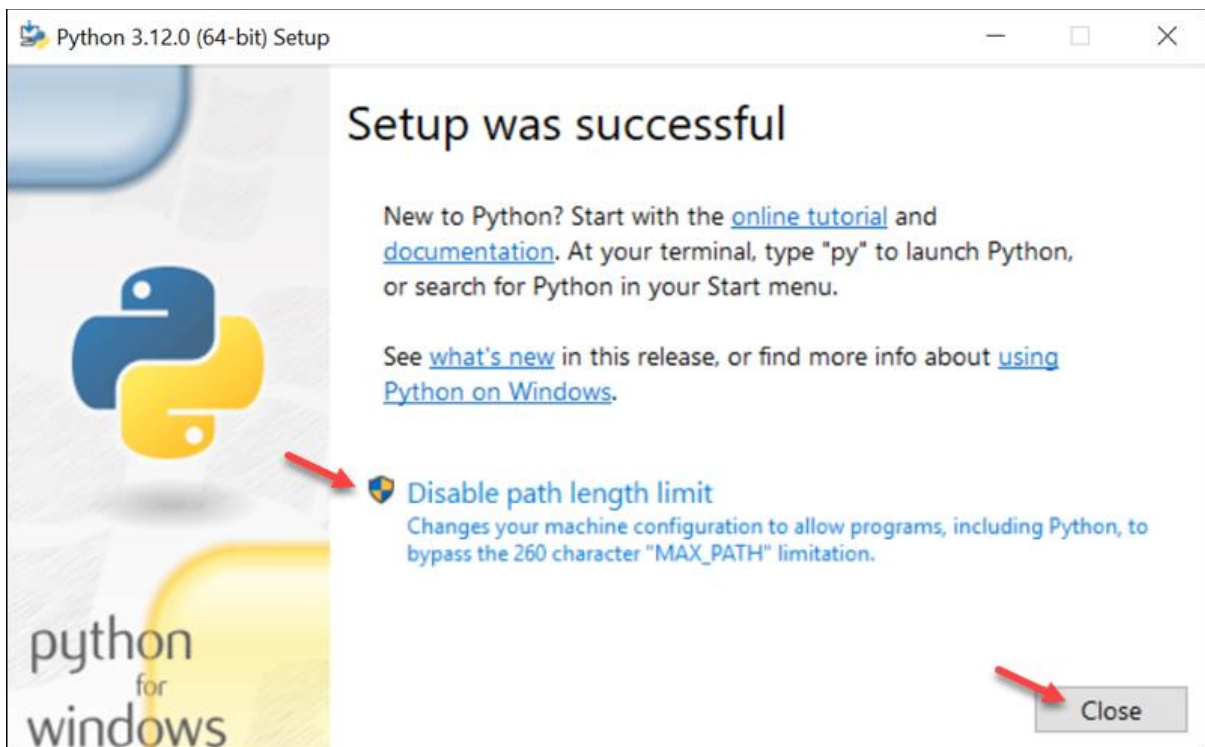
Other advanced options include creating shortcuts, file associations, and adding Python to PATH.





After picking the appropriate options, click **Install** to start the installation.

6. Select whether to disable the path length limit. Choosing this option will allow Python to bypass the 260-character **MAX\_PATH** limit.



The option will not affect any other system settings, and disabling it resolves potential name-length issues. We recommend selecting the option and closing the setup.

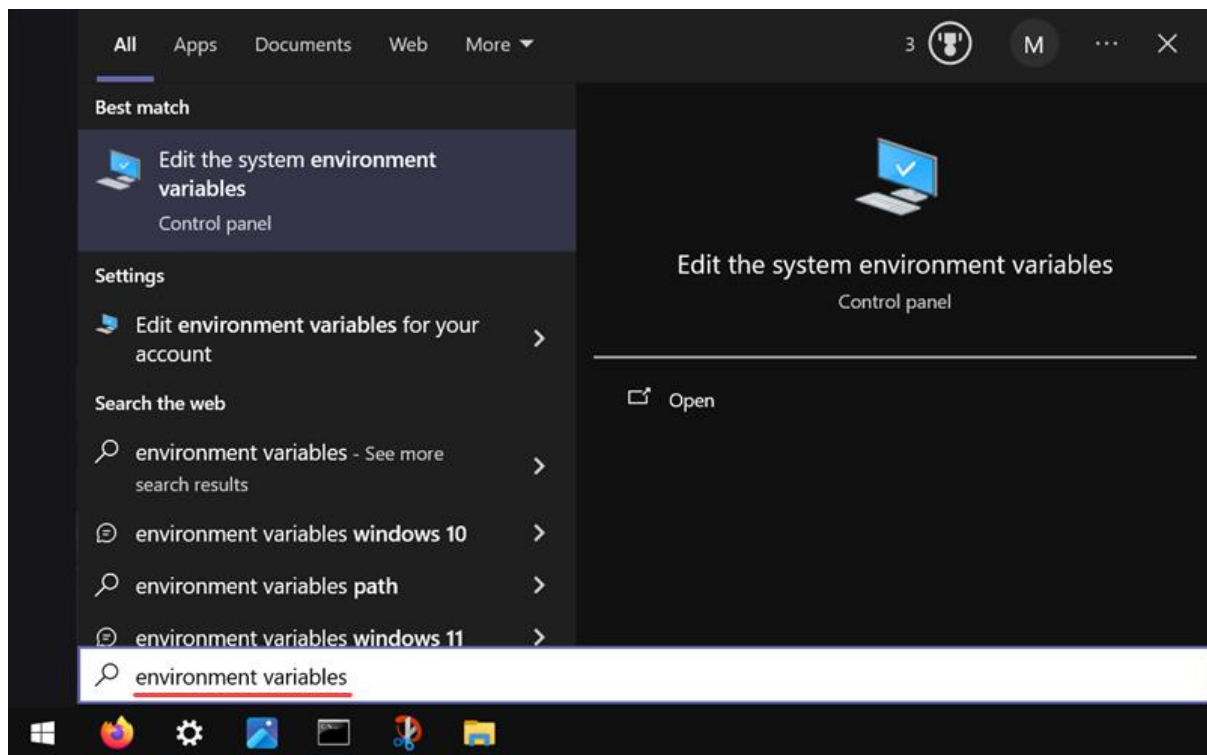
#### Step 4: Add Python to Path (Optional)

If the Python installer does not include the **Add Python to PATH** checkbox or you have not selected that option, continue in this step. Otherwise, skip to the next step.

