Python Guideline for Beginner

Trung Nguyen

Contents

1 Introduction

Currently (March 2020) Python is one of the most popular programming languages. According to the PYPL PopularitY of Programming Language Index, a search for a tutorial looks for a Python tutorial in 31.17% of the cases [PYPL] [1].

Python is a *script* language, whereby each line of the script is read and executed by the *Python interpreter*. Python scripts may *import* other Python scripts in the form of *packages*. The collection of packages used in a Python program should fit together and constitutes a *Python environment*. Buiding an environment and choosing the right versions of Python packages for a task is best left to a *package manager*.

Anaconda is a free and open-source tool for Python package management and *deployment*. Anaconda is platform-independent and will help managing your Python libraries on Windows, Linux, and Mac OS. This means that it will handle module dependencies and provides you with a Python environment. The virtual environment manager allows you to install an independent development

environment for each project. You can switch between environments like with other tools (*e.g.* virtualenv).
This document will guide the starter on how to set up the environment from scratch, starting with the installation. If you have already installed Anaconda, skip the Software Installation and start immediately by going to the Getting Started section.

2 Software Installation

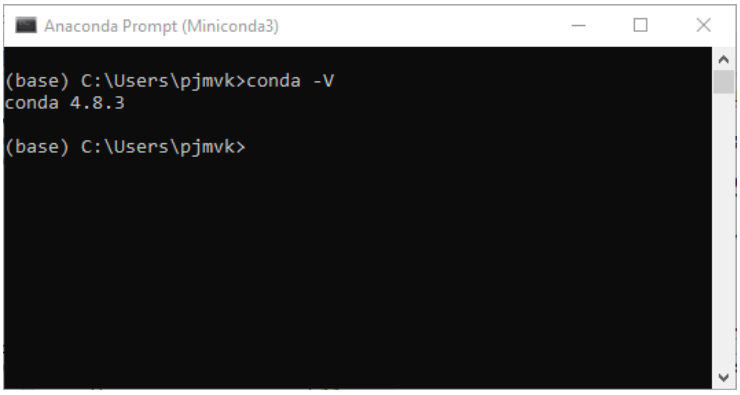
Installation of the Anaconda package manager can be done in two different ways:

- installation of Miniconda, the console version. Download from Miniconda
- installation of Anaconda Navigator, the GUI version. Download from Anaconda.

In both cases, the package manager itself is named: **conda**.
This document covers the scope of installing the software on Linux. However, installation on Windows and MacOS is very similar (consult Google on how to install Miniconda or Anaconda Navigator for these operating systems).

For Miniconda, the console version, select the Linux installer (Python 3.x, 64-bit) and download. Install “Just for me” and a ”∼/Miniconda3” folder is created in your home directory. Note: your username should NOT contain spaces. Choose default options for all possible installer choices.

Once installed, open a terminal window and type: ”conda -V” and see if the installation has been succesful. If so, you see the current version of the conda package manager and the (base) prompt, to indicate you are in the base environment, which you will, however, not use very often.



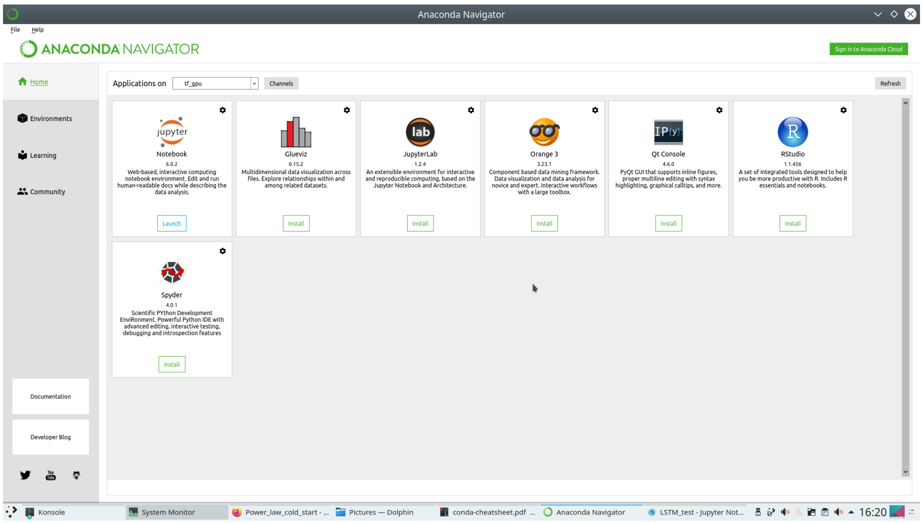
```
(base) C:\Users\pjmvk>conda -V
conda 4.8.3

(base) C:\Users\pjmvk>
```

