

Installing Python (Windows)

For Windows:

If your laptop is new, and you have never used it to compile a Python file, it is most likely that it does not have it installed already, but it is always good practice to check first.

1. Open your Windows terminal. For Windows, you can access different types of terminals (Windows PowerShell, Windows Terminal, Command Prompt). They have different uses, but for this, we will look for the command prompt. It should look something like this:

comand_prompt

When you open it, on the screen, you can write `python --version` or `python -V`. If you have Python installed in your computer, is going to show you a message of the version and some features for your computer.

KEEP IN MIND:

If the version you have on your computer is not the newest one, do not worry about it. We will discuss how we can manage that with Python environments.

CP_pythonVersion

If you do not have Python installed, the prompt will take you automatically to the Microsoft Store. If not, you can follow the next steps.

Microsoft Store install (Easiest):

If you are new to Python and looking to get started quickly, you can install it directly on the Microsoft Store page. If you followed the last steps to check if you have Python and you do not have it on your computer, most likely, the Microsoft Store app launched on its own. If not, you can search for it in the search bar. It should look something like this:

MS_python

IMPORTANT

Please ensure the application you select is created by **Python Software Foundation**. The official software is free, so if the application costs money, then it is the wrong one. Select the newest version.

Once you have found the version you need to install, click get, and wait for the application to finish downloading. The `Get` button will then be replaced by `Install on my devices`, where you can decide if you want to install only on the current user or all the computers. After you select them, click install.

IMPORTANT

Please keep in mind that this is only for first-time installations. If you have Python already and want to upgrade it to a newer version, it is a completely different process that we will discuss in later posts.

If the installation was successful, you will see the message `This product is installed`. You can now access Python, including `pip` and `IDLE`. This allows you to run Python scripts in your terminal directly.

Downloading the installer (intermediate):

The installation via the official page for Python is suited for more experienced developers, as it offers a lot more customization and control over the installation.

IMPORTANT

Please note that this installation requires you to have previous knowledge of how `PATH` works. If you do not know what `PATH` is, we strongly recommend you use the Microsoft Store Package instead of this installer.

Once you download the `.exe` file of the latest Python version (right now, it is not important to look for different versions), follow the installation guide. Remember to select either the Windows x86-64 executable installer for 64-bit or the Windows x86 executable installer for 32-bit, depending on your own computer specifications. You can see which one you have by following the next steps:

1. Click the start button at the bottom left corner.
2. Select or search for `Settings`

Windows_screen

3. In `Settings`, select the `System` tab.

systems_page

4. Scroll all the way down on the left panel and click "About". The information you need is under "Device specifications" in the System type.

System_about

IMPORTANT

Please note that drivers made for the 32-bit version of Windows will not work correctly on a computer running on a 64-bit version and vice versa.

Now that we have the correct installer for your computer run the file. A dialogue window will appear, and there are some different things we can do with it.

Python_setup

- The default path for installation is in the directory of the current Windows user. The Install launcher for all users (recommended) checkbox is checked default. This means every user on the machine will have access to the py.exe launcher. You can uncheck this box to restrict Python to the current Windows user.
- There is another checkbox that is unchecked by default called Add Python to PATH. There are several reasons why you might not want Python on PATH, so make sure you understand what this does before clicking on it.

IMPORTANT

If you do not know what PATH is, we strongly recommend you use the Microsoft Store Package instead of this installer. The Microsoft package is directed to people new to Python and focused primarily on learning the language rather than building professional software.

