**TRANSFORMING SMART JULIE FROM CODE TO APPLICATION**


## Contextualization

Hello, welcome to our second stage with **Julie**, before I didn't explain the reasons that led me to choose this name, this is due to a tribute to a friend of the same name and as smart as her!

However, in this step you don't need to code practically anything, you just need attention to implement it properly and talk to **Julie** asking her by voice or written messages everything you want to know!

Point out that our friend **Julie** understands English and Spanish well written or well pronounced including phrasal verbs that is everything.

# WHAT DO I NEED TO TALK TO HER?

Well, I could even say that you don't need anything, but you do. The question is what if everything is already done, of course everything is done, but for him to hear you he must make sure that your computer's microphone is working correctly otherwise you can connect an external microphone that's all. Also remember that at this stage it can be run on windows or Linux except android.

**HOW CAN I SAY GOODBYE TO JULIE AND END OUR CONVERSATION?**

Obviously when we want to say goodbye to a friend there are specific expressions so it's no different with **Julie**, for that purpose we need to use one of the following expressions: *bye, I must go, see you later, see off, or end.*

**HOW TO IMPLEMENT THE PROJECT?**

Most of the necessary tools have already been described in the previous article, so we just added a few technologies described below.

## 1. Necessary tools

For this phase we need following tools:

- **Tkinter** the Tkinter library is based on the Tk toolkit, which is a cross-platform GUI library. Tk provides the necessary building blocks for creating a graphical interface, and Tkinter is an abstraction layer that allows the use of Tk in Python.
  Since Tkinter is part of the standard Python library, there is no need to install anything additional to use it. It is available on virtually all platforms where Python runs, including Windows, macOS, and Linux.
- **Threading** The "threading" library provides a simple API for working with threads. Developers can create threads using the "Thread" class and implement the thread's logic within a method. Additionally, the library provides mechanisms for synchronization and communication between threads, such as locks, semaphores, and condition variables, which help avoid concurrency issues like race conditions and simultaneous access to shared resources.

It is available on virtually all platforms where Python runs, including Windows, macOS, and Linux.

- **Auto-py-to-exe** Auto-py-to-exe is a Python utility that allows you to convert a Python script into a standalone executable file. It simplifies the process of converting your Python code into an executable format, making it easier to distribute and run on systems that don't have Python installed.

  Simplifies the usage of PyInstaller by providing a user-friendly interface and pre-configured settings. It abstracts away some of the complexities and technical details involved in the conversion process, making it accessible to users with varying levels of Python and packaging knowledge. To install run: ***pip install auto-py-to-exe*** on your terminal.

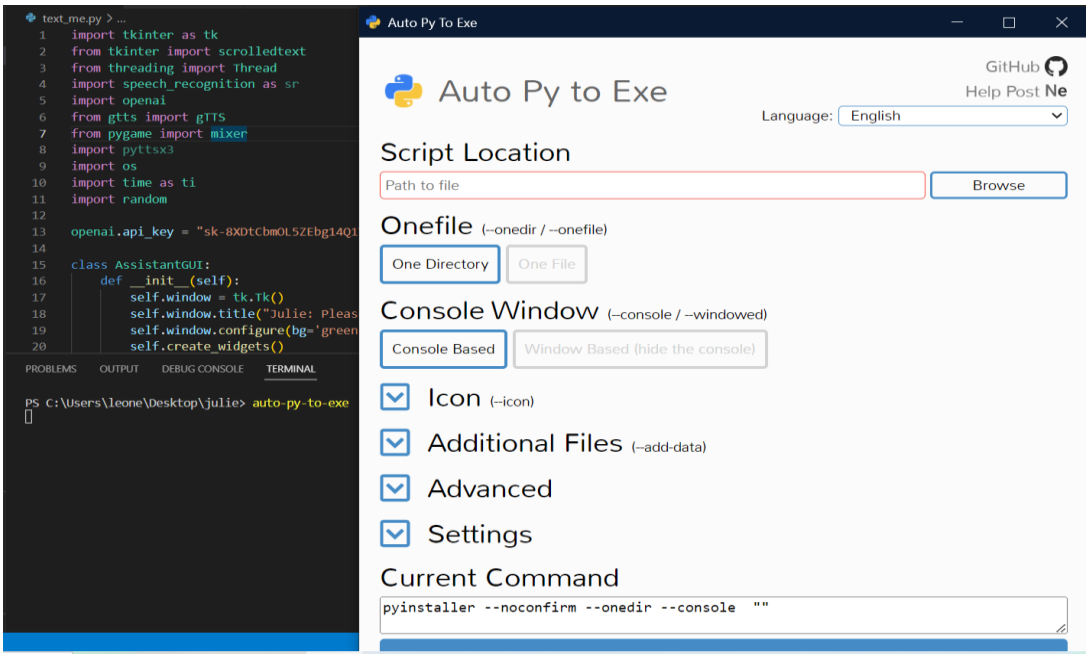
## CONVERTING CODE TO EXECUTABLE FILE

As we explained earlier, at this point we will be using the auto-py-to-exe library. as you can see in the figure below:

