

packages vs. package_dir

井民全, Jing, mqjing@gmail.com

Key:

packages: 告訴這次你要安裝的 **package** 名稱列表

package_dir: 告訴系統他們對應的目錄名稱為何

有時候, 你會希望 整個 python 原始碼 package 集中放在 src 目錄下, 這時候只要更改 package root 的 convention 設定就好了.

例如:

- `package_dir{":":"src"}` # 說明 *package root*, 從目前目錄. 換成在目錄 *src* 下找

通常 foo package 下的模組, 會放在 foo 目錄下, 但是若你希望讓 foo package 都放到 lib 目錄下. 也是使用 package_dir 這個設定.

例如:

- `package_dir{'foo':'lib'}` # 說明 *package foo* 換成在目錄 *lib* 下找

Detail:

定義: **packages**: specified your python modules can be found in the packages list.

Ex1:

- **packages** = ["foo"]:
表示要處理的 package 叫做 foo. 系統在 foo 目錄下, 預期有 __init__.py. 也就是 foo/__init__.py

Ex2:

- **packags** = find_packages("src"):
find_packages 會傳回 src 目錄下的每一個 package name list, 在那些 name 目錄下都有一個 __init__.py

定義: **package_dir**: to tell the distutils about your **convention**

Ex1: 換 package root 到其他目錄的例子

- **package_dir**{"src"} # 說明 package root, 從目前目錄. 換成在目錄 src 下找
- **packages** = {'foo', 'foo.bar'}
 - 意思是 package foo 放在在目錄 src/foo/__init__.py
 - 意思是 package foo.bar 放在目錄 src/foo/bar/__inti__.py

Ex2: 換 package name 與 目錄對應名稱的例子

- `package_dir{'foo':'lib'}` # 說明 *package foo* 換成在目錄 *lib* 下找
- `packages = {'foo', 'foo.bar'}`
 - 意思是 package foo 在目錄 `lib/__init__.py`
 - 意思是 package foo.bar: 在目錄 `lib/bar/__inti__.py`