Python Installation Instructions

**Contents**

[1 Introduction 2](#_Toc80632250)

[2 Install Python 2](#_Toc80632251)

[3 PyCharm 5](#_Toc80632252)

[3.1 Download & Install 5](#_Toc80632253)

[3.2 Create a New Project 10](#_Toc80632254)

[3.2 Install Packages 12](#_Toc80632255)

[4 Jupyter Notebook Installation Tutorial 14](#_Toc80632256)

[3.1 Understanding Jupyter Notebook 25](#_Toc80632257)

# 1 Introduction

This document is designed to teach you how to install Python and IDEs in order to write and execute scripts in Python. This tutorial will cover Linux, Windows, and Mac.

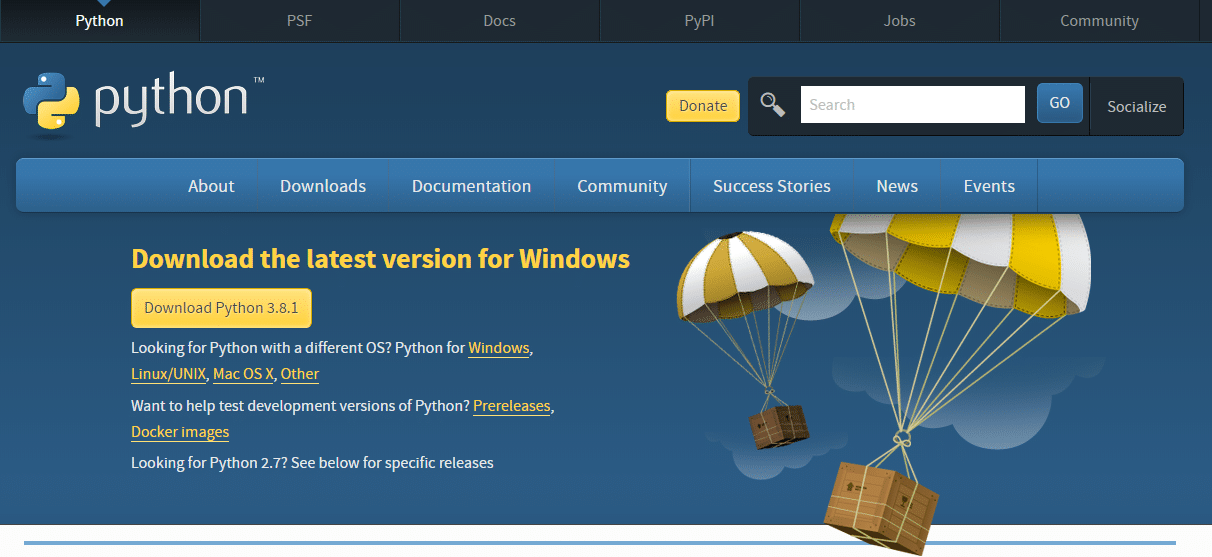
Before installing an IDE, you must first install Python. For Mac and Windows, navigate to <https://www.python.org/downloads/release/python-385/>for the version 3.8.5 download. At the bottom of the page, select the appropriate installer file for your operating system and architecture.

# 2 Install Python

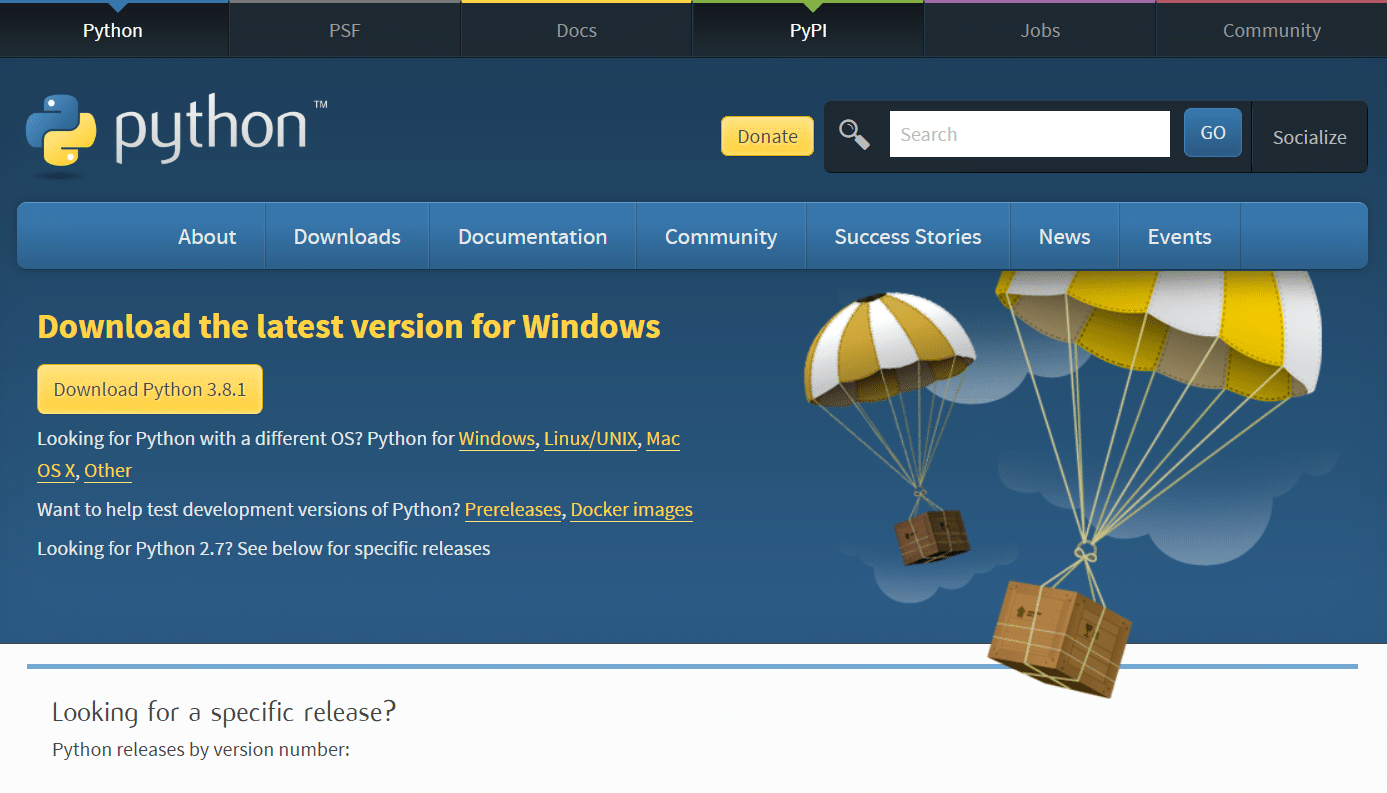
Steps to Installing Python

Step 1: Go to www.python.org

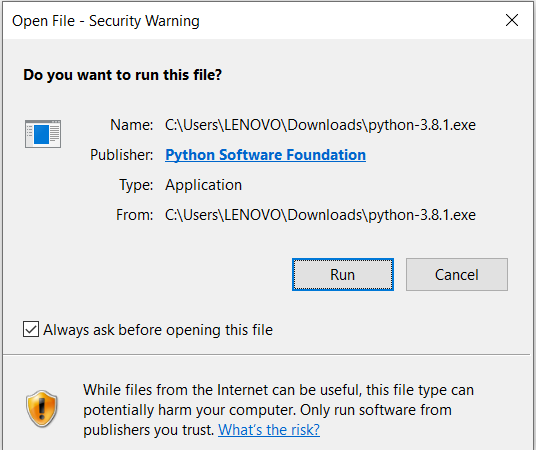
Step 2: Select ‘Downloads’ from the toolbar



