Create a *Python Virtual Environment* (PVE)

Learning outcomes:
- ► Learn how to install Python with *miniconda*.
- ► Learn how to create and use a Python Virtual Environment (PVE) with the *conda* command.

Expected duration:
- ► 30 minutes (depending on your Internet connection).

Summary

► Interest

The state of the art in Python programming (Data processing, Machine Learning…) is to work within a **Python Virtual Environment** (PVE) to encapsulate each project in a dedicated and persistent environment containing a specific installation of Python:

- • independent of other Python installations likely to coexist on the same machine,
- • independent of computer updates.

A PVE is based on a dedicated disk tree that houses the version of the Python interpreter and speccific versions of the Python modules that you need for your project. You can create, delete and re-create a PVE very easily, without impacting other Python installations possibly present on your computer.

► Tools

The two most often used tools to create PVE are:

- • the `conda` command, available if you have installed [miniconda](#) or [Anaconda](#) on your computer
- • the `venv` Python module (see [venv](#)).

The advantage of *miniconda* for numerical computation is that it transparently installs the [MKL](#) library which provides Intel processors optimization for linear algebra libraries ([BLAS](#), [LAPACK](#) …) that determine the performance modules like *numpy* and *tensorflow*.

Another advantage of *miniconda* is that you can create a PVE of any version : for example with the last *miniconda Python 3.10* version you can create a "Python 3.8 PVE", or a "Python 3.7 PVE" or even a "Python 2.7 PVE"… and they can all coexist on the same computer!

▶ Understanding how to create a PVE with conda

Any PVE can be created thanks to the *miniconda* command : this is detailed later in the section [Work do to] for your operating system (Linux, macOS or Windows). For now the goal is just to understand the 4 steps for creating a PVE.

> ⚠️ Don't do the job **now**!
> Just understand the commands below, do the job **later** for real in the section [Work do to].

1/ How to create a PVE

```
conda create -n <pve_name> pip python=<version>
```

- <pve_name>: (free) name of your PVE, often a mnemonic like *pyml* (for *Python machine learning*) or *tf2* (for a project under tensorflow2)… or what you want.
- <version>: the version of Python you want to install in your PVE (example: *3.6* or *3.6.8* or *3.8*…).

2/ How to activate a PVE

```
conda activate <pve_name>
```

Activating a PVE results in the *prompt* being prefixed with the string (<pve_name>) :
- [Windows] if the current prompt is C:\Users\LOGNAME>, activating the PVE named pyml modifies the prompt which becomes: (pyml) C:\Users\LOGNAME>
- [GNU/Linux]: a prompt like logname@host $ becomes (pyml) logname@host $
- [macOS]: a prompt like Mac:~ logname$ becomes (pyml) Mac:~ logname$.

3/ How to install all the desired Python modules in a PVE with a YAML file

Among the many ways to install modules within a PVE, one can use a YAML file listing the modules to install (and possibly their version). For example:

```
name: <pve_name>
channels:
  - defaults
dependencies:
  - python=3.8
  - tensorflow==2.9.*
  - pandas
  - matplotlib
  - opencv
  - jupyter
  - notebook
```

To install all the Python modules listed in the file use the command:

```
conda env update -n <pve_name> --file myfile.yml
```

4/ How to add "manually" Python modules to the PVE

⚠️ The important point is that PVE concerned must already be activated.

In case you miss one particular module in your activated PVE, you can install a modules 'by hand', with the PVE activated by typing:

```
conda install <module>
conda install <module==version>              # if you want to force the module version
```

- this command downloads and installs the Python module named <module> within the activated PVE. Not all the Python modules have a conda version.

```
pip install <module>
pip install <module==version>                # if you want to force the module version
```

- this command downloads and installs the Python module named <module> within the activated PVE. All the Python modules are available with the pip installer.

⚠️ The drawback is that mixing conda install … and pip install … can lead to conflicts between the installed modules.

Now you can jump to the section corresponding to your OS, Windows, macOS or GNU/Linux.

▶ Install a PVE under Windows

How does a Virtual Environment work

The installation of the *miniconda3* package under Windows creates a specific version of the terminal (*aka* the "*command windows*") named the "Anaconda Prompt". In this terminal, the PATH environment variable is modified to first reference the directory containing the conda executable like C:\Users\LOG-NAME\Miniconda3\condabin for example.

When you activate the PVE pve_name, The PATH environment variable is modified again to first reference the PVE root directory, like C:\Users\LOGNAME\Miniconda3\envs\pve_name:

- all the Python-related commands (*python*, *pip* …) are first searched under the PVE root directory tree,
- any installation of a Python module with *conda* or *pip* installs the module under the PVE root directory.

