How to Turn Your Python Script into an Executable File

Rohit Saroj

Are you a data scientist who wants to share your Python code with others without revealing your source code or requiring them to install Python and dependencies? If so, you might be interested in converting your Python script into an executable file that can run on any Windows machine.

An executable file is a file that can be installed or run on a computer without requiring any additional software or libraries. It has the .exe extension and is commonly used for software applications on Windows. By converting your Python script into an executable file, you can protect your code from being modified or stolen, make it easier for others to use your program, and schedule tasks to run automatically.

In this article, I will show you two simple methods to convert a Python file to an executable file using PyInstaller and auto-py-to-exe. These are two popular Python libraries that can create standalone executables from Python scripts. You will need Python 3.6 or higher installed on your computer to follow along.

Method 1: Using PyInstaller:

Reference link

PyInstaller is a Python library that can analyze your code and bundle it with the required modules and libraries into a single executable file. It supports many platforms, including Windows, Linux, and Mac OS X. PyInstaller can also handle complex cases such as importing data files, hidden imports, GUI applications, etc.

To use PyInstaller, you need to install it first using pip:

pip install pyinstaller

Then, you need to write your Python script and save it with the .py extension. For this example, I will use a simple script that prints "Hello World" and saves it as hello.py:

print("Hello World")

Next, you need to open a command prompt and navigate to the directory where your script is located. Then, you need to run the following command:

```
pyinstaller hello.py
```

This will create a folder called dist that contains the executable file hello.exe. You can double-click on this file to run it or share it with others.

If you want to create a single-file executable that does not require any additional files or folders, you can use the --onefile option:

```
pyinstaller --onefile hello.py
```

This will create a single hello.exe file in the dist folder that contains all the necessary code and resources.

If you want to hide the console window when running the executable file, you can use the --noconsole option:

```
pyinstaller --noconsole --onefile hello.py
```

This will create a windowless executable file that runs in the background.

You can also customize other aspects of the executable file, such as the icon, the name, the version, etc. by using various options or creating a spec file. You can refer to the PyInstaller documentation for more details.

Method 2: Using auto-py-to-exe:

Reference link

auto-py-to-exe is another Python library that can convert Python scripts to executable files. It is based on PyInstaller but provides a graphical user interface (GUI) that makes it easier to use. You can simply select your script, choose your options, and click a button to generate the executable file.

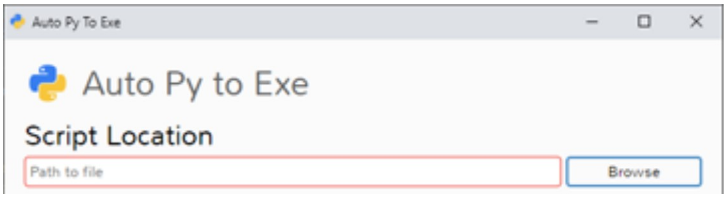
To use auto-py-to-exe, you need to install it first using pip:

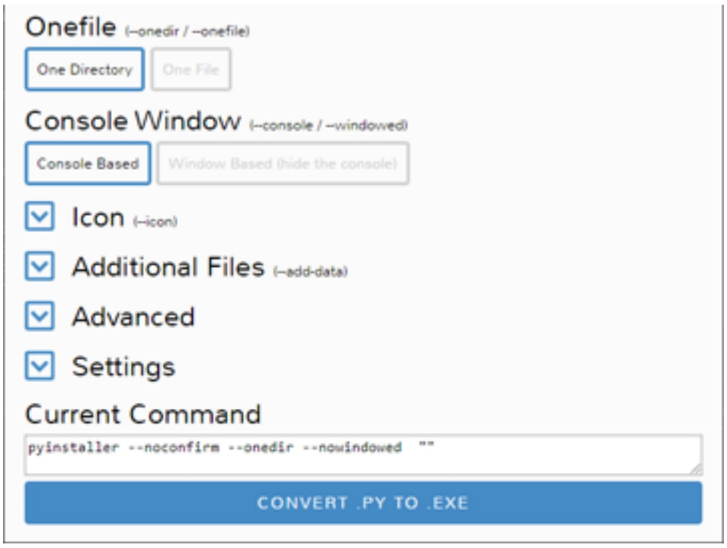
```
pip install auto-py-to-exe
```

Then, you need to run the following command to launch the GUI:

```
auto-py-to-exe
```

This will open a window that looks like this:





Onefile (--onedir / --onefile)

One Directory

Console Window (--console / --windowed)

Console Based

Current Command

```
pyinstaller --noconfirm --onedir --nowindowed  ""
```

One File
Window Based (hide the console)
Icon (--icon)
Additional Files (--add-data)
Advanced
Settings
CONVERT .PY TO .EXE

Auto PY to EXE simple graphical interface

Here, you can select your script by clicking on the Browse button next to Script Location. You can also choose whether you want a single-file or a one-folder executable by selecting One File or One Directory under Output Options.

You can also change other settings such as the icon, the name, the console window, etc. by clicking on the Advanced tab and modifying the fields under Additional Files or Window Based Options.

Once you are done with your settings, you can click on the Convert .py to .exe button at the bottom of the window. This will start the conversion process and show you the output in the Console tab.

When the conversion is finished, you can find your executable file in the output folder specified under Output Options. You can then run it or share it with others.

Conclusion:

In this article, I showed you how to convert a Python file to an executable file using PyInstaller and auto-py-to-exe. These are two simple methods that can help you protect your code, distribute your program, and automate your tasks.

I hope you found this article useful and learned something new. If you have any questions or feedback, please feel free to leave a comment below. Happy coding!

> Click ➡️ subscribe if you want to see more content like this, your support 🫶 fuels me to keep writing! 🌟👦💻🚀