How to use the Guitar Synth:

For basic usage:
- Download the code from https://github.com/ch98-1/Python_Guitar_Synth
- Open the "Guitar Synth.ipynb" in jupyter notebook
- Change the command file on line 9 to the file you want
- Run the script (in jupyter notebook) to generate the resulting audio files
    - result.wav will be the raw final results, and string.wav will be the output directly from string synthesis
    - Opus and mp3 format version of the file will be generated if ffmpeg is installed

For Changing Impulse Responses:
- Get the required impulse responses as 96kHz 32bit float mono wav file format
- For this, you could either acquire a impulse response from desired source and convert to the required format using software such as audacity or ffmpeg, or generate your own using the script provided in the IR generation folder. Impulse response file should be in the base directory.
    - For generating the IR for a system, the "Sine Sweep.wav" file in the IR generation folder should be played back at the highest volume that doesn't cause distortion into the system you want to measure the IR of.
    - Record the resulting sound in either mono for the instrument body's system and stereo for the room if stereo is desired.
    - For a stereo file, split in to 2 mono file using software such as audacity or ffmpeg, and process separately.
    - Open and edit the "Convoluter_Deconvoluter.ipynb" and change the name of the file inputs in "create_IR_wav" function call on line 40 of the 3rd cell
        - First file input will be the recorded sweep's file name (should be 96kHz 32bit float mono wav file)
        - Second file will be the original sine wave played back in to the system: the "Sine Sweep.wav" if it was not changed
        - Third file should be the output IR response file name
    - Run the first cell, then the 3rd cell of the Convoluter_Deconvoluter and the IR response file should be generated.
    - Put the impulse response file in the base directory

- Change Impulse Response files name to one you want in lines 322 to 324
    - The file on 322 will be the response for the instrument
    - The file on 323 should be the left channel of the room response
    - The file on 324 should be the right channel of the room response

For writing new input command file:
- The input command file is a csv with no space with specific format requirement to run.
- This is the file that should be changed to generate new music
- The command file works on 4 different commands: note, damp, pluck, and length.

- It should be formatted in such way that the first section will have note, damp, and pluck command, and the last line having the length command.
- Note command should be formatted as <time in seconds>,<string number>,note,<frequency in HZ>
    - This will change the note frequency playable on that string. Staying within reasonable limit for standard tuning on 24 fret guitar is highly recommended.
- Damp command should be formatted as <time in seconds>,<string number>,damp,<damping factor>
    - This sets the damping factor for that string. Setting the factor to 1 will play it like the string was plucked with no damping on a normal guitar, and 0 would almost immediately dampen the string down to 0. Value somewhere near 0.95 sounds mostly natural.
- Pluck command should be formatted as <time in seconds>,<string number>,pluck,<strength>
    - This plucks the string at the strength specified. Using 1 as the full volume and 0 as the 0 volume is recommended.
- Length command at the end should be formatted as length,<length in seconds>
    - This sets the total length of the music. This should be around 1 second longer than required with the last second being silence or near silence for the provided impulse response and at least as long as the impulse response if other impulse response is used.
- Each command should be in a seperate line, but there should not be any empty lines
- No line should have space in it
- Command file has to be at least as long as the impulse response or the code may break.
- Input.csv as well as test<string number>.csv is provided as and example