Figure 1: Miniconda

For the installation of Anaconda Navigator (GUI), select the linux installer (Python 3.x, 64-bit) and download. The GUI is automatically installed when you install Anaconda version 4.0.0 or higher. Install “Just for me” and a ”∼/Anaconda3” folder is created in your home directory. Note: your username should NOT contain spaces. Choose default options for all possible installer choices.

Note: If you have Miniconda or an older version of Anaconda installed, you can install Navigator from an Anaconda Prompt (console) by running the command:” conda install anaconda-navigator ”.

The Anaconda Navigator user interface looks like the picture below:



Notebook
6.0.2
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Glueviz
0.15.2
Multidimensional data visualization across files. Explore relationships within and among related datasets.

JupyterLab
1.2.4
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Orange 3
3.23.1
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

Qt Console
4.6.0
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

RStudio
1.1.456
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

Spyder
4.0.1
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Figure 2: Anaconda navigator

3 Environments

The initial Miniconda installation comes with a single **base** environment, which is visible in the terminal prompt.

The command: **(base)** ⟨*pwd*⟩**conda list** shows all installed packages in the **base** environment. ⟨*pwd*⟩ stands for the present folder, included in the terminal prompt as usual. Other **conda** commands can be given in the terminal to achieve full control over the Python environments and packages. A good practice is, to download a conda *cheat-sheet* (Google with key word conda cheat sheet). In such a document, you can find how to *create* a virtual environment and *activate* it.

A new environment is created with:

(base) ⟨*pwd*⟩**conda create –n** ⟨*env_name*⟩ **python=3.x**.

This creates an environment with the basic packages.

An important step is that you have to move to your new environment with:

(base) ⟨*pwd*⟩**activate** ⟨*env_name*⟩.

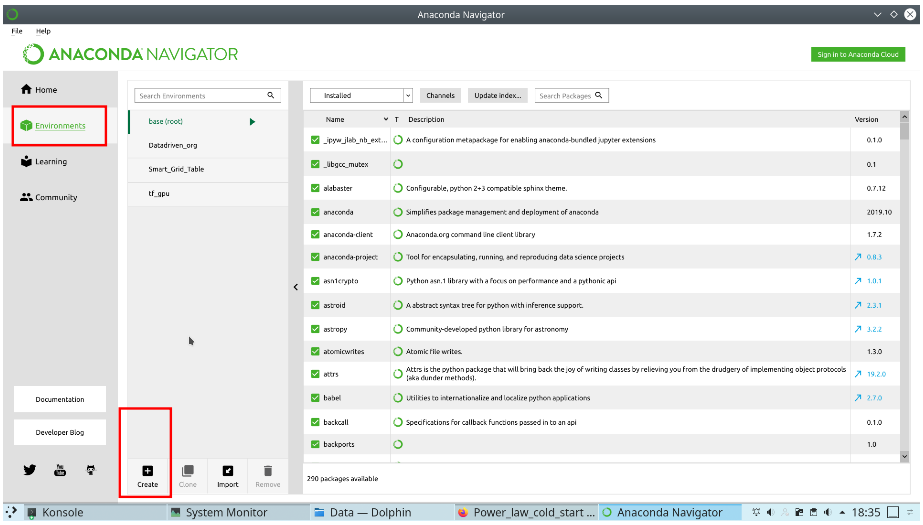
This changes the prompt into:

(new_name) ⟨*pwd*⟩**activate** ⟨*env_name*⟩.

and thus signals you the transfer to your newly created environment. Type **conda list** to see the basic packages of your new environment.

An alternative option is using the Anaconda Navigator graphical user interface. The folllowing link provides information on environments in Anaconda and the purpose of using them: Environments.

The Anaconda "Environments" window is shown in Figure 3. Have a look at the packages installed in the base environment. Create a new environment with the "Create" button in the lower left of the window. Have a look at the packages installed in your new environment. These steps are identical to the steps executed in Miniconda: under the hood, the GUI types and carries out the terminal commands for you!



