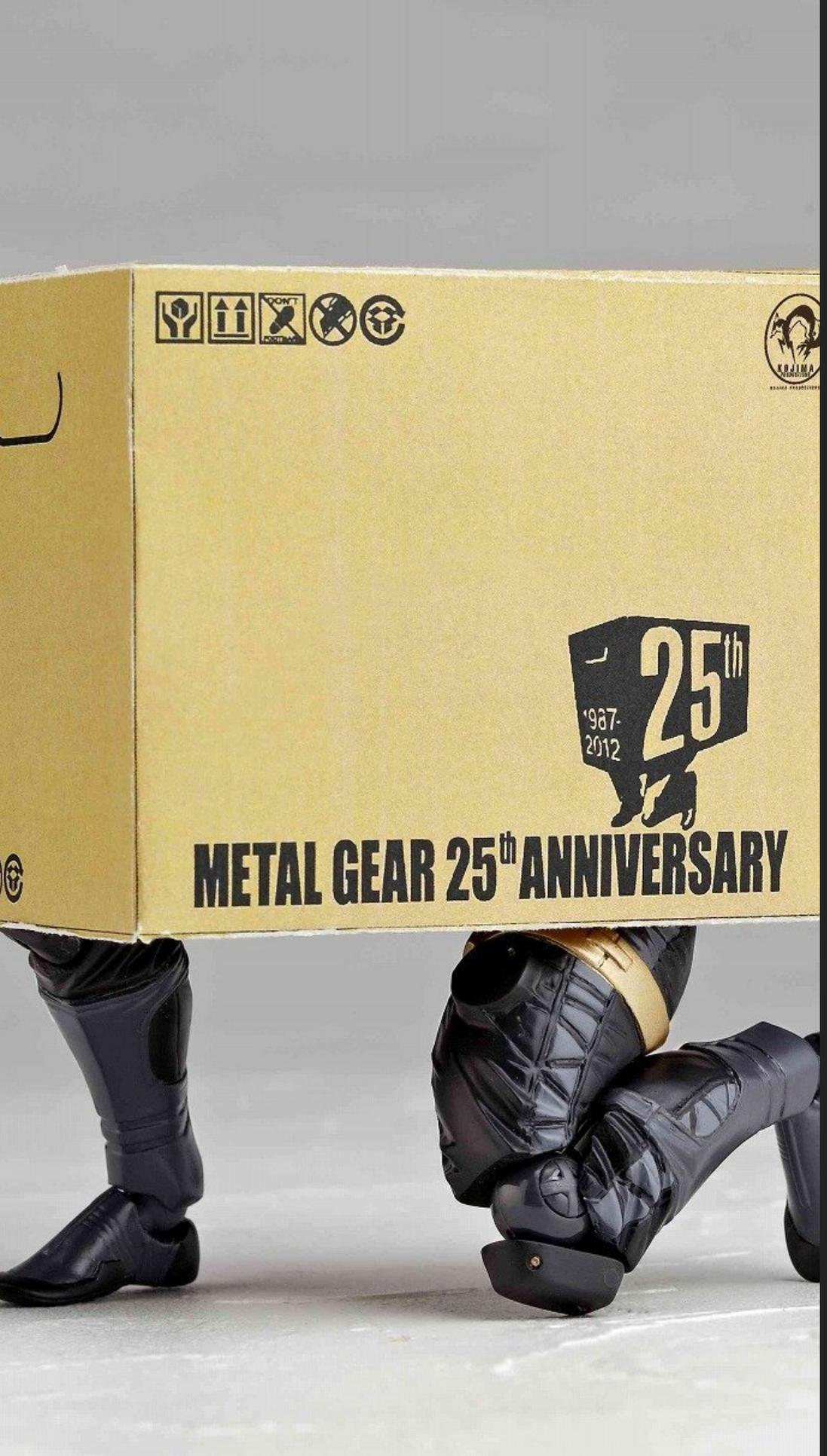

⚠️ Slides en franglais

STÉPHANE 'TWIDI' ANGEL,
JOACHIM JABLON

PACKAGER SON PROJET PYTHON

TEXT

POURQUOI PACKAGER UN PROJET ?

