BỘ CÔNG THƯƠNG

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP HỒ CHÍ MINH

Khoa: Công Nghệ Thông Tin

LAB REPORT 02

Student's ID	:	
Student's name	:	Hồ Phúc Lâm
Subject	:	PTHTDPT
Instructor	:	Nguyễn Thành Thái
Faculty	:	Công Nghệ Thông Tin
Completed Date	:	28/08/2024

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....

Điểm lab report:

LAB 2.1 AUDIO PROCESSING

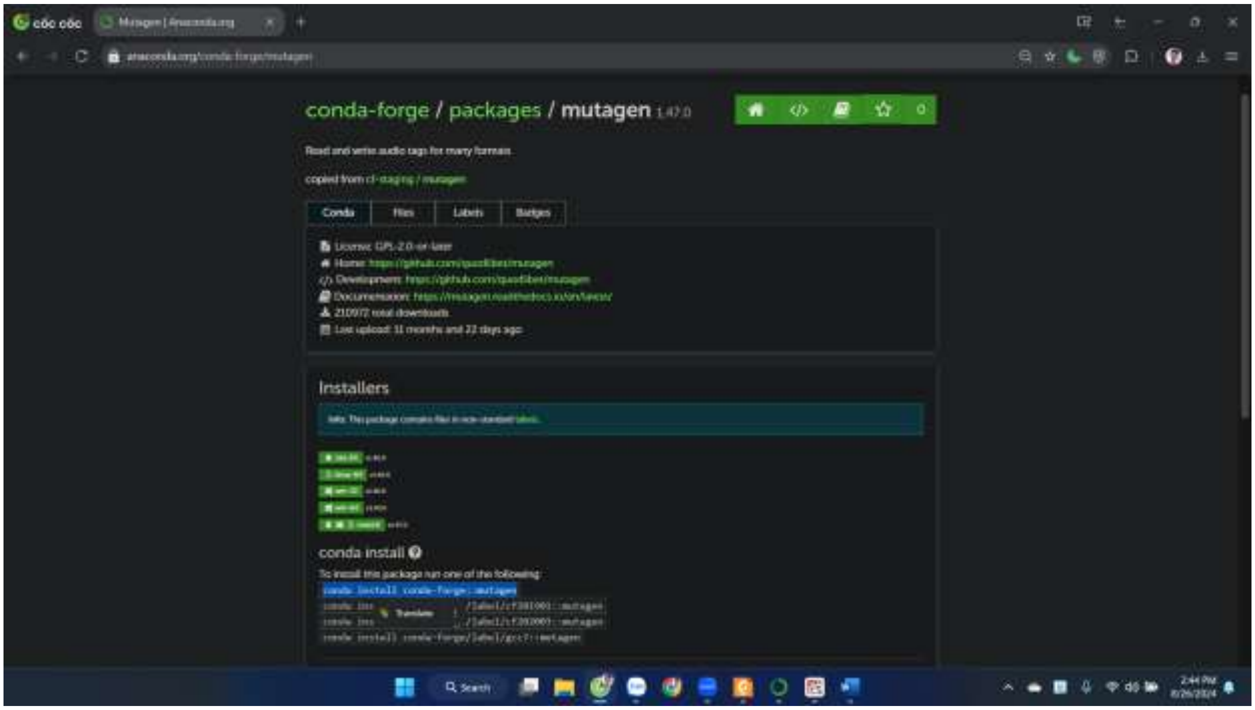
I. Thư viện mugtage.mp3

a) Cài thư viện mugtage

- Mở anaconda và base multimedia và mở cmd pompt

- Search từ khóa "how to install mutagen.mp3 on anaconda"

https://anaconda.org/conda-forge/mutagen



conda-forge / packages / mutagen 1.47.0

Read and write audio tags for many formats

copied from cf-staging / mutagen

Conda | Files | Labels | Badges

License: GPL-2.0-or-later
Home: https://github.com/quodlibet/mutagen
Development: https://github.com/quodlibet/mutagen
Documentation: https://mutagen.readthedocs.io/en/latest/
210972 total downloads
Last upload: 11 months and 22 days ago

Installers

Info: This package contains files in non-standard labels.

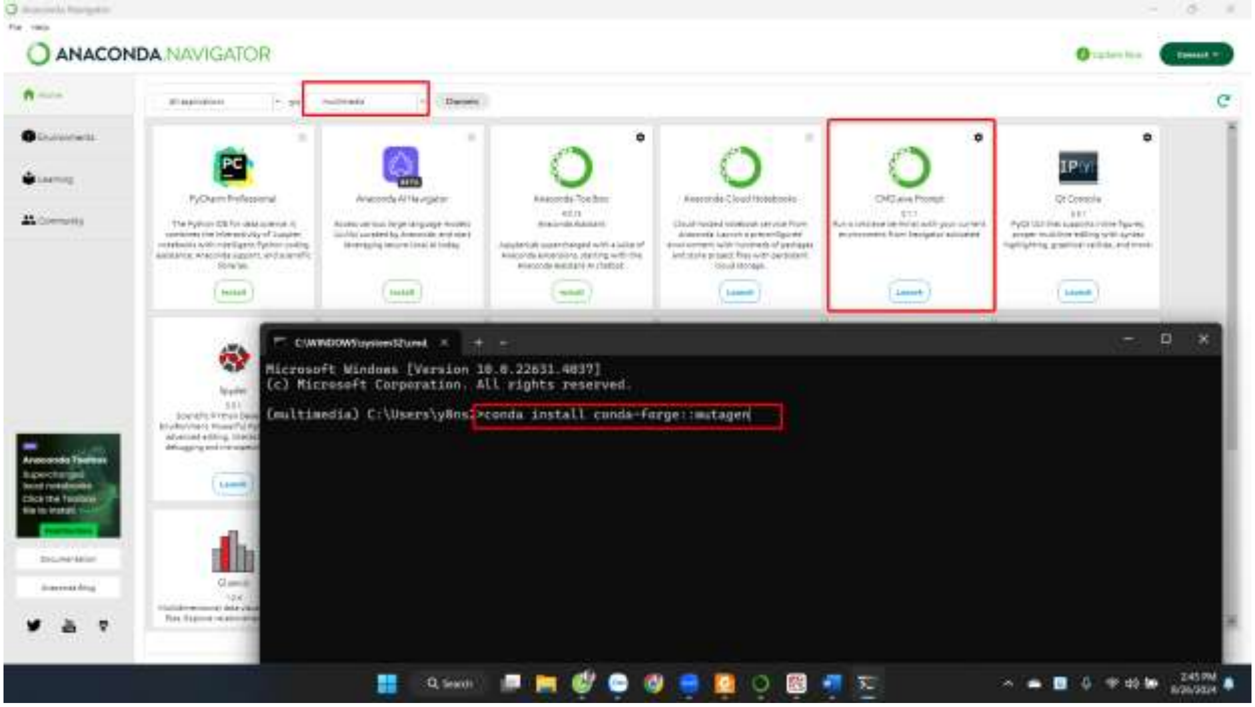
conda install

To install this package run one of the following:

```
conda install conda-forge::mutagen
conda ins                      /label/cf201901::mutagen
conda ins                      /label/cf202003::mutagen
conda install conda-forge/label/gcc7::mutagen
```

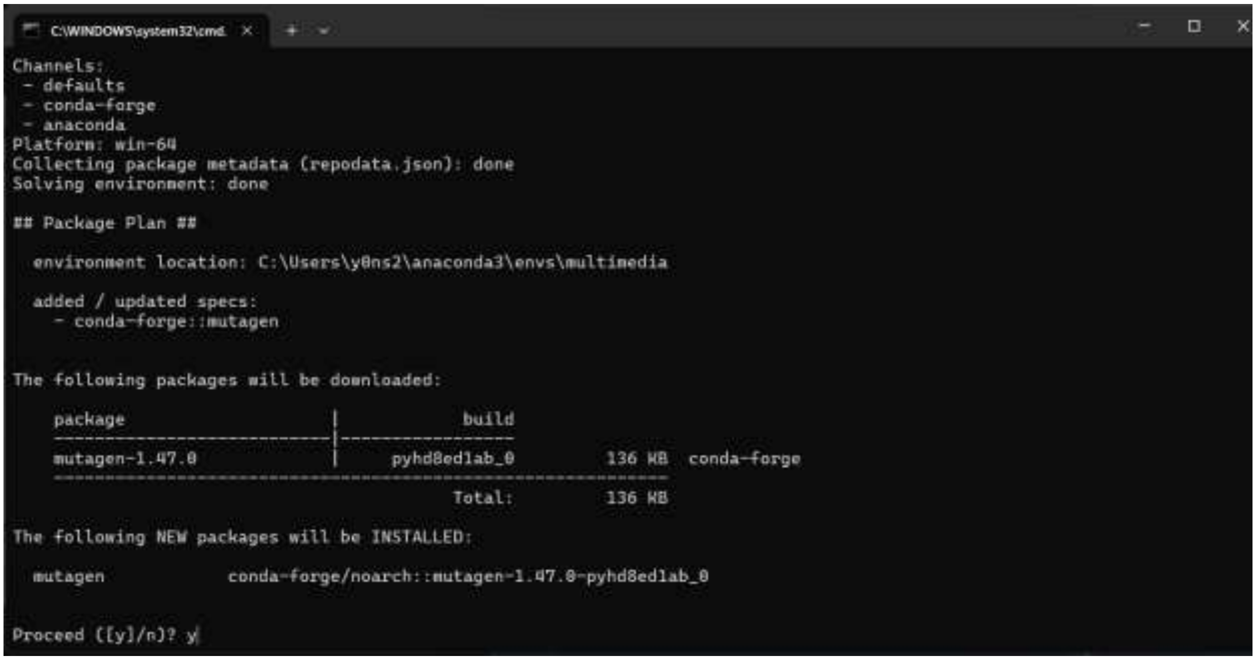
nhập lệnh:

conda install conda-forge::mutagen



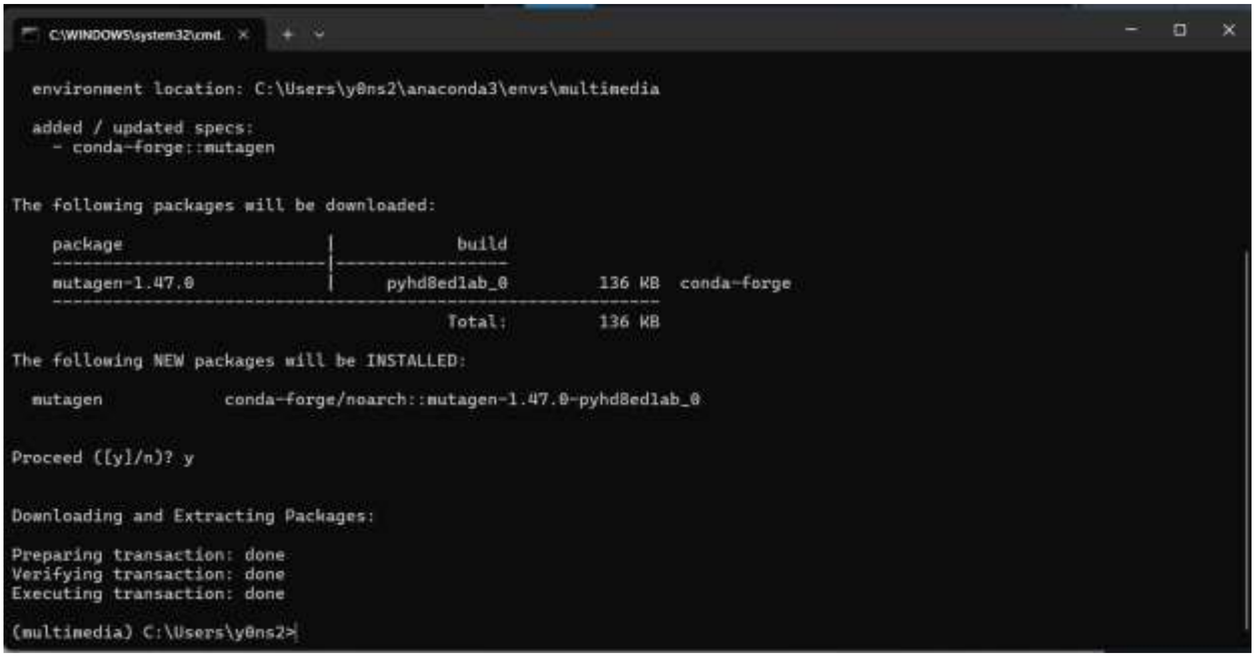
```
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

(multimedia) C:\Users\y0ns2>conda install conda-forge::mutagen
```