Do I need to install miniconda ?

If you don't find the Anaconda Prompt application when searching "anaconda prompt" in the Windows search bar, then you need to install the miniconda3 package (see below) else you just have to update the conda command. To do this, open an Anaconda Prompt window and type:

```
conda update -n base -c defaults conda -y
```

Now you can jump to the section ▶2 – Create a PVE dedicated to the DRL practical work.

🔨 Work to do in 3 steps

▶ 1 – Installation of `miniconda3`

If you don't find the Anaconda Prompt application, download the last version of *miniconda3* from doc.conda.io . Pay attention to these points for installing *miniconda3*:

- the Miniconda3 directory path (installation directory) must contain no space nor accented character (the default path on Windows is often: C:\Users\LOGNAME\Miniconda3\). See this frequently asked question here :

In what folder should I install Anaconda on Windows?

We recommend installing Anaconda or Miniconda into a directory that contains only 7-bit ASCII characters and no spaces, such as `C:\anaconda` . Do not install into paths that contain spaces such as `C:\Program Files` or that include `Unicode` characters outside the 7-bit ASCII character set. This helps ensure correct operation and no errors when using any open-source tools in either Python 3 or Python 2 conda environments.

- Install *miniconda3* "*just for me*".
- Keep unchecked the option "*Add Miniconda to my PATH environment variable*".
- If you don't have any other version of Python installed on your computer you can check the option "*Register Miniconda3 as my default Python ...*" else uncheck this option.
- At the end of the installation answer *yes* to the question "*Do you wish the installer to initialize Miniconda3 by running conda init? [yes | no]*".

- Advice: after the installation, you can disable the automatic launch of the PVE (base) by typing this command in the Anaconda Prompt window:

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

Now it's done. If you want to check your installation type the command 'conda info': you should get no error in return and see information on your *miniconda3* installation displayed on the screen.

▶ 2 – Create a PVE dedicated to ML with `tensorflow2`

Create then **activate** the PVE named pyml to work with Python 3.8 :

```
(base) C:\Users\LOGNAME> conda update -n base -c defaults conda -y
        ...some stuff...

(base) C:\Users\LOGNAME> conda create -n pyml pip python=3.8 -y
        ...some stuff...

(base) C:\Users\LOGNAME> conda activate pyml
(pyml) C:\Users\LOGNAME>
```

▶ 3 – Install the Python modules in the PVE pyml

Now install the Python modules required for this course :

- Open the Git repository https://github.com/cjlux/AI-ML_at_ENSPIMA.
- Download the ZIP archive with the button `Code ▾`.
- Extract the directory AI-ML_at_ENSPIMA-master from the ZIP archive and place it somewhere in your working tree.
- Rename AI-ML_at_ENSPIMA-master as AI-ML_at_ENSPIMA.
- Install all the modules with these commands:

```
(pyml) C:\Users\LOGNAME> cd <path_of_AI-ML_at_ENSPIMA>
(pyml) <path_of_AI-ML_at_ENSPIMA> conda env update -n pyml --file pyml.yml
```

▶ Install a PVE under macOS

How does a Virtual Environment work

The installation of *miniconda3* modifies your session config file (.bashrc or .zshrc): the PATH environment variable is modified to mention first the directory /Users/<logname>/opt/miniconda3/condabin containing the conda command. When you activate the PVE pve_name, The PATH variable is also modified to reference first the PVE root directory /Users/<logname>/opt/miniconda3/envs/pve_name:
- all the Python-related commands (*python*, *pip* …) are first searched under the PVE root directory tree
- any installation of a Python module with *conda* or *pip* installs the files under the PVE root directory.

Do I need to install miniconda ?

If you cannot run the conda command in a terminal then you need to install *miniconda3* else you can skip the *miniconda3* installation.

🔨 Work to do in 3 steps

▶ 1 – Installation of `miniconda3`

If the conda command does not work in a terminal, download and install miniconda on your computer from https://docs.conda.io/en/latest/miniconda.html. Pay attention to these points:

- the *installation path* of the miniconda3 directory must contain no space nor accentuated character. (the default installation path on MacOs is: /Users/<logname>/opt/miniconda3)

- At the end of the installation answer *yes* to the question *"Do you wish the installer to initialize Miniconda3 by running conda init? [yes | no]"*

- Start a new terminal to inherit changes from your .bashrc file: the conda command now becomes available in the terminal.

- Advice: you can disable the automatic launch of the PVE (base) by typing the command:

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

Now it's done. If you want to check your installation, launch a new terminal and try the command conda info: you should get no error in return and see information on your *miniconda3* displayed on the screen.

