

pip 使用

記事本:	NAL		
已建立:	2018/8/23 下午 05:01	已更新:	2018/9/24 下午 01:02
作者:	anong9420@gmail.com		
URL:	https://translate.googleusercontent.com/translate_c?depth=1&hl=zh-TW&prev=search&rurl=translate.go...		

python管理包發展

1. distutils

distutils 是 python 標準庫的一部分，2000年發布。使用它能夠進行python模塊的安裝與發佈。setup.py 就是利用 distutils 的功能寫成。

2. setuptools 和 distribute

setuptools 是一個為了增強 distutils 而開發的集合，2004年發布。它包含了 easy_install 這個工具。

ez_setup.py 是 setuptools 的安裝工具。

setuptools 像是一個項目名稱，類似基礎套件。而 easy_install 是這個項目中提供的工具，依賴基礎套件工作。

3. pip

pip 是目前 python 包管理的標準，2008年發行。用作 easy_install 的替代品，但仍有大量的功能建立在 setuptools 套件之上。

pip 可以利用 requirements.txt 實現dependency的安裝。在 setup.py 中，也存在install_requires 表來指定相依套件的安裝。

4.wheel

wheel 本質上是zip 包格式，它使用 .whl 擴展名，用於 python module的安裝，是為了替代 Eggs。

<https://packaging.python.org/discussions/wheel-vs-egg/>

	pip	easy_install
Installs from Wheels	Yes	No
Uninstall Packages	Yes (pip uninstall)	No
Dependency Overrides	Yes (Requirements Files)	No
List Installed Packages	Yes (pip list and pip freeze)	No
PEP 438 Support	Yes	No

Installation format	'Flat' packages with egg-infometadata.	Encapsulated Egg format
sys.path modification	No	Yes
Installs from Eggs	No	Yes
pylauncher support	No	Yes
Multi-version installs	No	Yes
Exclude scripts during install	No	Yes
per project index	Only in virtualenv	Yes, via setup.cfg

上傳套件到PyPI

- 將python package或是python script,利用PyPI將之散播到全世界。

```
MyProject-|--README.md
            |--setup.py <-- 加入setup.py
            |--setup.cfg <-- 加入setup.cfg
            |--.pypirc <-- 加入.pypirc(注意,此檔不要加入stage提交出去!)
            |--runner
            |--pkgA---|--__init__.py
            |           |--modA_1.py
            |           |--modA_2.py
            |--pkgB---|--__init__.py
                       |--modB.py
```

setup.py

```
from distutils.core import setup

setup(
    name = 'MyProject',
    packages = ['pkgA', 'pkgB'],
    scripts = ['runner'],
    version = '1.0',
    description = 'My first project',
    author = 'dokelung',
    author_email = 'dokelung@gmail.com',
    url = 'https://github.com/dokelung/MyProject',
    download_url = 'https://github.com/dokelung/MyProject/tarball/v1.0',
    keywords = ['Good Project'],
    classifiers = [],
)
```

script 是專案的執行檔,所謂的安裝就是將script放到可執行路徑裡(usr/local/bin)

- 設定setup.cfg

因為readme是markdown file所以需要新增一個setup.cfg的檔案:

```
[metadata]
description-file = README.md
```

HTML 是一種發布的格式，Markdown是一種編寫的格式
Markdown 語法分為:標準語法和擴展語法。



• 建立.pypirc設定檔

.pypirc

```
[distutils]index-servers =  
pypi  
  
[pypi]  
repository: https://pypi.python.org/pypi  
username: dokelung  
password:
```

• 上傳到PyPI

```
$ python setup.py register -r pypi  
$ python setup.py sdist upload -r pypi
```

結論:

1. 撰寫 setup.py
2. 撰寫 setup.cfg 如果 README 是 markdown file
3. 撰寫 .pypirc
4. 用 `python setup.py register` 註冊
5. 用 `python setup.py sdist upload` 上傳

Linux系統上安裝

1. 到pypi.python.org上搜索pip，下載最新版本的source code壓縮包。
2. 在local解壓縮，進入script路徑，執行：`python setup.py install`即可安裝。
3. 測試：打開cmd，輸入：`pip install web.py`
4. 進入python命令行，輸入：`import web`，若導入成功則說明pip安裝成功。

注：ubuntu系統可以直接使用該命令安裝：`sudo apt-get install python-pip`

pip常用命令

- 安裝軟件包：`pip install 包名`
- 升級pip自身：`pip install --upgrade pip`
- 查看已經通過pip安裝的包：`pip list`
- 顯示當前已經通過pip安裝的包及版本號：`pip freeze`
- 將pip freeze命令的結果重定向到requirements.txt文件中：`pip freeze > requirements.txt`
- 使用已有的requirements.txt文件在另一個環境上安裝各種包（比如在一台新電腦上）：`pip install -r requirements.txt`
- 查看某個已經安裝的包的詳情：`pip show 包名`
- 查看過期的包：`pip list --outdated`
- 安裝包到用戶目錄：`pip install 包名--user`
- 安裝本地的安裝包：`pip install 目录|文件名`
- 卸載包：`pip uninstall 包名`
- 升級包：`pip install 包名--upgrade`
- 顯示包所在目錄：`pip show -f 包名`
- 搜索包：`pip search 关键字`
- 查詢可升級的包：`pip list -o`
- 下載包但不安裝：`pip install 包名-d 目录`
- 打包：`pip wheel 包名`

`yes | pip install packagename`
`pip uninstall --yes packagename`