- Chọn y (yes) để tiếp tục



```
Channels:
 - defaults
 - conda-forge
 - anaconda
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\y0ns2\anaconda3\envs\multimedia

  added / updated specs:
    - conda-forge::mutagen


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    mutagen-1.47.0             |     pyhd8ed1ab_0         136 KB  conda-forge
    ------------------------------------------------------------
                                           Total:         136 KB

The following NEW packages will be INSTALLED:

  mutagen            conda-forge/noarch::mutagen-1.47.0-pyhd8ed1ab_0


Proceed ([y]/n)? y
```

Kết quả: đã cài đặt được thư viện mutagen



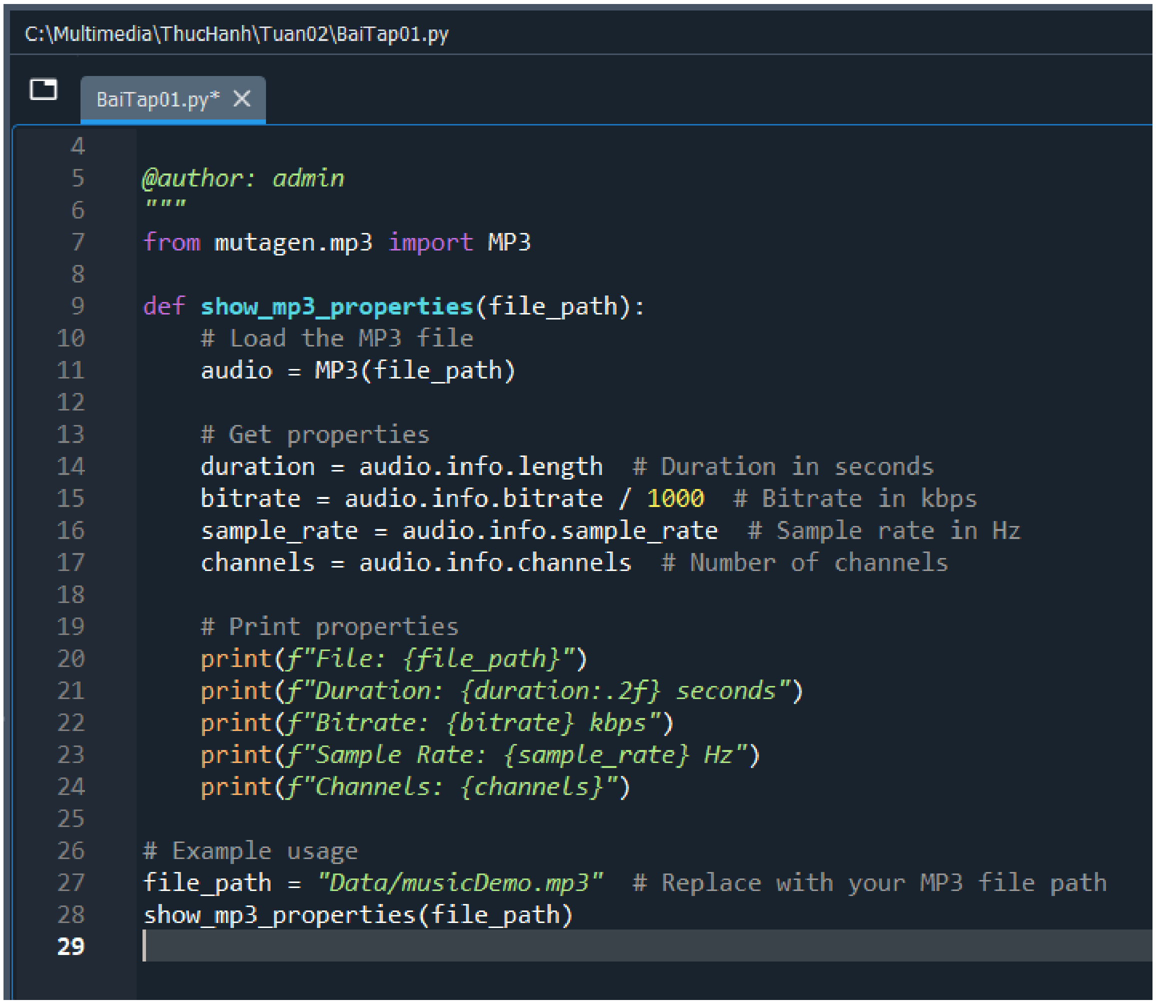
```
  environment location: C:\Users\y0ns2\anaconda3\envs\multimedia

  added / updated specs:
    - conda-forge::mutagen


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    mutagen-1.47.0             |     pyhd8ed1ab_0         136 KB  conda-forge
    ------------------------------------------------------------
                                           Total:         136 KB

The following NEW packages will be INSTALLED:

  mutagen            conda-forge/noarch::mutagen-1.47.0-pyhd8ed1ab_0


Proceed ([y]/n)? y


Downloading and Extracting Packages:

Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(multimedia) C:\Users\y0ns2>
```

b) Thực hiện lệnh code python sau:

```
from mutagen.mp3 import MP3

def show_mp3_properties(file_path):
    # Load the MP3 file
    audio = MP3(file_path)

    # Get properties
    duration = audio.info.length  # Duration in seconds
    bitrate = audio.info.bitrate / 1000  # Bitrate in kbps
    sample_rate = audio.info.sample_rate  # Sample rate in Hz
    channels = audio.info.channels  # Number of channels

    # Print properties
    print(f"File: {file_path}")
    print(f"Duration: {duration:.2f} seconds")
    print(f"Bitrate: {bitrate} kbps")
    print(f"Sample Rate: {sample_rate} Hz")
    print(f"Channels: {channels}")

# Example usage
file_path = "example.mp3"  # Replace with your MP3 file path
show_mp3_properties(file_path)
```

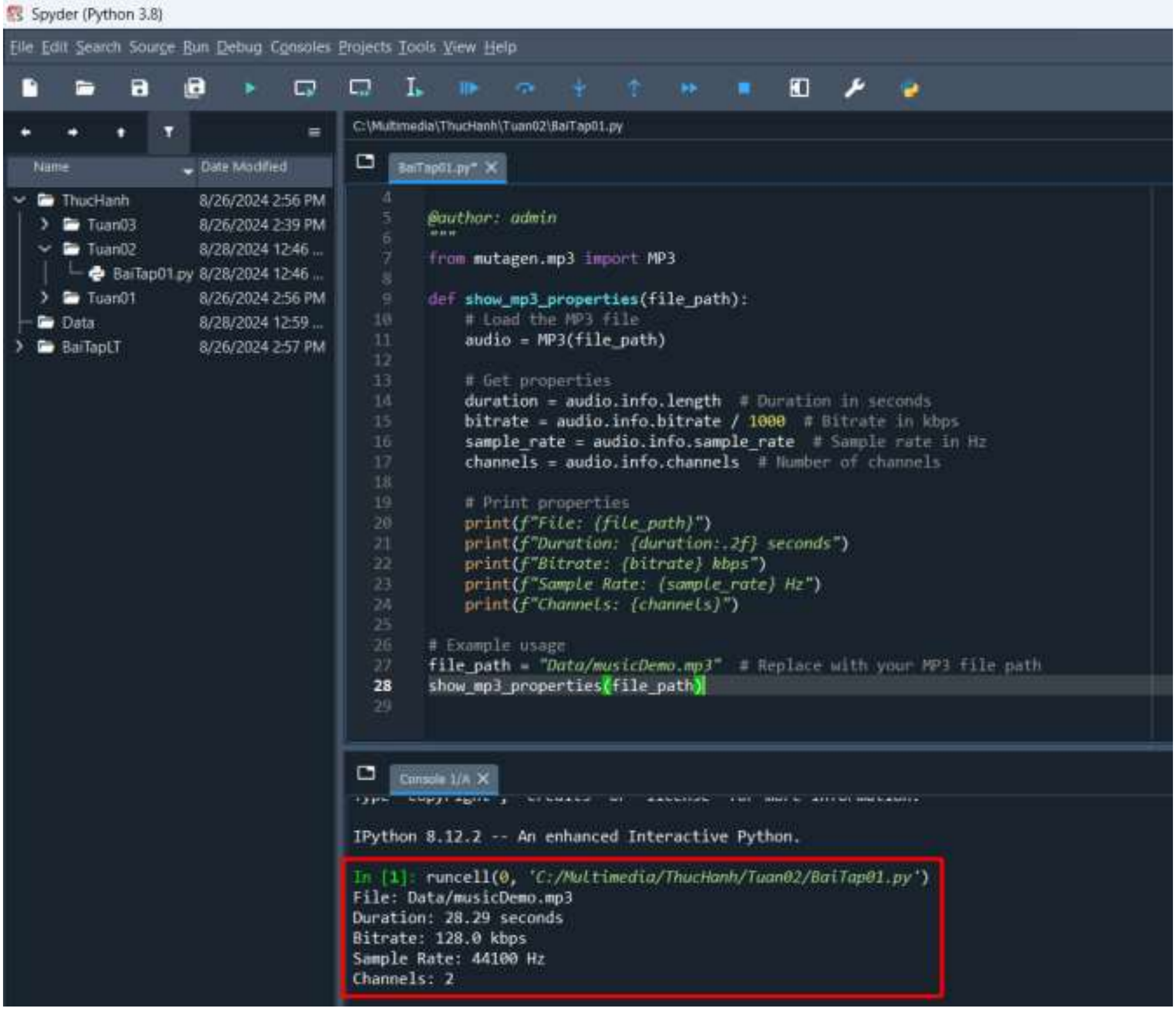
- Tải xuống 1 file .mp3 có độ dài khoảng 30s và cho vào Data của thư mục multimedia. Sau đó lấy đường dẫn Copy Relative Path rồi cho vào phần demo example usage:

file_path = "Data/musicDemo.mp3" # Replace with your MP3 file path



```

@author: admin
"""
from mutagen.mp3 import MP3

def show_mp3_properties(file_path):
    # Load the MP3 file
    audio = MP3(file_path)

    # Get properties
    duration = audio.info.length  # Duration in seconds
    bitrate = audio.info.bitrate / 1000  # Bitrate in kbps
    sample_rate = audio.info.sample_rate  # Sample rate in Hz
    channels = audio.info.channels  # Number of channels

    # Print properties
    print(f"File: {file_path}")
    print(f"Duration: {duration:.2f} seconds")
    print(f"Bitrate: {bitrate} kbps")
    print(f"Sample Rate: {sample_rate} Hz")
    print(f"Channels: {channels}")

# Example usage
file_path = "Data/musicDemo.mp3"  # Replace with your MP3 file path
show_mp3_properties(file_path)

```

Kết quả:


Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
Name
Date Modified
ThucHanh 8/26/2024 2:56 PM
Tuan03 8/26/2024 2:39 PM
Tuan02 8/28/2024 12:46 ...
BaiTap01.py 8/28/2024 12:46 ...
Tuan01 8/26/2024 2:56 PM
Data 8/28/2024 12:59 ...
BaiTapLT 8/26/2024 2:57 PM
C:\Multimedia\ThucHanh\Tuan02\BaiTap01.py
BaiTap01.py*
Console 1/A

```python

@author: admin
"""
from mutagen.mp3 import MP3

def show_mp3_properties(file_path):
    # Load the MP3 file
    audio = MP3(file_path)

    # Get properties
    duration = audio.info.length  # Duration in seconds
    bitrate = audio.info.bitrate / 1000  # Bitrate in kbps
    sample_rate = audio.info.sample_rate  # Sample rate in Hz
    channels = audio.info.channels  # Number of channels

    # Print properties
    print(f"File: {file_path}")
    print(f"Duration: {duration:.2f} seconds")
    print(f"Bitrate: {bitrate} kbps")
    print(f"Sample Rate: {sample_rate} Hz")
    print(f"Channels: {channels}")

# Example usage
file_path = "Data/musicDemo.mp3"  # Replace with your MP3 file path
show_mp3_properties(file_path)

```

```
IPython 8.12.2 -- An enhanced Interactive Python.

In [1]: runcell(0, 'C:/Multimedia/ThucHanh/Tuan02/BaiTap01.py')
File: Data/musicDemo.mp3
Duration: 28.29 seconds
Bitrate: 128.0 kbps
Sample Rate: 44100 Hz
Channels: 2
```

II. Thực hành bài Lab 2.1

1)Mục đích yêu cầu :

+Củng cố kiến thức về SIGNAL, AUDIO, sampling, sampling-rate, bit-rate,…

+Tiếp cận thư viện xử lý tín hiệu Audio, các API hỗ trợ,…

Uncompressed Audio Formats	
WAV (Waveform Audio File Format):	Commonly used on Windows systems.
AIFF (Audio Interchange File Format):	Developed by Apple for Mac systems.
PCM (Pulse-Code Modulation):	Raw digital audio format used in CDs and DVDs.

Lossless Compressed Audio Formats	
FLAC (Free Lossless Audio Codec):	Popular for high-quality audio storage.
ALAC (Apple Lossless Audio Codec):	Apple’s lossless format.
WAVPACK:	Another lossless format that supports hybrid compression.

Lossy Compressed Audio Formats	
MP3 (MPEG Audio Layer III):	Widely used for music files.
AAC (Advanced Audio Coding):	Commonly used in Apple devices and streaming.
OGG (Vorbis):	Open-source alternative to MP3.

2)Tài liệu tham khảo

a)https://realpython.com/playing-and-recording-sound-python/#playsound

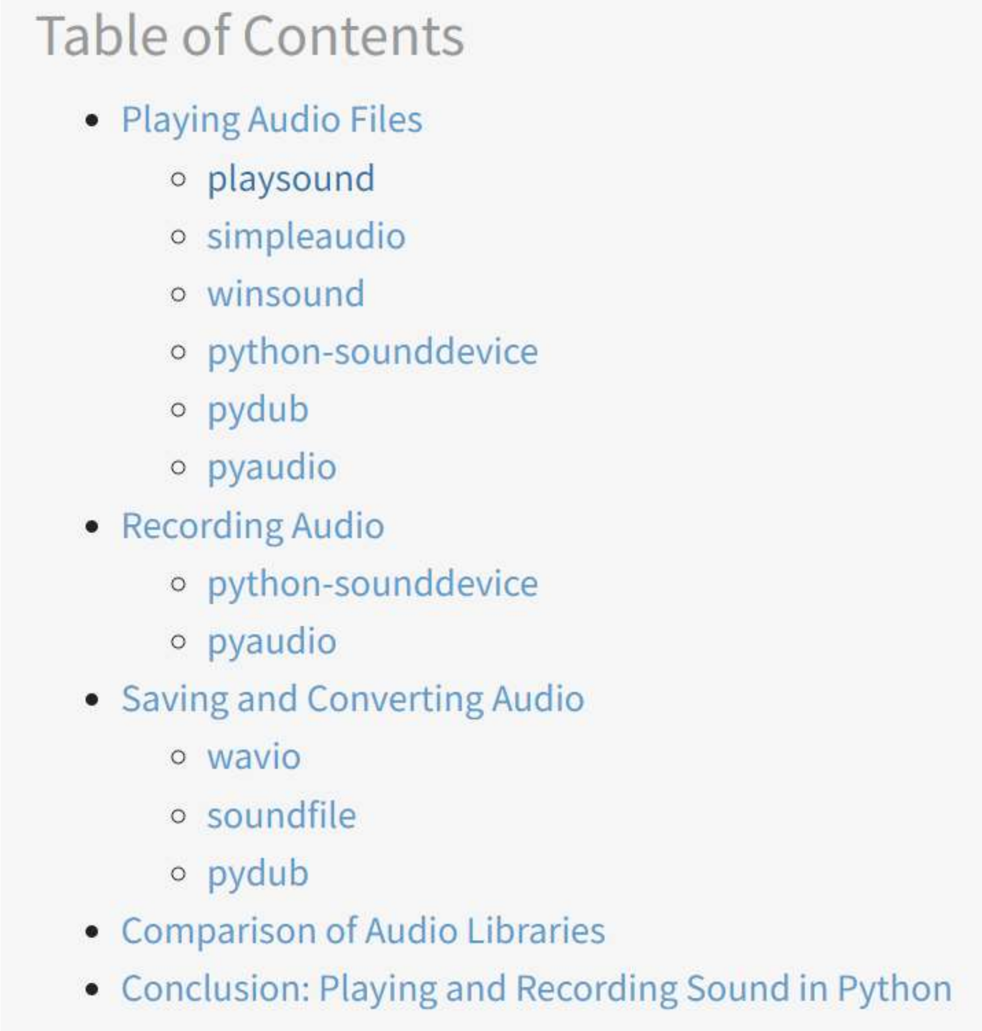
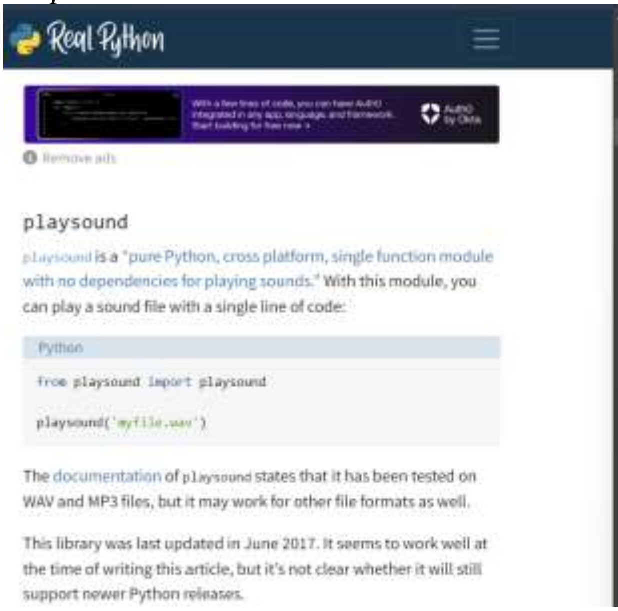


Table of Contents

- Playing Audio Files
  - playsound
  - simpleaudio
  - winsound
  - python-sounddevice
  - pydub
  - pyaudio
- Recording Audio
  - python-sounddevice
  - pyaudio
- Saving and Converting Audio
  - wavio
  - soundfile
  - pydub
- Comparison of Audio Libraries
- Conclusion: Playing and Recording Sound in Python

Phát tập tin âm thanh

- *phát âm thanh*



With a few lines of code, you can have Auth0 integrated in any app, language, and framework. Start building for free now >
Auth0 by Okta
Remove ads

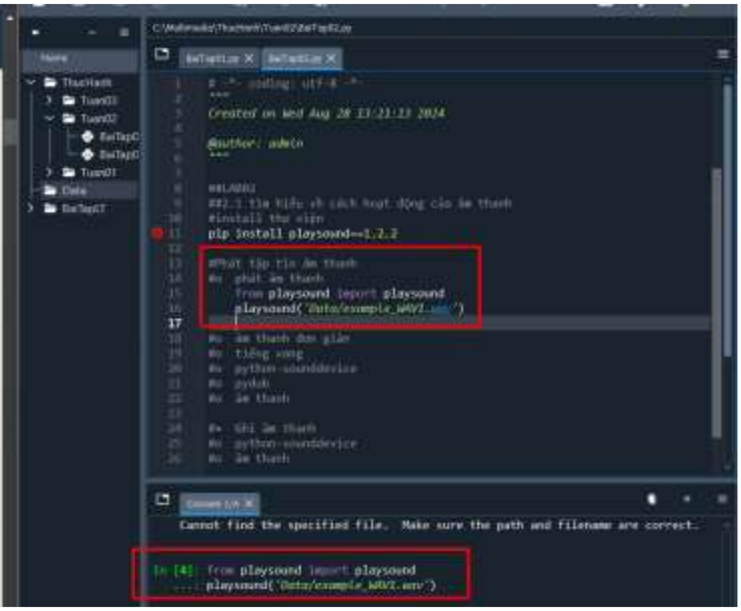
playsound

playsound is a "pure Python, cross platform, single function module with no dependencies for playing sounds." With this module, you can play a sound file with a single line of code:

```python
from playsound import playsound

playsound('myfile.wav')
```

The documentation of playsound states that it has been tested on WAV and MP3 files, but it may work for other file formats as well.

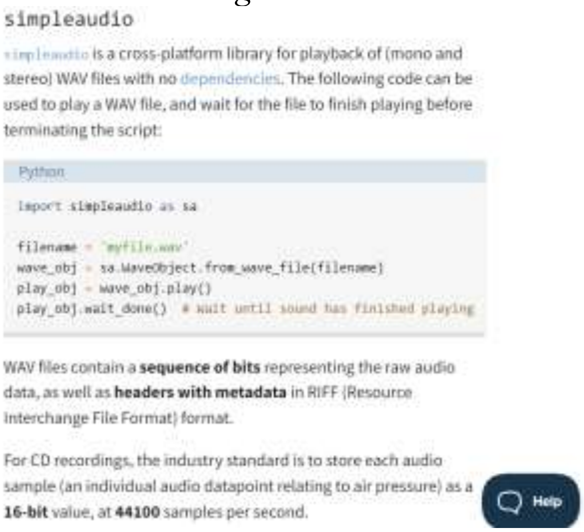
This library was last updated in June 2017. It seems to work well at the time of writing this article, but it's not clear whether it will still support newer Python releases.



```
# -*- coding: utf-8 -*-
"""
Created on Wed Aug 28 13:21:13 2024

@author: admin
"""

##LAB02
##2.1 tìm hiểu về cách hoạt động của âm thanh
#install thư viện
pip install playsound==1.2.2

#Phát tập tin âm thanh
#o  phát âm thanh
    from playsound import playsound
    playsound('Data/example_WAV1.wav')

#o  âm thanh đơn giản
#o  tiếng vang
#o  python-sounddevice
#o  pydub
#o  âm thanh

#+  Ghi âm thanh
#o  python-sounddevice
#o  âm thanh
```

```
Cannot find the specified file.  Make sure the path and filename are correct.

In [4]: from playsound import playsound
   ...: playsound('Data/example_WAV1.wav')
```

- *âm thanh đơn giản*



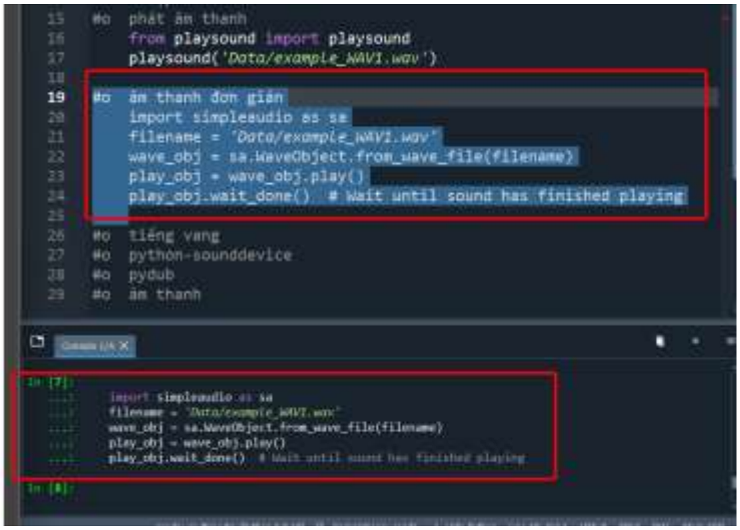
simpleaudio

simpleaudio is a cross-platform library for playback of (mono and stereo) WAV files with no dependencies. The following code can be used to play a WAV file, and wait for the file to finish playing before terminating the script:

```python
import simpleaudio as sa

filename = 'myfile.wav'
wave_obj = sa.WaveObject.from_wave_file(filename)
play_obj = wave_obj.play()
play_obj.wait_done()  # Wait until sound has finished playing
```

WAV files contain a **sequence of bits** representing the raw audio data, as well as **headers with metadata** in RIFF (Resource Interchange File Format) format.