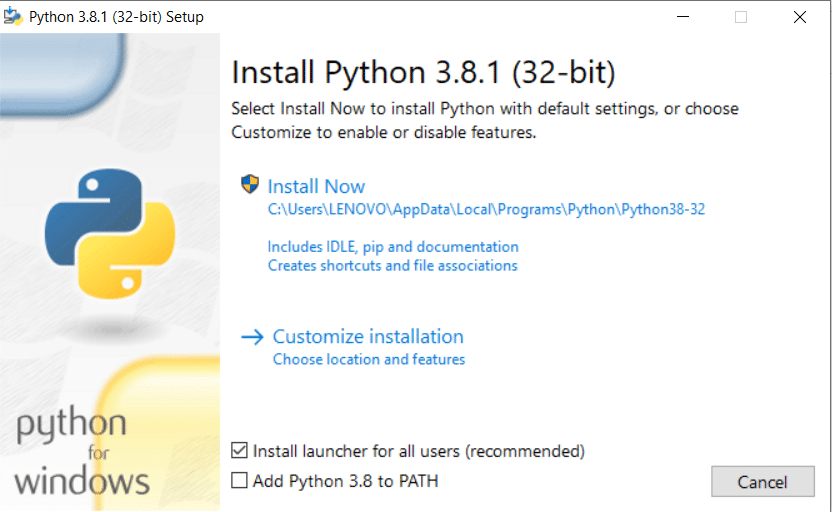
**Step 3**: Click on ‘Download Python 3.8.1’ or the latest version available



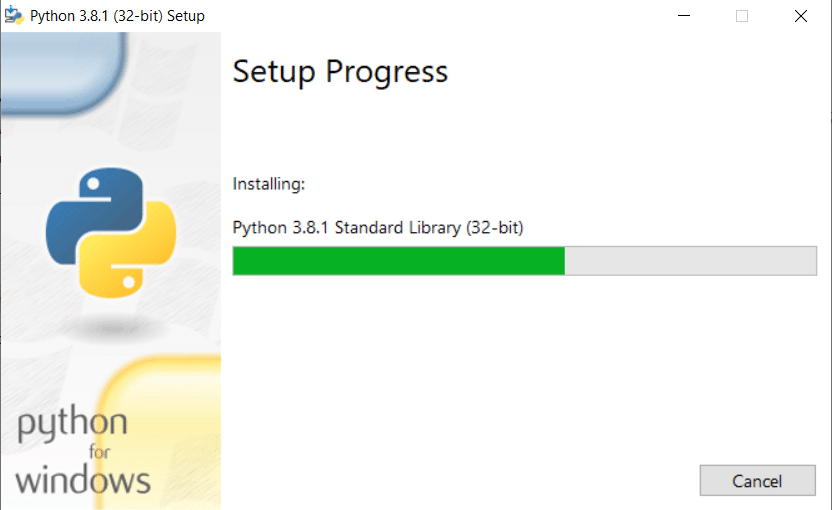
Step 4: Then, go to the File option. After that, a security dialog box will appear as shown below. Click on ‘Run’ to continue the installation process.



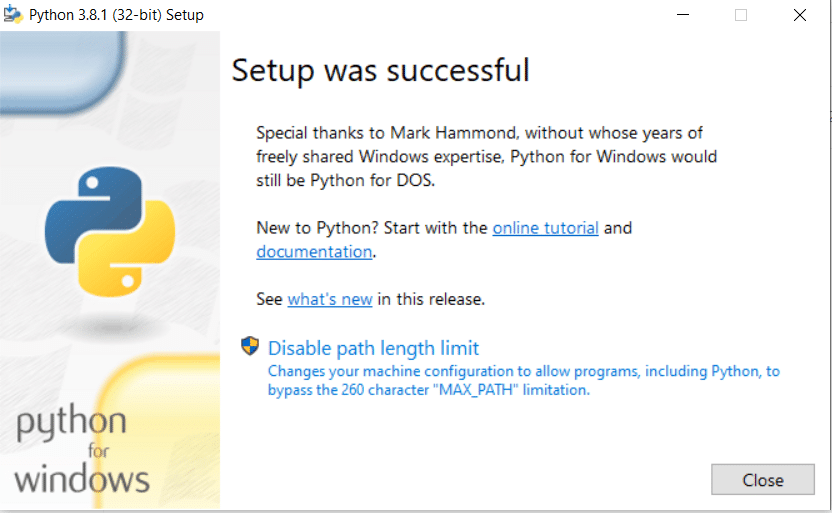
Step 5: Click on ‘Install Now’



Once you do that, you can see the setup in progress as in the below screenshot:



Step 6: After the installation of Python, when you see a window with the message ‘Setup was successful’, click on the ‘Close’ button.



Now, you are ready with Python 3.8.1 installed in your system.

Further, we will move on to the installation of PyCharm.

# 3 PyCharm

For developing applications in Python using PyCharm, you need to install it in your system.

PyCharm is available in two versions:

* **Professional version**: It is a full-featured IDE used by professional developers.
* **Community version**: It is an open-source and free software that is not full-featured. You can use it for learning Python application development.

The Professional version is a paid tool and comes with a free trial. You have to buy a license key for it after your free trial period is over.

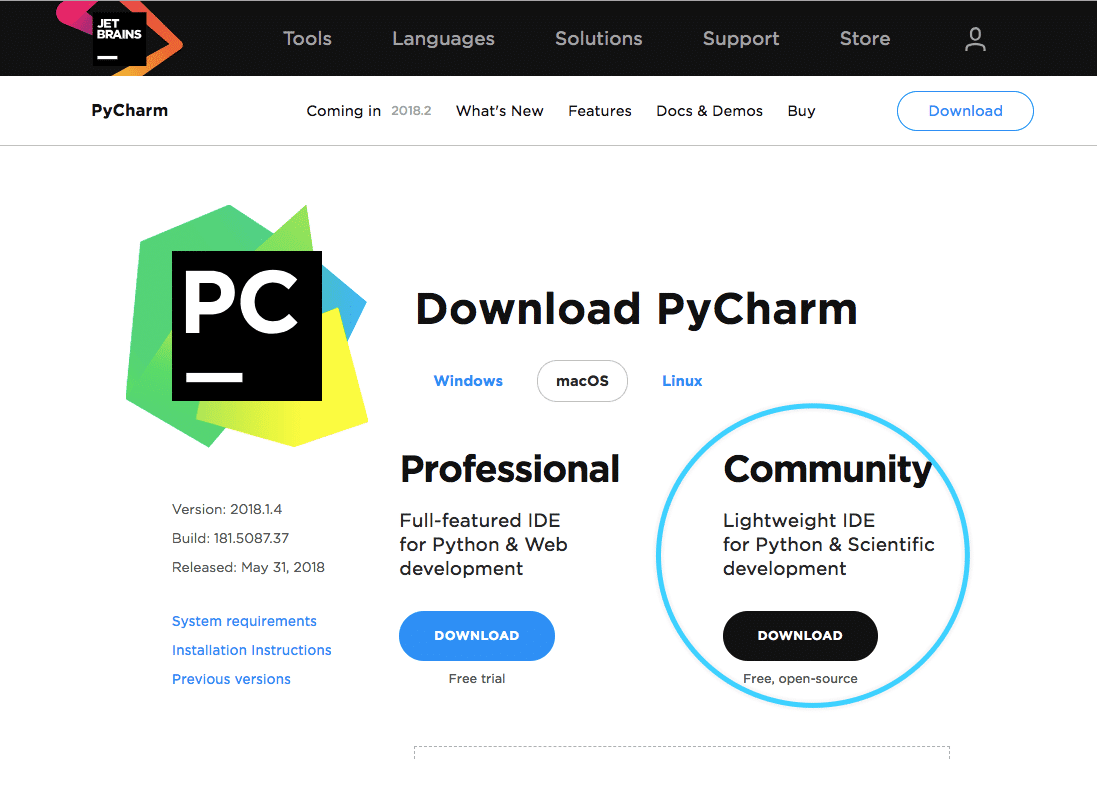
The Community version, on the other hand, is open-source without any subscription charge.

