

Formation Python : Environnement virtuel

par Bernard CHAMBON

CC-IN2P3, Lyon - France

12, 13, 14 novembre 2018

- Outil d'installation (`pip`) et format de packaging (`wheel`)
- Environnement virtuel : `virtualenv`
- Mise en pratique

■ Outil d'installation : `easy_install` et `pip`

- `easy_install`, de 2004
Outil qui permet d'installer des packages
Peut fonctionner avec `PyPI` (Python Package Index), repository de package python
Outil ancien (2004), qui ne reconnaît que le format `eggs`
- `pip`, de 2008 (`pip3` avec Python3)
C'est l'outil d'installation de package
`pip` permet d'installer, de désinstaller, d'upgrader, de lister
Ne supporte que le format `wheels`, ne supporte pas le format `egg` (voir convertisseur)

Voir comparaison [easy_install VS pip](#)

- `PyPI` : Python Package Index : <https://pypi.python.org/pypi>

■ Formats de packaging : `eggs` | `wheels`

- `eggs` (`.egg`)
"A 'Python egg' is a logical structure embodying the release of a specific version of a Python project, comprising its code, resources, and metadata.
Ancien, date de 2004
- `wheels` (`.whl`)
*"wheel is currently considered the standard for built and binary packaging for Python."
Wheels are the new standard of python distribution and are intended to replace eggs.*
Date de 2012

- Mise en place d'env. isolé : `virtualenv`
Pour disposer d'un env. de travail sous python et dans lequel on pourra installer des modules
Par exemple dans votre `$HOME`, sans droit d'admin, sans perturber votre système
- Qu'est ce qui vient avec `virtualenv`
 - L'interpréteur Python
La version sera celle avec lequel (`virtualenv`) il a été préparé
Il est possible de définir une autre version (`virtualenv -python ...`), pour peu que cette version soit disponible sur votre machine
⇔ pour disposer d'une nouvelle version de python en utilisant `virtualenv`, il faut d'abord disposer localement (télécharger [et builder]) cette nouvelle version de Python
 - Des outils d'installation (ex `pip`)

■ Création d'un environnement virtuel dans /tmp/DEMO_VE

- Disponibilité de virtualenv, installation si non trouvé

```
> which virtualenv
```

```
si virtualenv n'existe pas, on peut l'installer via pip
```

```
> pip install virtualenv
```

```
> which virtualenv
```

```
/usr/local/bin/virtualenv
```

- Création et activation de l'environnement virtuel

```
Création d'un env. virtuel
```

```
> virtualenv /tmp/DEMO_VE
```

```
New python executable in /tmp/DEMO_VE/bin/python
```

```
Installing setuptools, pip, wheel...done.
```

```
Activation de l'env. virtuel
```

```
> source /tmp/DEMO_VE/bin/activate.csh
```

```
[DEMO_VE] [ccbc:/tmp] bchambon
```

```
Vérification
```

```
[DEMO_VE] [ccbc:/tmp] bchambon > which python
```

```
/tmp/DEMO_VE/bin/python
```

```
Désactivation de l'env. virtuel et verification
```

```
[DEMO_VE] [ccbc:/tmp] bchambon > deactivate
```

```
> which python
```

```
/Users/bchambon/anaconda3/bin/python
```

■ Installation d'un module dans l'environnement virtuel créée

- Activation de l'env. virtuel et installation d'un module

```
> source /tmp/DEMO_VE/bin/activate.csh
```

```
[DEMO_VE] [ccbc:/tmp] bchambon> pip install /Users/bchambon/.../dist/TrainingSessionRegistration-0.1-py2.py3-none-any.whl
Processing /Users/bchambon/.../dist/TrainingSessionRegistration-0.1-py2.py3-none-any.whl
Installing collected packages: TrainingSessionRegistration
Successfully installed TrainingSessionRegistration-0.1
```

- Verification de la disponibilité du module

Verification par un code (voir main.py) utilisant le module installé

```
> cd /tmp
```

```
> [DEMO_VE] [ccbc:/tmp] bchambon% python /tmp/main.py
```

Programming with Python for 3 days at CC-IN2P3

Name	Lab	Registration date
Me	CC-IN2P3	2018/09/19/-16:05:04
You	Lab++	2018/09/19/-16:05:04
Him	SuperLab	2018/09/19/-16:05:04

- Désactivation de l'env. virtuel et verification de la NON disponibilité du module

Désactivation de l'env. virtuel

```
[DEMO_VE] [ccbc:/tmp] bchambon deactivate
```

Verification (que le module n'est plus trouvé)

```
> python /tmp/main.py
```

```
Traceback (most recent call last):
```

```
  File "/tmp/main.py", line 3, in <module>
```

```
    from pck1.tsr import TrainingSessionRegistration
```

```
ModuleNotFoundError: No module named 'pck1'
```

- Pour information : le code précédemment utilisé (main.py)

```
from pck1.tsr import TrainingSessionRegistration

def main():
    training_session = TrainingSessionRegistration()
    training_session.register_attendee("You", "Lab++")
    training_session.register_attendee("Him", "SuperLab")
    training_session.register_attendee("Me", "CC-IN2P3")

    training_session.display_info()
    training_session.display_attendees()

if __name__ == '__main__':
    main()
```

- Démo : En utilisant PyCharm (projet EX5_DemoVirtualEnv)
(voir le README pour la liste des actions)