

Installing Python on Windows 10

Romit Maulik

September 2017

Different individuals have different preferences when it comes to their development environment of choice. Personally, I have found the PyCharm IDE most intuitive and productive for my needs. This document will help a new Python user install Python (2.X or 3.X) on their Windows machine, integrate it with PyCharm, download commonly used discrete math libraries (Numpy, Scipy) and plotting tools (Matplotlib). Finally, we can implement a 1D Heat equation problem using the method of finite differences by using the aforementioned libraries in PyCharm. A video detailing the installation of Python and PyCharm can be found at <https://www.youtube.com/watch?v=puBXzcWJIQ>.

1 Setting up Python

Assuming you have no distribution of Python on your Windows machine at this point of time, the following steps can help you install Python on your system:

- Go to <https://www.python.org/downloads/> and download your choice of a Python release. This example is built using Python 2.7.13 but should work with no major issues for Python 3.6.2 (most up to date releases at the time of the drafting of this document). You will now have a *.msi file in your choice of the download directory.
- The *.msi file is essentially an executable installer and will proceed to install python (2.X or 3.X) to a desired destination (preferably C:). An **important** point to remember here is to add Python to PATH. This will save us a lot of headache later. You should then receive a “Setup was successful” message after this.
- Validate the proper installation of Python by opening up a Command Prompt in Windows and typing `python`. You should receive a message showing your Python release as well as the architecture of choice (win32 or win64). You may exit this command line by typing `exit()`.
- Now we install PyCharm. Go to <https://www.jetbrains.com/pycharm-edu/download/#section=windows> to install the free community edition for PyCharm (currently version 4.0). Run the installer and follow the prompts (make sure to associate .py extensions with PyCharm).
- Run PyCharm (search for it in your main menu). For this first time, it will ask you if you want to import settings: I assume that you have no settings to import at this time and you can proceed without any import. It then allows you to edit the Keymap scheme and IDE theme (mainly for aesthetics). You can proceed with the default or use the preview to choose something you fancy.
- You should now obtain a screen which says “PyCharm Community Edition” with options for creating or opening projects. At this stage, click the ‘Configure’ button and go to ‘Settings’.
- Go to the option title ‘Project Interpreter’. Click the settings icon and then ‘Add Local’ and navigate to your installation of Python and point to ‘python.exe’. If everything has worked so far then we are in good shape.

2 Hello World!

- Click to open PyCharm and create a new project with an appropriate directory name (avoid starting this with numbers) at a location of your choice.
- On the left side of your window you will have a panel showing a folder icon with your project name and 'External Libraries'. Right click on the folder for your project name and select 'new -> Python File'.
- Name your file Hello.py . You should now have the screen turn into a different color letting you type on it.
- Type `print("Hello world!")` and run the file by right clicking and selecting run. If everything has been done correctly, you should now have the console show you 'Hello world!'.

3 Installing Numpy and Matplotlib

By default your installation of Python should come with the 'pip' installer. With this tool - you may install any libraries of your choosing from <https://pypi.python.org/pypi>. At the time of the making of this document there were 117060 packages here. There is a very very high probability that whatever tool you need is already stored on these repositories.

- Open up a Windows command prompt or powershell and type and enter `pip install --upgrade pip`. This upgrades the pip installer to the latest version before using to download libraries from the online repository.
- You can now type and enter `python -m pip install numpy` which installs numpy on your machine. You can follow this up with `python -m pip install matplotlib` which installs matplotlib for plotting.
- In a command prompt run Python by typing and entering `python`. Once the python command line is active, type and enter `import numpy as np`, followed by `np.version.version` which should give you the version of numpy you have installed.
- Exit the python command line by type `exit()`. You are now ready to do scientific computing (for the most part) in Python!

4 Testing Numpy and Matplotlib

Follow the steps in Section 'Hello World!' to create a new project and paste the following code into a new python file.

```
import numpy as np
import matplotlib.pyplot as plt

y = np.random.rand(10)
x = np.linspace(0,9,10)

plt.figure(1)
plt.interactive(False)
plt.xlabel('x')
plt.ylabel('y')
plt.title('My First Scatter ')
plt.scatter(x, y)
plt.show()
```

If you obtain a scatter plot of random numbers you are good to go.