

# Demo 5 Exercises: Play a Wave file

DSP Lab (ECE 4163 / ECE 6183)

Fall 2018

Demo files:

```
play_wave_mono.py  
play_wave_stereo.py  
demo_sys.py
```

The demo python program `play_wave_mono.py` shows how to read and play a wave file using PyAudio. It is assumed the wave file is mono (single channel). The program `play_wave_stereo` plays a stereo (two channel) wave file.

In this demo, we use a `while` loop to write signal values to the audio output. Inside the loop, we use `unpack()` and `pack()` to convert between binary strings and lists of integers. In the `while` loop, we use the variable `gain` to amplify the signal and the function `clip16()` to keep the value within the range of a signed 16-bit integer to avoid potential overflow run-time errors. Correspondingly the format string `h` (`hh` for stereo) is used in the `unpack()` and `pack()` methods to set the encoding format.

Documentation for the `wave` module is at

<https://docs.python.org/3/library/wave.html>

## Exercises

1. **Single program for mono and stereo.** Write a single Python program to play both mono and stereo wave files. The program should determine the number of channels by reading the wave file information. SUBMIT

Verify that your program can play both mono and stereo wave files encoded with 16-bits per sample.

2. Modify your previous program so it can be used at the command line like

```
>> python my_play_wave.py filename.wav
```

You will need to import the `sys` module.

For example, consider the Python demo program `demo_sys.py`

```
1 | # demo_sys.py
2 |
3 | import sys
4 |
5 | for i in range(len(sys.argv)):
6 |     print('Argument %d is %s ' % ( i, sys.argv[i] ) )
```

We run this at the terminal command line:

```
>> python demo_sys.py abc.wav 10 h20
Argument 0 is demo_sys.py
Argument 1 is abc.wav
Argument 2 is 10
Argument 3 is h20
```