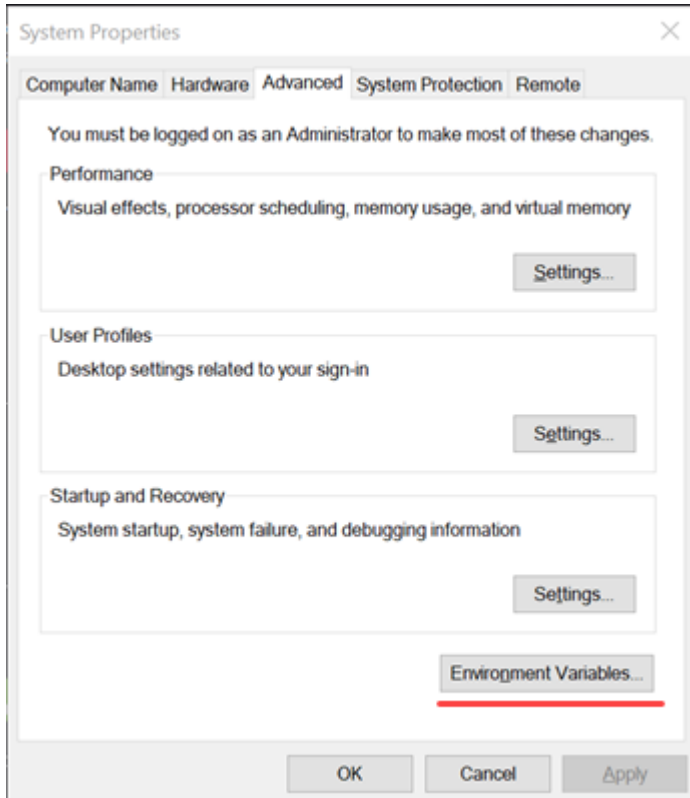
Adding the Python path to the PATH variable alleviates the need to use the full path to access the Python program in the command line. It instructs Windows to review all the folders added to the PATH environment variable and to look for the *python.exe* program in those folders.

To add Python to PATH, do the following:

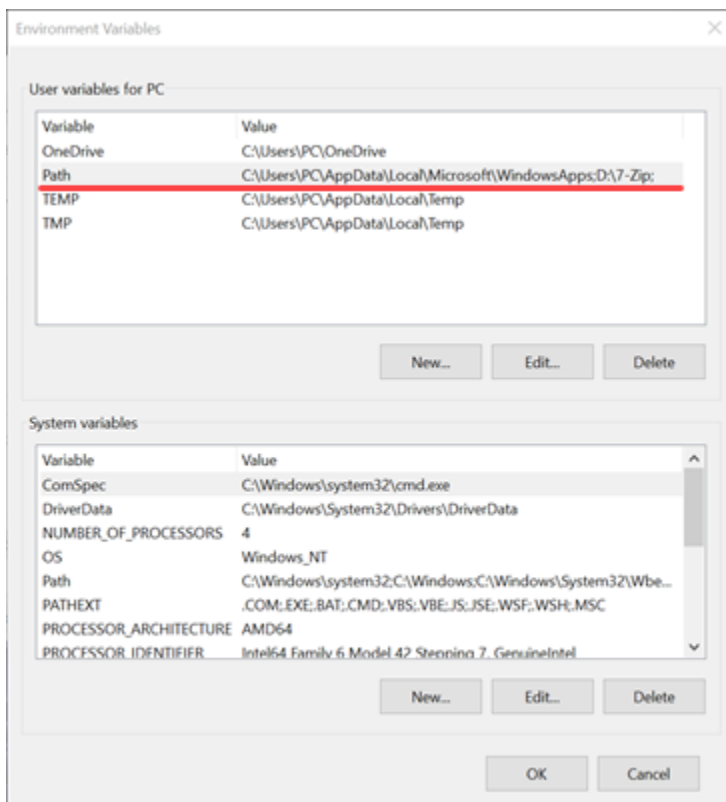
1. In the **Start** menu, search for **Environment Variables** and press **Enter**.



2. Click **Environment Variables** to open the overview screen.



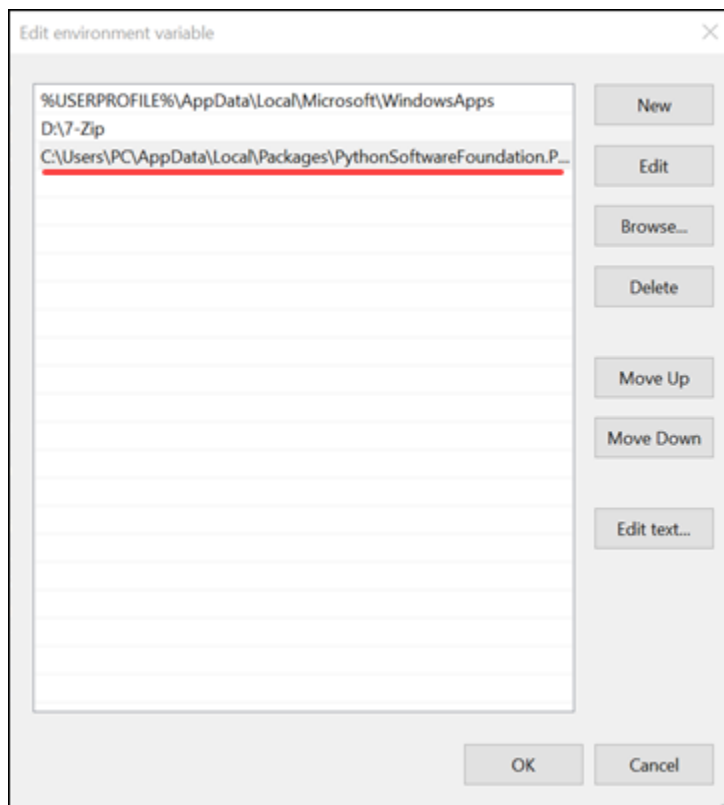
3. Double-click **Path** on the list to edit it.