- Version
- Partage
- Déploiement

COMMENT VERSIONNER UN PROJET ?

- `cp -a projet "projet-`date +%F`"`

- `git tag `date +%F``

COMMENT NE PAS VERSIONNER UN PROJET ?

- `cp -a projet "projet-`date +%F`"`

- `git tag `date +%F`*`

*Bon ok, à la rigueur

TEXT

COMMENT PARTAGER SON PROJET ?

- Partage réseau
- Copie sur clé USB
- Transfert FTP
- Transfert SCP

COMMENT NE PAS PARTAGER SON PROJET ?

- Partage réseau
- Copie sur clé USB
- Transfert FTP
- Transfert SCP

TEXT

COMMENT DÉPLOYER SON PROJET ?

- Coder en prod
- Transfert FTP
- Transfert SCP
- `git pull`

COMMENT NE PAS DÉPLOYER SON PROJET ?

- Coder en prod
- Transfert FTP
- Transfert SCP
- `git pull`

SOLUTION :

PACKAGER

LE PACKAGING

- apporte la gestion des dépendances
- apporte la gestion des versions
- permet de savoir où trouver le package
- rend la publication simple et rapide

TEXT

STRUCTURE D'UN PROJET

```
── py_foo                    -> répertoire principal (dépôt git...)
   ├── foo                   -> le projet à packager
   │   ├── some_folder
   │   │   ├── __init__.py
   │   │   └── (...)
   │   ├── __init__.py
   │   ├── helpers.py
   │   └── real_stuff.py
   ├── tests                 -> les tests hors du module
   │   ├── test_helpers.py
   │   └── test_real_stuff.py
   ├── README.md             -> pour dire de quoi le projet parle
   └── setup.py              -> fichier qui permet de faire le package
```

TEXT

LE FICHIER SETUP.PY

```
from setuptools import find_packages, setup

setup(
    name='foo',
    version='0.0.1',
    packages=find_packages(exclude=["tests"]),
)
```

LE FICHIER SETUP.PY

```python
from setuptools import find_packages, setup

setup(
    name='foo',
    version='0.0.1',
    packages=find_packages(exclude=["tests"]),
    include_package_data=True,
    description='Foo, Bar Baz!',
    long_description='Foo, a deer, a female deer... '
                     'Bar a note to follow Foo... '
                     'Baz, a name, I call myself...',
    url='https://github.com/myself/foo/',
    author='My Self',
    author_email='foo@bar.com',  # *
)
```

* Rappelez-vous d'aller un jour voir http://bar.com. Non, pas maintenant, merci ;)

LE FICHIER SETUP.PY

Création automatique

```
$ python -m createsetup --name=foo --version=0.0.1
```

LE FICHIER SETUP.PY

Création automatique

```
$ python -m createsetup --name=foo --version=0.0.1
```

MOUAHAHAHA

LE FICHIER SETUP.PY

Pas de création automatique

~~$ python -m createsetup --name=foo --version=0.0.1~~

CRÉER UN FICHIER SETUP.PY

1. trouver sur le web un fichier `setup.py`
2. copier coller son contenu
3. faire les modifications nécessaires
4. publier le package
5. modifier ce qui ne l'a pas été à l'étape 3
6. republier le package

LE FICHIER SETUP.PY EN DÉTAIL

Au programme :

- nom
- version
- description
- dépendances
- fichiers non-python

TEXT

LE PROBLÈME DU NOM

- Le nom du repo GitHub ?
- Le nom du module qu'on `import` ?
- Le nom qu'on `pip install` ?

VERSION

```
1.0.dev456
1.0a1
1.0a2.dev456
1.0a12.dev456
1.0a12
1.0b1.dev456
1.0b2
1.0b2.post345.dev456
1.0b2.post345
1.0rc1.dev456
1.0rc1
1.0
1.0+abc.5
1.0+abc.7
1.0+5
1.0.post456.dev34
1.0.post456
1.0.1
1.1.dev1
```