# 

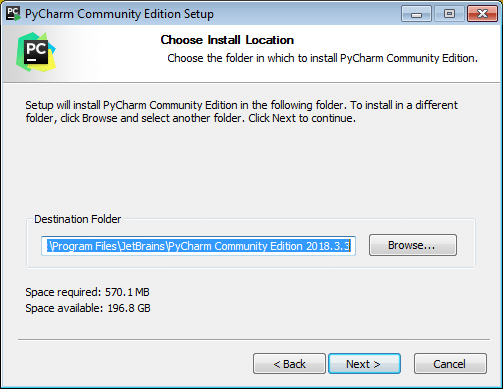
## Download & Install

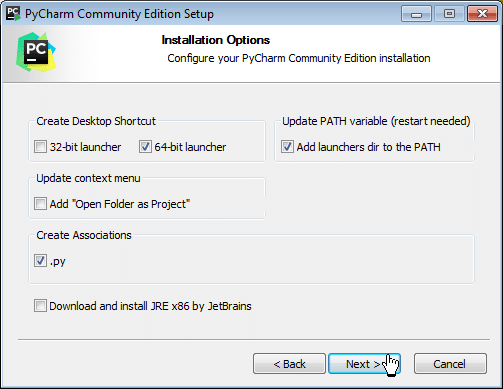
* + 1. Navigate to <https://www.jetbrains.com/pycharm/download/>.
    2. Select the tab for your operating system (Windows, Mac, or Linux).

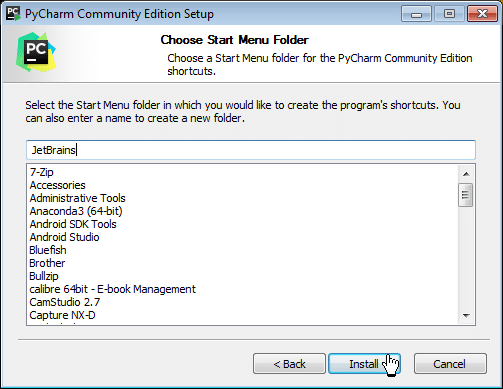
1. Under the *Community* header, press the **Download button**. The Professional version has some extra support, but requires a paid license. The download should automatically start.

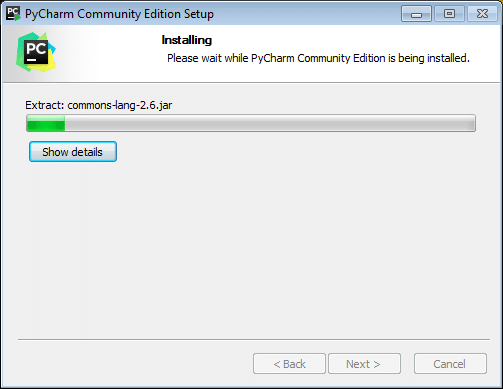


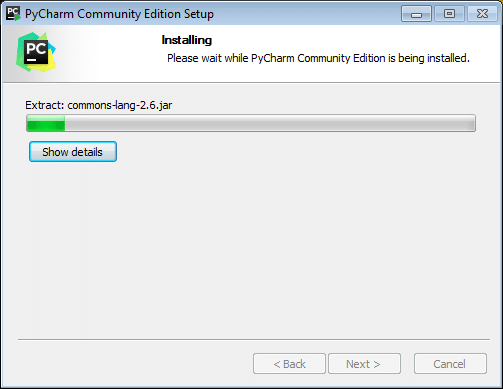
1. If you are on Windows, run the executable and press **Next** until you reach the **Install** button. On the second to last page, you can select some options (e.g., create file associations with Python scripts ending in the .py file extension), but none of these are required. Finally, follow the prompts (optionally select theme and featured packages) to finish the installation.











1. **If you are on Linux**, extract the .tar.gz directory and follow the instructions in the

Install-Linux-tar.txt file:

* + Navigate to the bin/ directory.
  + Run the PyCharm bash script with the command ./pycharm.sh.
  + Follow the prompts to finish the installation. None of them are required and you can instead click the skip button in the bottom left to complete.
  + If you chose to create a global command, you can use the command charm to run PyCharm. Though, a welcome window should open up after the installation completes.
  + If you did not choose to create a global command, you can instead run the command ./pycharm.sh from the download directory again and this will run PyCharm as it detects it is already installed. If your installation was successful and you chose to open PyCharm, you should now have a window as show



## 3.2 Create a New Project

* + 1. When first running PyCharm, you should see an window as shown in [Figure 1.](#_bookmark3)
    2. To create a new project, press the **New Project** button. If you have already created a new project, select the menu option File > New Project. Following either method, a window as shown in [Figure 2](#_bookmark5) will open. The location values will be different based on your operating system (these screenshots are for Linux).

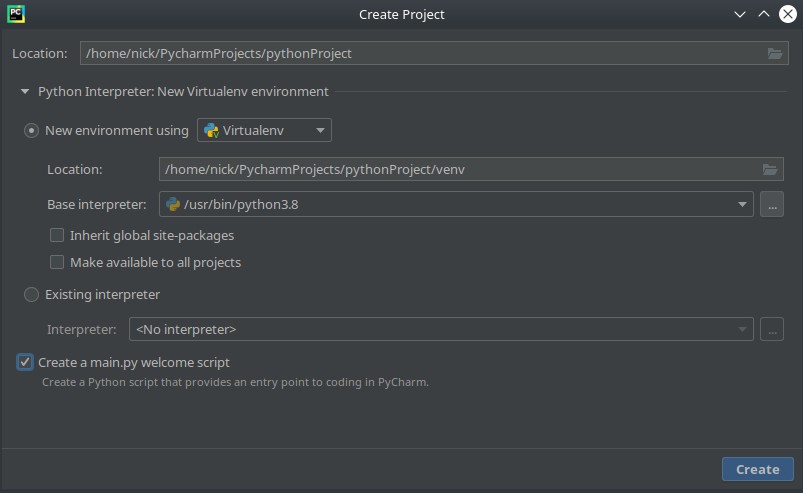


Figure 2: PyCharm create project window.

* + 1. Change the directory as desired in the Location: field. Since we are using Python 3.8, we can leave everything else default. If you wish to initialize a main.py script file, leave the last check box selected. If you wish to create your own, uncheck this box. When you are done, click the **Create** button.
    2. PyCharm should now open with a view as shown inFigure [3.](#_bookmark6)

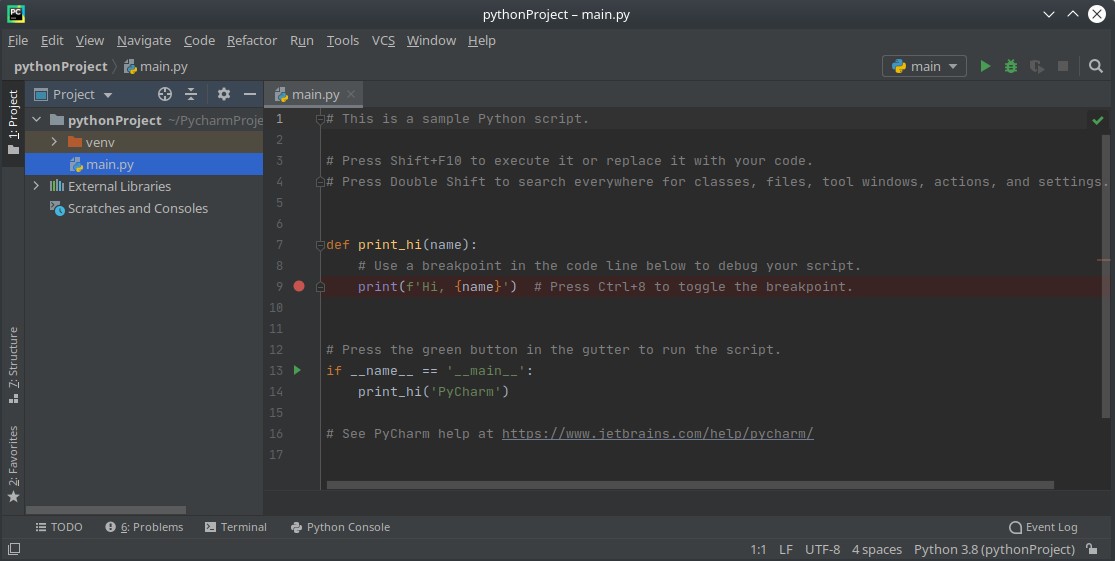


Figure 3: PyCharm development view.