Alternatively, select the variable and click the **Edit** button.



4. Double-click the first empty field and paste the Python installation folder path.



Alternatively, click the **New** button instead and paste the path.

5. Click **OK** to save the changes. If the command prompt is open, restart it for the following step.

### Step 5: Verify Python Was Installed on Windows

The first way to verify that Python was installed successfully is through the command line. Open the command prompt and run the following command:

```
python --version
```

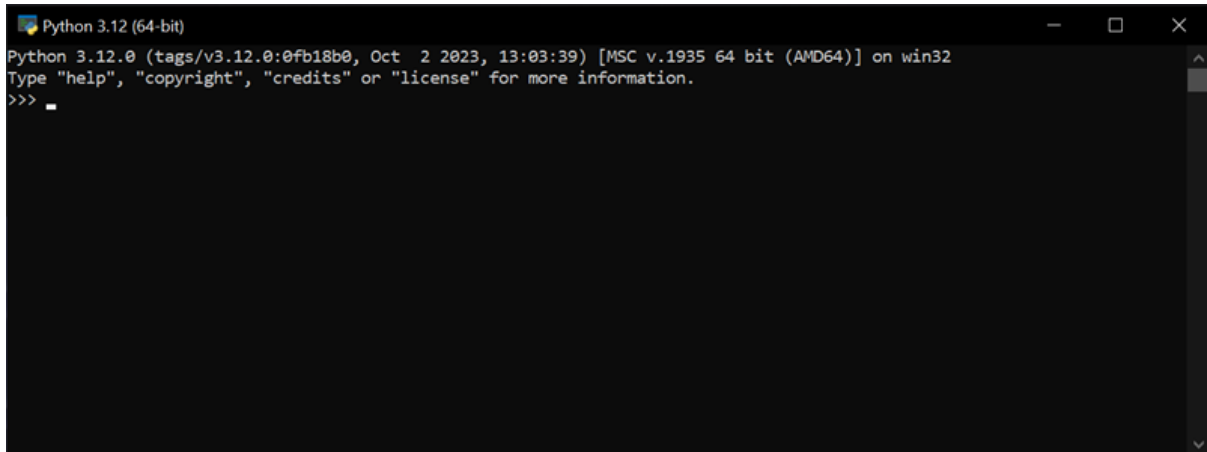
```
C:\Users\PC>python --version
Python 3.12.0
```

The output shows the installed Python version.

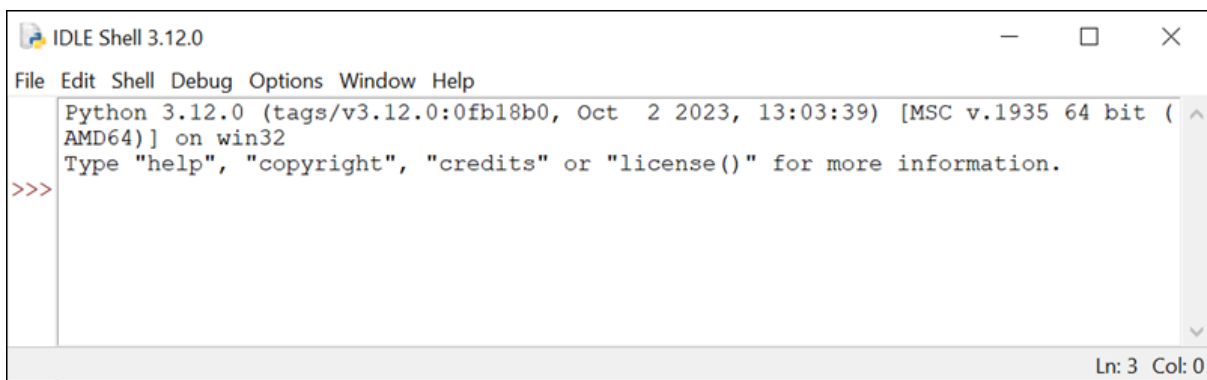
The second way is to use the GUI to verify the Python installation. Follow the steps below to run the Python interpreter or IDLE:

1. Navigate to the directory where Python was installed on the system.

2. Double-click *python.exe* (the Python interpreter) or IDLE.
3. The interpreter opens the command prompt and shows the following window:



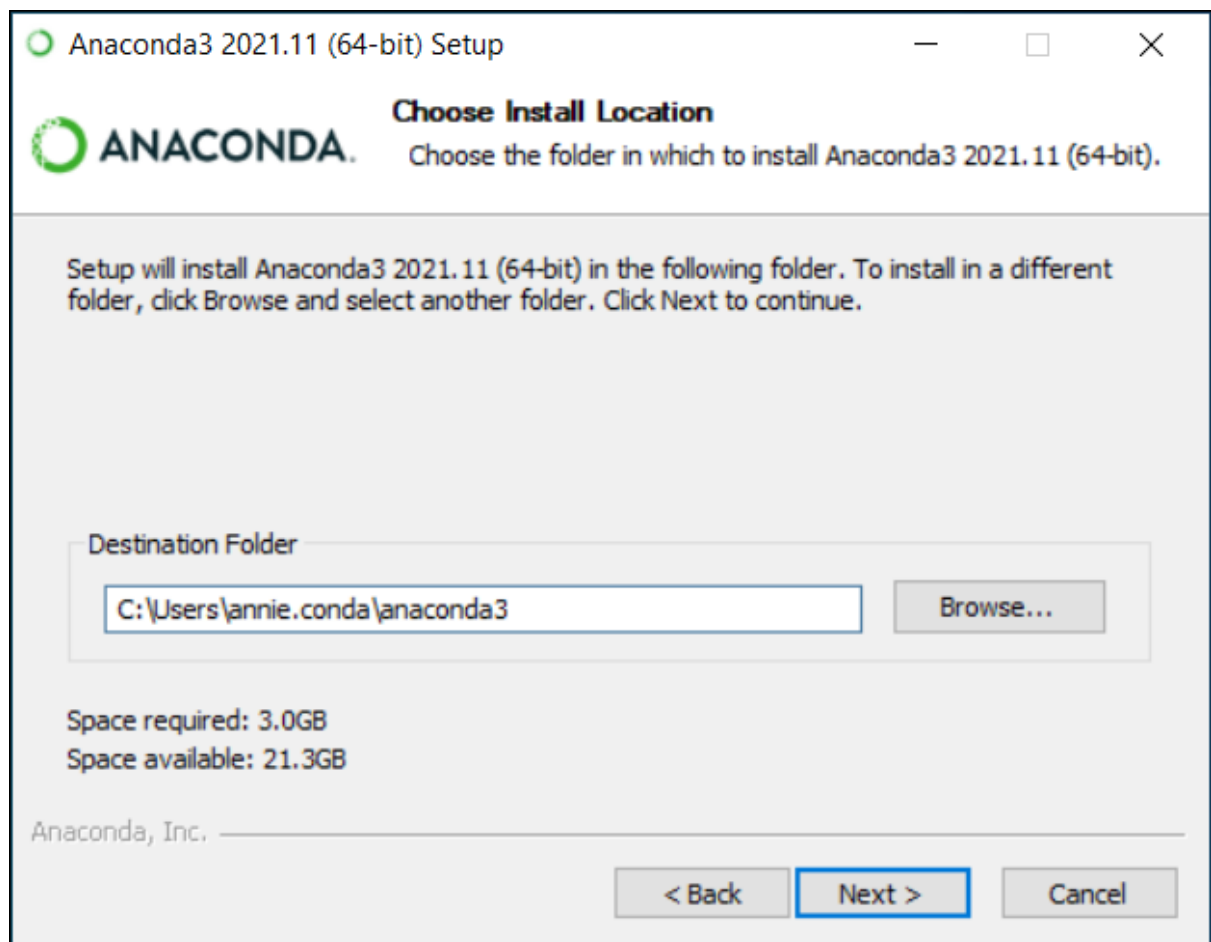
Running IDLE opens Python's built-in IDE:



In both cases, the installed Python version shows on the screen, and the editor is ready for use.

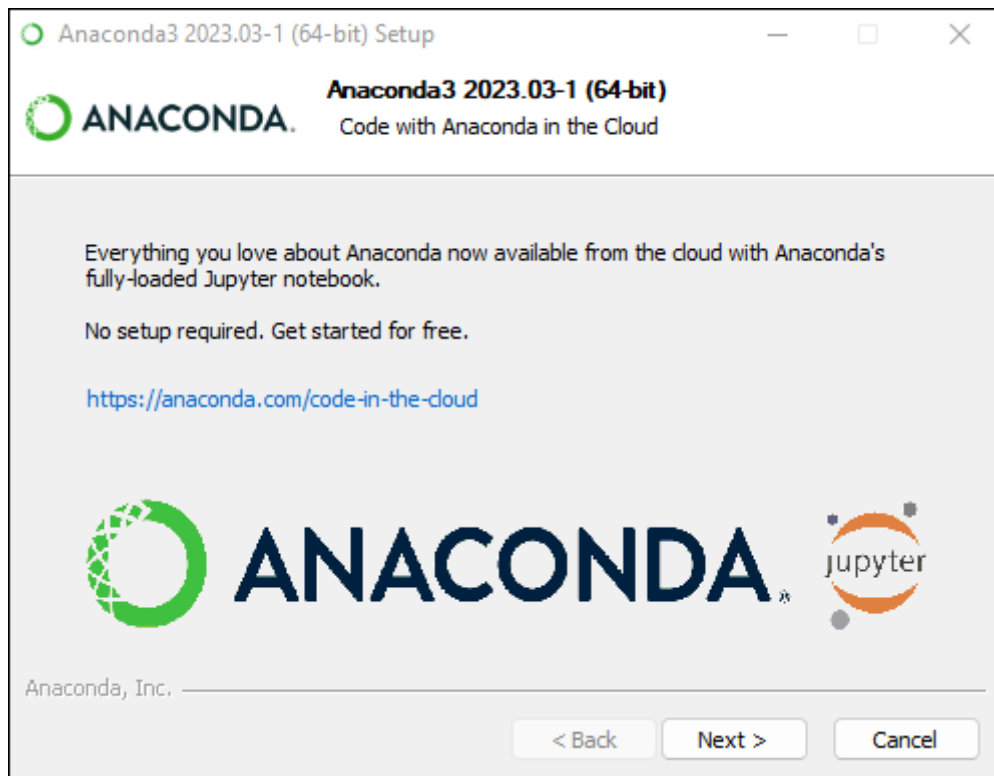
## Installation of Anaconda:

1. Download the Anaconda installer.
2. Go to your Downloads folder and double-click the installer to launch. To prevent permission errors, do not launch the installer from the Favorites folder.
3. Click Next.
4. Read the licensing terms and click I Agree.
5. It is recommended that you install for Just Me, which will install Anaconda Distribution to just the current user account. Only select an install for All Users if you need to install for all users' accounts on the computer (which requires Windows Administrator privileges).
6. Click Next.
7. Select a destination folder to install Anaconda and click Next. Install Anaconda to a directory path that does not contain spaces or unicode characters.
8. Do not install as Administrator unless admin privileges are required.



