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 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

Prior to the creation of a PVE, you will have to install the *miniconda* package on your computer. You will see this later in the present document, in the section [Work do to] for your operating system (Linux, macOS or Windows). For now the goal is just to understand the main steps for creating a PVE. With *miniconda* installed on your computer, you can create & configure as many PVE as you want following the 3 steps procedure explained bellow.

> Don't do the job **now**!
>
> Just understand the commands syntax and arguments presented in the 3 points 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> is the (free) name of your PVE, often a mnemonic like *pyml*(for *Python machine learning*) or *tf2* (for a project under tensorflow2)…
- <version> is the version of Python you want to install in your PVE (for 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

You can use a YAML file listing the modules to install with possibly their version. For exemple, you can use this myfile.yml file:

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

to install all the Python modules listed in the file with the command:

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

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

The important point is that PVE conerned 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

If you don't find the Anaconda Prompt application when search-
ing "anaconda prompt" in the Windows search bar, then you need
to install the *miniconda3* package else you can skip the *miniconda*
installation.
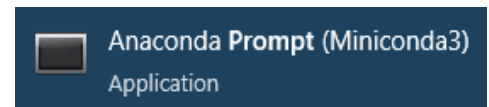
▶ How does a Virtual Environment work

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

When you activate the PVE pve_name, The PATH environment variable is modified again to first refer-
ence 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.

Work to do in 3 steps

▶ 1 – Installation of `miniconda3`

If you alredy have Anaconda or Miniconda installed on your computer, you just have to update conda. To
do this, open an Anaconda Prompt window and type:

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

That's all, you can jump to the next point.

If you don't find the Anaconda Prompt application, download the last version of *miniconda3* from
doc.conda.io (take care to choose the 64 bits version, unless your PC is a 32 bits one… which is very un-
likely). Pay attention to these points for installing *miniconda3*:

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

- 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 op-
  tion "*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, launch a new terminal and type the command 'conda info': you should get no error in return and see informations on your *miniconda3* installation displayed on the screen.
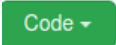
▶ 2 – Create a PVE dedicated to `tensorflow2`

With *conda* available on your computer, **create** then **activate** the PVE named pyml-pm to work with Python 3.8 :

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

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

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

Now install the Python modules required for this course :
- Open the Git repository https://github.com/cjlux/AI-Machine_Learning_at_ENSPIMA
- Download the ZIP archive with the button Code ▾
- Extract the directory AI-Machine_Learning_at_ENSPIMA-master from the ZIP archive somewhere in your working tree
- Rename AI-Machine_Learning_at_ENSPIMA-master as AI-Machine_Learning_at_ENSPIMA
- install all the modules with these commands:

```
(pyml-pm) C:\Users\LOGNAME> cd <path_of_folder AI-Machine_Learning_at_ENSPIMA>
(pyml-pm) C:\Users\LOGNAME> conda env update -n pyml-pm --file pyml-pm.yml
```

▶ 3 – Install the *idlex* minimalist IDE

In the following, C:<path_to_Miniconda3> stands for the path of the installation directory of Miniconda.

Create a shortcut that runs idlex.py within the activated pyml-pm PVE:
- Right-click on the Windows desktop.
- Move your mouse cursor over New > ShortCut in the pop-up menu and browse to select the file C:<path_to_Miniconda3>\envs\pyml-pm\Scripts\idlex.py ,
- (pyml-pm) idlex.py
- Find the icon (pyml-pm) idlex.py on your desktop and righ-click on it to edit its properties :

- in the Target field, copy all the string:

C:<path_to_Miniconda3>\condabin\conda.bat activate pyml-pm & C:<path_to_Miniconda3>\envs\pyml-pm\python.exe
C:<path_to_Miniconda3>\envs\pyml-pm\Scripts\idlex.py

replacing C:<path_to_Miniconda3> by the actual path of the Miniconda3 directory on your computer.

- In the Start In field copy the path of your directory C:\Users\LOGNAME (replacing LOGNAME by the actual value on your computer).

- Click on the Change icon button and browse to select C:<path_to_Miniconda3>\Lib\idlelib\ Icons\idle.ico

Now you can double-click on the (pyml-pm) idlex icon : you get the *Interpreter window* and with the menu File > New File, you get the *Editor windows*.

All the work made with this icon is done within the (pyml-pm) PVE.

▶ Install a PVE under MacOS

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

▶ How does a Virtual Environment work

The installation of *miniconda3* modifies the `.bashrc` file in your home directory. The `PATH` environment variable is modified to mention first the directory containing the `conda` command: `/Users/<logname>/opt/miniconda3/condabin`).

When you activate the PVE `pve_name`, The `PATH` variable is modified again 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.

Work to do in 3 steps

▶ 1 – Installation of *miniconda3*

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 `tensorflow2`

With *conda* available on your computer, create and activate the PVE named `pyml-pm` to work with Python 3.8 :

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


(base) Mac:~ logname$ conda update -n base -c defaults conda
… some stuff…


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

Now install the Python modules required for this course :

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

- Download the ZIP archive with the button   Code ▾

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

- Rename AI-Machine_Learning_at_ENSPIMA-master as   AI-Machine_Learning_at_ENSPIMA

- install all the modules with these commands:

```
(pyml-pm) Mac:~ logname$ cd  <path_of_folder AI-Machine_Learning_at_ENSPIMA>
(pyml-pm) Mac:~ logname$ conda env update -n pyml-pm --file pyml-pm.yml
```

If you get an error like "*No matching distribution found tensorflow==2.6*" try to lower the tensorflow version to the highest one mentioned in the error message (2.5.4, 2.4.3 ?) and retry the installation with this version number.

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….

▶ 3 – Install the *idlex* minimalist IDE

To run idlex, simply type:

```
(pyml-pm)  Mac:~ logname$ idlex
```

you get the *Interpreter window* and with the menu File > New File, you get the *Editor windows*.
All work done following this procedure is done in the pyml-pm PVE.

▶ Install a PVE under GNU/Linux

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

▶ 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 containing the `conda` command: `/home/<logname>/ miniconda3/condabin` 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.

Work to do in 3 steps

1 – Install `miniconda`

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 `tensorflow2`

With *conda* available on your computer, create and activate the PVE named `pyml-pm` to work with Python 3.8 :

```
(base) logname@host $ conda create -n pyml-pm pip python=3.8 -y
…some stuff …
…
(base) logname@host $ conda update -n base -c defaults conda
… some stuff…

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

Now install the Python modules required for this course :

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

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

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

- Rename AI-Machine_Learning_at_ENSPIMA-master as AI-Machine_Learning_at_ENSPIMA

- install all the modules with these commands:

```
(pyml-pm) logname@host $ cd  <path_of_folder AI-Machine_Learning_at_ENSPIMA>
(pyml-pm) logname@host $conda env update -n pyml-pm --file pyml-pm.yml
```

▶ Install the *idlex* minimalist IDE

To run idlex, simply type:

```
(pyml-pm) logname@host $ idlex
```

you get the *Interpreter window* and with the menu File > New File, you get the *Editor windows*.
All work done following this procedure is done in the pyml-pm PVE.

▶ Useful commands

command	description
conda info	Display informations about *conda*
conda env list	List the PVEs known by *conda*
conda deactivate	Deactivate the currenttly activated PVE
conda activate <pve_name>	Activate the PVE named <pve_name>
conda list	conda view of the list of installed packages **for the activated PVE**
pip list	pip view of the list of installed packages for the activated PVE
conda search <name>	Find 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>