* + 1. The project view is in the left pane. This will show your current project and its source files. The main window is your editor. If you kept the create a main.py welcome script option checked, your project will contain this script and the editor will show its contents. This simple script calls the print\_hi() function and prints the text Hi, PyCharm to the output.
    2. To run the script, click on the green right-facing triangle in the upper left corner or press Shift+F10. A new view will show up on the bottom of the window with the output of the program. This view can be opened and closed by pressing Alt+4 or through the menu option View > Tool Windows > Run. The output of running this script is shown in[Figure 4.](#_bookmark7)

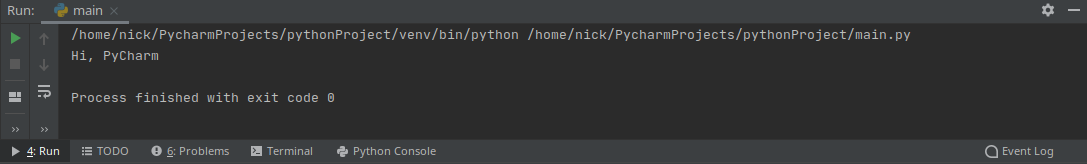


Figure 4: Output from PyCharm’s main.py welcome script.

* + 1. If your script successfully completes, the output will end with exit code 0. If unsuccessful, PyCharm will display an error message in the output along with the line it occurred on and the process will exit with code 1.
    2. To create new projects, go through the menu option File > New Project and follow the above steps. To add more scripts to your project, right-click on the project folder in the project view and select New > File.

## 3.2 Install Packages

1. To install packages for your project, navigate to the settings page in

File > Settings or use the shortcut Ctrl+Alt+S.

1. Navigate to Project:$(your\_project) > Python Interpreter.

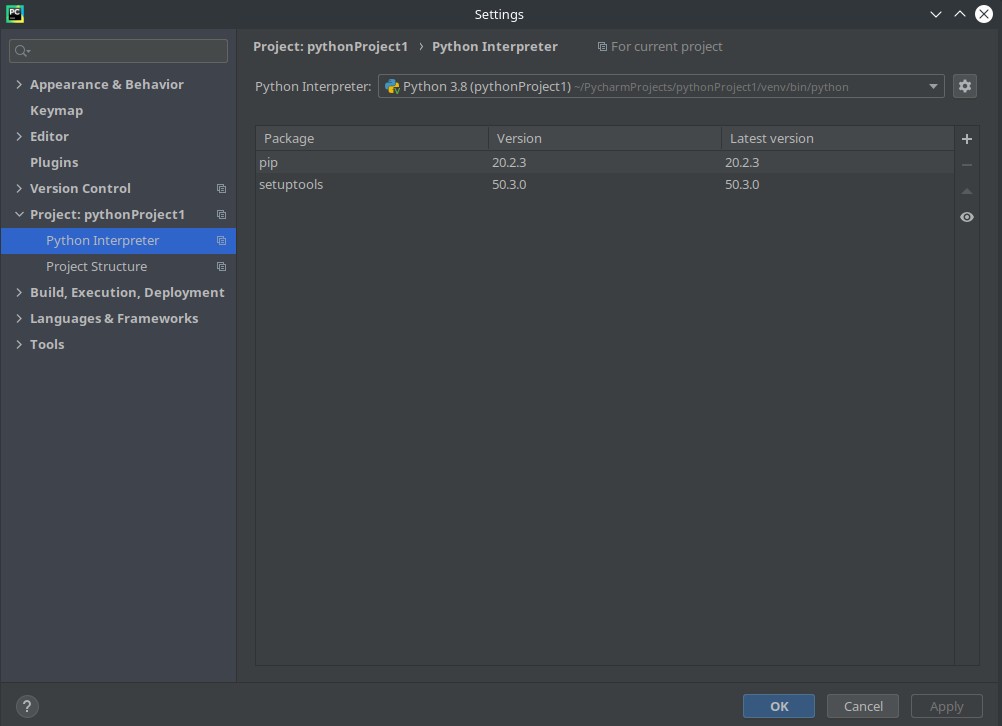


Figure 5: Package list view for the current virtual environment.

1. Click the + button in the upper right to open the available packages dialog.
2. In the available packages dialog, the list of all available packages is shown. Type the name of the package you want to install into the search field. The list will now show matching packages only. The search results for the numpy package are shown in Fi[gure 6.](#_bookmark9)

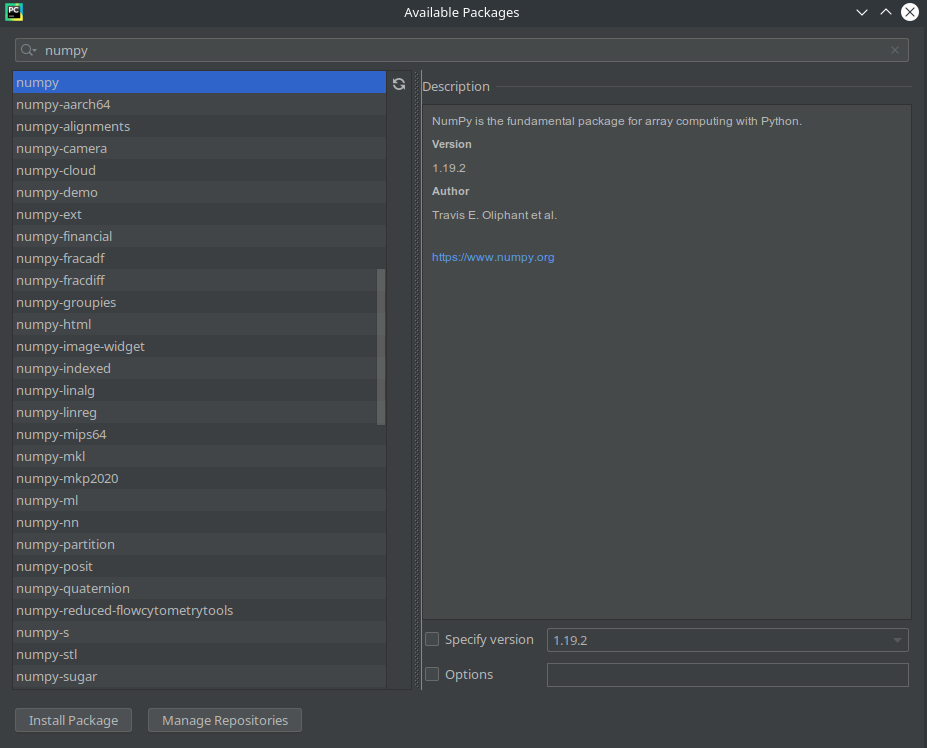


Figure 6: Search results for the numpy package.

1. To install the selected package, click the **Install Package** button.
2. Once the success message appears, you can close the available packages view and close the settings window.
3. To install packages more quickly, first type the import statement in your script. If the package is not already installed, the package name will be underlined with a red squiggly line.
4. Hovering over the underlined package for a short duration will display a popup box. Move your mouse directly onto the box and click the **Install Package $(package\_name)** button. Alternatively, press Alt+Shift+Enter when the popup box displays to install the package.