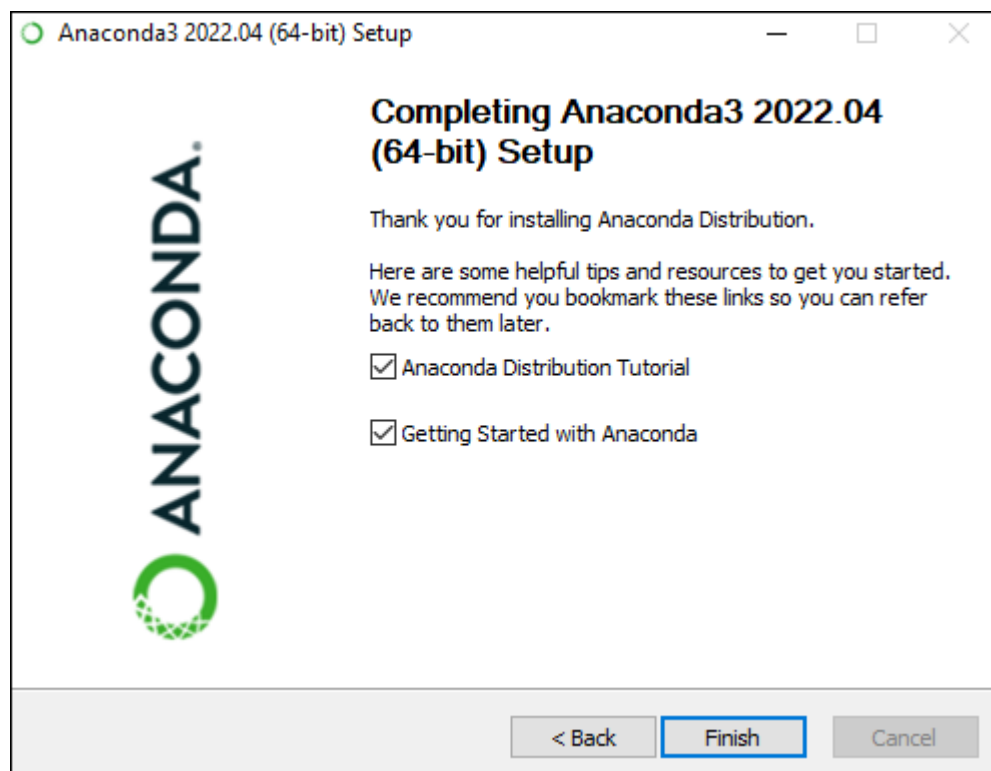
9. Choose whether to add Anaconda to your PATH environment variable or register Anaconda as your default Python.
10. Click Install. If you want to watch the packages Anaconda is installing, click Show Details.
11. Click Next.

12. Optional: To learn more about Anaconda's cloud notebook service, go to <https://www.anaconda.com/code-in-the-cloud>.



Or click Continue to proceed.

13. After a successful installation you will see the “Thanks for installing Anaconda” dialog box:



14. If you wish to read more about Anaconda.org and how to get started with Anaconda, check the boxes “Anaconda Distribution Tutorial” and “Learn more about Anaconda”. Click the Finish button.
15. Verify your installation.

### **Jupyter Notebook Install Using pip on Windows**

If you prefer not to use conda, you can install Jupyter Notebook using pip package manager for Python.

#### **Step 1: Install Python programming language**

Download and install the latest version of Python from the official website (<https://www.python.org/downloads/>). Make sure to check the option “Add Python to PATH” during the installation.

#### **Step 2: Install Jupyter Notebook**

Open the Command Prompt and run the following command using pip install to install Jupyter Notebook:  
`pip install jupyter`

#### **Step 3: Start Jupyter Notebook**

Type the following command in the Command Prompt to launch Jupyter Notebook:  
`jupyter notebook`

### **How to Install Pycharm**

Here is a step by step process on how to download and install Pycharm IDE on Windows:

**Step 1)** To download PyCharm visit the website <https://www.jetbrains.com/pycharm/download/> and Click the “DOWNLOAD” link under the Community Section.

