1、使用pip install Pyinstaller  命令安装

2、使用命令 pyinstaller -F  \*.py打包成exe

3、在\dist文件夹下找到exe;

一、pyinstaller相关参数

-F, –onefile 打包一个单个文件，如果你的代码都写在一个.py文件的话，可以用这个，如果是多个.py文件就别用  
-D, –onedir 打包多个文件，在dist中生成很多依赖文件，适合以框架形式编写工具代码，我个人比较推荐这样，代码易于维护  
-K, –tk 在部署时包含 TCL/TK  
-a, –ascii 不包含编码.在支持Unicode的python版本上默认包含所有的编码.  
-d, –debug 产生debug版本的可执行文件  
-w,–windowed,–noconsole 使用Windows子系统执行.当程序启动的时候不会打开命令行(只对Windows有效)  
-c,–nowindowed,–console   
使用控制台子系统执行(默认)(只对Windows有效)

pyinstaller -c  xxxx.py

pyinstaller xxxx.py --console

-s,–strip 可执行文件和共享库将run through strip.注意Cygwin的strip往往使普通的win32 Dll无法使用.  
-X, –upx 如果有UPX安装(执行Configure.py时检测),会压缩执行文件(Windows系统中的DLL也会)(参见note)  
-o DIR, –out=DIR 指定spec文件的生成目录,如果没有指定,而且当前目录是PyInstaller的根目录,会自动创建一个用于输出(spec和生成的可执行文件)的目录.如果没有指定,而当前目录不是PyInstaller的根目录,则会输出到当前的目录下.  
-p DIR, –path=DIR 设置导入路径(和使用PYTHONPATH效果相似).可以用路径分割符(Windows使用分号,Linux使用冒号)分割,指定多个目录.也可以使用多个-p参数来设置多个导入路径，让pyinstaller自己去找程序需要的资源  
–icon=<FILE.ICO>   
将file.ico添加为可执行文件的资源(只对Windows系统有效)，改变程序的图标  pyinstaller -i  ico路径 xxxxx.py

–icon=<FILE.EXE,N> 将file.exe的第n个图标添加为可执行文件的资源(只对Windows系统有效)  
-v FILE, –version=FILE 将verfile作为可执行文件的版本资源(只对Windows系统有效)