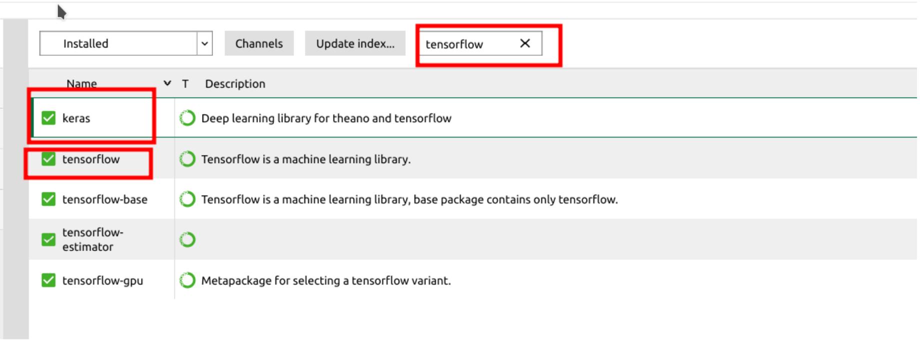
	Name	T	Description	Version
☑	_ipyw_jlab_nb_ext...	○	A configuration metapackage for enabling anaconda-bundled jupyter extensions	0.1.0
☑	_libgcc_mutex	○		0.1
☑	alabaster	○	Configurable, python 2+3 compatible sphinx theme.	0.7.12
☑	anaconda	○	Simplifies package management and deployment of anaconda	2019.10
☑	anaconda-client	○	Anaconda.org command line client library	1.7.2
☑	anaconda-project	○	Tool for encapsulating, running, and reproducing data science projects	↗ 0.8.3
☑	asn1crypto	○	Python asn.1 library with a focus on performance and a pythonic api	↗ 1.0.1
☑	astroid	○	A abstract syntax tree for python with inference support.	↗ 2.3.1
☑	astropy	○	Community-developed python library for astronomy	↗ 3.2.2
☑	atomicwrites	○	Atomic file writes.	1.3.0
☑	attrs	○	Attrs is the python package that will bring back the joy of writing classes by relieving you from the drudgery of implementing object protocols (aka dunder methods).	↗ 19.2.0
☑	babel	○	Utilities to internationalize and localize python applications	↗ 2.7.0
☑	backcall	○	Specifications for callback functions passed in to an api	0.1.0
☑	backports	○		1.0

290 packages available

Figure 3: create environment

In general, upon creating a new environment, a few (Miniconda) or quite a lot (Anaconda Navigator) standard packages will be installed in the new environment. However, in both cases most of the useful packages (libraries) need to be installed manually.

There are multiple ways to install the missing packages.Search for the missing packages and install with anaconda-navigator. Look at instruction on pictures below:



	Name	T	Description
☑	keras	○	Deep learning library for theano and tensorflow
☑	tensorflow	○	Tensorflow is a machine learning library.
☑	tensorflow-base	○	Tensorflow is a machine learning library, base package contains only tensorflow.
☑	tensorflow-estimator	○	
☑	tensorflow-gpu	○	Metapackage for selecting a tensorflow variant.

Figure 4: install packages

Installing packages with Console. Keep in mind that the virtual environment needs to be *activated* before installing the missing packages.

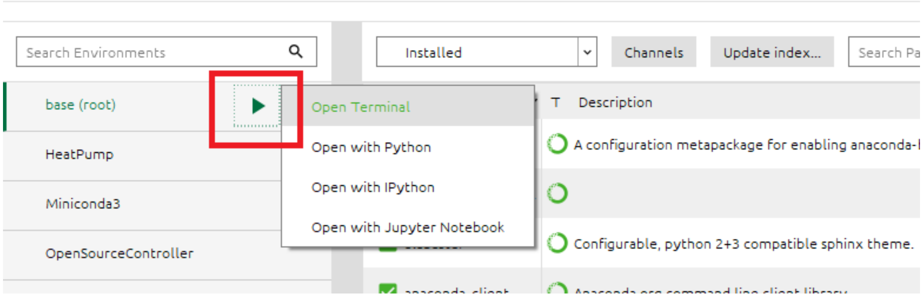
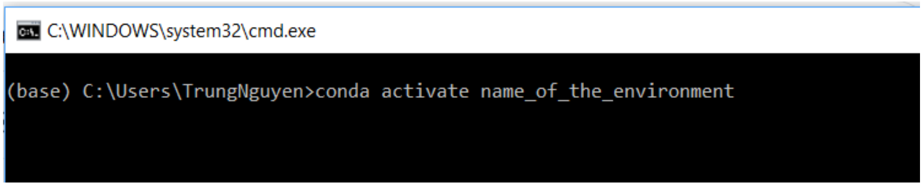


