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. Choosing one over the other might be better depending on your project details. If there are no constraints, choose whichever one you prefer.

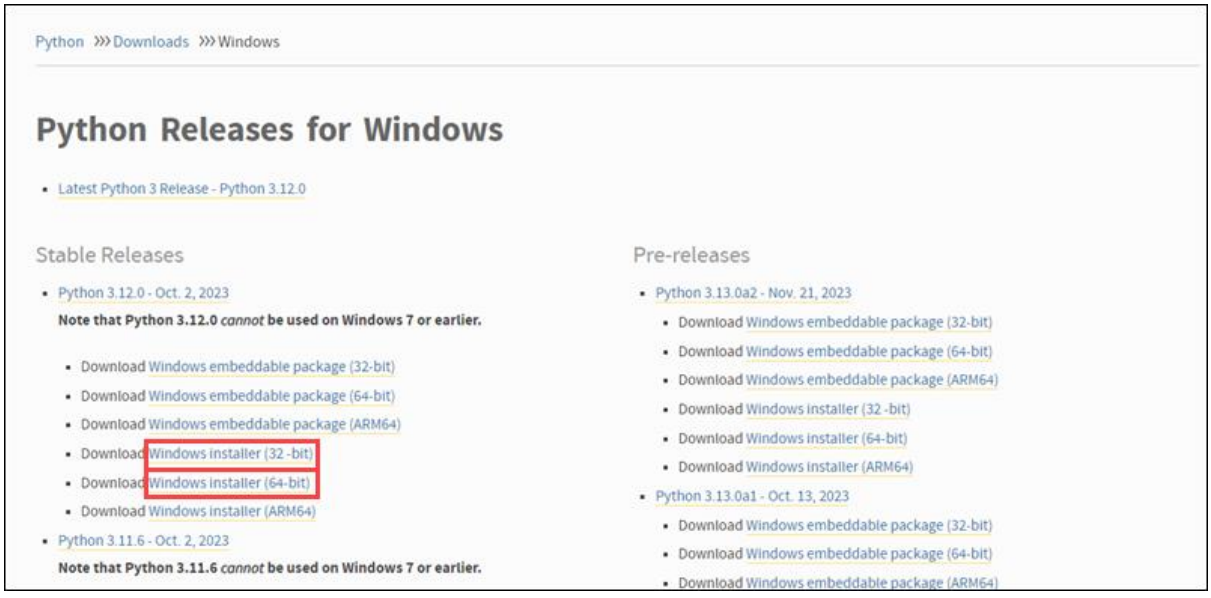
We recommend Python 3, as Python 2 reached its end of life in 2020. Download Python 2 only if you work with legacy scripts and older projects. Also, choose a stable release over the newest since the newest release may have bugs and issues.

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.