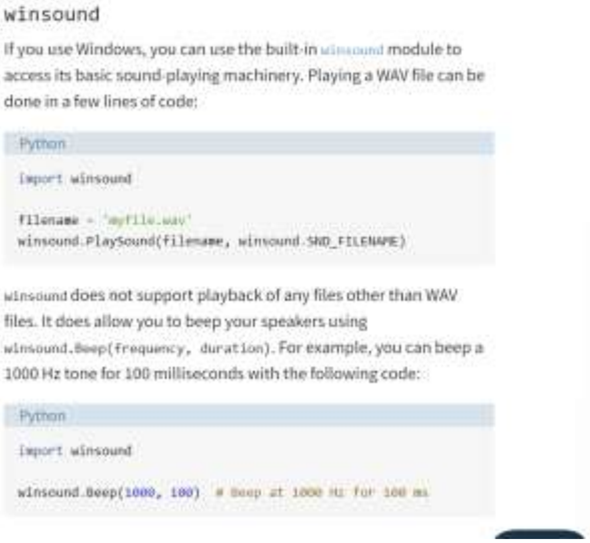
For CD recordings, the industry standard is to store each audio sample (an individual audio datapoint relating to air pressure) as a **16-bit** value, at **44100** samples per second.



```
#o  phát âm thanh
    from playsound import playsound
    playsound('Data/example_WAV1.wav')

#o  âm thanh đơn giản
    import simpleaudio as sa
    filename = 'Data/example_WAV1.wav'
    wave_obj = sa.WaveObject.from_wave_file(filename)
    play_obj = wave_obj.play()
    play_obj.wait_done()  # Wait until sound has finished playing

#o  tiếng vang
#o  python-sounddevice
#o  pydub
#o  âm thanh
```

```
In [7]:
   ...:     import simpleaudio as sa
   ...:     filename = 'Data/example_WAV1.wav'
   ...:     wave_obj = sa.WaveObject.from_wave_file(filename)
   ...:     play_obj = wave_obj.play()
   ...:     play_obj.wait_done()  # Wait until sound has finished playing

In [8]:
```

- *tiếng vang*



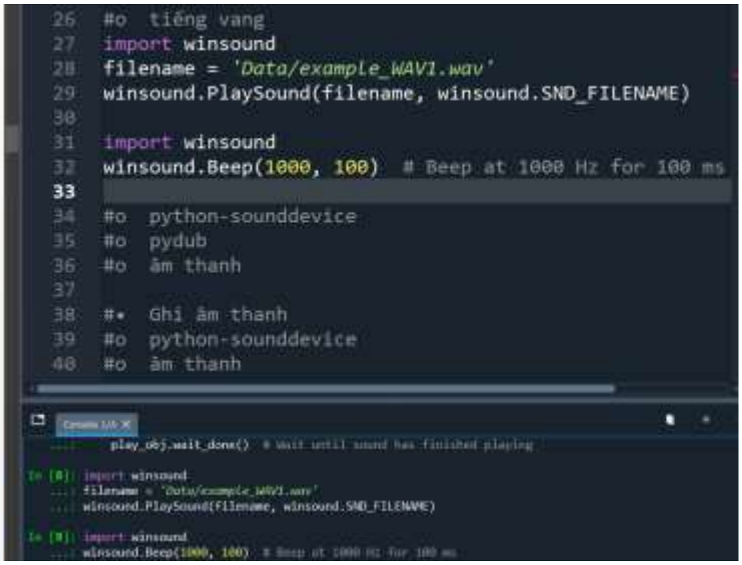
winsound

If you use Windows, you can use the built-in winsound module to access its basic sound-playing machinery. Playing a WAV file can be done in a few lines of code:

```python
import winsound

filename = 'myfile.wav'
winsound.PlaySound(filename, winsound.SND_FILENAME)
```

winsound does not support playback of any files other than WAV files. It does allow you to beep your speakers using winsound.Beep(frequency, duration). For example, you can beep a 1000 Hz tone for 100 milliseconds with the following code:

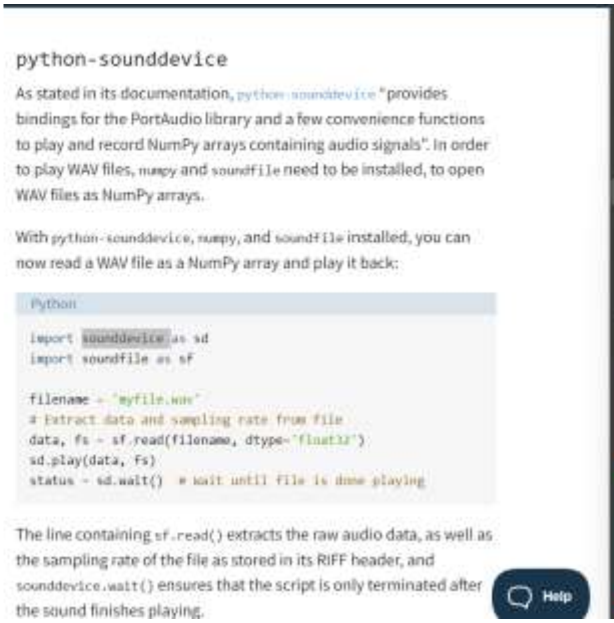
```python
import winsound

winsound.Beep(1000, 100)  # Beep at 1000 Hz for 100 ms
```



```python
#o  tiếng vang
import winsound
filename = 'Data/example_WAV1.wav'
winsound.PlaySound(filename, winsound.SND_FILENAME)

import winsound
winsound.Beep(1000, 100)  # Beep at 1000 Hz for 100 ms

#o  python-sounddevice
#o  pydub
#o  âm thanh

#•  Ghi âm thanh
#o  python-sounddevice
#o  âm thanh
```

```
   ...:     play_obj.wait_done()  # Wait until sound has finished playing

In [8]: import winsound
   ...: filename = 'Data/example_WAV1.wav'
   ...: winsound.PlaySound(filename, winsound.SND_FILENAME)

In [9]: import winsound
   ...: winsound.Beep(1000, 100)  # Beep at 1000 Hz for 100 ms
```

- *python-sounddevice*



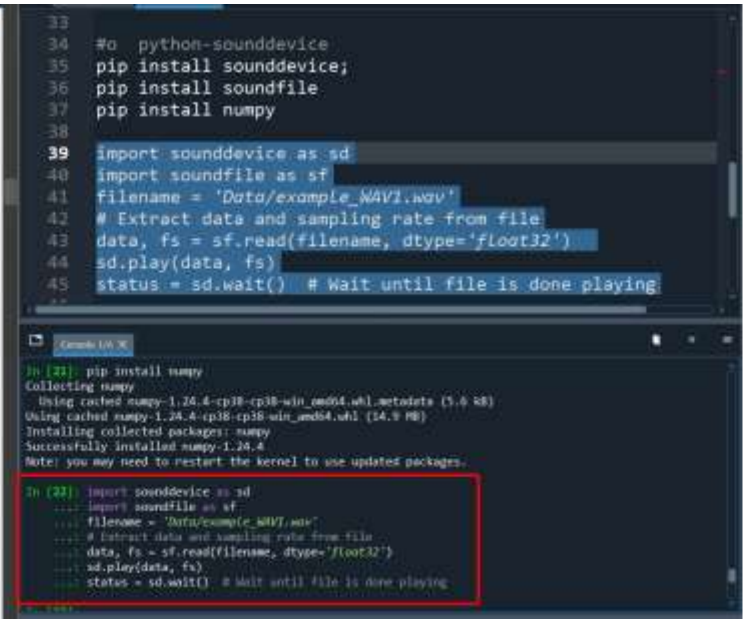
python-sounddevice

As stated in its documentation, python-sounddevice "provides bindings for the PortAudio library and a few convenience functions to play and record NumPy arrays containing audio signals". In order to play WAV files, numpy and soundfile need to be installed, to open WAV files as NumPy arrays.

With python-sounddevice, numpy, and soundfile installed, you can now read a WAV file as a NumPy array and play it back:

```python
import sounddevice as sd
import soundfile as sf

filename = 'myfile.wav'
# Extract data and sampling rate from file
data, fs = sf.read(filename, dtype='float32')
sd.play(data, fs)
status = sd.wait()  # Wait until file is done playing
```

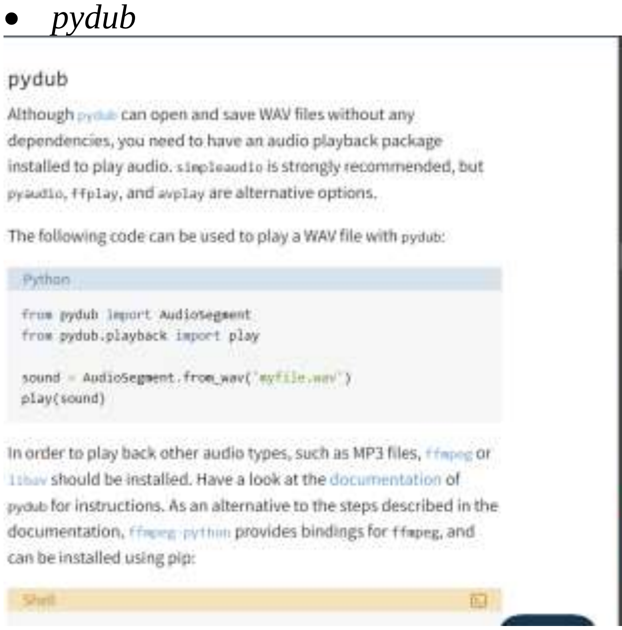
The line containing sf.read() extracts the raw audio data, as well as the sampling rate of the file as stored in its RIFF header, and sounddevice.wait() ensures that the script is only terminated after the sound finishes playing.



```python
#o  python-sounddevice
pip install sounddevice;
pip install soundfile
pip install numpy

import sounddevice as sd
import soundfile as sf
filename = 'Data/example_WAV1.wav'
# Extract data and sampling rate from file
data, fs = sf.read(filename, dtype='float32')
sd.play(data, fs)
status = sd.wait()  # Wait until file is done playing
```

```
In [21]: pip install numpy
Collecting numpy
  Using cached numpy-1.24.4-cp38-cp38-win_amd64.whl.metadata (5.6 kB)
Using cached numpy-1.24.4-cp38-cp38-win_amd64.whl (14.9 MB)
Installing collected packages: numpy
Successfully installed numpy-1.24.4
Note: you may need to restart the kernel to use updated packages.

In [22]: import sounddevice as sd
    ...: import soundfile as sf
    ...: filename = 'Data/example_WAV1.wav'
    ...: # Extract data and sampling rate from file
    ...: data, fs = sf.read(filename, dtype='float32')
    ...: sd.play(data, fs)
    ...: status = sd.wait()  # Wait until file is done playing
```