# Download PyCharm

Windows

macOS

Linux

## Professional

Full-featured IDE  
for Python & Web  
development

**DOWNLOAD**

Free trial

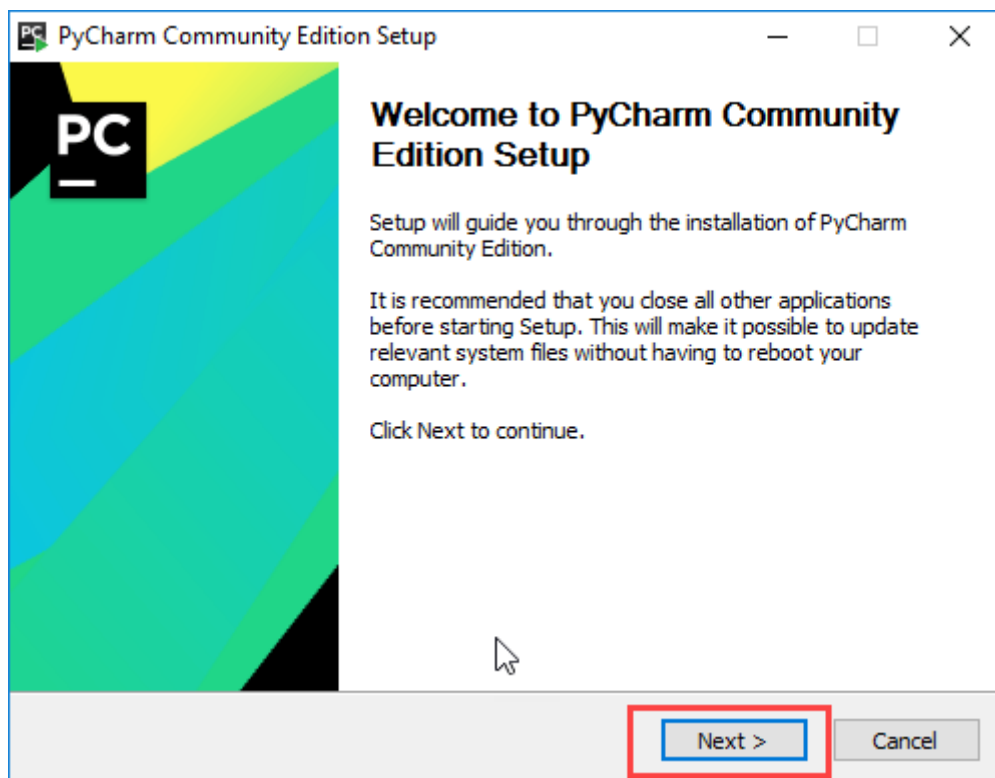
## Community

Lightweight IDE  
for Python & Scientific  
development

**DOWNLOAD**

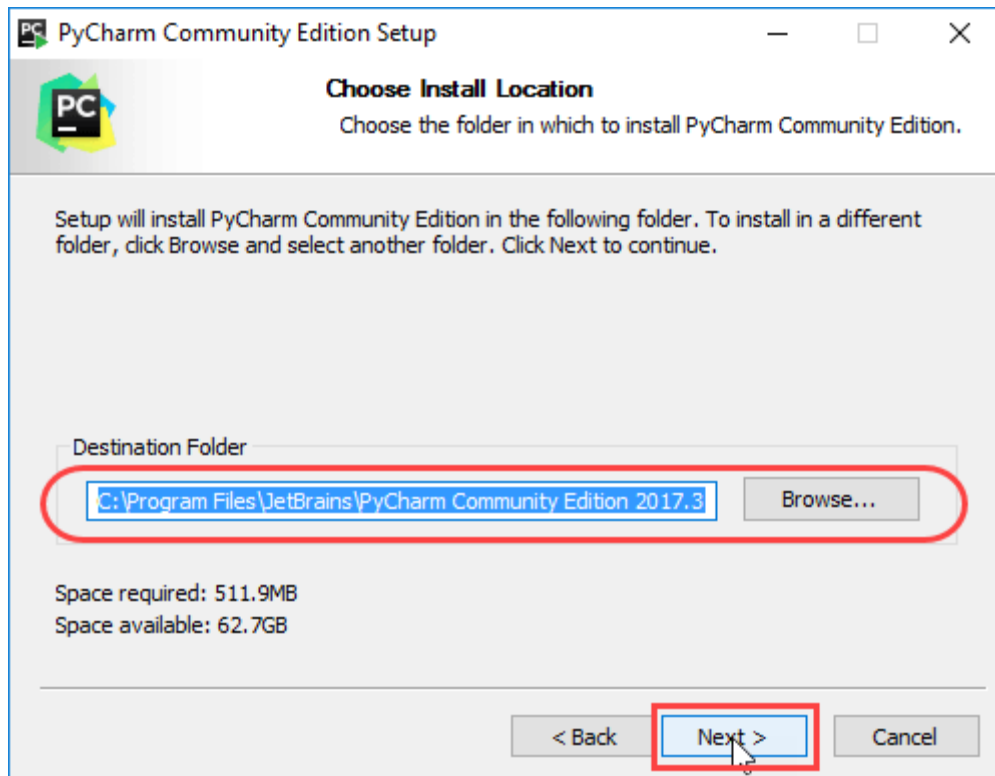
Free, open-source

**Step 2)** Once the download is complete, run the exe for install PyCharm. The setup wizard should have started. Click “Next”.

