

Freezing the program

Applications developed in the Python language can be frozen for Windows, Linux, and Mac systems using the PyInstaller tool, introduced on pages 182-183, to create a single standalone executable file. Apps that require the inclusion of additional files, such as image files, require the script file to be modified so the PyInstaller tool can readily locate them at their absolute address:



The `file` attribute is preceded by TWO underscore characters and followed by TWO underscore characters.

- 1 Open the `lotto.py` program file and add this utility function definition at the very start of the script


```
def resource_path( relative_path ) :
    absolute_path = os.path.abspath( __file__ )
    root_path = os.path.dirname( absolute_path )
    base_path = getattr( sys, '_MEIPASS', root_path )
    return os.path.join( base_path, relative_path )
```
- 2 Modify the assignment to the `img` variable – to identify the image file’s location using the added utility function


```
img = PhotoImage( file = resource_path( 'logo.gif' ) )
```

```
lucky.py - C:\MyScripts\lucky.py
File Edit Format Run Options Window Help
def resource_path( relative_path ) :
    absolute_path = os.path.abspath( __file__ )
    root_path = os.path.dirname( absolute_path )
    base_path = getattr( sys, '_MEIPASS', root_path )
    return os.path.join( base_path, relative_path )

# Widgets:
from tkinter import *
window = Tk()
img = PhotoImage( file = resource_path( 'logo.gif' ) )
```



lucky.py

- 3 Save the modified file as “lucky.py” in your MyScripts folder, then open a Command Prompt in that folder
- 4 Begin a command with `pyinstaller --onefile --noconsole` to instruct PyInstaller to create a single file, and to prevent the program also opening a console window
- 5 Continue the command with `--add-binary logo.gif,.` to include the image file in the single executable
- 6 Complete the command with the `lucky.py` script name – so the command now looks like that on the opposite page