- *pydub*



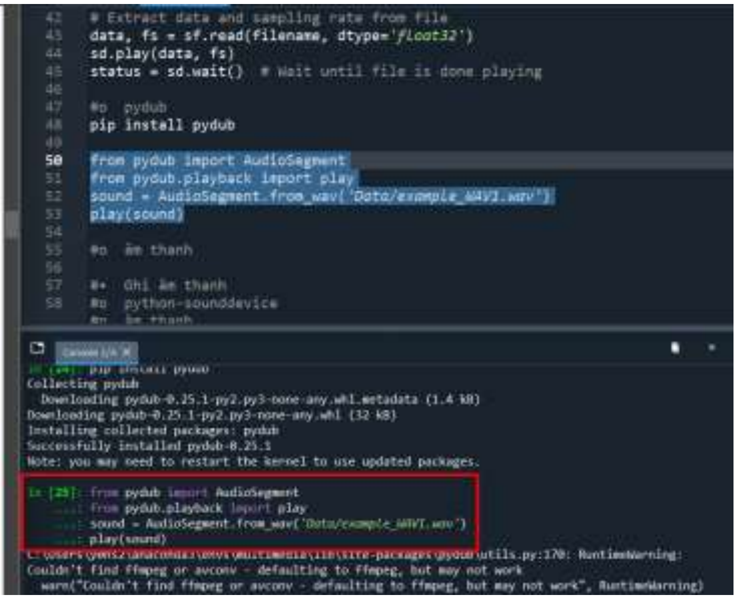
pydub

Although pydub can open and save WAV files without any dependencies, you need to have an audio playback package installed to play audio. simpleaudio is strongly recommended, but pyaudio, ffplay, and avplay are alternative options.

The following code can be used to play a WAV file with pydub:

```python
from pydub import AudioSegment
from pydub.playback import play

sound = AudioSegment.from_wav('myfile.wav')
play(sound)
```

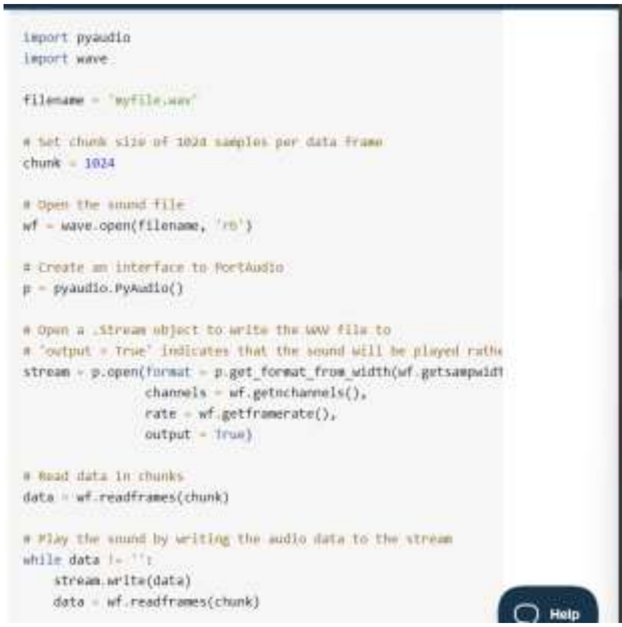
In order to play back other audio types, such as MP3 files, ffmpeg or libav should be installed. Have a look at the documentation of pydub for instructions. As an alternative to the steps described in the documentation, ffmpeg-python provides bindings for ffmpeg, and can be installed using pip:



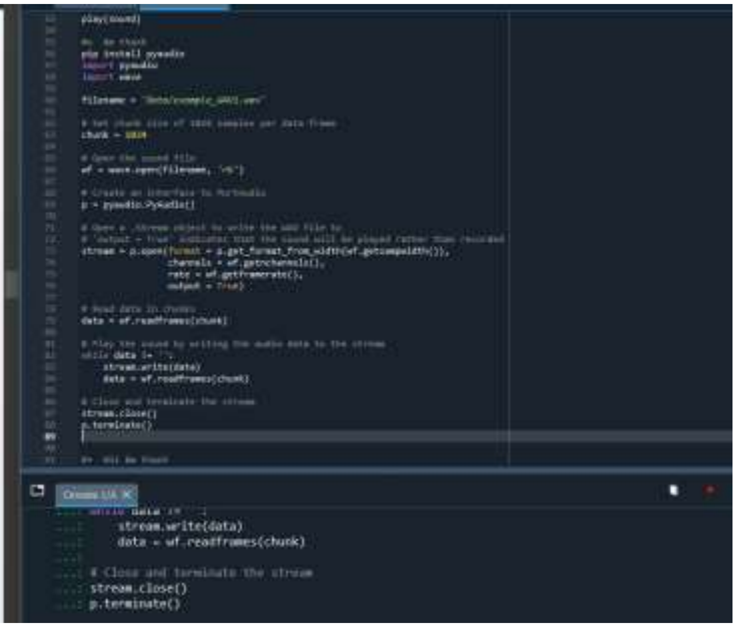
```
# Extract data and sampling rate from file
data, fs = sf.read(filename, dtype='float32')
sd.play(data, fs)
status = sd.wait()  # Wait until file is done playing

#o  pydub
pip install pydub

from pydub import AudioSegment
from pydub.playback import play
sound = AudioSegment.from_wav('Data/example_WAV1.wav')
play(sound)

#o  âm thanh

#•  Ghi âm thanh
#o  python-sounddevice
#o  âm thanh
```

```
In [24]: pip install pydub
Collecting pydub
  Downloading pydub-0.25.1-py2.py3-none-any.whl.metadata (1.4 kB)
Downloading pydub-0.25.1-py2.py3-none-any.whl (32 kB)
Installing collected packages: pydub
Successfully installed pydub-0.25.1
Note: you may need to restart the kernel to use updated packages.

In [25]: from pydub import AudioSegment
    ...: from pydub.playback import play
    ...: sound = AudioSegment.from_wav('Data/example_WAV1.wav')
    ...: play(sound)
C:\Users\[illegible]\anaconda3\envs\multimedia\lib\site-packages\pydub\utils.py:170: RuntimeWarning:
Couldn't find ffmpeg or avconv - defaulting to ffmpeg, but may not work
  warn("Couldn't find ffmpeg or avconv - defaulting to ffmpeg, but may not work", RuntimeWarning)
```

- *âm thanh*



```python
import pyaudio
import wave

filename = 'myfile.wav'

# Set chunk size of 1024 samples per data frame
chunk = 1024

# Open the sound file
wf = wave.open(filename, 'rb')

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

# Open a .Stream object to write the WAV file to
# 'output = True' indicates that the sound will be played rathe
stream = p.open(format = p.get_format_from_width(wf.getsampwidt
                channels = wf.getnchannels(),
                rate = wf.getframerate(),
                output = True)

# Read data in chunks
data = wf.readframes(chunk)

# Play the sound by writing the audio data to the stream
while data != '':
    stream.write(data)
    data = wf.readframes(chunk)
```



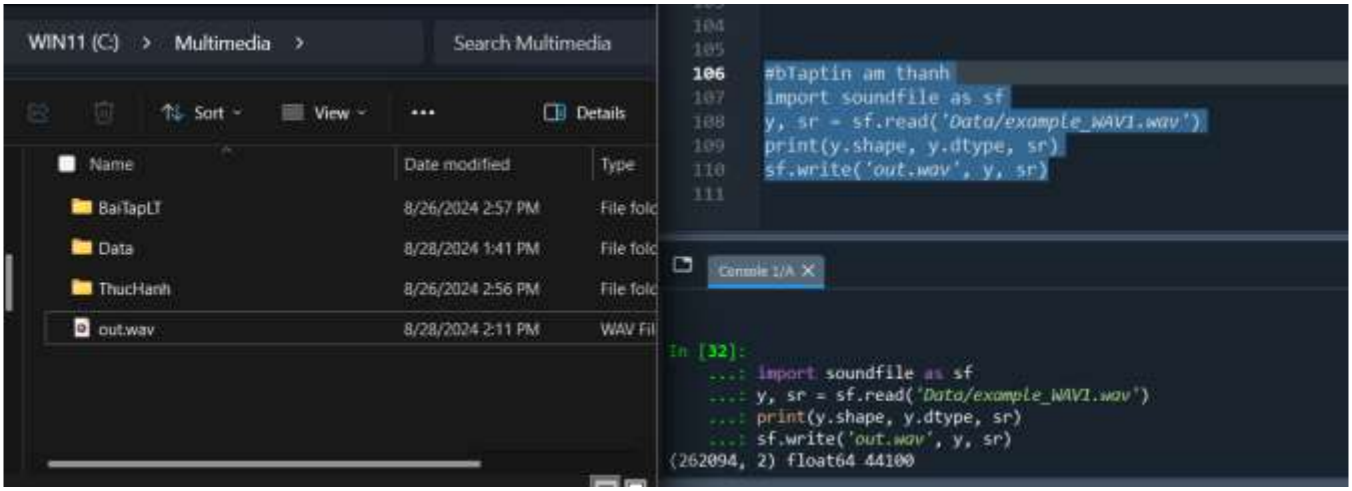
```
play(sound)

#o  âm thanh
pip install pyaudio
import pyaudio
import wave

filename = 'Data/example_WAV1.wav'

# Set chunk size of 1024 samples per data frame
chunk = 1024

# Open the sound file
wf = wave.open(filename, 'rb')

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

# Open a .Stream object to write the WAV file to
# 'output = True' indicates that the sound will be played rather than recorded
stream = p.open(format = p.get_format_from_width(wf.getsampwidth()),
                channels = wf.getnchannels(),
                rate = wf.getframerate(),
                output = True)

# Read data in chunks
data = wf.readframes(chunk)

# Play the sound by writing the audio data to the stream
while data != '':
    stream.write(data)
    data = wf.readframes(chunk)

# Close and terminate the stream
stream.close()
p.terminate()


#•  Ghi âm thanh
```

```
   ...: while data != '':
   ...:     stream.write(data)
   ...:     data = wf.readframes(chunk)
   ...:
   ...: # Close and terminate the stream
   ...: stream.close()
   ...: p.terminate()
```

- **Ghi âm thanh**
- *python-sounddevice*
- *âm thanh*
- **Lưu và chuyển đổi âm thanh**
- *Wavio*
- *tập tin âm thanh*
- *pydub*
- **So sánh các thư viện âm thanh**
- **Kết luận: Phát và ghi âm thanh trong Python**

b)https://www.it-jim.com/blog/audio-processing-basics-in-python/ {thực hành}

Soundfile:

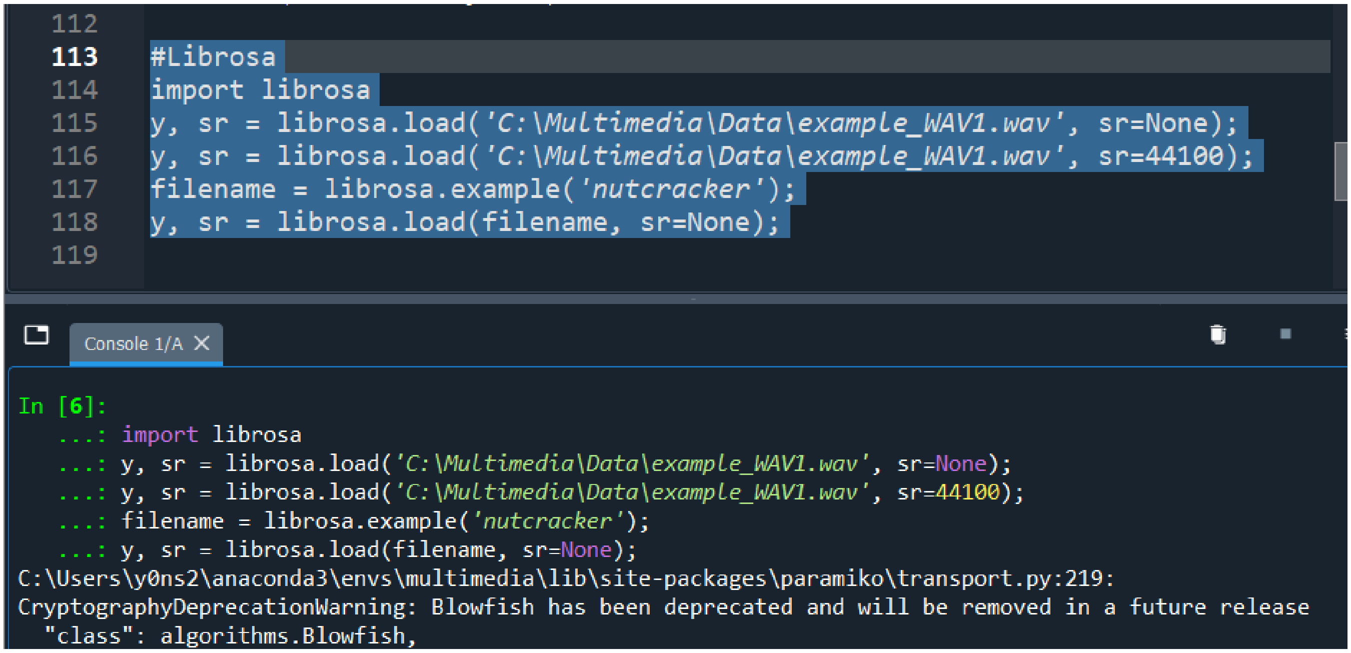


Name	Date modified	Type
BaiTapLT	8/26/2024 2:57 PM	File fold
Data	8/28/2024 1:41 PM	File fold
ThucHanh	8/26/2024 2:56 PM	File fold
out.wav	8/28/2024 2:11 PM	WAV Fil

```
#bTaptin am thanh
import soundfile as sf
y, sr = sf.read('Data/example_WAV1.wav')
print(y.shape, y.dtype, sr)
sf.write('out.wav', y, sr)
```

```
In [32]:
   ...: import soundfile as sf
   ...: y, sr = sf.read('Data/example_WAV1.wav')
   ...: print(y.shape, y.dtype, sr)
   ...: sf.write('out.wav', y, sr)
(262094, 2) float64 44100
```

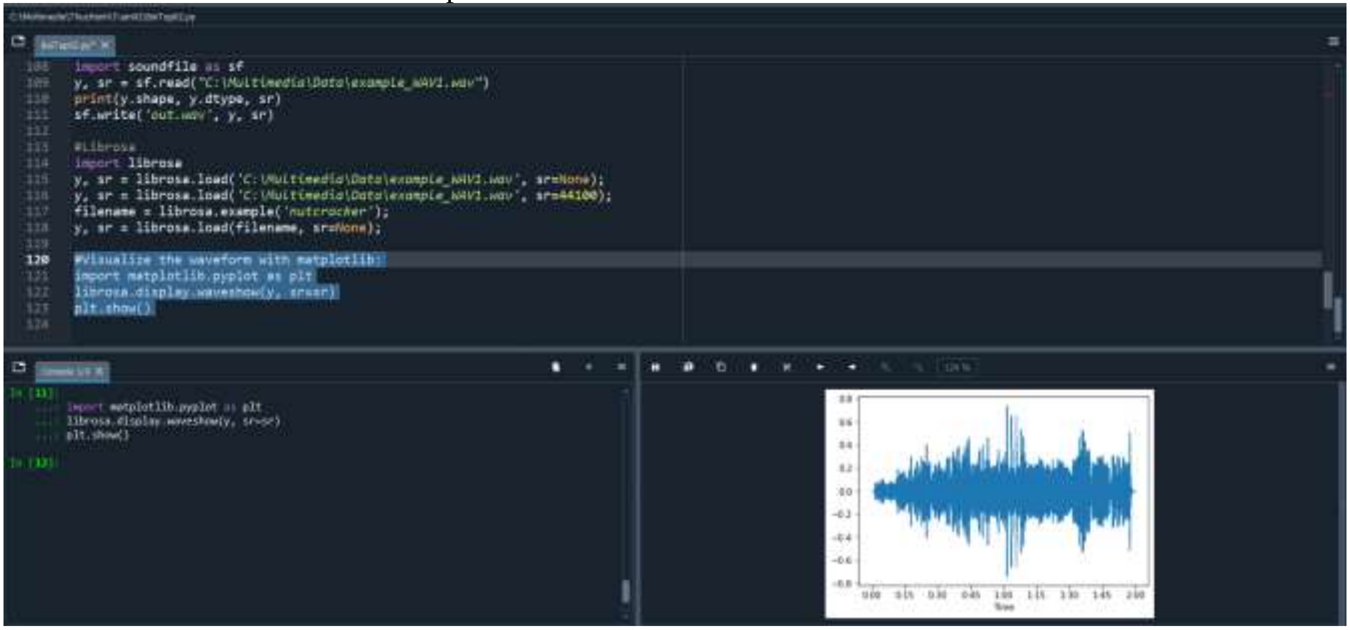
Librosa:



```
#Librosa
import librosa
y, sr = librosa.load('C:\Multimedia\Data\example_WAV1.wav', sr=None);
y, sr = librosa.load('C:\Multimedia\Data\example_WAV1.wav', sr=44100);
filename = librosa.example('nutcracker');
y, sr = librosa.load(filename, sr=None);
```

```
In [6]:
   ...: import librosa
   ...: y, sr = librosa.load('C:\Multimedia\Data\example_WAV1.wav', sr=None);
   ...: y, sr = librosa.load('C:\Multimedia\Data\example_WAV1.wav', sr=44100);
   ...: filename = librosa.example('nutcracker');
   ...: y, sr = librosa.load(filename, sr=None);
C:\Users\y0ns2\anaconda3\envs\multimedia\lib\site-packages\paramiko\transport.py:219:
CryptographyDeprecationWarning: Blowfish has been deprecated and will be removed in a future release
  "class": algorithms.Blowfish,
```

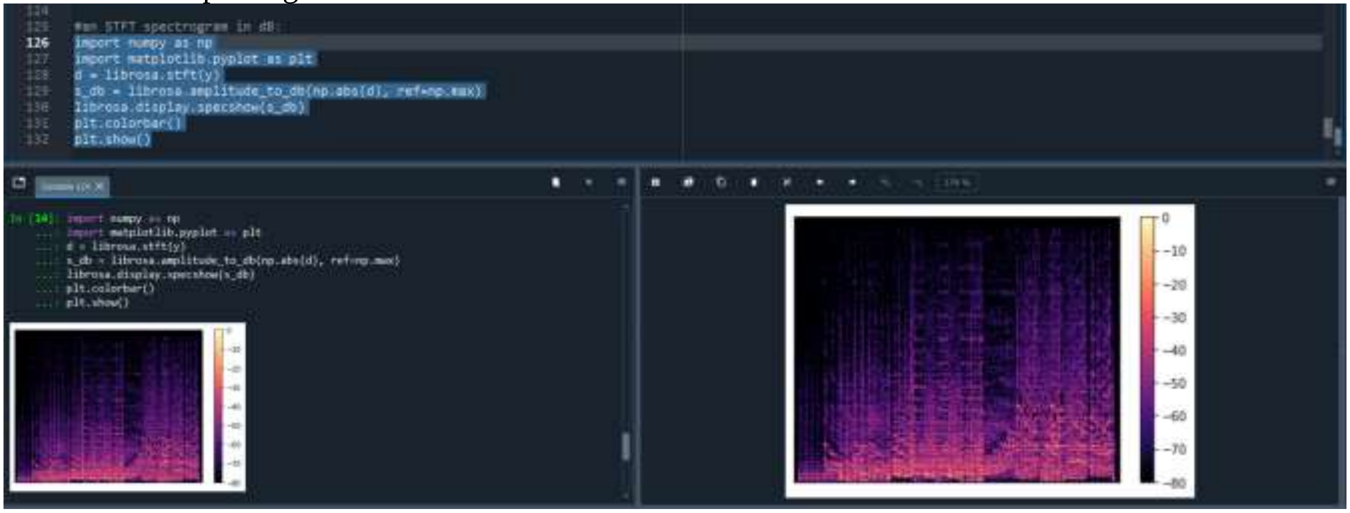
#Visualize the waveform with matplotlib:



```
import soundfile as sf
y, sr = sf.read("C:\Multimedia\Data\example_WAV1.wav")
print(y.shape, y.dtype, sr)
sf.write('out.wav', y, sr)

#Librosa
import librosa
y, sr = librosa.load('C:\Multimedia\Data\example_WAV1.wav', sr=None);
y, sr = librosa.load('C:\Multimedia\Data\example_WAV1.wav', sr=44100);
filename = librosa.example('nutcracker');
y, sr = librosa.load(filename, sr=None);

#Visualize the waveform with matplotlib:
import matplotlib.pyplot as plt
librosa.display.waveshow(y, sr=sr)
plt.show()
```

```
In [11]:
   ...: import matplotlib.pyplot as plt
   ...: librosa.display.waveshow(y, sr=sr)
   ...: plt.show()

In [12]:
```