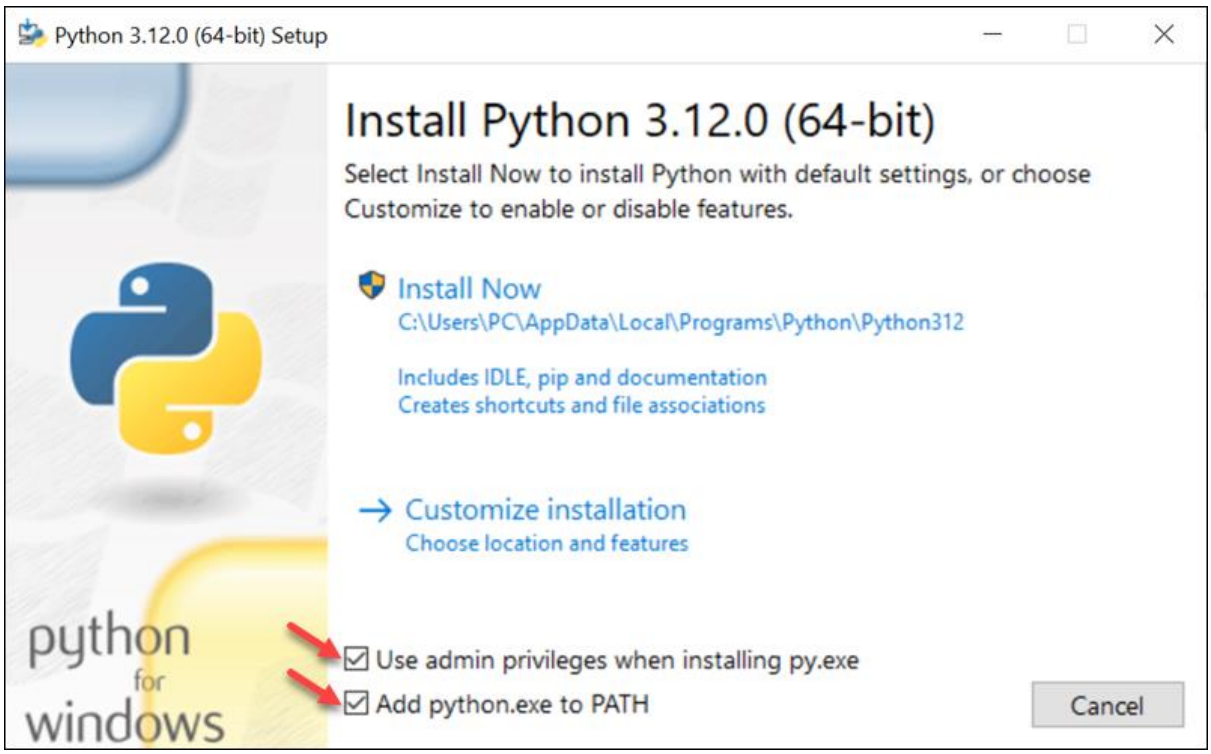
Step 3: Run Executable Installer

The steps below guide you through the installation process:

1. Run the downloaded **Python Installer**.

2. The installation window shows two checkboxes:

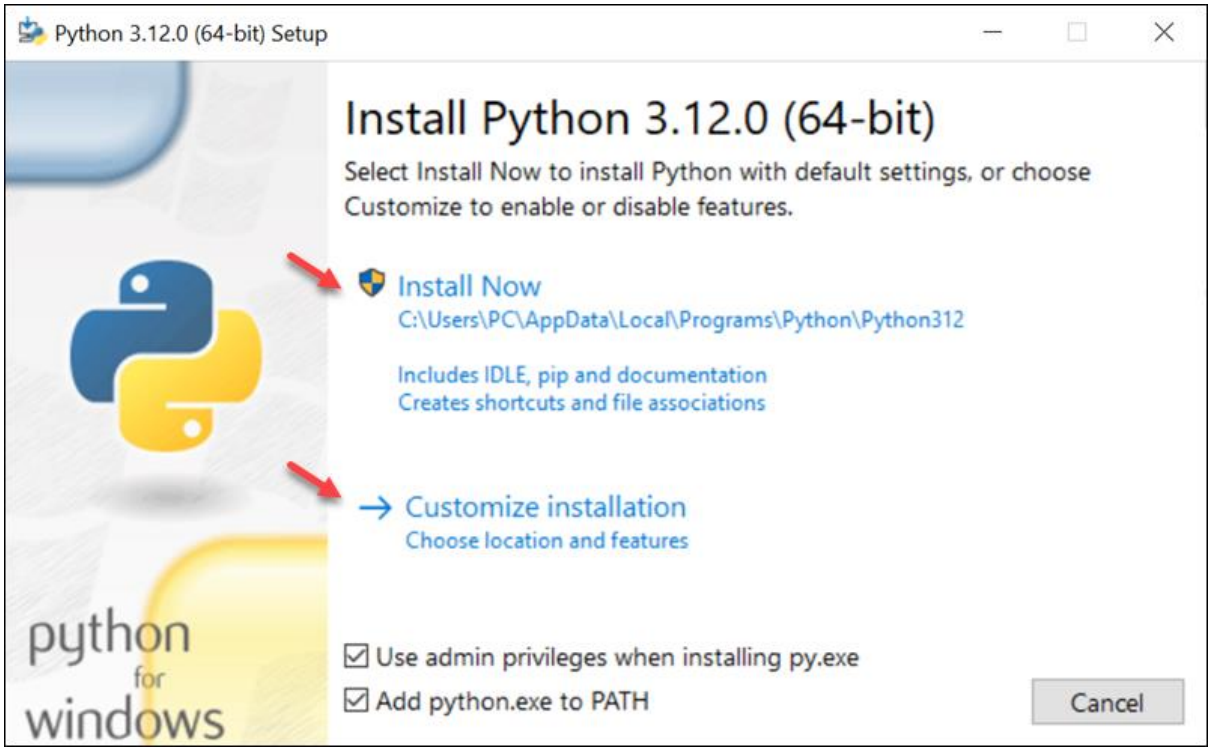
- **Admin privileges**. The parameter controls whether to install Python for the current or all system users. This option allows you to change the installation folder for Python.
- Add Python to PATH. The second option places the executable in the PATH variable after installation. You can also add Python 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).

