

# 將你的python code 打包 - pyinstaller



CW Lin

Sep 26, 2019 · 5 min read

讓沒有python 的電腦也能執行你的程式!



Photo by Safar Safarov on Unsplash

python 有很多不同的版本，更別說有一堆package了，當你寫了一支python code 想在很多地方部署使用，去一台一台電腦安裝相關環境實在太費時，這時我們可以透過pyinstaller 將執行這支程式相關的所有東西給打包成一個.exe執行檔，讓其他電腦可以直接執行!

## 簡介

執行pyinstaller 後他會先分析你的code，看有哪些需要的module、library、interpreter，然後蒐集相關的資源，最後全部包起來生成exe檔。

這邊提醒一下:

1. 引入套件時盡量使用 `from ...import ...`，如果直接使用`import`的話會把整個包都打包進去，而無謂的增加工具的大小。
2. `pyinstaller` 打包後的`exe` 只能在與打包時相同的作業系統下執行，在`windows` 下打包的無法在`linux`使用，另外 64位元的電腦下打包也無法在 32位元上使用(但反過來似乎可以)。

## 安裝&使用

### 安裝install PyInstaller from PyPI

```
pip install pyinstaller
```

### 相關參數

```
-h :help  
-F :將程式打包成單一執行檔 (適合較簡易的代碼或只有單一.py檔)  
-D :打包多個文件，exe檔及依賴的東西會一起放置在dist資料夾內 (適合框架形式的程式)
```

基本上我只有用到這幾個參數~其他更多參數可以參考這篇。

若你的程式屬於框架型的，有很多子程式及依賴模組，建議使用`-D`，若使用`-F` 則會產生一個容量很大的`exe`檔，且開啟時速度很緩慢。

### Example

比如我寫了一個小工具計算 1到 user input的正整數的總和的程式並命名為 `my_sum.py` :

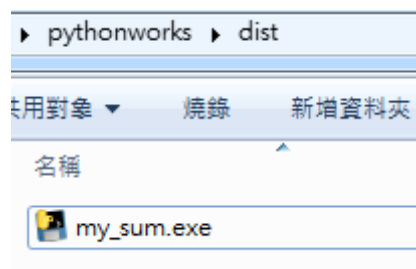
```
while True:  
    num_input = input('please input an positive integer:')  
    try:  
        num = int(num_input)  
        if num>0:  
            ans = (1+num)*num/2  
            print('sum from 1 to %d is: %d \n'%(num,ans))  
        else:  
            print('your input is not positive integer')  
            continue  
    except:  
        print('your input is not positive integer')  
        continue
```

然後在該路徑下執行:

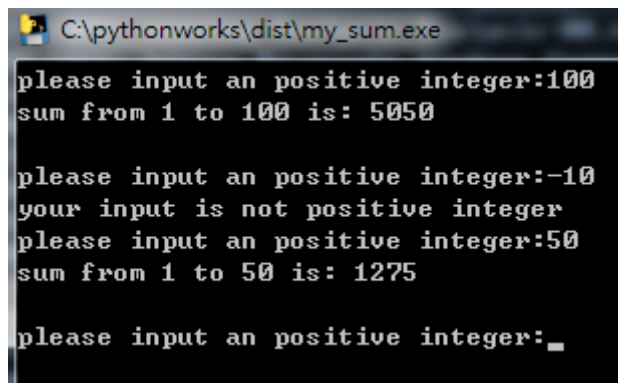
```
pyinstaller -F my_sum.py
```

接著你會發現:

1. 在路徑下產生了my\_sum.spec: 這個是打包時相關的設定，後面再細講
2. 建立build 資料夾，log紀錄檔與相關檔案於資料夾 build 中
3. 建立dist 資料夾
4. 建立執行檔exe file 於dist資料夾內



這個exe檔就可以直接在其他電腦上使用了



## .spec 檔

spec檔是你打包時的一些相關設定

```
a = Analysis(['my_sum.py'],
             pathex=['C:\\pythonworks'],
             binaries=[],
             datas=[], 除了程式外需要一起打包近來的檔案路徑
             hiddenimports=[], 有些module打包時會有問題，在這邊hidden
             hookspath=[],
             runtime_hooks=[],
             excludes=[],
             win_no_prefer_redirects=False,
             win_private_assemblies=False,
             cipher=block_cipher)
```

```

    cipher=block_cipher,
    noarchive=False)
pyz = PYZ(a.pure, a.zipped_data,
    cipher=block_cipher)
exe = EXE(pyz,
    a.scripts,
    a.binaries,
    a.zipfiles,
    a.datas,
    [],
    name='my_sum',
    debug=False,
    bootloader_ignore_signals=False,
    strip=False,
    upx=True,
    upx_exclude=[],
    runtime_tmpdir=None,
    console=True ) 若你的執行檔想要在背景執行而不顯示那個黑框框的話可以使用console=False

```

當你程式不是那麼簡單時，打包時很可能遇到很多error，尤其像是”no module names XXX”這種，這時候就需要一直google。

我目前的經驗，spec 裡 ['cython', 'sklearn', 'sklearn.ensemble', 'sklearn.neighbors.typedefs', 'sklearn.neighbors.quad\_tree', 'sklearn.tree.\_utils', 'sklearn.utils.\_cython\_blas'] 這些都給他設定在hiddenimport 裡應該可以擋掉大部分。

datas的寫法為(路徑，名稱)，如: [('mtcnn-model\\\*', 'mtcnn-model'), ('img\_for\_registered\\\*', 'img\_for\_registered')], 這樣他就會把這些folder給打包進dist。如果你是使用 -D 的模式，你也可以不寫在datas裡，當他包好後在自己手動拉到dist 資料夾內也行。

另外有些module像是mxnet好像也無法直接自動幫你包進去，所以你需要自己到你python 的site-package裡找mxnet的資料夾，把它整個複製到dist裡就可以了～

當你改寫好spec檔後，可以透過

```
pyinstaller -D XXX.spec
```

指定用這個spec 來重新打包。

總之，當你的程式越複雜，呼叫到越多module時，越可能踩到一堆坑，通常都google 的到，相信你不會是第一個踩到坑的人!

以下是我將 ArcFace 的程式整理好後透過 pyinstaller 把它打包成exe檔，做成一個方便大家使用的人臉辨識工具：

## Reference

【Python】將Python打包成exe檔

pyinstaller参数介绍以及总结

Python    Pyinstaller

About   Help   Legal

Get the Medium app