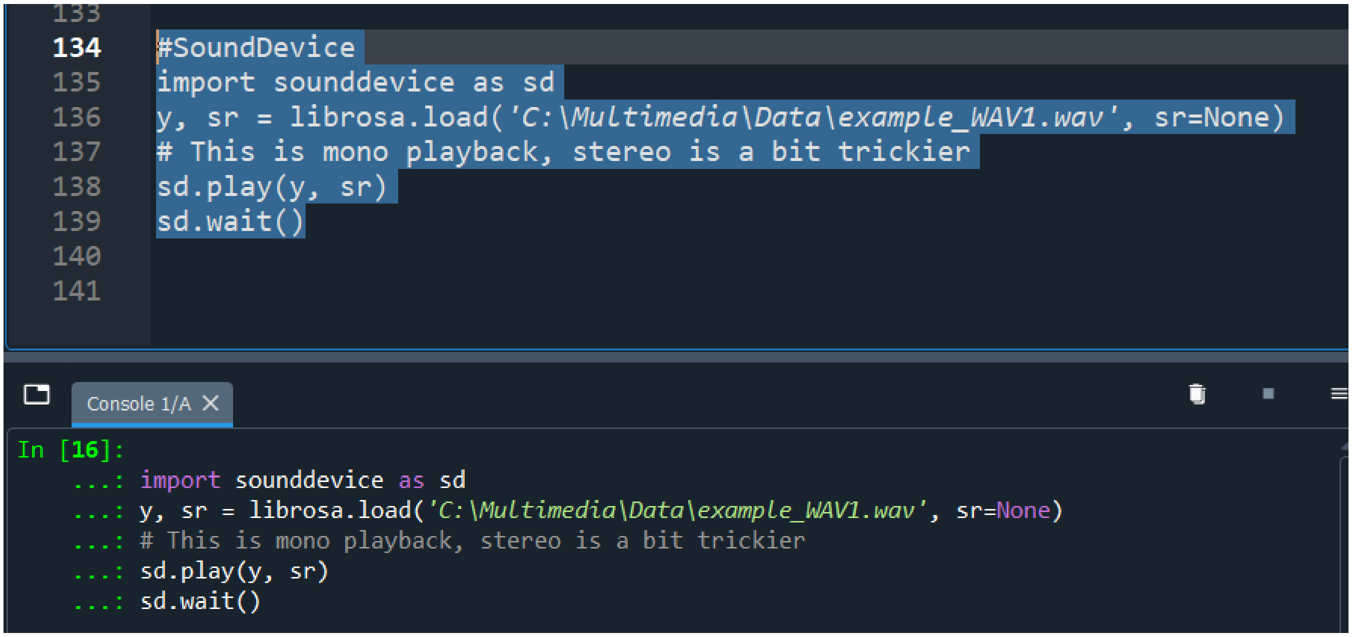
# Or an STFT spectrogram in dB:



```
#an STFT spectrogram in dB:
import numpy as np
import matplotlib.pyplot as plt
d = librosa.stft(y)
s_db = librosa.amplitude_to_db(np.abs(d), ref=np.max)
librosa.display.specshow(s_db)
plt.colorbar()
plt.show()
```

```
In [14]: import numpy as np
    ...: import matplotlib.pyplot as plt
    ...: d = librosa.stft(y)
    ...: s_db = librosa.amplitude_to_db(np.abs(d), ref=np.max)
    ...: librosa.display.specshow(s_db)
    ...: plt.colorbar()
    ...: plt.show()
```

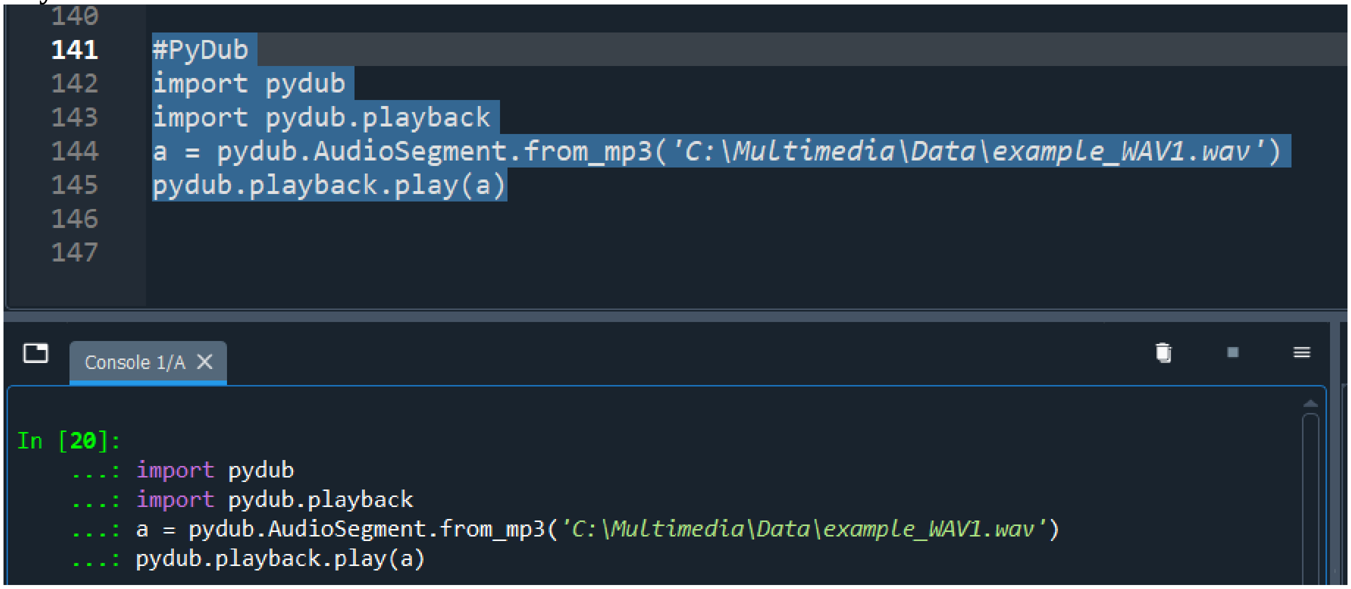
#SoundDevice



```
#SoundDevice
import sounddevice as sd
y, sr = librosa.load('C:\Multimedia\Data\example_WAV1.wav', sr=None)
# This is mono playback, stereo is a bit trickier
sd.play(y, sr)
sd.wait()
```

```
In [16]:
    ...: import sounddevice as sd
    ...: y, sr = librosa.load('C:\Multimedia\Data\example_WAV1.wav', sr=None)
    ...: # This is mono playback, stereo is a bit trickier
    ...: sd.play(y, sr)
    ...: sd.wait()
```

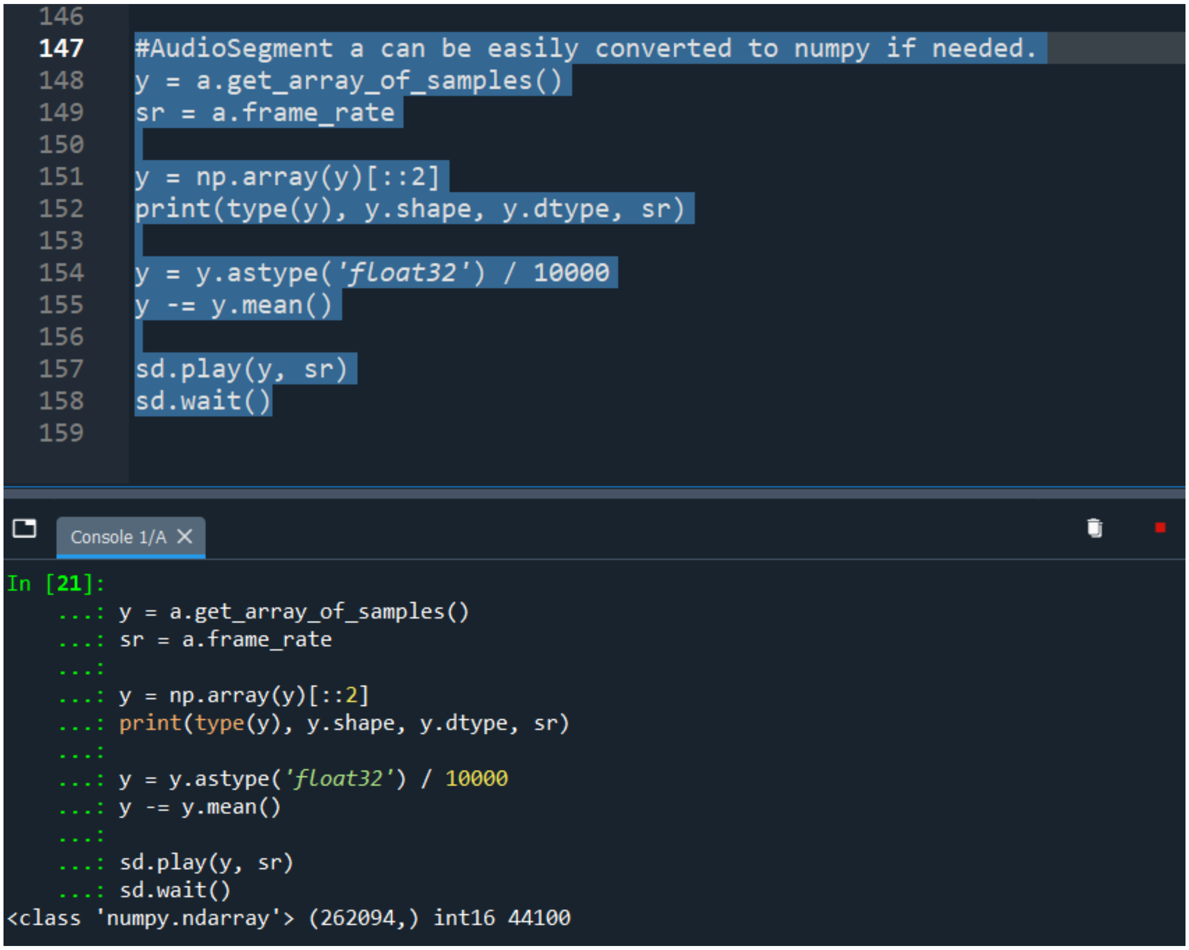
#PyDub



```
#PyDub
import pydub
import pydub.playback
a = pydub.AudioSegment.from_mp3('C:\Multimedia\Data\example_WAV1.wav')
pydub.playback.play(a)
```

```
In [20]:
    ...: import pydub
    ...: import pydub.playback
    ...: a = pydub.AudioSegment.from_mp3('C:\Multimedia\Data\example_WAV1.wav')
    ...: pydub.playback.play(a)
```

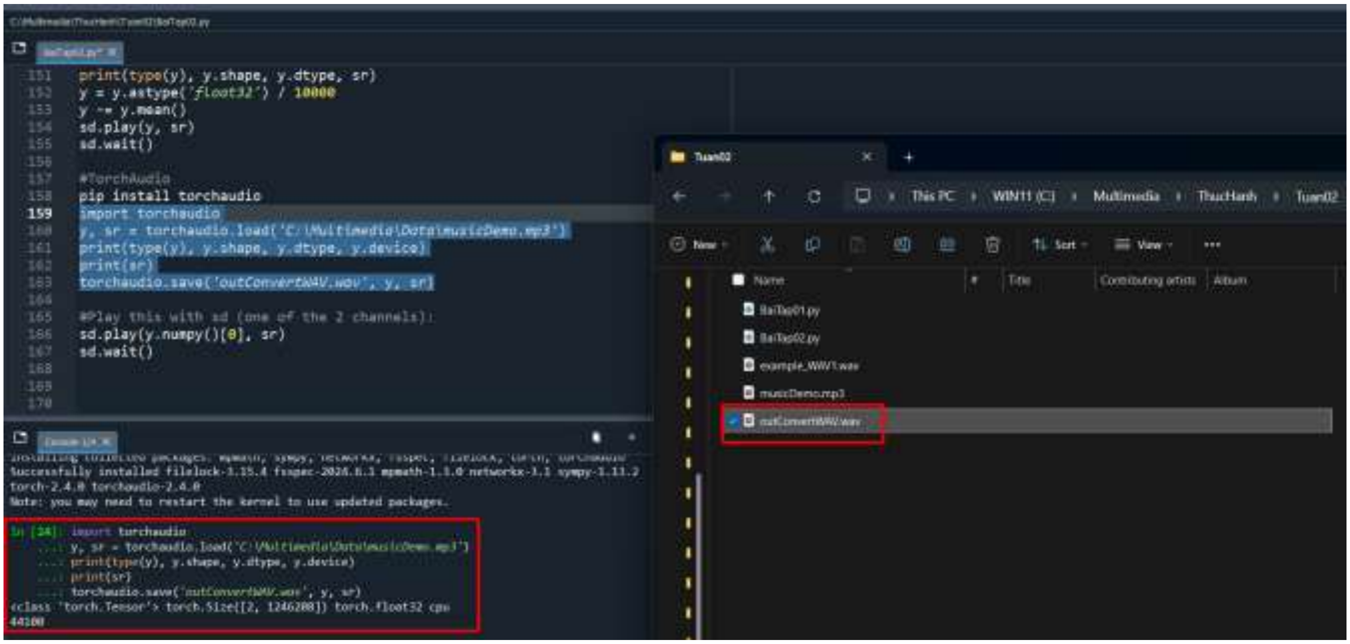
AudioSegment **a** can be easily converted to numpy if needed.



```
#AudioSegment a can be easily converted to numpy if needed.
y = a.get_array_of_samples()
sr = a.frame_rate

y = np.array(y)[::2]
print(type(y), y.shape, y.dtype, sr)

y = y.astype('float32') / 10000
y -= y.mean()

sd.play(y, sr)
sd.wait()
```

```
In [21]:
    ...: y = a.get_array_of_samples()
    ...: sr = a.frame_rate
    ...:
    ...: y = np.array(y)[::2]
    ...: print(type(y), y.shape, y.dtype, sr)
    ...:
    ...: y = y.astype('float32') / 10000
    ...: y -= y.mean()
    ...:
    ...: sd.play(y, sr)
    ...: sd.wait()
<class 'numpy.ndarray'> (262094,) int16 44100
```

