How to use pyinstaller?

Asked 7 years, 1 month ago Modified 2 years, 11 months ago Viewed 85k times

22

Okay so I'm a complete noob in programming and I'm trying to compile a simple program I wrote that takes in a string and prints out the string in morse code it's called morse.py. I installed pyinstaller using

```
pip install pyinstaller
```

and I am trying to compile my program using pyinstaller.

Now I've searched a bit and it says that I need to write pyinstaller morse.py, but I don't really know where to write that. I tried moving to the directory of my program and doing that in CMD but it didn't work. I tried making a python program in the same directory and doing that and that also didn't work. I couldn't find anything very helpful to tell me exactly how to compile the file.

Can someone please help?

python compilation pyinstaller

Share Edit Follow Flag

edited Dec 24, 2015 at 13:27
phts
3,869 ● 1 ● 18 ● 31

asked Dec 24, 2015 at 13:21
user3333708
367 ● 1 ● 5 ● 10

4 Answers

Sorted by: Highest score (default)

29

✓

I would suggest to first read the **[Using Pyinstaller section in the documentation](#)** of the module itself.

You can also use some **tutorials** (e.g. Matt Borgerson's one).

In order to recap you should:

- **write your script** and make sure that it works
- run from the **command line**:

  ~\ pyinstaller *your_file_name*.py
- this command will generate a *your_file_name*.spec file where you can include all the dll required by your application and any custom settings (Using Spec Files)

- once you have decided what to include in your .exe application you can run from the command line

  ~\ pyinstaller [*option1*] [*option2*] *your_file_name*.py

You can find the full list the **options** in the documentation. An example could be *pyinstaller.exe --onefile --windowed --icon=app.ico app.py* where:

- --onefile: Create a one-file bundled executable.
- --windowed: Parameter to chooseif you are compiling in Mac OS X or Windows
- --icon= : Choose the file to use as icon for file.

You can create your exe file very easily also with **py2exe**.

Share Edit Follow Flag

answered Dec 24, 2015 at 14:29
mabe02
2,596 ● 2 ● 20 ● 33

3

Try to write full path to pyinstaller (for example = C:\Users\user\AppData\Local\Programs\Python\Python35-32\Scripts\pyinstaller.exe)

full cmd string must look like:

```
C:\Users\user\AppData\Local\Programs\Python\Python35-32\Scripts\pyinstaller.exe --onefile myscript.py
```

Share Edit Follow Flag

answered Jan 17, 2018 at 17:54
Андрей Вовченко
31 ● 1

2

I don’t have much experience in Python. I am able to write simple but useful scripts. I was able to convert them to .exe using pyinstaller. But the more programs I write, the more python’s versions I have (and packages and virtual environments). What was working in the beginning suddenly stopped working and got me very confused. But jumping from one page to another I figured it out. My post is to help another newbies. I’m using Windows, but some rules are the same probably for other systems. I’m writing it in March 2020. Pyinstaller doesn’t work properly with the newest python version which is now 3.8. So firstly you need to install Python 3.7. Secondly choose the version accordingly to type of your system. It can be 32 or 64. You can install more than one Python’s version. In command line when you type “py” your computer will look for the newest Python’s version - I mean the one with the highest version number. It’s because of the file C:\Windows\py.exe which appears when you install your first Python. If today you have installed only python 3.7, “py” will run Python 3.7. But next day when you install Python 3.8 and you type “py” it will run Python 3.8. If you want to use the older one just type py -3.7

`C:\Users\Ania>py` Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AM D64)] on win32 Type "help", "copyright", "credits" or "license" for more information.

```
C:\Users\Ania>py -3.7
```

Python 3.7.7 (tags/v3.7.7:d7c567b08f, Mar 10 2020, 10:41:24) [MSC v.1900 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information.

Now to work with your project you need to create a virtual environment for it. Remember to use the correct python version. First let’s make a folder for it.

`D:\PYTHON>mkdir my_new_great_project`

Go to that folder:

```
D:\PYTHON>cd my_new_great_project
```

Create the venv:

```
D:\PYTHON\my_new_great_project>py -3.7 -m venv venv
```

The first “venv” above is the name of a python module which we are using. The second “venv” is the name of the virtual environment which we are creating.

Activate the venv:

```
D:\PYTHON\my_new_great_project>venv\Scripts\activate.bat
```

You should see:

```
(venv) D:\PYTHON\my_new_great_project>
```

Now you can use pip to install whatever external packages your project needs.

```
(venv) D:\PYTHON\my_new_great_project>pip install pandas
```

Pip “knows” that you need packages for python 3.7. When you create your project files don’t put them in “venv” folder.

To run your project from the venv just type its name

```
(venv) D:\PYTHON\my_new_great_project>great_project.py
```

If you are done with your project, everything works perfectly and you need .exe file, it's time to install pyinstaller in your venv.

```
(venv) D:\PYTHON\my_new_great_project>pip install pyinstaller
```

To check what external packages you've already installed just type "pip list":

```
(venv) D:\PYTHON\my_new_great_project>pip list
```

Package Version

---

altgraph 0.17 future 0.18.2 numpy 1.18.2 pandas 1.0.3 pefile 2019.4.18 pip 19.2.3 PyInstaller 3.6 python-dateutil 2.8.1 pytz 2019.3 pywin32-ctypes 0.2.0 setuptools 41.2.0 six 1.14.0

Now from venv you can run pyinstaller to make an .exe file:

```
(venv) D:\PYTHON\my_new_great_project>pyinstaller --onefile --name my_project
great_project.py
```

Now your project is available as an single .exe file in the folder called "dist".

I hope it helps someone.

Share Edit Follow Flag

edited Mar 21, 2020 at 16:52

answered Mar 21, 2020 at 15:22
aniaszka
21 ●3

1

Hello i made a code in python, i used it to turn itself into a exe.

Make sure to have it in same directory as the file you want converted.

```
import subprocess
import shutil
import os

cmd = input("file name: ") # asks user for filename
extra = input("extra commands? eg -w -F if using two or more put a space between:
") # asks for extra options
suros = "pyinstaller "+ extra + " " + cmd + ".py" # sets run command
current_workin_path = os.getcwd() # gets current working path
dirt = current_workin_path + "/"+ cmd # set current working path for folder
creation
os.mkdir(dirt) # creates folder for working path and for a copy
shutil.copy(cmd + ".py", current_workin_path + "/" + cmd + "/" + cmd + ".py") #
creats copy of python file in the newly set path
os.chdir(current_workin_path + "/" + cmd + "/") # changes current working path to
```

```
newly created one
subprocess.run(suros) # runs the command set eariler
os.remove(cmd + ".py") # delete the copy of python file
```

Hope this helps someone.

Share Edit Follow Flag

answered Mar 7, 2020 at 3:39

Darthbeaverdog

21 ● 1