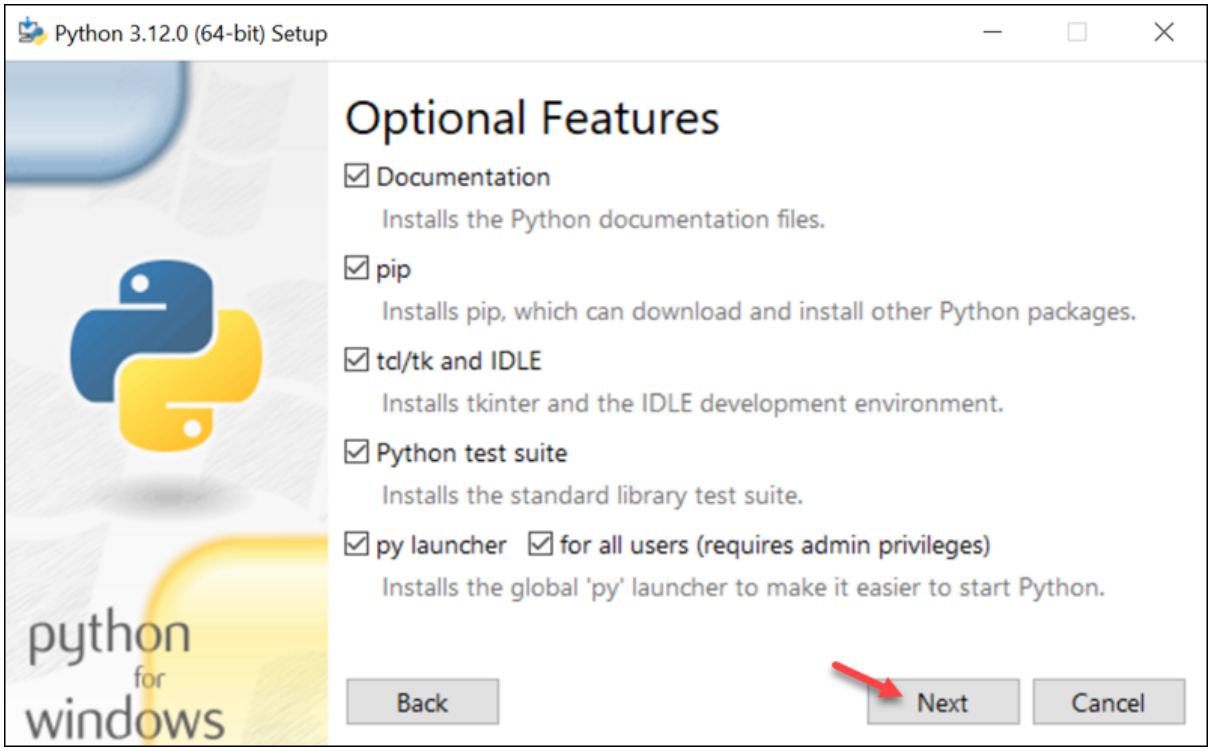
To adjust the default installation options, choose **Customize installation** instead and proceed to the following step.