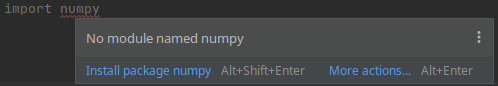


Figure 7: Search results for the numpy package

YEAH, Congratulations! Now, you can start to create a new project. Good luck!

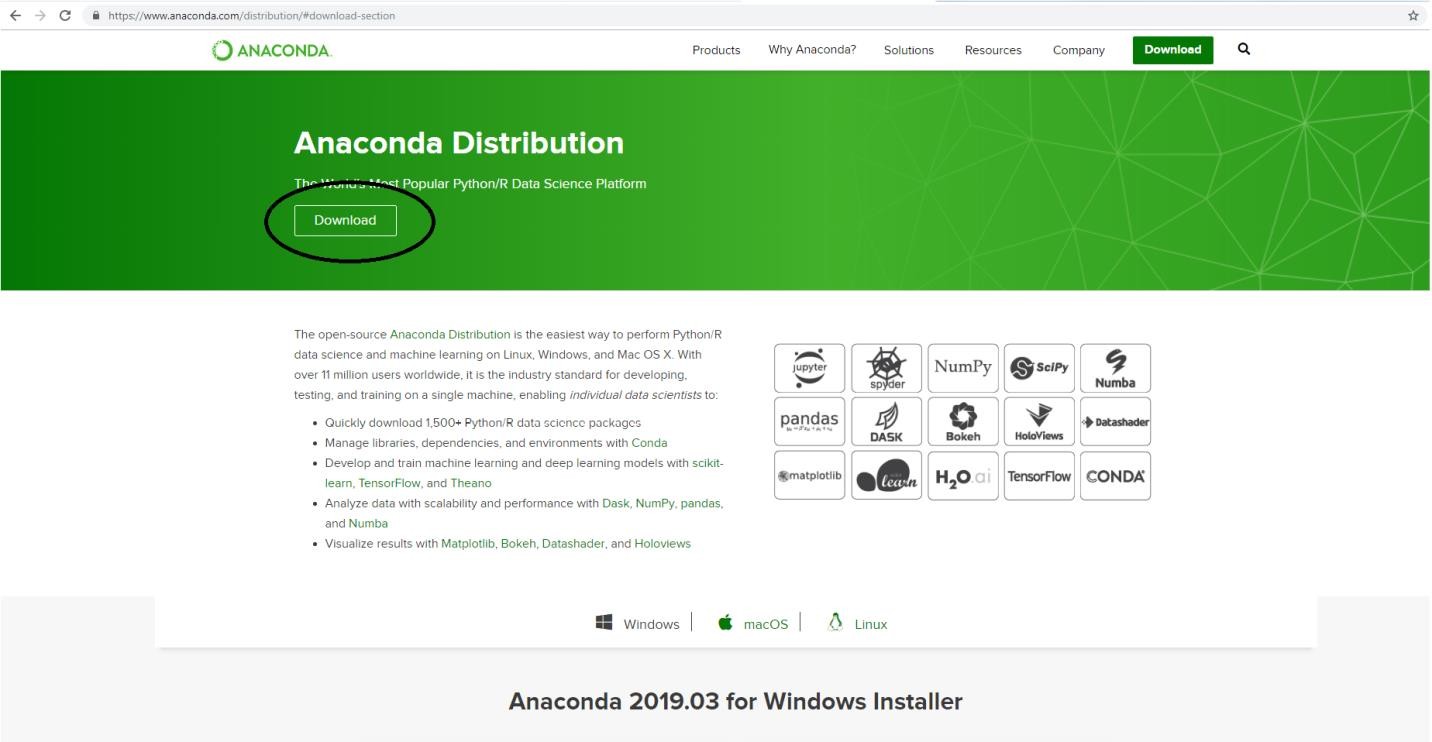
# 4 Jupyter Notebook Installation Tutorial

In this tutorial, we will show how to install Jupyter Notebook on your system. **The Jupyter Notebook is free of cost.**

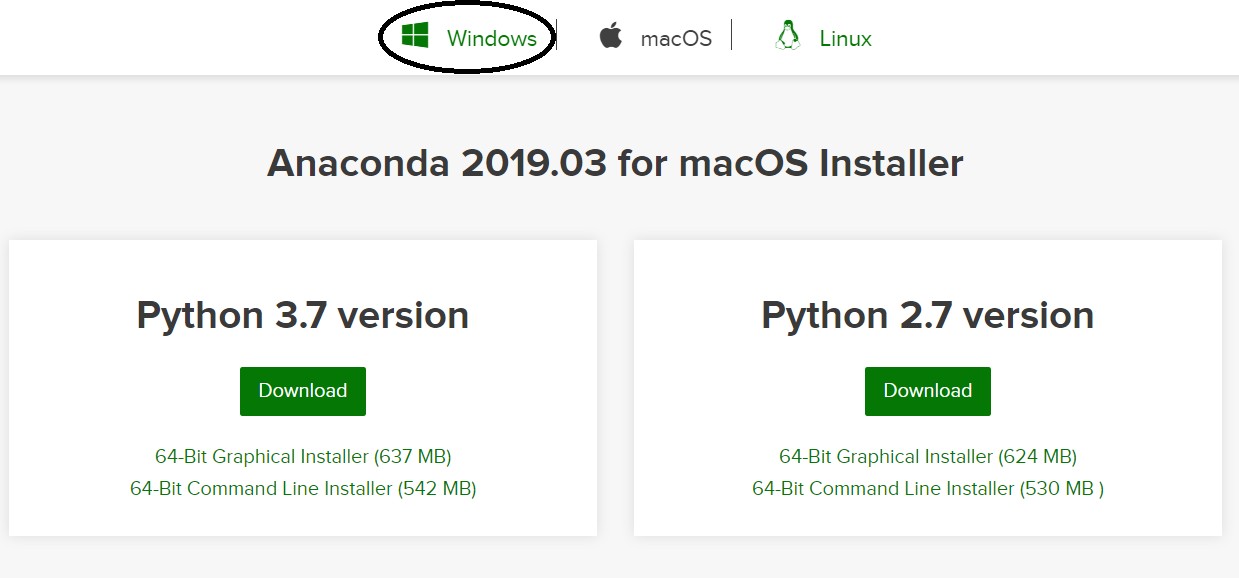
**Step 1: Install Anaconda**

If you already have Anaconda Installed, you can skip to **Step no 8** directly. Else follow below steps

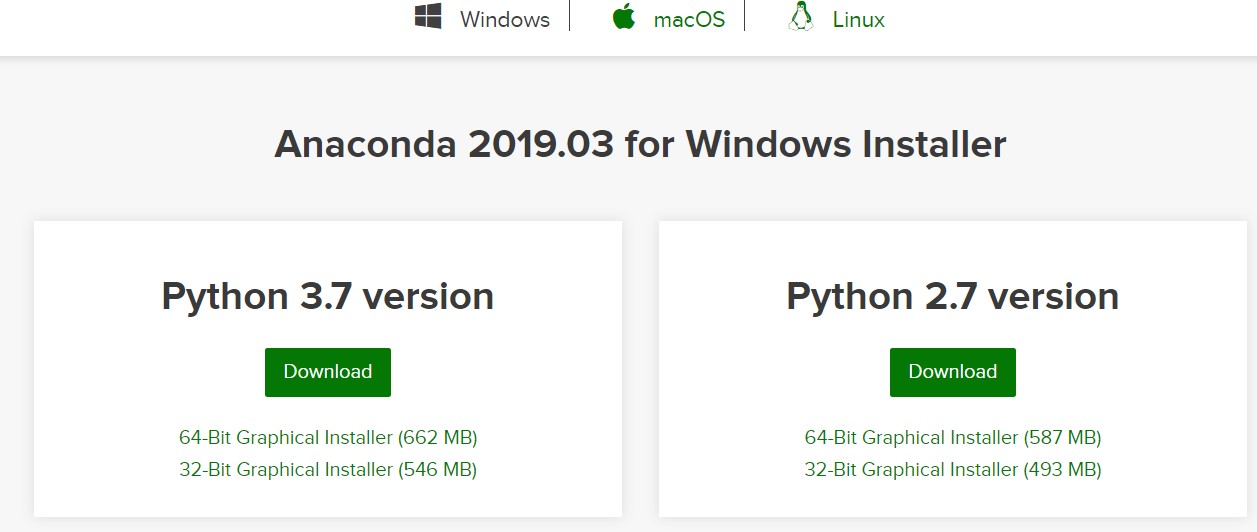
Open your browser and go to Anaconda website (<https://www.anaconda.com/distribution/>) to download and install Anaconda. You will see a page like this. Click on download.



