Create a *Python Virtual Environment* (PVE)

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

Expected duration:
- 30-45 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. Each PVE provides a dedicated computing 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 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*.

Another advantage of *miniconda* is that you can create PVE of any version : for example with the last *miniconda Python 3.9* 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. For now the goal is just to understand the main steps of 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, 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 PVE results in the prompt being prefixed with the string (<pve_name>).
  For example under Windows, if the current prompt is C:\Users\\LOGNAME$, activating the PVE named pydrl modifies the prompt which becomes: (pydrl) C:\Users\LOGNAME$.
  Under GNU/Linux, a prompt like logname@host $ becomes (pydrl) logname@host $ and under MacOS a prompt like Mac:~ logname$ becomes (pydrl) Mac:~ logname$.

3/ How to add Python modules to the PVE

```
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.

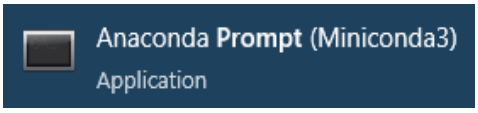
```
pip install <module>
pip install <module==version>                      if you want to force the module version
```

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

The important point is that your PVE must be activated!

▸ Install a PVE under Windows

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 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 variable is modified to first reference the directory containing the conda executable: C:\Users\LOGNAME\Miniconda3\condabin.

When you activate the PVE pve_name, The PATH environment variable is modified to first reference the PVE root directory 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 2 steps

▸ 1 – Installation of `miniconda3`

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 unlikely).

Pay attention to these points:

- Miniconda3 directory path must not contain spaces or accented characters (the default path on Windows is: 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: you can disable the automatic launch of the PVE (base) by typing the command in a terminal (an 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 try the command 'conda info': you should get no error in return and see informations on your *miniconda3* displayed on the screen.

▸ 2 – Create a PVE dedicated to the DRL practical work

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

```
(base) C:\Users\LOGNAME$ conda  create  -n  pydrl  pip  python=3.8
...some stuff ... answer 'y'  to proceed to the installation...

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

(base) C:\Users\LOGNAME$ conda activate pydrl
(pydrl) C:\Users\LOGNAME$
```

Now install the Python modules required for the pratical work:

- Open the Git repository https://github.com/cjlux/DRL_at_ENSEIRB-MATMECA
- Download the ZIP archive with the button Code
- Extract the directory DRL_at_ENSEIRB-MATMECA-master from the ZIP archive somewhere in your working tree.
- Rename DRL_at_ENSEIRB-MATMECA-master as DRL_at_ENSEIRB-MATMECA.
- Install all the modules with these commands:

```
(pydrl) C:\Users\LOGNAME$ cd  <path_of_folder DRL_at_ENSEIRB-MATMECA>
(pydrl) <path_of_DRL_at_ENSEIRB-MATMECA>$ pip install -r requirements.txt
... some stuff…
```

Now you can run jupyter notebook or jupyter lab to work with the notebooks...

▸ Install a PVE under MacOS

If you cannot run the miniconda 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 not contain any spaces or 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 or type the command source ~/.bashrc 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 the DRL practical work

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

```
(base) Mac:~ logname$ conda create -n pydrl pip python=3.8
...some stuff ... answer 'y' to proceed to the installation…

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

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

Now install the Python modules required for the pratical work:

- Open the Git repository https://github.com/cjlux/DRL_at_ENSEIRB-MATMECA
- Download the ZIP archive with the button Code
- Extract the directory DRL_at_ENSEIRB-MATMECA-master from the ZIP archive somewhere in your working tree.
- Rename DRL_at_ENSEIRB-MATMECA-master as DRL_at_ENSEIRB-MATMECA.
- Install all the modules with these commands:

```
(pydrl) Mac:~ logname$ cd  <path_of_folder DRL_at_ENSEIRB-MATMECA>
((pydrl) Mac:<path_of_folder DRL_at_ENSEIRB-MATMECA> logname$$ pip install -r requirements.txt
... some stuff…
```

Now you can run jupyter notebook or jupyter lab to work with the notebooks…

▸ Install a PVE under GNU/Linux

If you cannot run the miniconda 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 or type the command source ~/.bashrc 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 the DRL practical work

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

```
(base) logname@host $ conda create -n pydrl pip python=3.8
...some stuff ... answer 'y' to proceed to the installation...
...
(base) logname@host $ conda update -n base -c defaults conda
... some stuff...

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

Now install the Python modules required for the pratical work:

- Open the Git repository https://github.com/cjlux/DRL_at_ENSEIRB-MATMECA
- Download the ZIP archive with the button Code
- Extract the directory DRL_at_ENSEIRB-MATMECA-master from the ZIP archive somewhere in your working tree.
- Rename DRL_at_ENSEIRB-MATMECA-master as DRL_at_ENSEIRB-MATMECA.
- Install all the modules with these commands:

```
(pydrl) logname@host $ cd  <path_of_folder DRL_at_ENSEIRB-MATMECA>
(pydrl) logname@host $ pip install -r requirements.txt
... some stuff…
```

Now you can run jupyter notebook or jupyter lab to work with the notebooks…

▸ 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