VERSION: HARD CODING

foo/foo/__init__.py

```
#...
__version__ = '0.0.1a1'
 VERSION = tuple(__version__.split('.'))
# VERSION est maintenant ('0', '0', '1a1')
#...
```

foo/setup.py

```
setup(
   # ...
   version='0.0.1a1',
   # ...
)
```

VERSION: HARD CODING

foo/foo/__init__.py

```
#...
__version__ = '0.0.1a1'
 VERSION = tuple(__version__.split('.'))
# VERSION est maintenant ('0', '0', '1a1')
#...
```

foo/setup.py

```
setup(
   # ...
   version='0.0.1a1',
   # ...
)
```

NOT DRY

Don't Repeat Yourself

VERSION: IMPORTING

foo/foo/__init__.py

```
#...
__version__ = '0.0.1a1'
 VERSION = tuple(__version__.split('.'))
# VERSION est maintenant ('0', '0', '1a1')
#...
```

foo/setup.py

```
from foo import __version__
setup(
    version=__version__,
)
```

VERSION: IMPORTING

foo/foo/__init__.py

```
#...
__version__ = '0.0.1a1'
 VERSION = tuple(__version__.split('.'))
# VERSION est maintenant ('0', '0', '1a1')
#...
```

foo/setup.py

```
from foo import __version__
setup(
    version=__version__,
)
```

YOU'RE GONNA HAVE A BAD TIME

VERSION: EXEC

foo/foo/about.py

```
__version__ = '0.0.1a1'
```

foo/foo/__init__.py

```
from .about import *
```

foo/setup.py

```
about = {}
with open("foo/about.py") as f:
    exec(f.read(), about)
# ...
setup(
    version=about["__version__"],
)
```

DESCRIPTION

`description=`

Nom qui apparaîtra par exemple
dans les résultats de pip search

`long_description=`

Texte qui apparaîtra par exemple
sur les pages web de PyPI

DESCRIPTION: HARD CODING

foo/foo/__init__.py

```
b"""Foo, Bar Baz!"""  # docstring
#...
```

foo/README.md

```
Foo, a deer, a female deer...
Bar a note to follow Foo...
Baz, a name, I call myself...
```

foo/setup.py

```
setup(
    description="Foo, Bar Baz!",
    long_description='Foo, a deer, a female deer... '
                     'Bar a note to follow Foo... '
                     'Baz, a name, I call myself...',
)
```

DESCRIPTION: HARD CODING

NOT DRY

DESCRIPTION: HARD CODING

NOT DRY

DESCRIPTION: IMPORTING

YOU'RE GONNA HAVE A BAD TIME

DESCRIPTION: EXEC

foo/foo/about.py

```
b"""Foo, Bar Baz!"""
#...
```

foo/setup.py

```
setup(
  description=about["__doc__"],
  long_description=open("README.md").read()
)
```

TEXT

ÉVOLUTION DE NOTRE SETUP.PY

```
package_name = 'foo'
about = {}
with open("foo/about.py") as f:
    exec(f.read(), about)

setup(
    name=package_name,
    version=about["__version__"],
    packages=find_packages(exclude=["tests"]),
    include_package_data=True,
    description=about["__doc__"],
    long_description=open('README.md').read(),
    url=about["__url__"],
    author=about["__author__"],
    author_email=about["__author_email__"]
)
```

TEXT

DÉPENDANCES

foo/requirements.txt

```
django==1.11.3
psycopg2==2.7.1
django-extended-choices==1.1.1
pytest==3.1.2
```

DÉPENDANCES

POUR LES CONTRIBUTEURS

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

POUR LES UTILISATEURS

```
$ pip install foo
```

DÉPENDANCES

foo/setup.py

```
setup(
#...
   install_requires=[
      'django==1.11.3',
      'psycopg2==2.7.1',
      'django-extended-choices==1.1.1',
      'pytest==3.1.2',
   ],
#...
)
```

DÉPENDANCES

foo/setup.py

```
setup(
#...
   install_requires=[
      'django==1.11.3',
      'psycopg2==2.7.1',
      'django-extended-choices==1.1.1',
      'pytest==3.1.2',
   ],
#...
)
```

A-T-ON VRAIMENT BESOIN DE ÇA ?