Figure 5: Open terminal

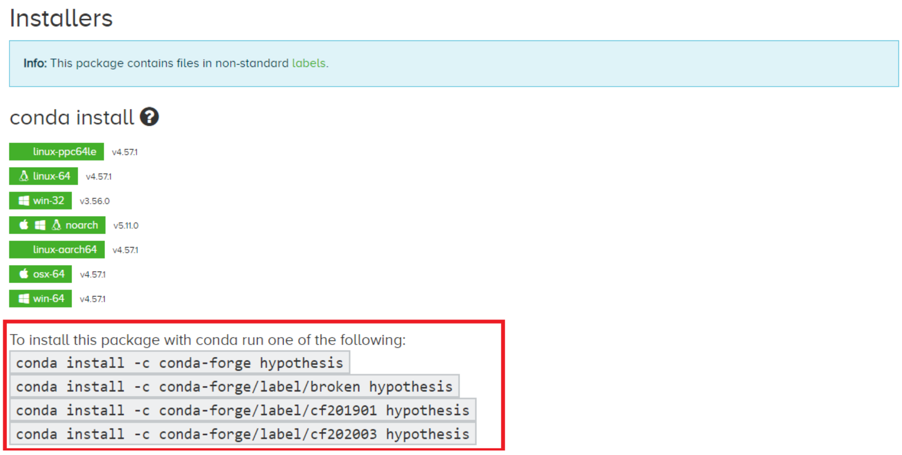


C:\WINDOWS\system32\cmd.exe

```
(base) C:\Users\TrungNguyen>conda activate name_of_the_environment
```

Figure 6: Activate Environment

Find the missing packages from :Anaconda repo.
Copy one of the command in the red box and paste to the terminal.



Installers

Info: This package contains files in non-standard labels.

conda install

linux-ppc64le v4.57.1
linux-64 v4.57.1
win-32 v3.56.0
noarch v5.11.0
linux-aarch64 v4.57.1
osx-64 v4.57.1
win-64 v4.57.1

To install this package with conda run one of the following:

```
conda install -c conda-forge hypothesis
conda install -c conda-forge/label/broken hypothesis
conda install -c conda-forge/label/cf201901 hypothesis
conda install -c conda-forge/label/cf202003 hypothesis
```

