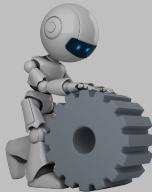


# Machine learning with Python: Create, install & use a Python Virtual Environment (PVE)

DuMAS department day – 2023/09/22

Jean-Luc.Charles@mailo.com





## Programming *Machine Learning* (ML) in Python3

- **miniconda3** allows you to install a **dedicated Python Virtual Environment** (PVE) on your laptop GNU/Linux, macOS or Windows.
- IDE<sup>a</sup> interesting for ML in Python:
  - **jupyter notebook**: for creating Python *notebooks*  $\leadsto$  files `*.ipynb` for ML, data processing, reports... Used in most tutorials on the internet.
  - **idlex**: the simplest IDE for creating/running `*.py` files (a "Python interpreter" window and a "program editor" window)
  - **VSCode**, *a.k.a Visual Studio code* from Microsoft: multi-language, very powerful, requires some work (time) to get started, especially to make it work with PVE...
  - **pycharm**, **pyzo**, **spyder** and many others [here](#) ...

---

<sup>a</sup>Integrated Development Environment

# The Python installer : Miniconda3

## Installation of Miniconda3

- Download the latest version of **Miniconda3** for your OS at [docs.conda.io/en/latest/miniconda.html](https://docs.conda.io/en/latest/miniconda.html).
- Start the installation of **Miniconda3**... **note the path for the installation folder miniconda3** ~ this will be used useful later...

[Linux, in a terminal type:

```
bash ...some_where.../miniconda3-latest-Linux-x86_64.sh]
```

**Warning** : the path of the installation folder **miniconda3** must not contain any **space** or any **accented characters**!



Windows :

C:\Miniconda3 ou C:\Users\Marie\miniconda3 ~ OK

C:\Yoann\Mes install\miniconda3 ~ not OK (space)

C:\Users\Léon\miniconda3 ~ not OK (accentuated e)

MacOSX & GNU/Linux :

/home/moi/miconda3 ou /Users/moi/opt/miniconda3 ~ OK

/home/moi/Mes install/miniconda3 ~ not OK (space)

/Users/Léon/miniconda3 ~ not OK (accentuated e)

## miniconda3 post-Installation

In the terminal, or the "Anaconda prompt" window:

- to disable the automatic activation of the **base** default PVE:

```
conda config --set auto_activate_base false
```

- to get information on the **Miniconda3** installation:

```
conda info
```

# How to create a PVE (Python Virtual Environment)

## PVE creation

In a **NEW terminal** (macOS, Linux) or an "Anaconda prompt" window (Windows), create the **dumas1** PVE:

```
conda create -n dumas1 python=3.8 -y
```

## PVE activation

Once the **dumas1** PVE is created, you must **activate** it to use it:

- in the terminal, or the "Anaconda prompt" window type:

```
conda activate dumas1
```

- the *prompt* is now prefixed with **(dumas1)**:

```
Windows:      (dumas1) C:\Users\me>
```

```
macOs:        (dumas1) /Users/me>
```

```
GNU Linux:    (dumas1) user@home $
```

# Installation of Python modules



## Different methods to load modules into the **dumas1** PVE

- With an ASCII-YAML file (\*.yml) listing the modules to install and the command `conda env update -n dumas1 --file <file.yml>`:  
~ the most efficient for a PVE created with `conda`
- With an ASCII file (\*.txt) listing the modules to install and the command `pip install -r <file.txt>`:  
~ the most used on internet tutorials...  
~ but can lead to incompatibilities between `conda` and `pip`
- By hand with `conda install ...` or `pip install ...`:  
~ the most painful!

# Installation of Python modules

Examples of files to install some Python modules in the **dumas1** PVE:



## YAML format for conda

```
name: dumas1
channels:
  - defaults
dependencies:
  - python=3.8
  - tensorflow==2.8.*
  - pandas
  - matplotlib
  - opencv
  - jupyter
  - notebook
  - scikit-learn
  - seaborn
  - pip
```



## TXT format for pip

```
tensorflow==2.8.*
pandas
matplotlib
opencv
jupyter
notebook
scikit-learn
seaborn
```

# Installation of Python modules

## Preferred method: YAML file + command **conda**

- The **--file** option of the **conda env update** command takes the name of an ASCII file in YAML format containing the list of Python modules to install.
- It is imperative to designate the PVE concerned with the option:  
**-n <PVE\_name>**



## Populate the **dumas1** PVE using the file **dumas1.yml**

In a terminal, or "Anaconda prompt" window:

- go to the folder holding the YAML file with the **cd** command:  
**cd <path\_of\_the\_folder\_containing\_the\_file\_YAML>**
- then install the Python modules in the **dumas1** PVE:  
**conda env update -n dumas1 --file dumas1.yml**



## Windows: launch jupyter notebook

- In an "Anaconda prompt" window, with the **dumas1** PVE **activated**, type:

```
jupyter notebook
```

- Access folders on a disk partition other than C:\ (e.g. D:\)

```
jupyter notebook D:\
```

```
jupyter notebook D:\folder1\folder2
```

- Access folders on a USB key mounted on example E:\

```
jupyter notebook E:\
```

## macOS & GNU/Linux: launch jupyter notebook

- In a terminal with the **dumas1** PVE **activated**, type:

```
jupyter notebook
```

- Access a folder anywhere:

```
jupyter notebook /home/users/me/folder1/folder2
```