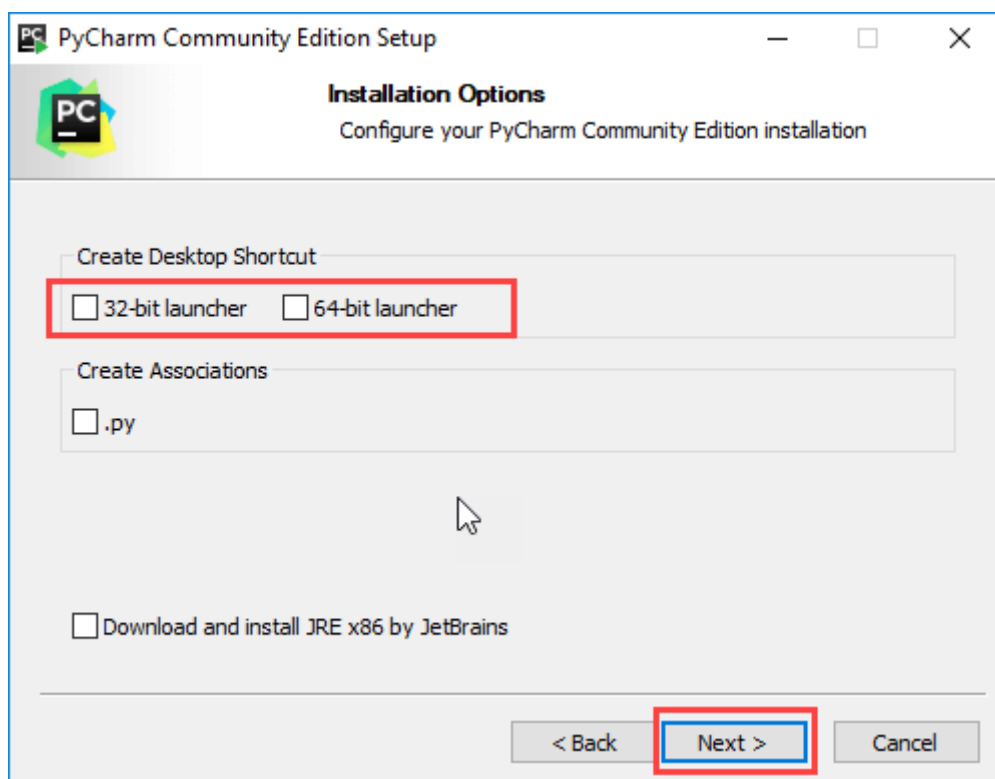


**Step 3)** On the next screen, Change the installation path if required. Click “Next”.

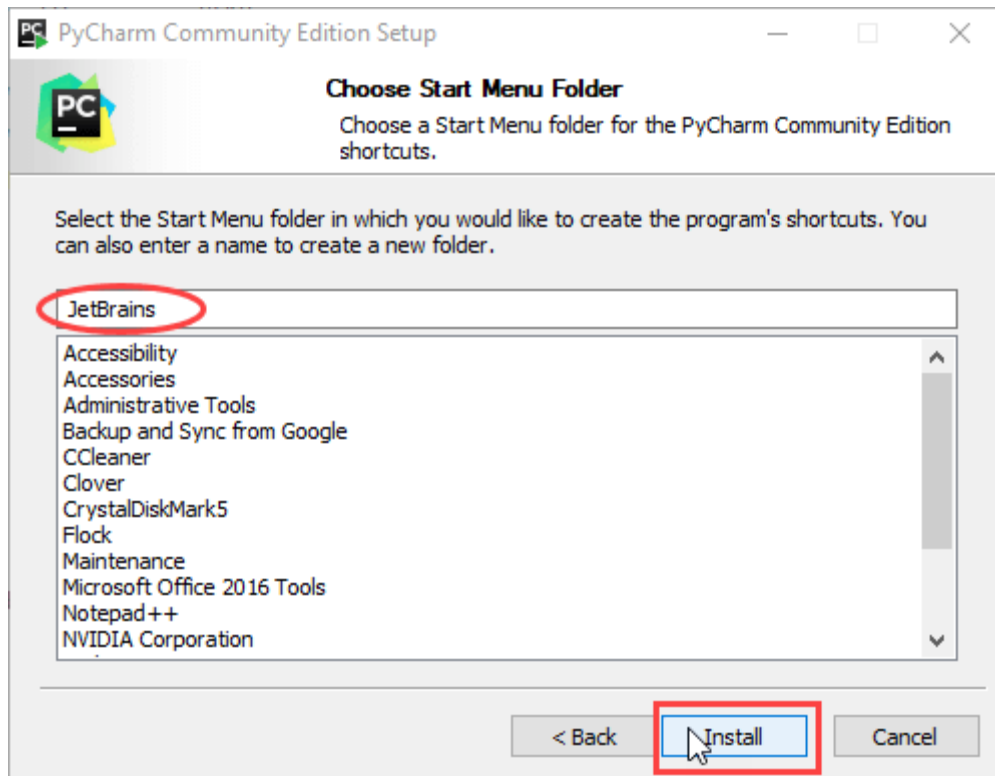




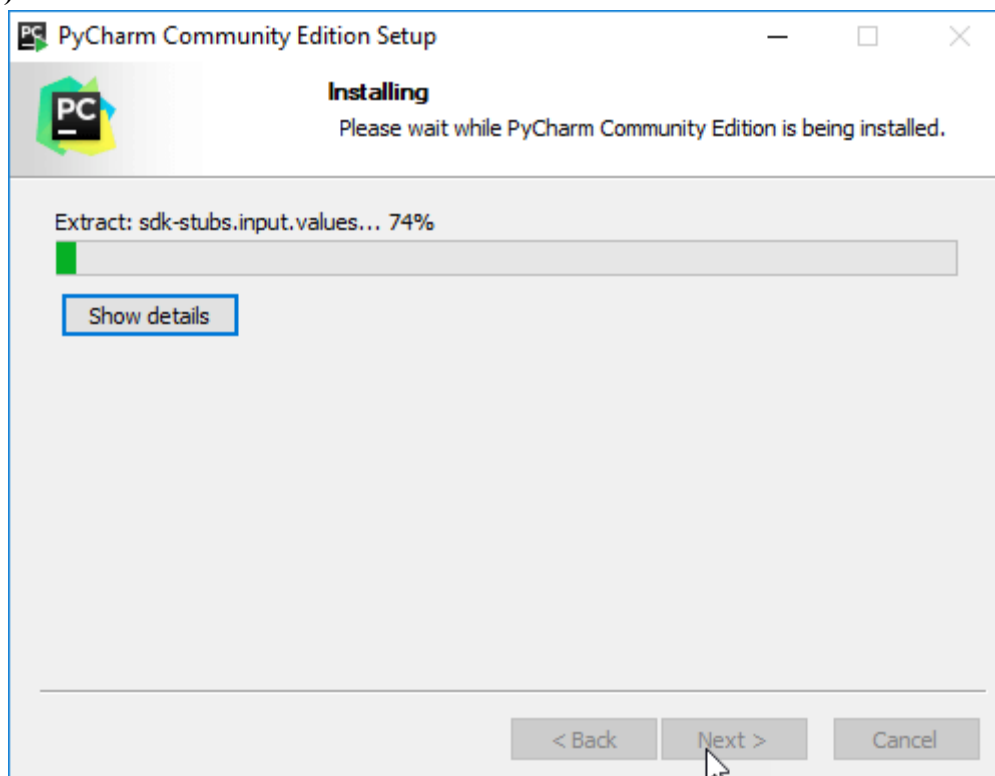
**Step 4)** On the next screen, you can create a desktop shortcut if you want and click on “Next”.