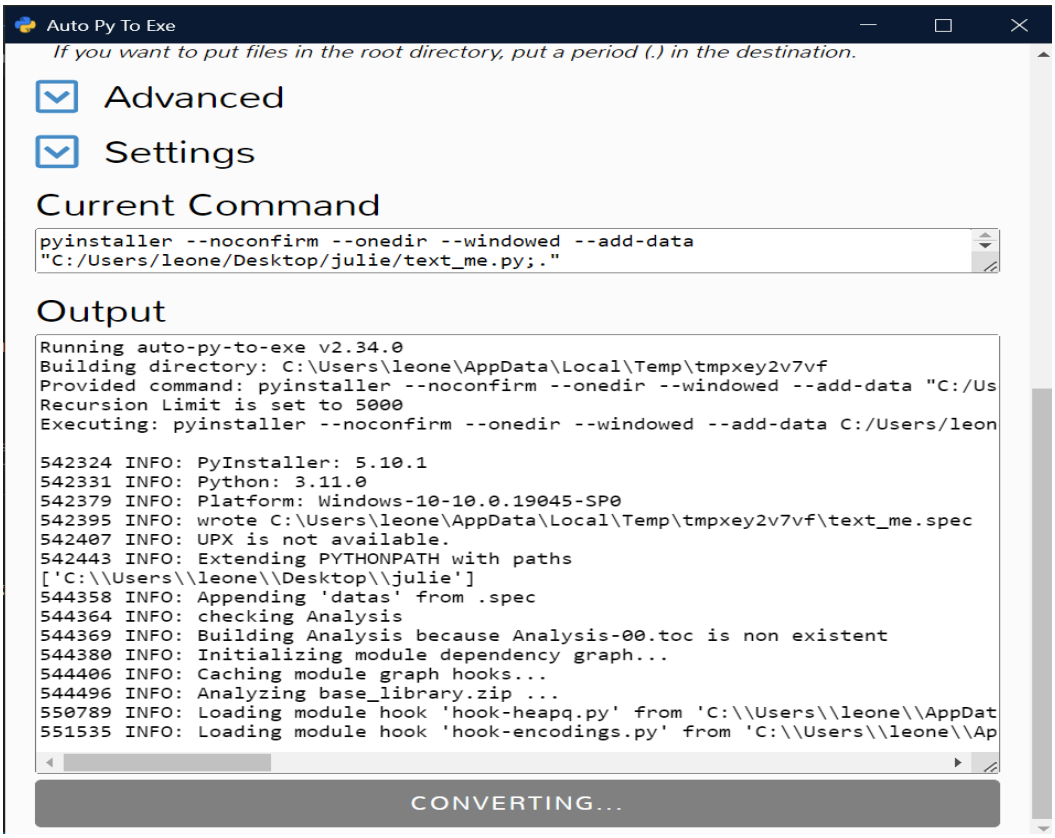
```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL                                                    powershell

PS C:\Users\leone\Desktop\julie> pip install --upgrade auto-py-to-exe
Requirement already satisfied: auto-py-to-exe in c:\users\leone\appdata\local\programs\python\python311\lib\site-packages (2.34.0)
Requirement already satisfied: Eel>=0.11.0 in c:\users\leone\appdata\local\programs\python\python311\lib\site-packages (from auto-py-to-exe) (0.16.0)
Requirement already satisfied: pyinstaller>=5.8.0 in c:\users\leone\appdata\local\programs\python\python311\lib\site-packages (from auto-py-to-exe) (5.10.1)
Requirement already satisfied: bottle in c:\users\leone\appdata\local\programs\python\python311\lib\site-packages (from Eel>=0.11.0->auto-py-to-exe) (0.12.25)
Requirement already satisfied: bottle-websocket in c:\users\leone\appdata\local\programs\python\python311\lib\site-packages (from Eel>=0.11.0->auto-py-to-exe) (0
.2.9)
Requirement already satisfied: future in c:\users\leone\appdata\local\programs\python\python311\lib\site-packages (from Eel>=0.11.0->auto-py-to-exe) (0.18.3)
Requirement already satisfied: pyparsing in c:\users\leone\appdata\local\programs\python\python311\lib\site-packages (from Eel>=0.11.0->auto-py-to-exe) (3.0.9)
Requirement already satisfied: whichcraft in c:\users\leone\appdata\local\programs\python\python311\lib\site-packages (from Eel>=0.11.0->auto-py-to-exe) (0.6.1)
Requirement already satisfied: setuptools>=42.0.0 in c:\users\leone\appdata\local\programs\python\python311\lib\site-packages (from pyinstaller>=5.8.0->auto-py-t
o-exe) (65.5.0)
Requirement already satisfied: altgraph in c:\users\leone\appdata\local\programs\python\python311\lib\site-packages (from pyinstaller>=5.8.0->auto-py-to-exe) (0.
17.3)
Requirement already satisfied: pyinstaller-hooks-contrib>=2021.4 in c:\users\leone\appdata\local\programs\python\python311\lib\site-packages (from pyinstaller>=5
.8.0->auto-py-to-exe) (2023.2)
Requirement already satisfied: pefile>=2022.5.30 in c:\users\leone\appdata\local\programs\python\python311\lib\site-packages (from pyinstaller>=5.8.0->auto-py-to
```

