Creating Standalone Python Applications

Using auto-py-to-exe

`auto-py-to-exe` is a graphics program that enables you to create executable Python applications. It wraps a graphical interface around a program called `pyinstaller`. Using pyinstaller on its own involves working in a console, so using `auto-py-to-exe` is more straightforward.

In order to use `auto-py-to-exe` you will need to have the Python interpreter installed.

Installing `auto-py-to-exe` does involve working in a console, but once it is installed you will have a graphical interface with which to create your executable files.

In windows, the easiest way to open a console is by typing **cmd** in the search bar. If you are using a mac, the console is usually referred to as a terminal, and it can be opened in a variety of ways - for example by clicking the **Launchpad** icon in the **Dock** and typing "Terminal" in the search field, and then clicking **Terminal**.

To install `auto-py-to-exe`, you will need to use the `pip` program, which is located in a directory called `Scripts` within the folder where your python interpreter is located.

If you are not sure where your Python interpreter is located this is you can find it by opening a console and typing the following at the prompt:

```
where python
```

It may be that a path to the `Scripts` folder has been added to your system, in which case you can type the `pip` command from anywhere within the console window. If not, you can either navigate to this folder, or you can add the path by typing the following at the prompt:

```
set PATH=%PATH%;\your\path\here\
```

You will need to replace your\path\here\ with the correct path to your folder. See figure 1 as an example.

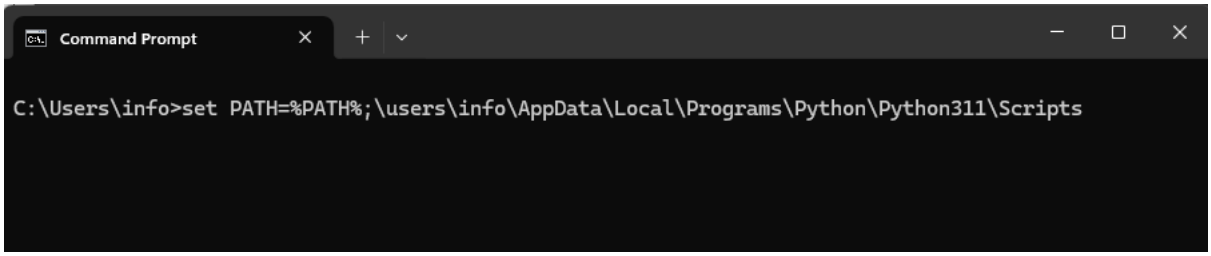

Fig. 1. Setting a path

To install `auto-py-to-exe` type the following command in the console window (see figure 2):

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

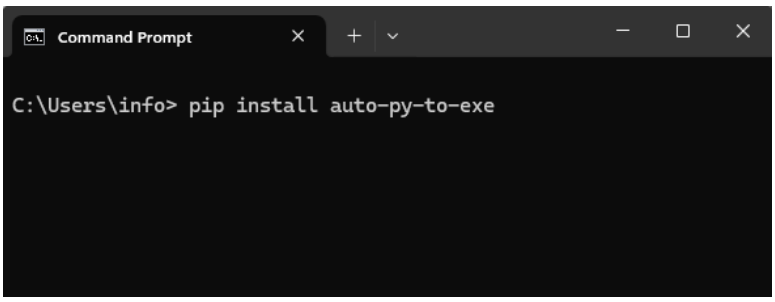

Fig. 2. Installing `auto-py-to-exe`

By default, `auto-py-to-exe` will be installed into the `Scripts` folder.

It is recommended that you create a folder in a location of your choice - called, for example, `StandAlone` - and copy any files that you want to make executable into that folder. You should also copy any files that your program imports. For example, if you wanted to make an executable version of the `RectagleGUI` class from the lecture, then as well as `RectangleGUI.py`, you will also need the `Rectangle.py` module.

When you launch `auto-py-to-exe`, you will see a rather cumbersome graphic, and you will need to scroll up and down to see all the options. Figure 3 shows a portion of this graphic, which you will find near the beginning and which you can locate by scrolling down a little way.