**Step 2:** You will see that following page appears. By default, Anaconda shows you the download link for Mac operating system. If you have Mac, then you can click “64-Bit Graphical Installer” under Python 3.7 version to start downloading the file. In this computer, Windows is the operating system, so we will select Windows as shown below. If you have Linux as operating system, then you can select Linux option and download file in similar manner as Mac. Mac and Linux users can skip **Step 3 & 4**.



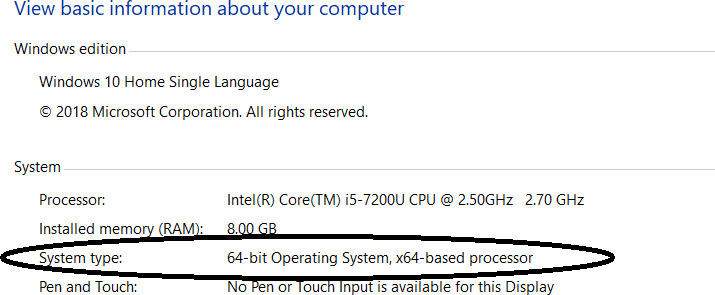
After clicking on Windows button, the following page will appear.



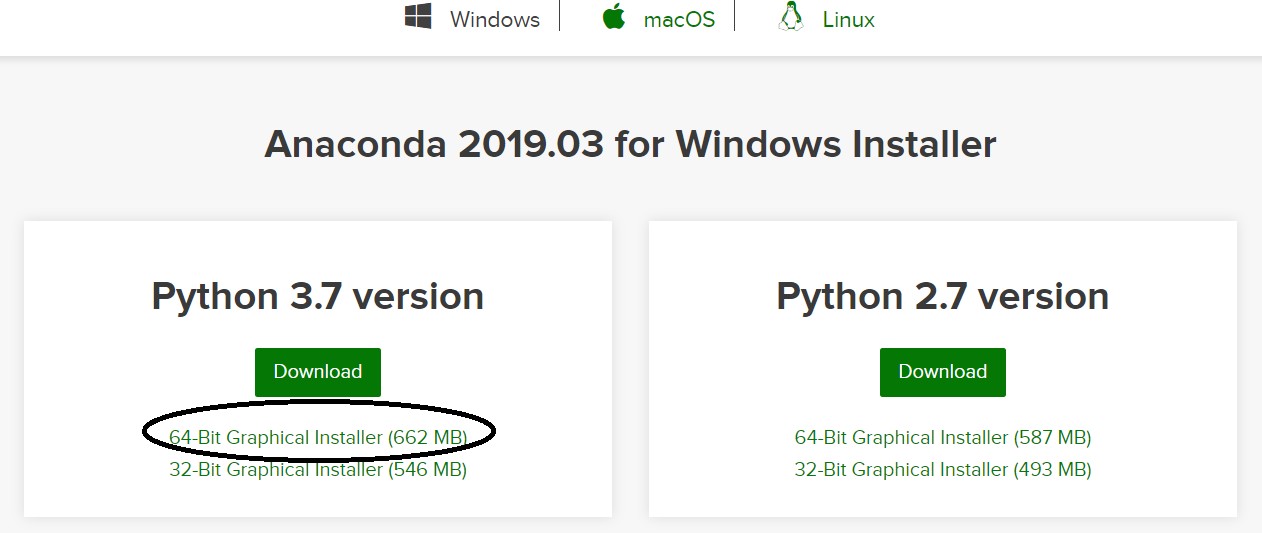
**Step 3:** You can see that there are two options for Windows: 64-Bit and 32-Bit. You need to find out whether your system is 64-Bit or 32-Bit and accordingly you need to select the file for your system. To do so, go to your desktop home screen, right click on ‘Computer’ icon, then select Properties.



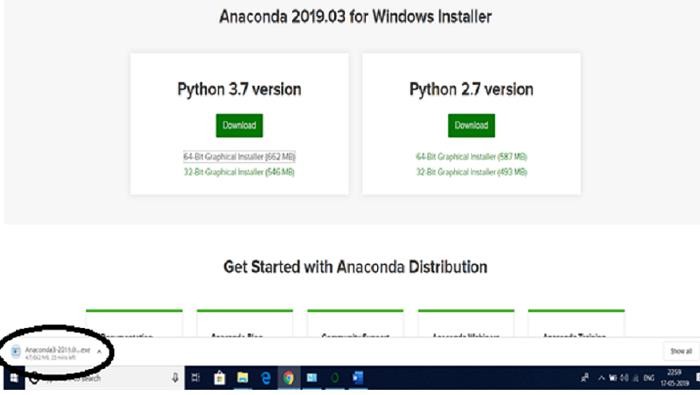
This will show you basic information about your system. Look for “System Type” as shown below and check whether it is 64-bit or 32-bit. For this computer, we see that Windows system type is 64 -bit.



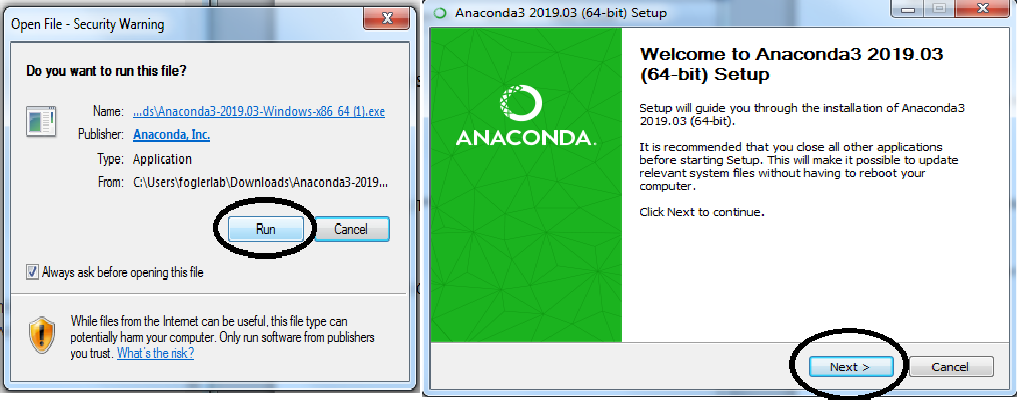
**Step 4:** Now, go back to your browser and then click “[64-Bit Graphical Installer (662 MB)](https://repo.anaconda.com/archive/Anaconda3-2019.03-Windows-x86_64.exe)” as this computer is 64 bit (as identified in Step 3)



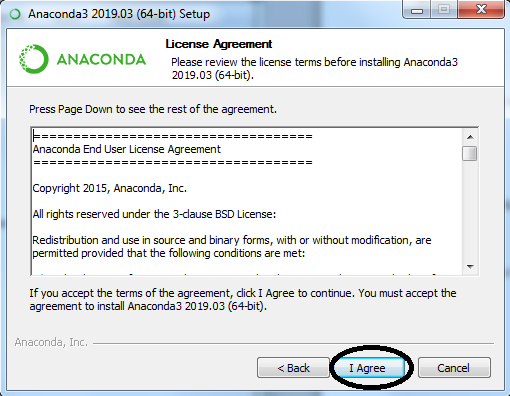
The installer will start downloading the file (this may take a while) and will appear in bottom left of your browser (if you are using google chrome) as shown below.