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.

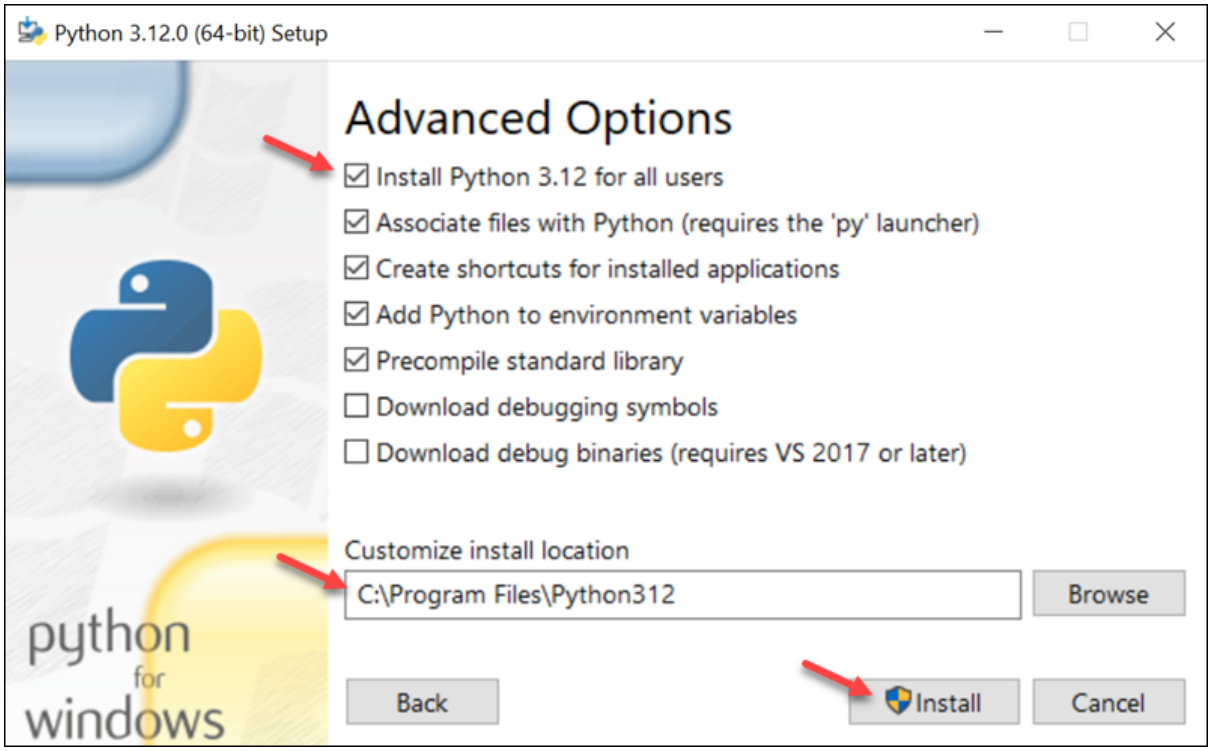


Click **Next** to proceed to the Advanced Options screen.