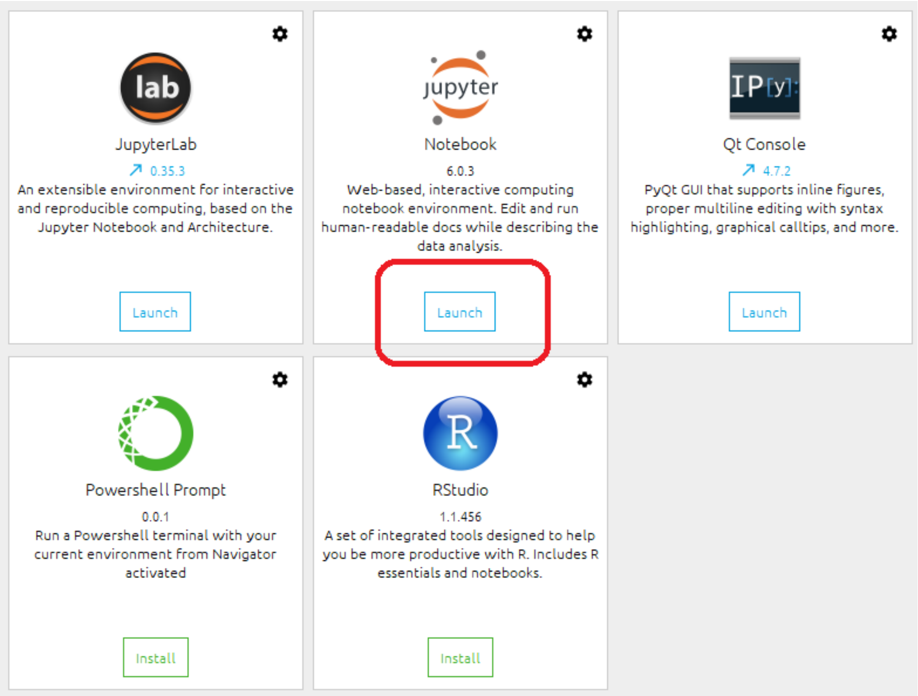
Figure 7: Install package

For some users, Miniconda will seem a more obvious option for installing new packages in an environment. The **activate** and **conda install -c** ⟨*channel*⟩ ⟨*package*⟩ commands can just be entered in the terminal window.

Note on using environments: It is customary to create a tailor-made environment for each Python project. Therefore, the information on installed packages needs to be shared among the project contributors. We'll see later in this document how that is done.

4 Getting started

Now you can install Jupyter Notebook and Sypder for working with Python.
The easiest way to get start is using a Jupyter notebook. "The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more." [2].
There are several alternative online Jupyter note book available for free such as Google Colab,Kaggle,...etc. The instruction on how to setup these online platform will be discusses in separate document.
Open Jupyter note book by click "Launch"



JupyterLab
↗ 0.35.3
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.
Launch

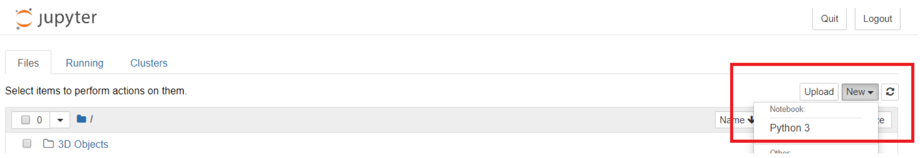
Notebook
6.0.3
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
Launch

Qt Console
↗ 4.7.2
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
Launch

Powershell Prompt
0.0.1
Run a Powershell terminal with your current environment from Navigator activated
Install

RStudio
1.1.456
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.
Install

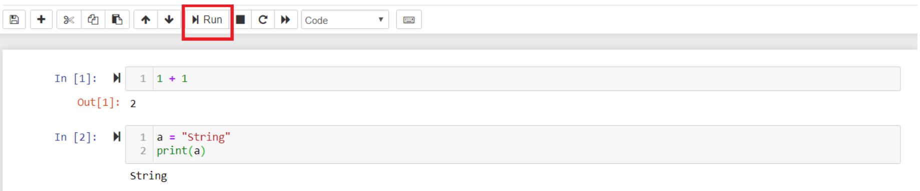
Figure 8: Open Jupyter notebook



Select items to perform actions on them.

Figure 9: Create a new python 3 project

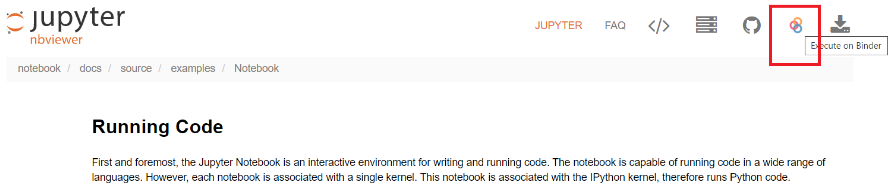
Now try to write something with Jupyter notebook and click Run button or ”Shift-Enter”



```
In [1]:  1 + 1
Out[1]:  2

In [2]:  a = "String"
         print(a)

         String
```

Figure 10: Simple example

Downloading Notebook example. Click execute on Binder to try it online on your browser



Running Code

First and foremost, the Jupyter Notebook is an interactive environment for writing and running code. The notebook is capable of running code in a wide range of languages. However, each notebook is associated with a single kernel. This notebook is associated with the IPython kernel, therefore runs Python code.

Figure 11: Online example

Github example folder.
Using google search with key world ”Jupyter notebook for beginner”

5 Link to relevant information

Anaconda user guide cheatsheet.
Navigator docs

6 List of Documentation

1. Python Guideline for Beginner.
2. Setting up Online Python Notebook
3. update...

7 Recommendation Courses

Machine learning by Andrew Ng.
Deep Learning using FastAI by Jeremy Howard.

References

[1] URL: http://pypl.github.io/PYPL.html.

[2] URL: https://jupyter.org/.