

Composants réutilisables dans la pratique : PyPI

S. Aicardi

ANF Mathrice, Angers, 21-25 mai 2012

Paquets Python

Un paquet Python est généralement un fichier `.tar.gz` qui contient le source et un fichier `setup.py`.

L'installation se fait alors par la commande suivante :

```
$ python setup.py install
```

Pour fonctionner, il faut au préalable installer la librairie `setuptools` ou `distribute`.

Python Package Index

PyPI, c'est avant tout un site web :

<http://pypi.python.org/pypi>

On peut y naviguer par catégorie (version de python, langue, OS, type d'utilisateurs, sujet).

easy_install

`easy_install` est une commande qui fait partie de la librairie `setuptools`.

Elle manipule les paquets sous la forme de *eggs*.

Elle permet d'installer un paquet et ses dépendances et de le mettre à jour :

```
$ easy_install Package  
$ easy_install --upgrade Package
```

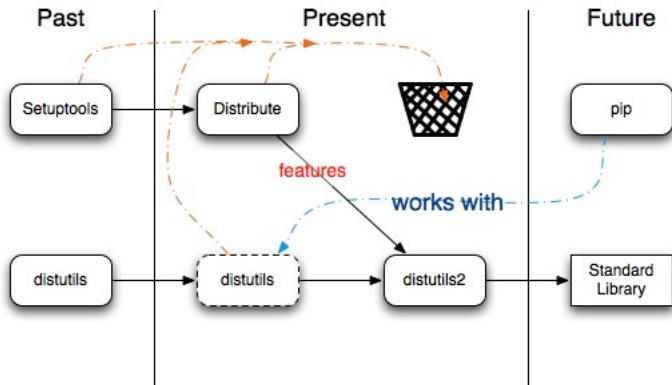
Pour supprimer un paquet, il suffit d'effacer le `.egg` à la main.

pip

pip (Pip Installs Python) est un nouveau gestionnaire de paquets qu'il faut privilégier :

- ▶ support de python3,
- ▶ support des systèmes de gestion de version,
- ▶ désinstallation d'un paquet,
- ▶ plus grande stabilité,
- ▶ suivi des paquets dans le temps,
- ▶ etc.

pip



pip : installation

Soit avec le système de paquets de votre distribution favorite.

Soit à la main : installez d'abord setuptools ou distribute, puis

```
$ curl https://raw.githubusercontent.com/pypa/pip/master  
/contrib/get-pip.py | python
```

pip : Usage

Installation

```
$ pip install Package[==Version]
```

Mise à jour

```
$ pip install --upgrade Package[==Version]
```

Désinstallation

```
$ pip uninstall Package[==Version]
```

Recherche

```
$ pip search Package
```

Liste des paquets installés

```
$ pip freeze
```


Fabrication de modules

Un module est soit un fichier python, soit un répertoire contenant un fichier `__init__.py`.

Pour faciliter le test des modules, il est bon de permettre une exécution de test du module.

```
#!/usr/bin/python

Class MaClasse:
    ...

if __name__=="__main__":
    toto = MaClasse()
    print toto
```

Fabrication de paquets

Il faut ajouter à côté du module, un fichier `setup.py` décrivant les métadonnées. On pourra également ajouter un fichier `README.txt` décrivant le paquet et un fichier `LICENSE.txt`.

Exemple de répertoire :

```
MonPaquet/  
  LICENSE.txt  
  MonModule/  
    __init__.py  
    test/  
      __init__.py  
  README.txt  
  setup.py
```

Fabrication de paquets

Le fichier `setup.py` aura la forme suivante :

```
from distutils.core import setup

setup(
    name='MonPaquet',
    version='1.0',
    author='Super Coder',
    author_email='Super.Coder@example.com',
    packages=['monmodule', 'monmodule.test'],
    url='http://pypi.python.org/pypi/MonPaquet/',
    license='LICENSE.txt',
    description='Un super Paquet.',
    long_description=open('README.txt').read(),
    install_requires=[
        "Django >= 1.1.1",
    ],
)
```

Fabrication de paquets

Pour produire une version du paquet :

```
$ python setup.py sdist
```

Pour s'enregistrer auprès de PyPI :

```
$ python setup.py register
```

Pour publier le paquet sur PyPI :

```
$ python setup.py upload
```

Fabrication de paquets

Pour plus de détails, voir

<http://guide.python-distribute.org/index.html>