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

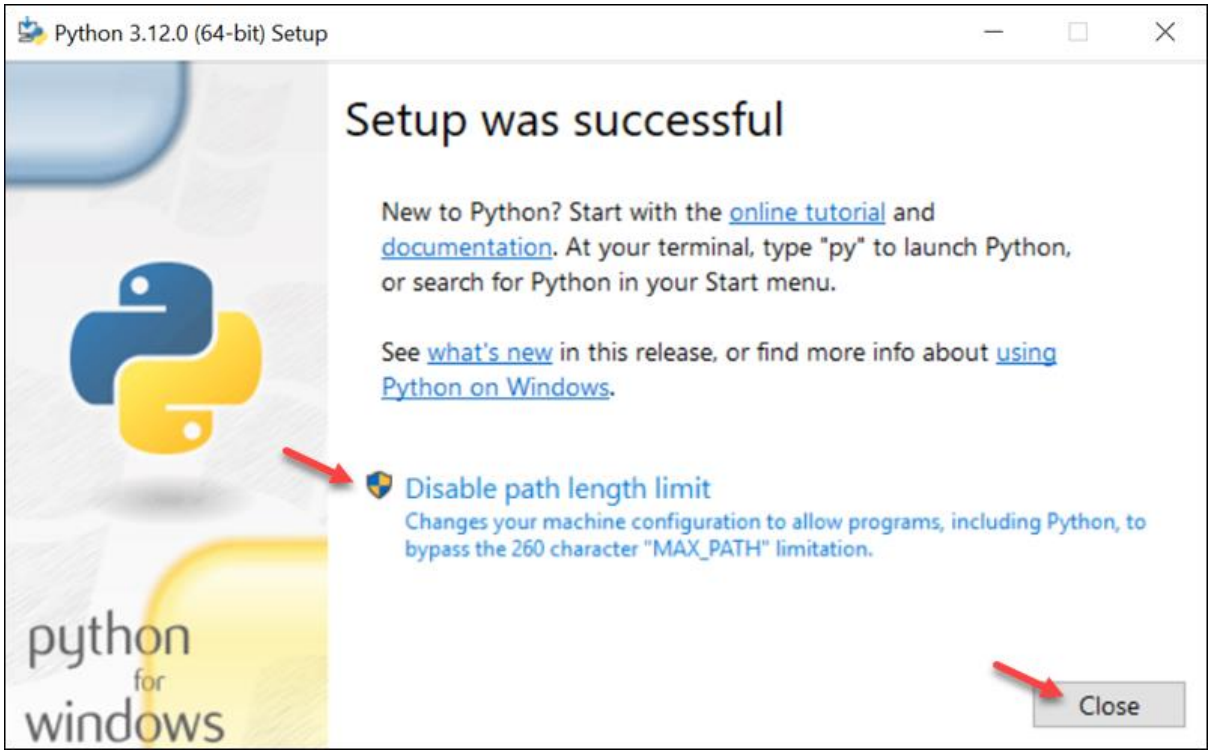
Choose whether to install Python for all users. The option changes the install location to *C:\Program Files\Python[version]*. If selecting the location manually, a common choice is *C:\Python[version]* because it avoids spaces in the path, and all users can access it. Due to administrative rights, both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

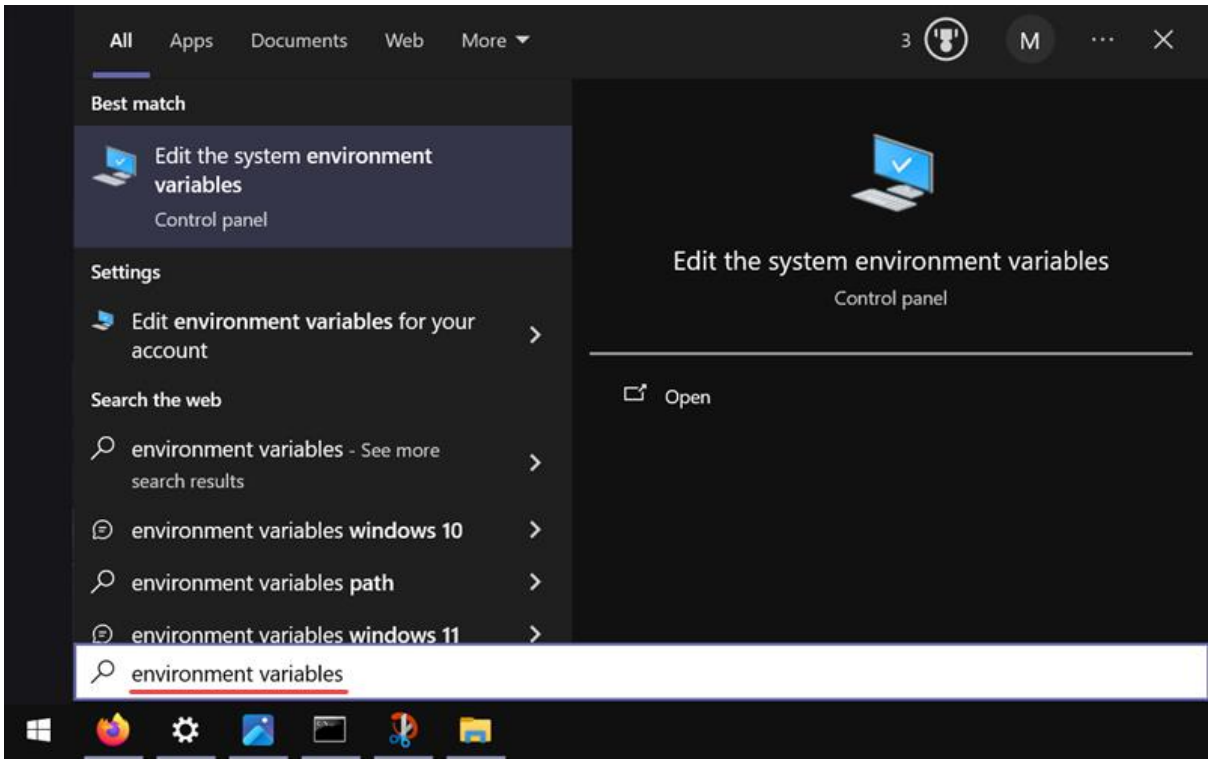
Step 4: Add Python to Path (Optional)