▶ 2 – Create a PVE dedicated to ML with `tensorflow2`

Create and **activate** the PVE named pyml to work with Python 3.8 :
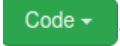
```
(base) Mac:~ logname$ conda update -n base -c defaults conda -y
                    ...some stuff...

(base) Mac:~ logname$ conda create -n pyml pip python=3.8 -y
                    ...some stuff...

(base) Mac:~ logname$ conda activate pyml
(pyml) Mac:~ logname$
```

▶ 3 – Install the Python modules in the PVE pyml

Now install the Python modules required for this course :

- Open the Git repository https://github.com/cjlux/AI-ML_at_ENSPIMA.

- Download the ZIP archive with the button `Code ▾`.

- Extract the directory AI-ML_at_ENSPIMA-master from the ZIP archive somewhere in your working tree.

- Rename AI-ML_at_ENSPIMA-master as AI-ML_at_ENSPIMA.

- Install all the modules with these commands:

```
(pyml) Mac:~ logname$ cd  <path_of_AI-ML_at_ENSPIMA>
(pyml) Mac:<path_of_AI-ML_at_ENSPIMA> logname$ conda env update -n pyml --file pyml.yml
        ...some stuff...
```

The installation of the module opencv on some MacOS laptop may require *Xcode* to compile the source code of the module. Install the *Xcode* development workbench on your laptop and retry the installation of the module, it takes a very long time to compile the module, don't worry, be patient….

▶ Install a PVE under GNU/Linux

How does Virtual Environment work

The installation of miniconda3 modifies the .bashrc file in your home directory: the PATH environment variable is modified to first reference the directory /home/<logname>/miniconda3/condabin containing the conda command on Ubuntu.
When you activate the PVE pve_name, The PATH variable is modified again to first reference the PVE root directory /home/<logname>/miniconda3/envs/pve_name:
- all the Python-related commands (*python, pip* …) are first searched under the PVE root directory tree,
- any installation of a Python module with *conda* or *pip* installs the files under the PVE root directory.

Do I need to install miniconda ?

If you cannot run the conda command in a terminal then you need to install *miniconda3* else you can skip the *miniconda3* installation.

🔨 Work to do in 3 steps

▶ 1 – Install `miniconda`

If the conda command does not work in a terminal, download and install miniconda on your computer from https://docs.conda.io/en/latest/miniconda.html. Pay attention to these points:

- The *installation path* for the miniconda3 directory must not contain spaces or accentuated characters (the default installation path on Ubuntu is: /home/<logname>/miniconda3)

- At the end of the installation answer *yes* to the question *"Do you wish the installer to initialize Miniconda3 by running conda init? [yes | no]"*

- Start a new terminal to inherit changes from your .bashrc file: the conda command now becomes available in the terminal.

- Advice: you can disable the automatic launch of the PVE (base) by typing the command:

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

Now it's done. If you want to check your installation launch, a new terminal and try the command conda info: you should get no error in return and see a information on *miniconda3* displayed on the screen.

▶ 2 – Create a PVE dedicated to ML with `tensorflow2`

Create and **activate** the PVE named pyml to work with Python 3.8 :

```
(base) logname@host $ conda update -n base -c defaults conda -y
                  ... some stuff…


(base) logname@host $ conda create -n pyml pip python=3.8 -y
                  ...some stuff ...


(base) logname@host $ conda activate pyml
(pyml) logname@host $
```

▶ 3 – Install the Python modules in the activated PVE

With the PVE pyml activated install the Python modules required for this course:

- Open the Git repository https://github.com/cjlux/AI-ML_at_ENSPIMA
- Download the ZIP archive with the button [Code ▾]
- Extract the directory AI-ML_at_ENSPIMA-master from the ZIP archive somewhere in your working tree
- Rename AI-ML_at_ENSPIMA-master as AI-ML_at_ENSPIMA
- install all the modules with these commands:

```
(pyml) logname@host $ cd <path_of_AI-ML_at_ENSPIMA>
(pyml) logname@host $ conda env update -n pyml --file pyml.yml
```

▶ Useful commands

command	description
conda info	displays informations about *conda*
conda env list	lists the PVEs known by *conda*
conda deactivate	deactivates the currenttly activated PVE
conda activate <pve_name>	activates the PVE named <pve_name>
conda list	conda view of the installed packages of **the activated PVE**
pip list	pip view of the installed packages of **the activated PVE**
conda search <name>	finds versions of the Python module named <name> compatible with the activated PVE
conda env remove -n <pve-name>	removes all the files of the PVE <pve-name>