



Images haven't loaded yet. Please exit printing, wait for images to load, and try to print again.

Oct 5, 2017 · 4 min read



Making a Stand Alone Executable from a Python Script using PyInstaller

Making a Stand Alone Executable from a Python Script using PyInstaller

There are plenty of tools available for converting python script into executable. For example, checkout:

- [PyInstaller](#)
- [py2exe](#)

For Python 2, I used to prefer py2exe. Its a neat tool that does the trick by making a stand alone executable from python script. The one problem I faced was, py2exe used to support only python 2! Then I moved on to PyInstaller. The good thing is, py2exe now provides support to Python 3 as well! I might share my py2exe experience as well but, that will be in another article.

This article is all about PyInstaller! Its a handy tool specially when you can create an .exe file for windows with only one command! Although I prefer Linux, I had to create executables for windows all the time targeting various Windows distributions (Yes, It was Clients' requirement)

Lets explore the process of creating executable using PyInstaller.

Installation:

Hassle free installation, all you have to do is to run the following command (if you have already installed python) and that's it!

```
pip install pyinstaller
```

Note: If you are working on Windows. You might need to install **PyWin32** as well! Download it from here:

<https://sourceforge.net/projects/pywin32/files/>

And, If you need additional help, you can always head right to the Official Documentation of PyInstaller:

<https://pyinstaller.readthedocs.io/en/stable/installation.html>

If you are using different package management tool such as **conda** (for Anaconda) then you will have to use the following commands instead:

```
conda install -c conda-forge pyinstaller  
conda install -c anaconda pywin32
```

Please, find suitable installation documentation for your package management tool for more details.

Create Executable:

Now that, you have installed PyInstaller all you have to do is find the python script that you want to convert to an executable. Just navigate to your python script directory. Now, open up your Terminal/Command Prompt in the script directory and, *test your python script*:

```
python your_script.py
```

Make sure, the script work as expected. Now, The command I prefer for compiling the script into executable is the following one!

Run the command:

```
pyinstaller --onefile <your_script_name>.py
```

This will create a standalone executable in the dist directory of your script folder. Don't worry, if the folder doesn't exist it will create one automatically.

Notice that we passed an argument “— onefile”. This argument tells PyInstaller to create only one file. If you don't specify this, the libraries will be distributed as separate file along with the executable.

NOTE: The format/extension of the executable will depend on which operating system you used for compilation. For example, if you run the PyInstaller command on windows the executable file will be **.exe**. If you run it on Linux the extension will depend on the distribution you are using.

You can only create executable for your Operating system, i.e. the Operating system you used to compile the executable.

For example, It is not possible to create a Windows executable (.exe) by directly running a Pyinstaller command on a Linux Distribution and vice versa. (If you know a way to do it, feel free to comment)

What you can do is run Virtualbox or similar application to run the OS virtually and, create executable in that virtual os and export it later. Also, Wine works as well! (Though, I didn't try it myself)

If your python script depends on additional executables say for example, phantomjs or chromedriver. You may have to put these executable in the same directory of your executable!

For these type of dependencies, when I'm working on Windows Platform. I usually package them using NSIS.

The procedure is like below (Assuming NSIS already installed):

- Create a ZIP of all the necessary files i.e script's executable, dependencies, readme etc.
- run NSIS
- Click on the make installer from zip archive option.
- Select the zip file
- Provide a name and create the Installer.exe/Setup.exe file.

This executable is basically an extractor that extracts all the necessary file in a directory. Cool!

Valuable resources:

- [Read the answers of this stackoverflow question](#)
- [PyInstaller Source Code](#)
- [PyInstaller Supported Python Packages](#)
- [py2exe official tutorial](#)
- [Read reddit discussion on this article](#)

Thanks for reading this far! If you liked this article please, feel free to clap and, spread the love! :)

Author:



Wasi Mohammed Abdullah

Thinker, Day Dreamer, Python Enthusiast, Javascript Admirer An Introvert with Exception!

CEO, Founder

Dreamcatcher IT

twitter: twitter.com/wasi0013

github: github.com/wasi0013

facebook: fb.me/wasi0013

Please feel free to contact if you need any help.