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.

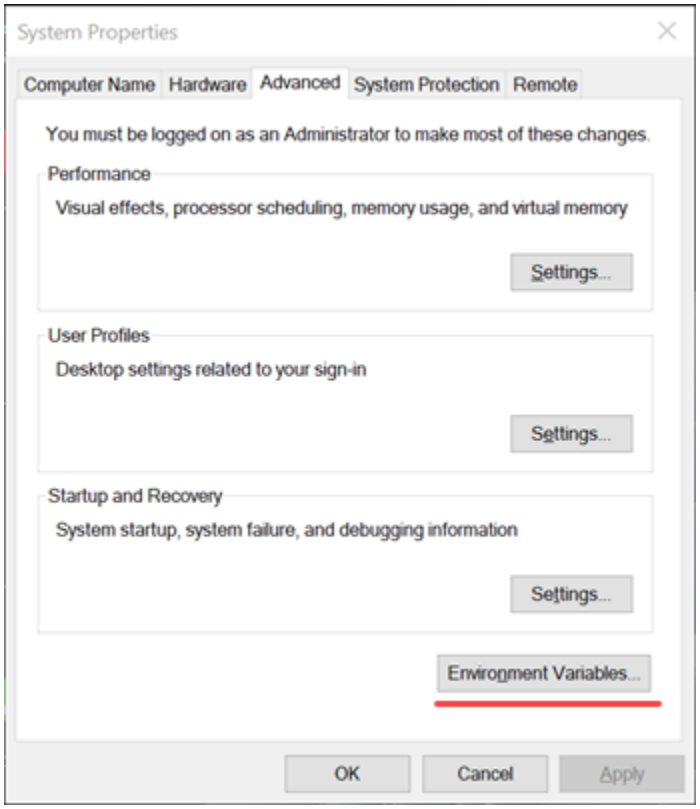
Adding the Python path to the PATH variable alleviates the need to use the full path to access the Python program in the command line. It instructs Windows to review all the folders added to the PATH environment variable and to look for the *python.exe* program in those folders.

