Play a WAVE file '''

import pyaudio

import wave

filename = 'path-to\_file.wav'

# Set chunk size of 1024 samples per data frame

chunk = 1024

# Open the soaudio/sound file

af = wave.open(filename, 'rb')

# Create an interface to PortAudio

pa = pyaudio.PyAudio()

# Open a .Stream object to write the WAV file

# 'output = True' indicates that the

# sound will be played rather than

# recorded and opposite can be used for recording

stream = pa.open(format = pa.get\_format\_from\_width(af.getsampwidth()),

                channels = af.getnchannels(),

                rate = af.getframerate(),

                output = True)

# Read data in chunks

rd\_data = af.readframes(chunk)

# Play the sound by writing the audio

# data to the Stream using while loop

while rd\_data != '':

    stream.write(rd\_data)

    rd\_data = af.readframes(chunk)

# Close and terminate the stream

stream.stop\_stream()

stream.close()

pa.terminate()

<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

import pyaudio

import wave

# Record in chunks of 1024 samples

chunk = 1024

# 16 bits per sample

sample\_format = pyaudio.paInt16

chanels = 2

# Record at 44400 samples per second

smpl\_rt = 44400

seconds = 4

filename = "path\_of\_file.wav"

# Create an interface to PortAudio

pa = pyaudio.PyAudio()

stream = pa.open(format=sample\_format, channels=chanels,

                 rate=smpl\_rt, input=True,

                 frames\_per\_buffer=chunk)

print('Recording...')

# Initialize array that be used for storing frames

frames = []

# Store data in chunks for 8 seconds

for i in range(0, int(smpl\_rt / chunk \* seconds)):

    data = stream.read(chunk)

    frames.append(data)

# Stop and close the stream

stream.stop\_stream()

stream.close()

# Terminate - PortAudio interface

pa.terminate()

print('Done !!! ')

# Save the recorded data in a .wav format

sf = wave.open(filename, 'wb')

sf.setnchannels(chanels)

sf.setsampwidth(pa.get\_sample\_size(sample\_format))

sf.setframerate(smpl\_rt)

sf.writeframes(b''.join(frames))

sf.close()

<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

**Recording Audio**

The [python-sounddevice](https://python-sounddevice.readthedocs.io/en/latest/) and [pyaudio](https://people.csail.mit.edu/hubert/pyaudio/) libraries provide ways to record audio with Python. python-sounddevice records to NumPy arrays and pyaudio records to bytes objects. Both of these can be stored as WAV files using the scipy and wave libraries, respectively.

python-sounddevice

[python-sounddevice](https://python-sounddevice.readthedocs.io/en/latest/) allows you to record audio from your microphone and store it as a NumPy array. This is a handy datatype for sound processing that can be converted to WAV format for storage using the [scipy.io.wavfile](https://docs.scipy.org/doc/scipy/reference/generated/scipy.io.wavfile.write.html) module. Make sure to install the scipy module for the following example (pip install scipy). This automatically installs NumPy as one of its dependencies:

import sounddevice as sd

from scipy.io.wavfile import write

fs = 44100 # Sample rate

seconds = 3 # Duration of recording

myrecording = sd.rec(int(seconds \* fs), samplerate=fs, channels=2)

sd.wait() # Wait until recording is finished

write('output.wav', fs, myrecording) # Save as WAV file

pyaudio

[Earlier in this article](https://realpython.com/playing-and-recording-sound-python/#pyaudio), you learned how to play sounds by reading a [pyaudio.Stream()](https://people.csail.mit.edu/hubert/pyaudio/docs/#class-stream). Recording audio can be done by writing to this stream instead:

import pyaudio

import wave

chunk = 1024 # Record in chunks of 1024 samples

sample\_format = pyaudio.paInt16 # 16 bits per sample

channels = 2

fs = 44100 # Record at 44100 samples per second

seconds = 3

filename = "output.wav"

p = pyaudio.PyAudio() # Create an interface to PortAudio

print('Recording')

stream = p.open(format=sample\_format,

channels=channels,

rate=fs,

frames\_per\_buffer=chunk,

input=True)

frames = [] # Initialize array to store frames

# Store data in chunks for 3 seconds

for i in range(0, int(fs / chunk \* seconds)):

data = stream.read(chunk)

frames.append(data)

# Stop and close the stream

stream.stop\_stream()

stream.close()

# Terminate the PortAudio interface

p.terminate()

print('Finished recording')

# Save the recorded data as a WAV file

wf = wave.open(filename, 'wb')

wf.setnchannels(channels)

wf.setsampwidth(p.get\_sample\_size(sample\_format))

wf.setframerate(fs)

wf.writeframes(b''.join(frames))

wf.close()