

Helper for installation and introduction for python

Sebastien Fromenteau (ICF-UNAM) and Mariana Vargas Magaña (IF-UNAM)

December 4, 2023

This note is to help you to install and start to use Python. Since the version 2.7 is not maintained anymore, we recommend to move to the python 3.10 even for the students who already use version 2.7. We also recommend to install and maintain your Python packages using Anaconda which works with Linux, Mac-OS and Windows. We will propose, at least for Linux and Mac-OS users, the possibility to easily install an environment for python 3.10 without change their version 2.7.

1 Why Anaconda?

The reason why we recommend to install Anaconda is in order to have a self consistent to maintain and upgrade the versions of Python and your packages using the "conda install" and "pip install" method. Can appear as a detail for the moment but will change your life after a while (we managed everything by ourself using port and brew before and we needed to move to anaconda at the end). You can install a graphic and a command line Anaconda and you can choose the most convenient for you.

2 First Installation of anaconda.

For all the OS, you have to go to the following link: [Anaconda](#) almost at the end of the web page on the section "Anaconda Installers".

2.1 Installer guide for Linux

A full guide for the Debian (Ubuntu is a Debian), RedHat, ArchLinux, OpenSuse and Gentoo users is provided her: [Guide for Linux](#)

2.2 Installer guide for Mac OS

A full guide for the mac OS users is provided her: [Guide for Mac OS](#).

2.3 Installer guide link for Windows

A full guide for Windows users is provided here : [Link Data Camp](#) and here: [Guide from Anaconda page](#)

For Linux and Mac OS users, you maybe need to restart your terminal after the installation in order to source your bash profile. If you know which file correspond to your bash profile, then you can just use the command:

```
myprompt$ source ~/.bash_profile
```

Before to go further we would like to precise that all the command line we will describe in the document should work for all the OS. The difference come from that Windows do not integrate a Linux/UNIX terminal but the Anaconda for Windows emulate it! So the Windows user have to find the terminal (or prompt) inside Anaconda application (it is directly on the main page normally).

3 Install Anaconda 2 with an python 3 environment, or install Python 2.7 from Anaconda 3 (Linux and Mac OS and Anaconda prompt in Windows)

This section is for the one who already installed Python 2.7 with anaconda and want to keep it while using also python 3. We do not recommend to install Anaconda 3 at the same time you have a Anaconda 2 already

installed. It is really easy to manage the two environment separately using your Anaconda 2. The same apply for the one who want use a Python 2.7 while having Anaconda 3.

If you want to create a Python 3 environment from Anaconda 2:

```
myprompt$ conda create -n py3 python=3.10
```

If you want to create a Python 2 environment from Anaconda 3:

```
myprompt$ conda create -n py2 python=2.7
```

Now you did it, you can see that your prompt is starting by a "(base)". If you are using Anaconda 2 then the base (by default) environment is using Python 2.7 while using Anaconda 3 the base is using Python 3. Just to be sure what you are using you can type the following command:

```
myprompt$ conda create -n py3 python=3.10
```

You can check the location and the version of the python code accessible when using "python" command as:

```
myprompt$ which python
```

```
myprompt$ python --version
```

Now you have installed your environment you can activate it typing:

Anaconda2 `(base) myprompt$ conda activate py3`

```
(base) myprompt$ -> (py3) myprompt$
```

Anaconda3 `(base) myprompt$ conda activate py2`

```
(base) myprompt$ -> (py2) myprompt$
```

and check the version of python now available as you did before. You will see that the location and the version are now different. In order to go back to the base version just typing:

Anaconda2 `(py3) myprompt$ conda activate py3`

```
(py3) myprompt$ -> (base) myprompt$
```

Anaconda3 `(py2) myprompt$ conda activate py2`

```
(py2) myprompt$ -> (base) myprompt$
```

4 Basic commands

The most interesting part on using Anaconda is the easy way to install new packages and maintain them up to date. In order to know the list of the packages available on the activated environment.

```
myprompt$ conda list
```

So, for the users whom defined different environment, you have to check which environment is activated at the moment you type this command (take a look to the left part of your prompt). In general, you will realize that a package is not installed when you will try to use it on your codes. When you want to install a new package (for example numpy) you will first check if it exist in the standard distribution typing:

```
myprompt$ conda search numpy
```

In the case it exists, will appear at least one element with the version of python needed. If it exist one with your version you can install it as following:

```
myprompt$ conda install numpy
```

Most of the know packages are easy to find using this method. That is the consequence that most of the developers upload their packages on the Anaconda package distribution. Sometimes it is not the case and you can first check if it exist on the "pip" packages. Indeed, pip was integrated in the Anaconda library which allows to install from pip without break the consistency of the packages we have.

```
myprompt$ pip search numpy
```

```
myprompt$ pip install numpy
```

When a package is dedicated for specific users it is usual to not encounter it on the main package distribution. In this case, the first thing to do is to Google it. You will generally get the information of the server you have to precise. For example when you want to install camb (something you will have to do), you will find that the package is on the server `conda-forge` and that you will need to type:

```
myprompt$ conda install -c conda-forge camb
```

The "-c" means that you specify the channel. Try to search camb with "conda search camb" command. The latest versions are able to find it and will also specify the channel "conda-forge".

5 Packages to install

We will need to install the following packages in the indicated order:

```
conda install -c conda-forge tensorflow
```

```
conda install -c anaconda jupyter
```

```
conda install -c anaconda pandas
```

```
conda install -c anaconda numpy
```

```
conda install -c conda-forge matplotlib
```

```
conda install -c anaconda scikit-learn
```

6 Jupyter!!

Python is mostly an interpreted language that allows us to use it interactively. In other words, you do not need to compile your program before to execute it. You can execute line by line the instructions. Thanks to that, it was possible to create a friendly version using the support of the internet browser : Jupyter (for Julia, Python and R but which is able to support other languages). It is a very friendly to develop codes with the possibility to plot intermediate results and run/rerun some blocks which allows an effective debugging.

In order to use the Jupyter Notebook, you have first to install the jupyter package with your anaconda. Some Anaconda distributions natively have the jupyter package. If you use a grphic version of Anaconda, it will probably appear with interactive button on the main page. For the others, you just need to type in your terminal (Anaconda prompt for Windows users):

```
myprompt$ jupyter-notebook
```

It should open a window in one of your browser. If you prefer to change the browser by default, you can easily find the instructions for your OS using Google.

In order to help you with the use of python and in particular inside a jupyter-notebook we provide to you a notebook which introduce most of the basic command, packages and functions you will use during this course:

[Click to download the Notebook](#)

[Click to download the file used on the notebook](#). Please download this file in the same folder than the notebook.