To add Python to PATH, do the following:

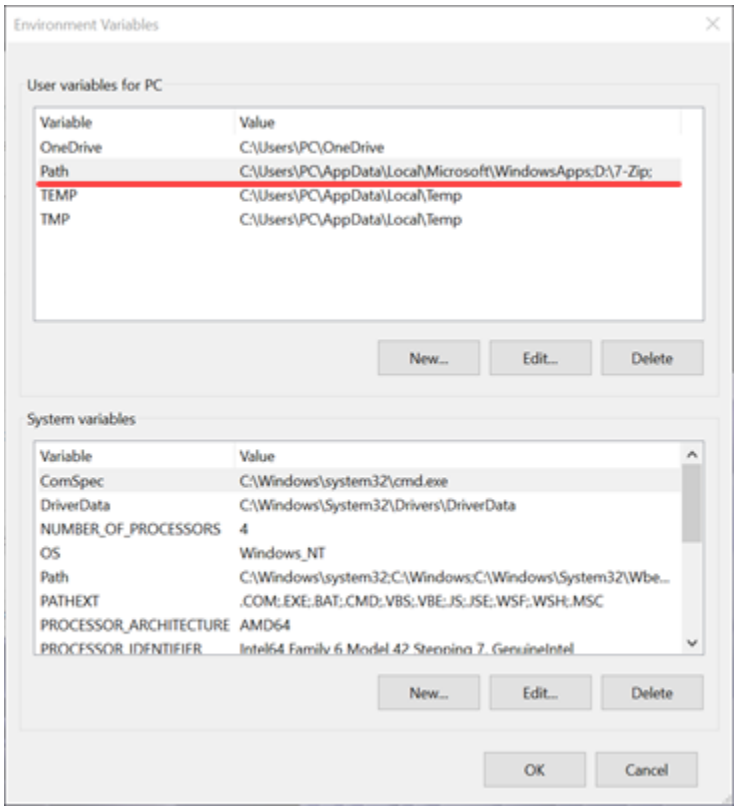
1. In the **Start** menu, search for **Environment Variables** and press **Enter**.