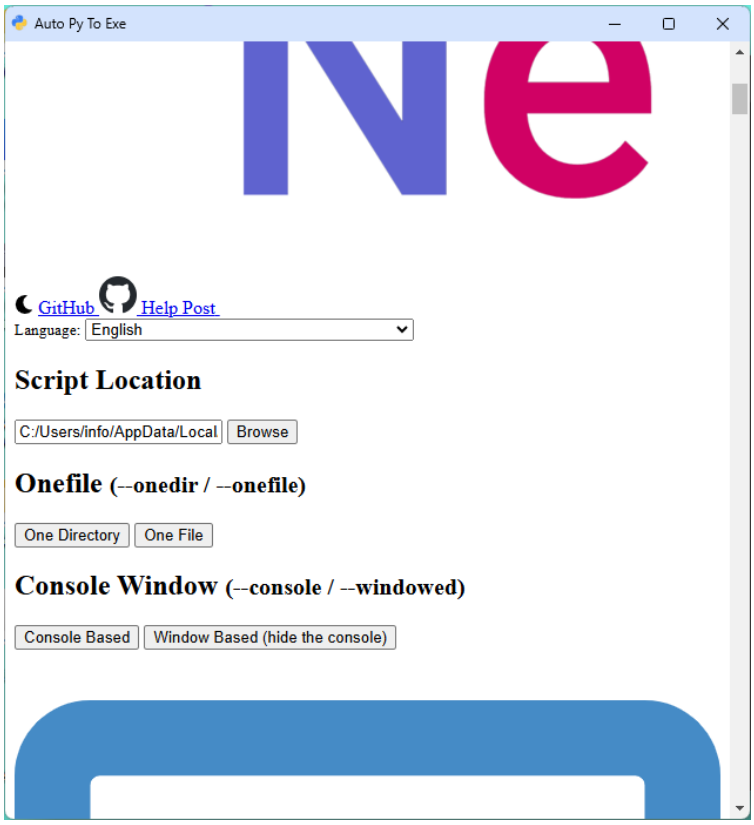

Fig. 3. Initial screen

The **Script Location** option allows you to browse to the file that you want to make executable. Once you have selected your file, you should select the next option - either **One Directory** or **One File**. Normally, you would choose **One File** so that you end up with a single executable file. You should then choose whether you want your application to be console based or window based. It is most likely that you will be creating a graphical program, in which case you should choose Window Based. If you were creating a console based program, you would choose the console based option[1].

Don't worry that your choices are not highlighted after clicking - they will have been recorded.

You can now scroll down to the bottom of the graphic (figure 4).

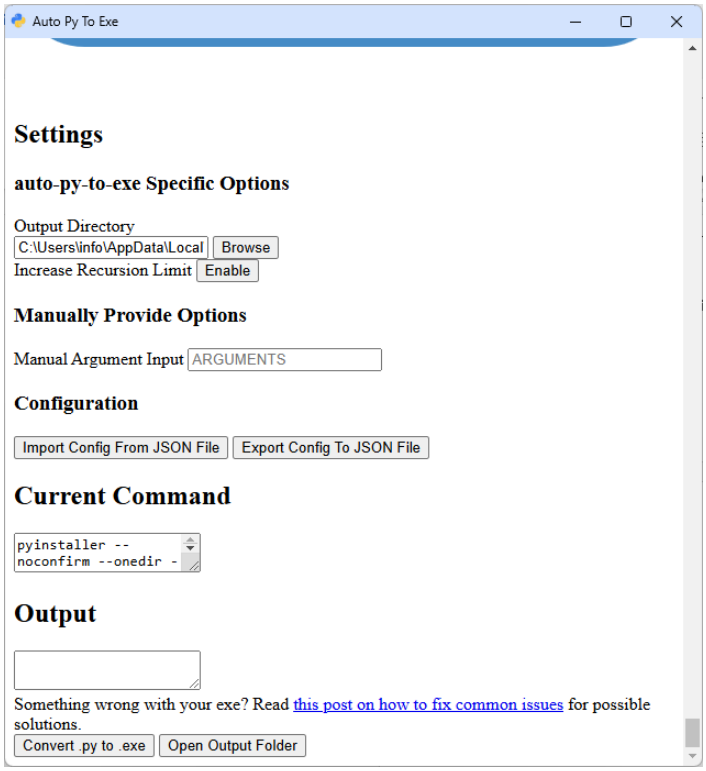

Fig. 4. Creating the executable file

In the **Settings** section[2] you can choose your output directory. Browse to the directory where you want the output file to be saved, and then press **Convert .py to exe**. Once the process has completed you can open the output folder - double clicking on the file will cause it to run.

The **Convert .py to exe** button will now have change to **Clear Output** - you can press this button and start again with a new file.

---

1 You should note that the console window will close once the program terminates, so if there is any text to be displayed before the program ends, you should add an extra line to your application that requires some input from the user - doing this will mean that the program can be terminated with a carriage return.

2 You might find that this section is collapsed, in which case you will need to click to expand it.