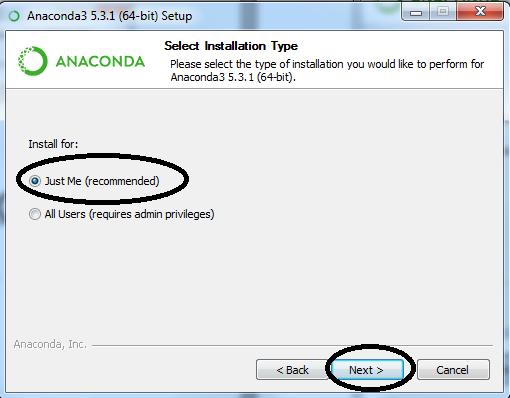
**Step 5:** When the file is completely downloaded, click on the file. You will see that following window appears. Click on ‘Run’, and then click ‘Next’ button.



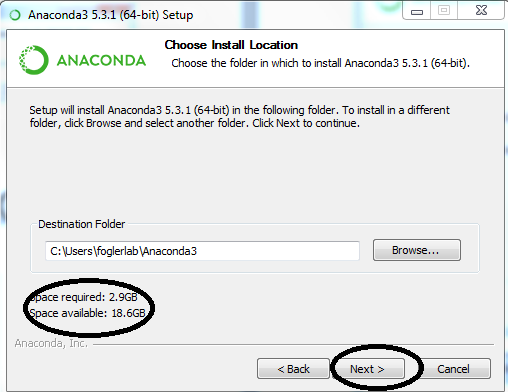
A new window will appear asking you to accept the terms of agreement, select “I Agree”.



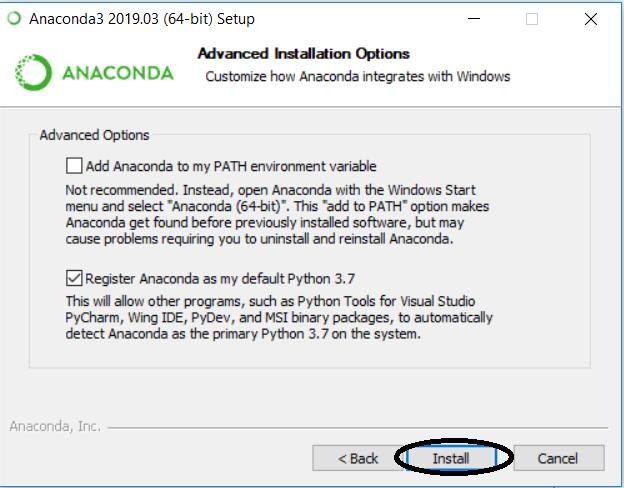
Select ‘Just Me’ which is recommended and then click Next.



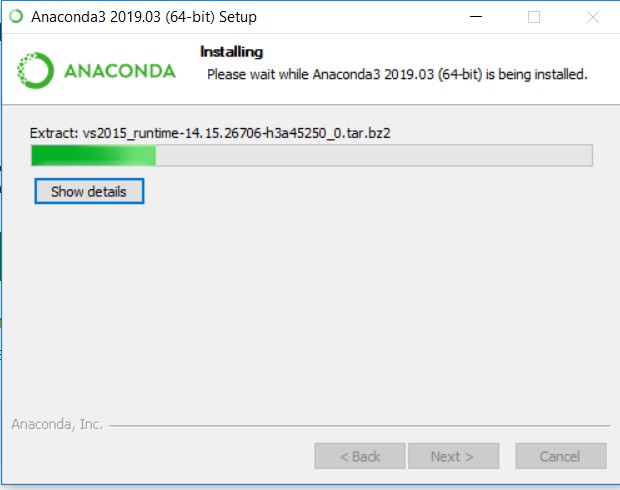
**Step 6:** Make sure you have the required free space for software installation. which you can check as shown below. Then click Next. (If you don’t have required space, then you need to delete some of your items to free the space)



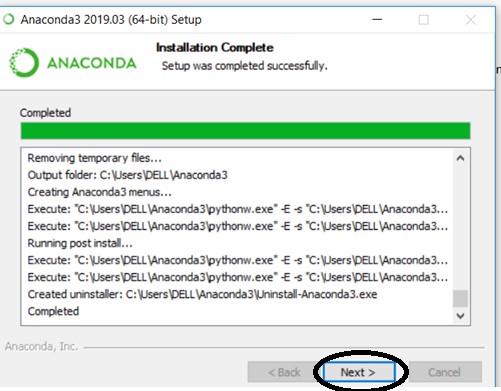
**Step 7:** You will see that following window appears. Click on Install.



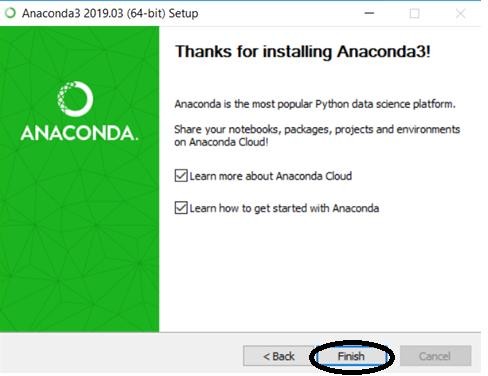
This will lead you to installation page showing the progress of installation. It will take some time for the software to get installed.



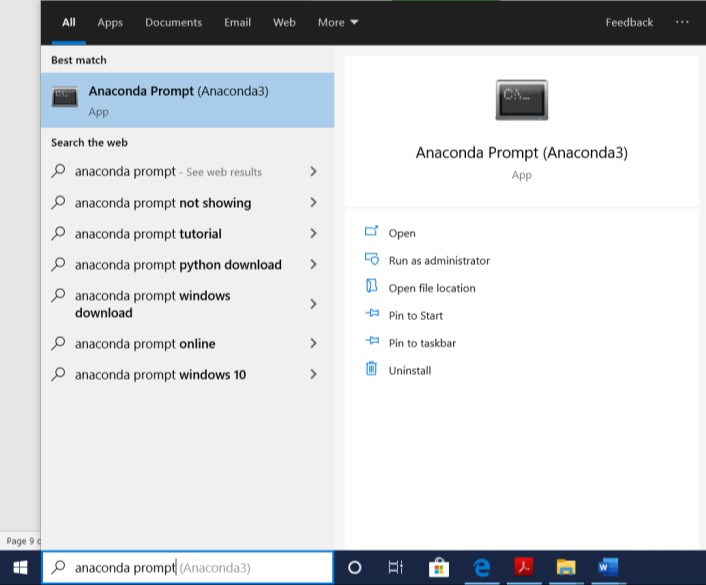
After all the files are extracted, the “Next” button will get enabled. Click on Next button



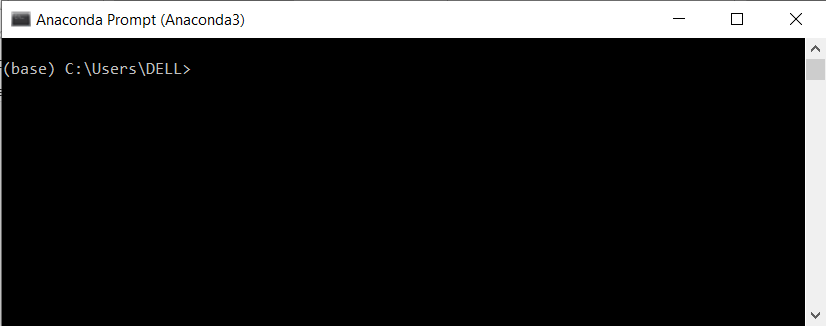
Then following window will appear. Click on Finish button to complete the installation. Now Anaconda has been installed on your computer.