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

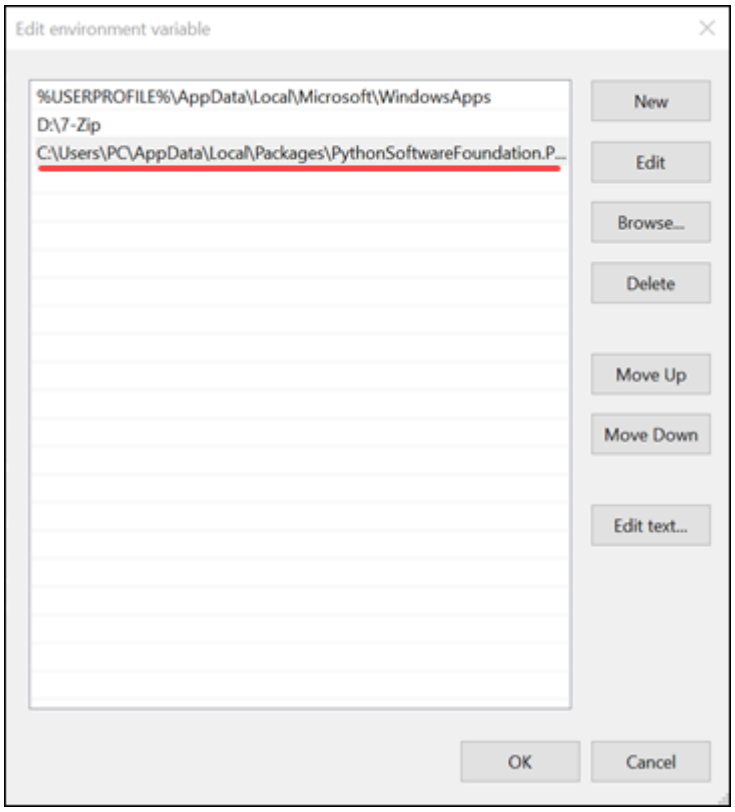


3. Double-click **Path** on the list to edit it.



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
```

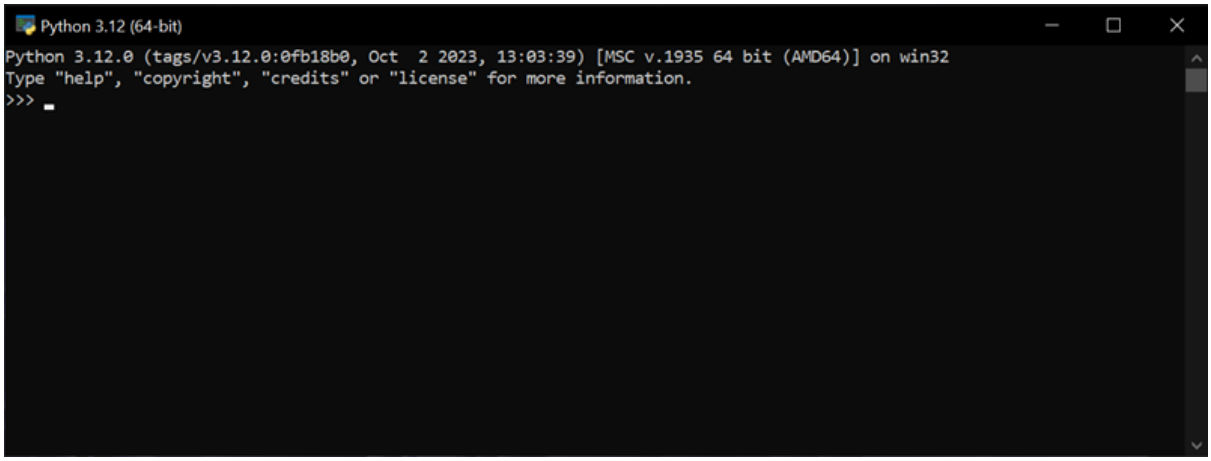
The output shows the installed Python version.

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

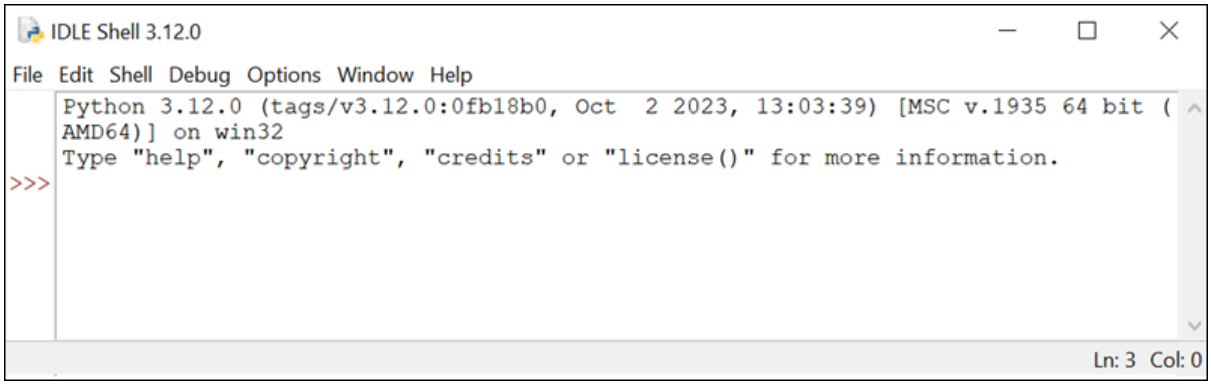
1. Navigate to the directory where Python was installed on the system.

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.

Step 6: Verify PIP Was Installed

To verify whether PIP was installed, enter the following command in the command prompt:

```
pip --version
```

If it was installed successfully, you should see the PIP version number, the executable path, and the Python version:

PIP has not been installed yet if you get the following output:

```
'pip' is not recognized as an internal or external command,
Operable program or batch file.
```

If an older version of Python is installed or the PIP installation option is disabled during installation, PIP will not be available. To install PIP, see our article How to Install PIP on Windows.

Step 7: Install virtualenv (Optional)

Python software packages install system-wide by default. Consequently, whenever a single project-specific package is changed, it changes for all your Python projects.

The `virtualenv` package enables making isolated local virtual environments for Python projects. Virtual environments help avoid package conflicts and enable choosing specific package versions per project.

To install `virtualenv`, run the following command in the command prompt:

```
pip install virtualenv
```