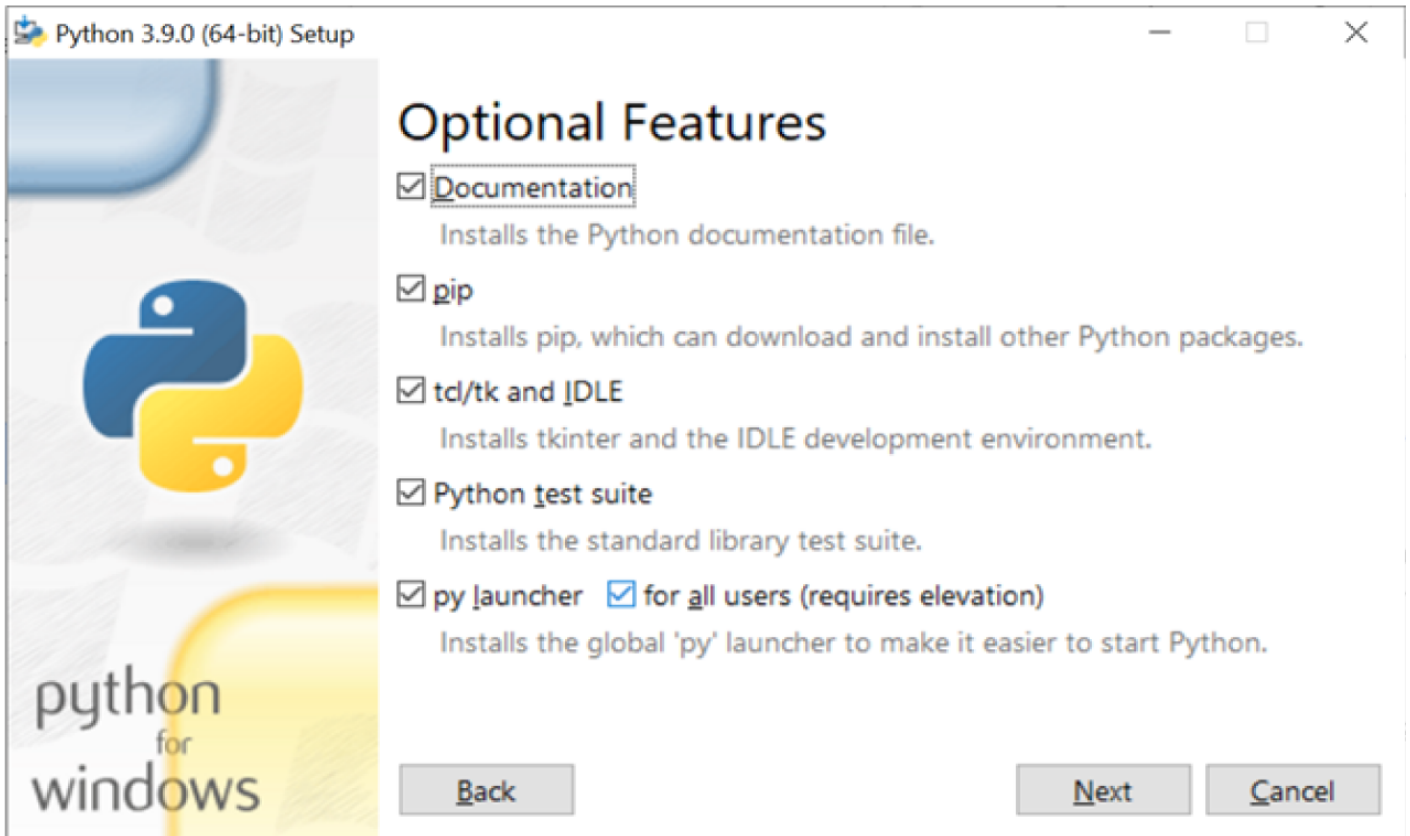
If you choose the customised installation, you have other different features that can be installed.

KEEP IN MIND

This option requires you to provide administrative credentials. If this is not your personal computer, you may not have the correct access.

This allows developers to have more control over other optional features for the manual installation:

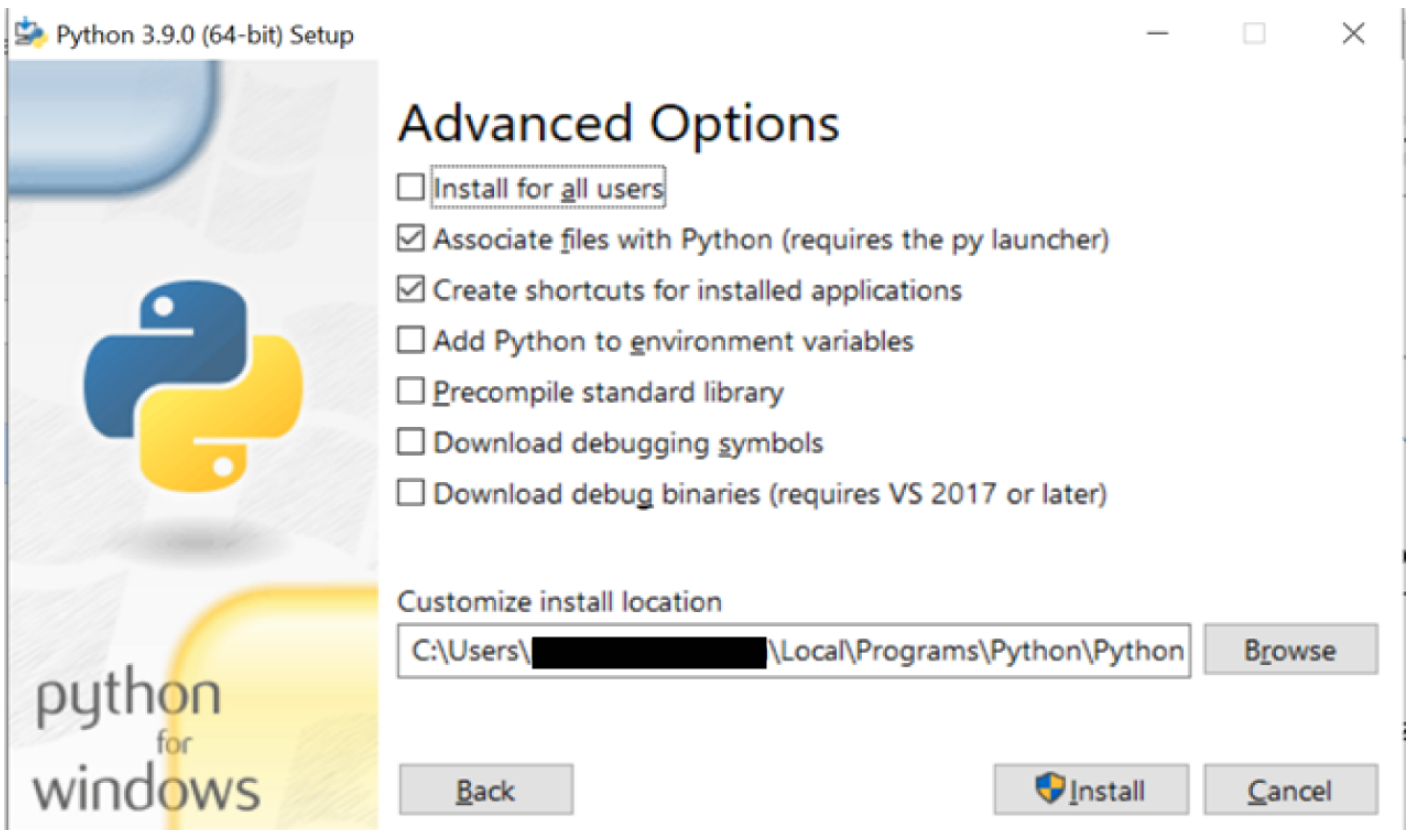
We will explain the different optional features and a general idea of what they represent, but if you do not understand or are unfamiliar with any of them, please go back to the automatic installation.



For each option, we have:

- **Documentation:** Will download the documentation of the Python version you are installing. All this information is also available on the Python documentation page, so it is unnecessary.
- **Pip:** Pip is a tool that helps to fetch packages for Python. When you install Python, you only have the standard library available, so you need a package manager to install specific libraries. Pip is the official supported one. We will discuss different package managers in later posts.
- **IDLE:** This stands for **I**ntegrated **D**evelopment **L**earning **E**nvironment. This function will install the "tkinter" toolkit, a Python default GUI (**G**raphical **U**ser **I**nterface) package.
- **Python test suite:** This installs all the standard libraries for Python application testing. The test package is meant for internal use by Python only. It is documented for the benefit of the core developers of Python. Any use of this package outside of Python's standard library is discouraged.
- **Py launcher:** Enables you to launch Python CLI (**C**ommand **L**ine **I**nterface) like command prompt or Windows shell.
- **For all users:** Same as the normal installer, it allows you to install the Python launcher for all system users.

After clicking next, we will have the last customizable screen, where developers can check all the additional features. Some may have already been checked depending on the choices made before.



- **Install for all users:** This installs the Python launcher for all the users in your system.
- **Associate files with Python:** This option will link all the files Python extensions like `.py`, `.pyd`, `.pyc` or `.pyo` with the Python launcher or editor.
- **Create a shortcut for installed applications:** It will automatically create a shortcut for applications on the Desktop.
- **Add Python to environment variables:** PATH is an environment variable that holds values related to the current user and operating system. It specifies the directories in which executable programs are located. So when you type Python, Windows gets its executable from PATH. Hence, you won't need to type the whole path to the file on the command line.

The last three options tell the Python installation wizard to install all the debug symbols and binaries along with the bytecode of all the standard libraries, which we will be using with our programs. Finally, we can change the location of the Python installation directory. The location you specify will be added to the environmental variable of our Windows operating system.

Installing it lets you go to your terminal and execute a Python script. Assuming that you know how to deploy and work with the terminal (command tool) in your computer, once you have an interpreter installed, you can run a Python code by going to the folder where the py file is stored (from the terminal), and run:

```
python my_script.py
```

Being 'my_script' the name of our dummy code. This will instruct the Python interpreter stored on your computer to read that file. Assuming that the code obeys the grammatical rules of Python and the instructions in the code have to provide an output, this will appear in the terminal window.

Again, this is an example of a very simple code. Some codes also require input from the user, and for that, we will need further tools.

Revision #3

Created 22 January 2024 11:02:32 by Mayra Cristina Berrones Reyes

Updated 29 April 2024 22:22:38 by Mayra Cristina Berrones Reyes