In my case I used upgrade because I already have it installed and I'm just updating it and in your case remove upgrade and run it.

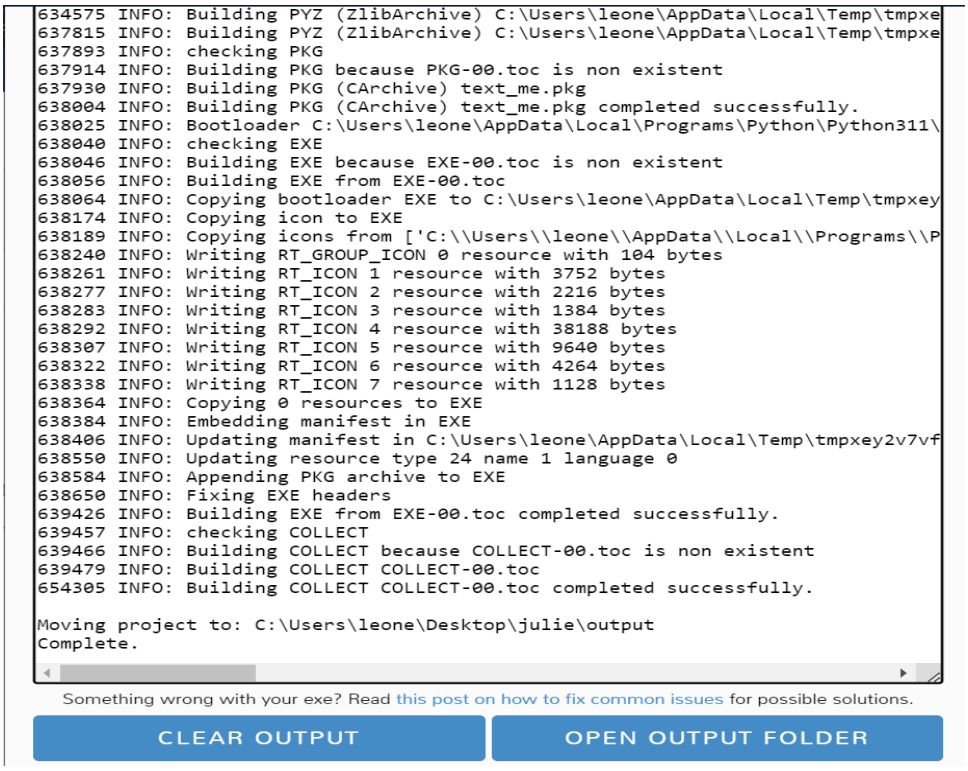
Then find full code here and run auto-to-py-exe to open the GUI window and load the file you want to convert. Like the figure below:




Once done, load your file, select the one file options for just one file and windows based for the graphical interface and in the additional file option insert another file. and click the convert py to exe option. As you can see the figure:



And finally locate your converted project in the previous project directory in an automatically generated folder with the name outpu or click the open output folder button. See it in the figure below:




## Final considerations

Finally, we can talk to Julie by simply downloading the files available here and running them on the operating systems. If you want to contribute to the progress of this project, contact me! Enjoy and visit us often for more python news.

**Keywords**

**Julie, convert, application, python, API, Virtual assistant**

## Reference List

- https://docs.python.org/3/library/tkinter.html
- https://docs.python.org/3/library/threading.html
- https://pypi.org/project/auto-py-to-exe/