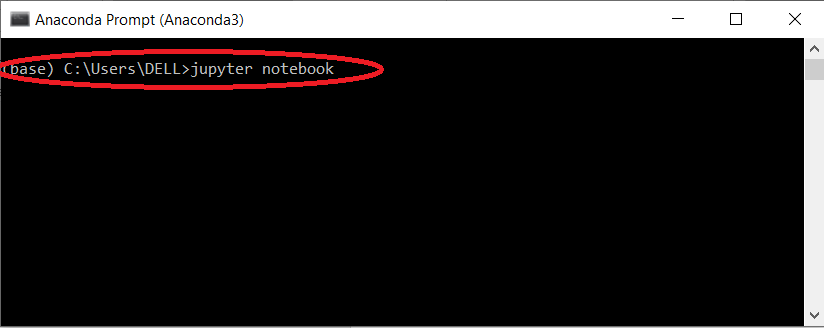
**Step 8:** Type ‘anaconda prompt’ in search box and click on the icon indicated below.



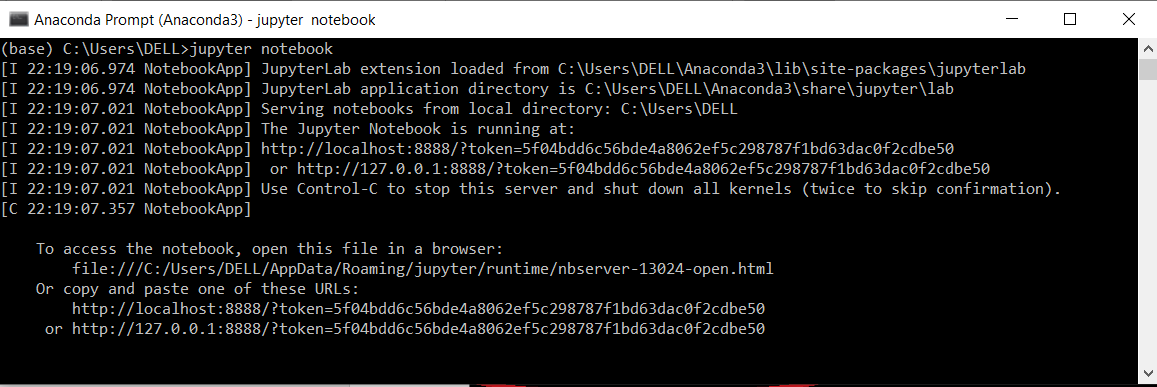
You will see that a command window opens. Just wait for few seconds until you see a file location (something like shown below)



**Step 9:** Type “jupyter notebook” in the command prompt and then Press Enter



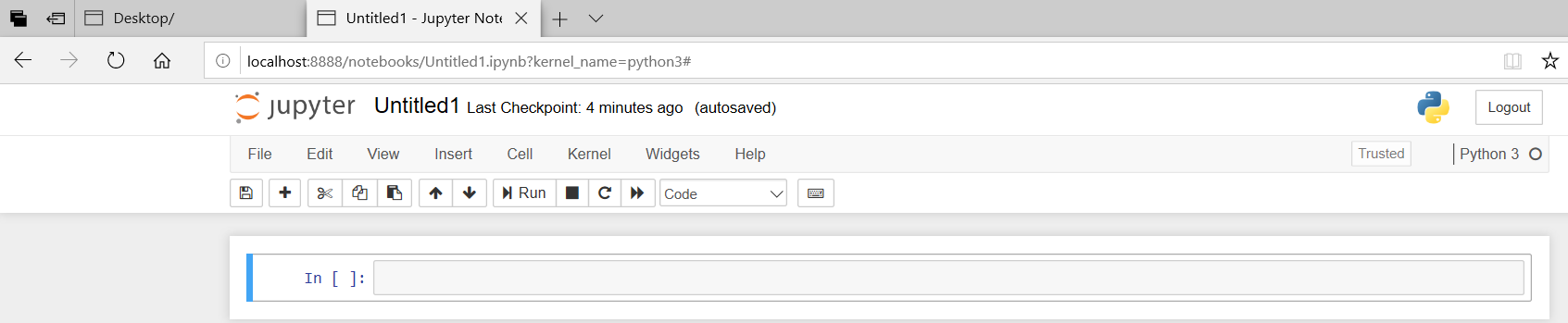
In few seconds, you will see that your command is executed as shown below.



At the same time, you will also see that your browser opens showing Jupyter Notebook Interface. Click on “New” located at upper right corner if you wish to open Jupyter Notebook.



This will open Jupyter Notebook in an another tab as shown below



**Step 10:** Now you can start creating your own code or open an LEP

## Understanding Jupyter Notebook

**What is an ipynb File?**

*The short answer*: each .ipynb file is one notebook, so each time you create a new notebook, a new .ipynb file will be created.

*The longer answer*: Each .ipynb file is a text file that describes the contents of your notebook in a format called JSON. Each cell and its contents, including image attachments that have been converted into strings of text, is listed therein along with some metadata.

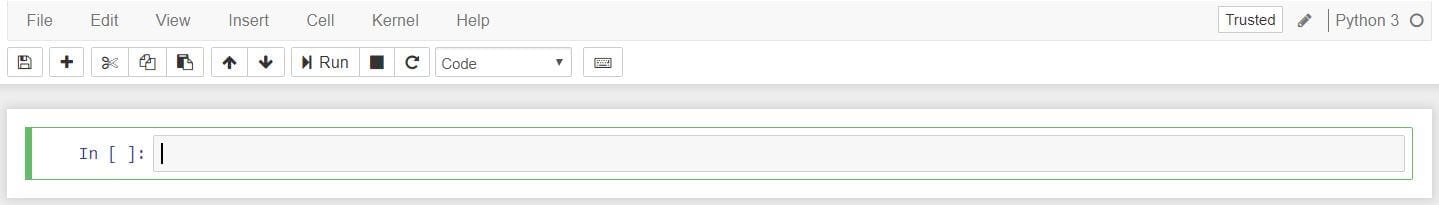
You can edit this yourself — if you know what you are doing! — by selecting “Edit > Edit Notebook Metadata” from the menu bar in the notebook. You can also view the contents of your notebook files by selecting “Edit” from the controls on the dashboard

However, the key word there is can. In most cases, there's no reason you should ever need to edit your notebook metadata manually.

**The Notebook Interface**

Now that you have an open notebook in front of you, its interface will hopefully not look entirely alien. After all, Jupyter is essentially just an advanced word processor.

Why not take a look around? Check out the menus to get a feel for it, especially take a few moments to scroll down the list of commands in the command palette, which is the small button with the keyboard icon (or Ctrl + Shift + P).



There are **two fairly prominent terms** that you should notice, which are probably new to you: **cells and kernels are key both to understanding Jupyter** and to what makes it more than just a word processor. Fortunately, these concepts are not difficult to understand.

**A kernel is a “computational engine”** that executes the code contained in a notebook document.

**A cell is a container for text** to be displayed in the notebook or code to be executed by the notebook’s kernel.

**Cells**

We’ll return to kernels a little later, but first let’s come to grips with cells. Cells form the body of a notebook. In the screenshot of a new notebook in the section above, that box with the green outline is an empty cell. There are two main cell types that we will cover:

* A code cell contains code to be executed in the kernel. When the code is run, the notebook displays the output below the code cell that generated it.
* A Markdown cell contains text formatted using Markdown and displays its output in-place when the Markdown cell is run.

The first cell in a new notebook is always a code cell.

Let’s test it out with a classic hello world example: Type print('Hello World!') into the cell and click the run button Notebook Run Button in the toolbar above or press Ctrl + Enter.

The result should look like this:

When we run the cell, its output is displayed below and the label to its left will have changed from In [ ] to In [1].

The output of a code cell also forms part of the document, which is why you can see it in this article. You can always tell the difference between code and Markdown cells because code cells have that label on the left and Markdown cells do not.

The “In” part of the label is simply short for “Input,” while the label number indicates when the cell was executed on the kernel — in this case the cell was executed first.