TEXT

DÉPENDANCES

foo/requirements.txt

```
-e .
pytest==3.1.2
```

foo/setup.py

```
setup(
#...
   install_requires=[
      'django==1.11.3',
      'psycopg2==2.7.1',
      'django-extended-choices==1.1.1',
   ],
#...
)
```

DÉPENDANCES: TO PIN OR NOT TO PIN ?

foo/setup.py

```
setup(
#...
   install_requires=[
      'django==1.11.3',  # pour un projet
      'django>=1.8,<2.0',  # pour une librairie
      'django',  # encore mieux pour une librairie
   ],
#...
)
```

ASSETS

- Templates
- CSS
- JS
- Images
- Fixtures JSON ?

ASSETS

foo/MANIFEST.in

```
include README.md
include LICENSE.md
recursive-include foo *.html
recursive-include foo/static *
recursive-exclude foo/media *
recursive-include foo/locale *.mo
recursive-exclude * __pycache__
recursive-exclude * *.py[co]
```

ENTRY POINT

- Votre lib est faite pour être appelée dans la console

```
$ python -m foo
Hello world !
```

Et si on en faisait un vrai exécutable ?

```
$ foo
Hello world !
```

TEXT

ENTRY POINT

foo/setup.py

```
setup(
    entry_points={
        'console_scripts': ['foo=foo:main'],
    },
)
```

foo/foo/__main__.py

```
def main():
  print("Hello World !")

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

TEXT

EN RÉSUMÉ

foo/setup.py

```
from setuptools import find_packages, setup

package_name = 'foo'

requirements = [
    'django==1.11.3',
    'psycopg2==2.7.1',
    'django-extended-choices==1.1.1',
]
```

EN RÉSUMÉ

foo/setup.py

```
classifiers = [
    "Development Status :: 4 - Beta",
    "Operating System :: OS Independent",
    "Intended Audience :: Developers",
    "License :: OSI Approved :: BSD License",
    "Programming Language :: Python :: 3 :: Only"
    "Programming Language :: Python :: 3.7",
]

about = {}
with open("foo/about.py") as f:
    exec(f.read(), about)
```

EN RÉSUMÉ

foo/setup.py

```
setup(
    name=package_name,
    version=about['__version__'],
    packages=find_packages(exclude=["tests"]),
    include_package_data=True,
    description=about['__doc__'],
    long_description=open('README.md').read(),
    url=about['__url__'],
    author=about['__author__'],
    author_email=about['__author_email__'],
    license=about['__license__'],
    classifiers=classifiers,
    install_requires=requirements,
    entry_points={'console_scripts': ['foo=foo:main']},
    zip_safe=True,
)
```

EN RÉSUMÉ

foo/foo/__init__.py

```
from .about import *
from .real_stuff import Foo

__all__ = ["Foo"]
```

foo/foo/about.py

```
"""Foo, Bar Baz!"""
__author__ = 'My Self'
__author_email__ = 'foo@bar.com'
__url__ = 'https://github.com/myself/foo/'
__license__ = 'BSD'
__version__ = '0.0.1'
VERSION = tuple(__version__.split('.'))
```

TEXT

PYPI

La toute première fois il faut "déposer" le nom du paquet

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

Ensuite, les releases

```
$ python setup.py sdist bdist_wheel upload
```

(C'est une bonne idée de tester les paquets avant upload)

IDENTIFIANTS

~/.pypirc

```
[distutils]
index-servers =
    pypi

[pypi]
username: ewjoachim
password: monP@sswordEnClair!!1!  # optionnel
```

INSTALL

```
$ pip install foo
```

TEXT

TOUT CE DONT CE TALK DÉJÀ LONG NE PARLERA PAS

- `pytest` - Les tests en version propre et simple
- `tox` - Lancement des tests dans toutes les configurations
- `prospector` - Pour un code au top

TEXT

TOUT CE DONT CE TALK DÉJÀ LONG NE PARLERA PAS

- Travis / CircleCI - Lancement des tests nuageux
- ReadTheDocs - Documentation nuageuse
- Codecov - Calcul de couverture nuageuse
- shields.io - Pour avoir des badges stylés dans GitHub
- CONTRIBUTING.md, CODE_OF_CONDUCT.md, …

@EWJOACHIM

@TWIDI

MERCI !

RETROUVEZ L'ORIGINAL SUR TWIDI.GITHUB.IO

ET MA VERSION SUR EWJOACH.IM