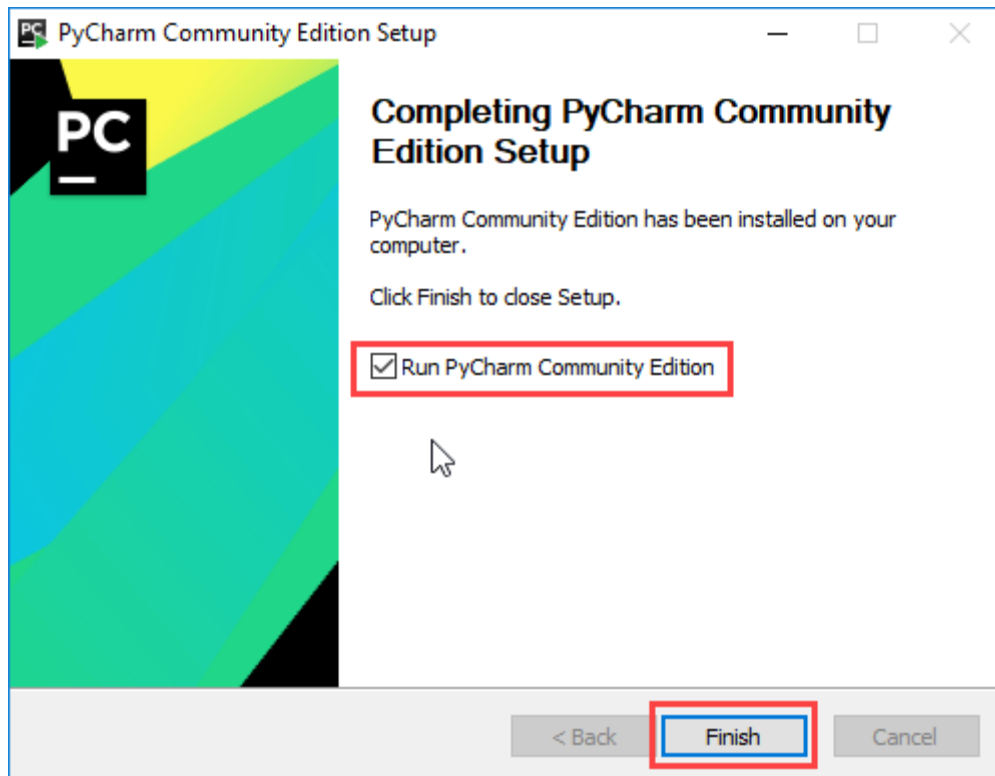
**Step 5)** Choose the start menu folder. Keep selected JetBrains and click on “Install”.



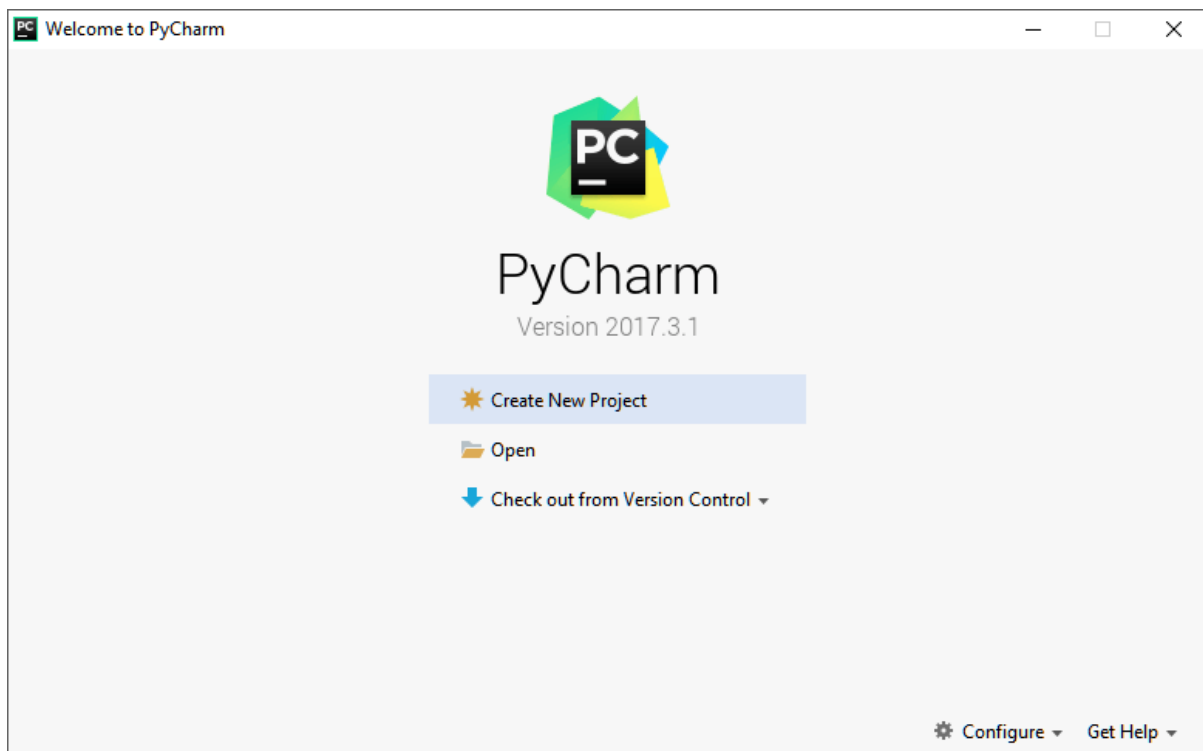
**Step 6)** Wait for the installation to finish.



**Step 7)** Once installation finished, you should receive a message screen that PyCharm is installed. If you want to go ahead and run it, click the “Run PyCharm Community Edition” box first and click “Finish”.



**Step 8)** After you click on “Finish,” the Following screen will appear.



## Conclusion

In this tutorial, we described how to install Python on Windows. The process is similar for all Python versions.