

Installer facilement des modules python

L'installation préalable de distributions complètes du type [Anacaonda](#), [Pythonxy](#), [Enthought](#) ou [ActivePython](#) est préférable !

Ces installations peuvent d'ailleurs l'installation des outils comme [setuptools](#), [distutils](#), [pip](#),...

Introduction

Des modules additionnels de Python peuvent être installés via des sites qui les proposent. Il s'agit de :

- créateurs de programmes, librairies
- firmes ou associations qui proposent des ensembles cohérents (comme [Sage](#), [Enthought](#),...)
- logithèques de systèmes l'exploitation (de distributions Linux comme Ubuntu par exemple)

Ces modes d'installation de modules ne sont pas toujours satisfaisants : difficultés de trouver certains modules, ensembles reprenant des modules non souhaités, versions de modules générant des conflits,... aussi le site officiel de Python a créé un catalogue et des archives de librairies Python, imitant en cela les grands sites d'archives comme :

- [LaTeX](#),
- [Perl](#),
- [R](#),
- [JavaScript](#),
- [Pythonxy](#), [Enthought](#) ou [ActivePython](#) est préférable !
- [php](#),
- ...

Ce grand répertoire se nomme [PyPI](#) (Python Package Index). Il reprend 112 860 paquets python (juillet 2017). Un programme nommé [Pip](#) permet de télécharger et ensuite installer des programmes ou librairies référencées dans le répertoire **PyPi**.

Pour vérifier une installation existante, n'hésitez pas à exécuter les commandes suivantes :

```
python --version
python2 --version
python3 --version
pip --version
pip2 --version
pip3 --version
virtualenv --version
```

Procédure sous Windows

Procédure sous Linux

Sous Ubuntu, Debian et distributions dérivées, pour python (v2) et python3, au cas où pip ne serait pas installé :

```
sudo apt-get install python-pip python3-pip
```

Pour mettre à jour :

```
sudo -H pip3 install --upgrade pip
sudo -H pip2 install --upgrade pip
```

Procédure sous Mac

À écrire, cf. [ici](#)

Utiliser Pip

C'est donc Pip qui va être utilisé pour installer des programmes et librairies. Par exemple, pour installer matplotlib, taper dans une fenêtre de commande :

```
pip install matplotlib
```

Attention, il est préférable d'utiliser les exécutables pour Windows proposés directement sur le site [Pypi](#), s'ils existent.

La plupart des librairies devraient être installées de manière locale, à l'exception de certaines, comme virtualenv. Donc sous GNU/Linux, il ne faut pas utiliser le "sudo".

Quelques commandes, remarques,... avec "nomlibrairie" comme nom de librairie, à remplacer :

- mise à jour à la dernière version :

```
pip install --upgrade nomlibrairie
```

- désinstaller :

```
pip uninstall nomlibrairie
```

- version de pip :

```
pip --version
```

- information à propos de la librairie :

```
pip show nomlibrairie
```

- génère une liste de librairies installées en "requirements" :

```
pip freeze > requirements.txt
```

- installe à partir d'une liste de "requirements" :

```
pip install -r requirements.txt
```

- lister les librairies :

```
pip list
```

- lister les librairies dépassées et les nouvelles versions :

```
pip list --outdated
```

Pour plus d'information, consulter le [Reference guide](#).

Virtualenv

venv, équivalent, est installé par défaut à partir de Python 3.3. Après activation d'un environnement virtuel, son nom est indiqué entre parenthèses devant le prompt.

Installation :

```
sudo apt-get install virtualenv
sudo apt install virtualenvwrapper
echo "source /usr/share/virtualenvwrapper/virtualenvwrapper.sh" >> ~/.bashrc
```

Configuration :

```
export WORKON_HOME=~/.virtualenvs
mkdir $WORKON_HOME
echo "export WORKON_HOME=$WORKON_HOME" >> ~/.bashrc
echo "export PIP_VIRTUALENV_BASE=$WORKON_HOME" >> ~/.bashrc
source ~/.bashrc
```

Utilisation (exemples) :

```
mkvirtualenv --python=/usr/bin/python3 test
deactivate
workon test
which python
deactivate
rmvirtualenv test
```

refs :

- <https://askubuntu.com/questions/244641/how-to-set-up-and-use-a-virtual-python-environment-in-ubuntu>
- <https://stackoverflow.com/questions/16123459/virtualenvwrapper-and-python-3>
- <http://sametmax.com/les-environnement-virtuels-python-virtualenv-et-virtualenvwrapper/>
- [<https://docs.python.org/3/tutorial/venv.html>]
- <http://python-guide-pt-br.readthedocs.io/en/latest/dev/virtualenvs/>
- <https://virtualenvwrapper.readthedocs.io/en/latest/>
- <http://apprendre-python.com/page-virtualenv-python-environnement-virtuel>
- À tester : <https://www.opendatascience.com/blog/setup-a-python-virtual-machine-with-ubuntu/>

Problèmes particuliers

Upgrade de Matplotlib par pip sous Ubuntu 14.04 :

Passer via sudo (sinon erreurs)

- <https://mlbernauer.wordpress.com/2014/12/06/upgrading-from-matplotlib-1-3-1-to-1-4-1-in-ubuntu-14-04/>
- solution :
 - `sudo apt-get install libfreetype6-dev`
 - `sudo pip install --upgrade matplotlib`
- Résultats :
 - Successfully installed cycler-0.9.0 matplotlib-1.5.1 numpy-1.10.4 pyparsing-2.0.7 python-dateutil-2.4.2 pytz-2015.7 six-1.10.0

références

- <http://coagul.org/drupal/publication/installer-facilement-modules-python>
- <http://stackoverflow.com/questions/4750806/how-to-install-pip-on-windows>
- <http://docs.python-guide.org/en/latest/starting/install/win/>
- <https://zignar.net/2012/06/17/install-python-on-windows/>
- <http://peak.telecommunity.com/DevCenter/EasyInstall>
- <http://www.pip-installer.org/en/1.0.1/index.html>
- <http://www.blog.pythonlibrary.org/2011/11/24/python-101-setting-up-python-on-windows/>
- <http://python-packaging-user-guide.readthedocs.org/en/latest/installing/>
- pip documentation
- <http://askubuntu.com/questions/431780/apt-get-install-vs-pip-install>
- <http://kazhack.org/?post/2014/12/12/pip-gem-install-without-sudo> (sudo or not sudo...)
- <http://www.dabapps.com/blog/introduction-to-pip-and-virtualenv-python/> (intro pip & virtualenv)
- virtualenv :
 - <http://docs.python-guide.org/en/latest/dev/virtualenvs/>
 - http://christrel.io/sandbox/2014/09/04/virtualenv_and_virtualenvwrapper_on_ubuntu_14_04.html
 - <http://www.sitepoint.com/virtual-environments-python-made-easy/>

From:

<https://dvillers.umons.ac.be/wiki/> - **Didier Villers, UMONS - wiki**

Permanent link:

<https://dvillers.umons.ac.be/wiki/floss:python:pip-pypi>

Last update: **2017/07/22 16:28**

