

原创 Python中的图片打包与pyinstaller中的spec文件简介

2018-08-25 16:36:57 SDCS\_Zyx 阅读数 5660

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。  
本文链接：[https://blog.csdn.net/SCDS\\_Zyx/article/details/82052396](https://blog.csdn.net/SCDS_Zyx/article/details/82052396)

一.Python中的图片打包

1.经过科学搜索后有以下几种打包方式：

①将图片转化为py文件中的变量，通过import该文件来生成需要的图片，使用后移除图片  
(该方法可以通过库base64和wxpython实现)

②将图片放在同一目录下的文件夹中，随程序一起使用

③使用工具对exe文件和图片进行二次打包

可能还有其他方式，在此只列取这几种

2.本人比较懒，选择的是较为方便好看的第一种方式，其他方式就不讨论了：

①base64

将图片转化为base64编码的py文件中的变量

```
1 import base64
2
3 def pic2py(picture_name):
4     open_pic = open("%s" % picture_name, 'rb')
5     b64str = base64.b64encode(open_pic.read())
6     open_pic.close()
7
8     # 注意这边b64str一定要加上.decode()
9     write_data = 'img = "%s"' % b64str.decode()
10    f = open('%s.py' % picture_name.replace('.', '_'), 'w+')
11    f.write(write_data)
12    f.close()
13
14 if __name__ == '__main__':
15     name = input("你想转化的图片的完整文件名：")
16     pic2py(name)
17
18 #使用方式如下
19
20 '''
21 from in_ico import img
22 ...
23 tmp = open('in.ico', 'wb')
24 tmp.write(base64.b64decode(img))
25 tmp.close()
26
27 #使用图片
28
29 os.remove('in.ico')
30 '''
```

②wxpython

安装wxpython库后，在命令框中使用命令：

python img2py.py -n = 引用图片名（含格式） -i = 生成py文件名（含.py）

得到一个py文件，图片已转化为该文件中的一个二进制编码的变量，通过类似上一种方法中的使用方法，解码后即可使用

二.pyinstaller中的spec文件

1.可以在[官网](#)了解一下

简单介绍:

通过pyi-makespec name.py命令来生成一个自定义的spec文件

```
1 # -*- mode: python -*-
2
3 block_cipher = None
4
5
6 a = Analysis(['name.py'],
7             pathex=['D:\\ME'],
8             binaries=[],
9             datas=[],
10            hiddenimports=[],
11            hookspath=[],
12            runtime_hooks=[],
13            excludes=[],
14            win_no_prefer_redirects=False,
15            win_private_assemblies=False,
16            cipher=block_cipher)
17 pyz = PYZ(a.pure, a.zipped_data,
18          cipher=block_cipher)
19 exe = EXE(pyz,
20          a.scripts,
21          exclude_binaries=True,
22          name='name',
23          debug=False,
24          strip=False,
25          upx=True,
26          console=True )
27 coll = COLLECT(exe,
28               a.binaries,
29               a.zipfiles,
30               a.datas,
31               strip=False,
32               upx=True,
33               name='name')
```

spec文件分为Analysis, pyz, exe, coll四个部分

- ①Analysis:首先是需要打包的py文件，通过一个字符串列表储存；然后是py文件的路径；datas是需要引用的文件（图片等）
- ②exe：要生成exe文件时，name是exe文件的名字， console是是否在打开exe文件时打开命令框
- ③coll：收集前三个部分的内容进行整合

pyinstaller通过最开始生成的spec文件进行打包，也可以自定义spec文件后，使用pyinstaller -F name.spec命令来打包

2.打包后的spec文件

如果使用了--add-data命令后，在Analysis中的datas中出现引用的图片的名称;使用-i 命令后，在exe在最后会出现icon = 图标名

**注意：**使用--add-data命令时，使用方式如：--add-data 图片名；位置（同一目录下可以用相对路径）  
该命令可以打包使用绝对路径引用的图片，但一旦通过该路径找不到图片时即会报错（如在其所在目录下打开exe文件，原路径上图片删除后）