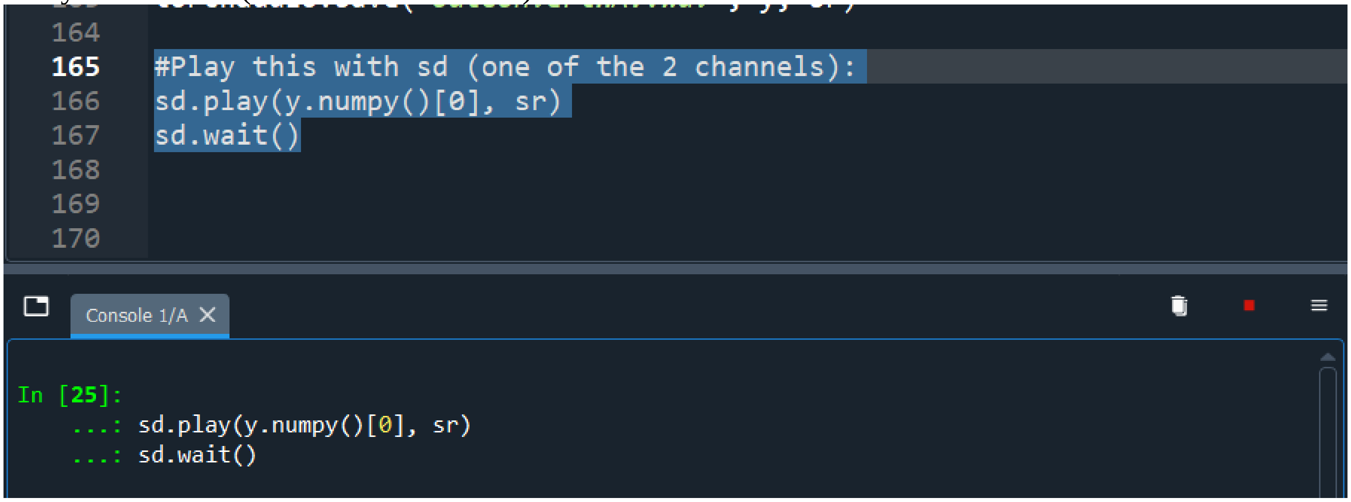
#TorchAudio



```
print(type(y), y.shape, y.dtype, sr)
y = y.astype('float32') / 10000
y -= y.mean()
sd.play(y, sr)
sd.wait()

#TorchAudio
pip install torchaudio
import torchaudio
y, sr = torchaudio.load('C:\\Multimedia\Data\musicDemo.mp3')
print(type(y), y.shape, y.dtype, y.device)
print(sr)
torchaudio.save('outConvertWAV.wav', y, sr)

#Play this with sd (one of the 2 channels):
sd.play(y.numpy()[0], sr)
sd.wait()
```

```
Installing collected packages: mpmath, sympy, networkx, fsspec, filelock, torch, torchaudio
Successfully installed filelock-3.15.4 fsspec-2024.6.1 mpmath-1.3.0 networkx-3.1 sympy-1.13.2
torch-2.4.0 torchaudio-2.4.0
Note: you may need to restart the kernel to use updated packages.

In [24]: import torchaudio
    ...: y, sr = torchaudio.load('C:\\Multimedia\Data\musicDemo.mp3')
    ...: print(type(y), y.shape, y.dtype, y.device)
    ...: print(sr)
    ...: torchaudio.save('outConvertWAV.wav', y, sr)
<class 'torch.Tensor'> torch.Size([2, 1246208]) torch.float32 cpu
44100
```

#Play this with sd (one of the 2 channels):



```
torchaudio.save('outConvertWAV.wav', y, sr)

#Play this with sd (one of the 2 channels):
sd.play(y.numpy()[0], sr)
sd.wait()
```

```
In [25]:
    ...: sd.play(y.numpy()[0], sr)
    ...: sd.wait()
```

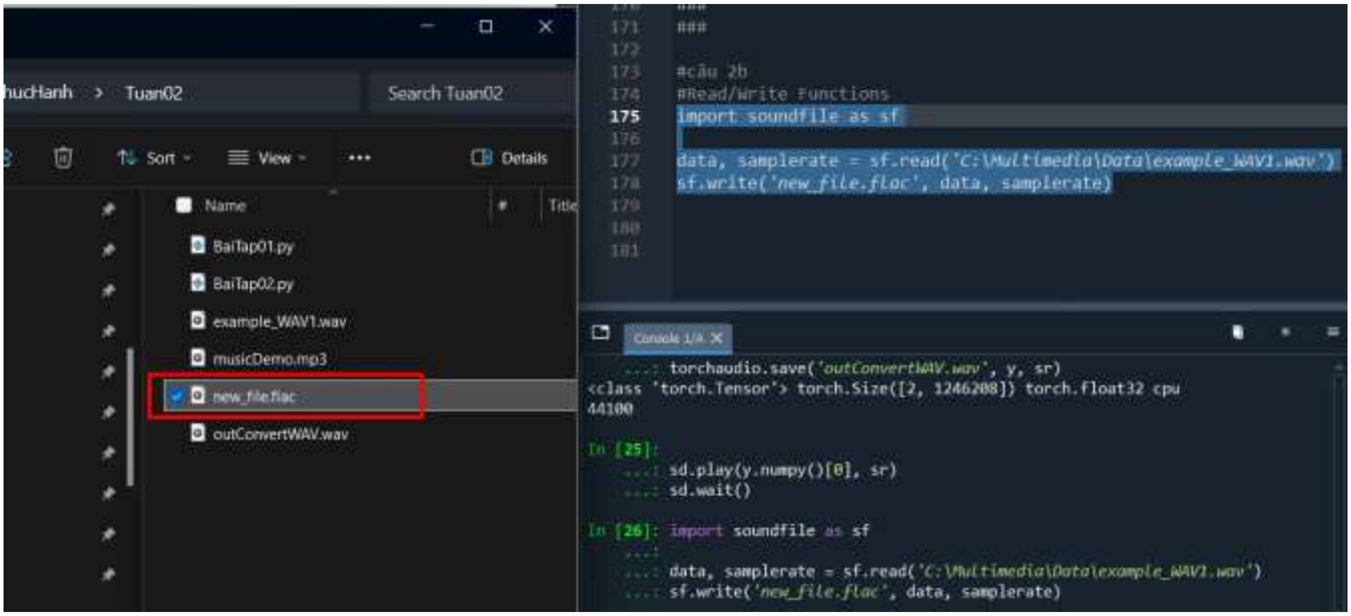
b1)https://pysoundfile.readthedocs.io/en/latest/ {thực hành}
Truy cập trang website sau:
https://python-soundfile.readthedocs.io/en/latest/

Breaking Changes

Installation:

pip install soundfile

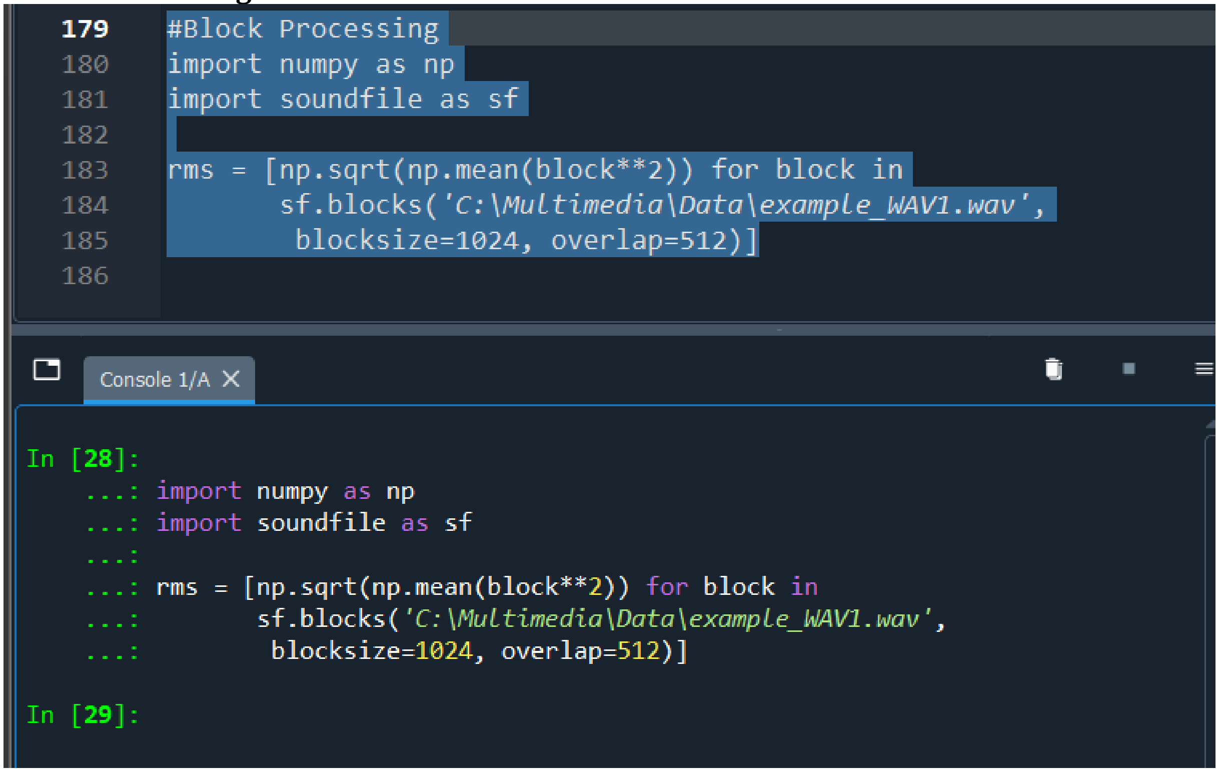
Read/Write Functions



```
###

#câu 2b
#Read/Write Functions
import soundfile as sf

data, samplerate = sf.read('C:\Multimedia\Data\example_WAV1.wav')
sf.write('new_file.flac', data, samplerate)
```

```
   ...: torchaudio.save('outConvertWAV.wav', y, sr)
<class 'torch.Tensor'> torch.Size([2, 1246208]) torch.float32 cpu
44100

In [25]:
   ...: sd.play(y.numpy()[0], sr)
   ...: sd.wait()

In [26]: import soundfile as sf
   ...:
   ...: data, samplerate = sf.read('C:\Multimedia\Data\example_WAV1.wav')
   ...: sf.write('new_file.flac', data, samplerate)
```

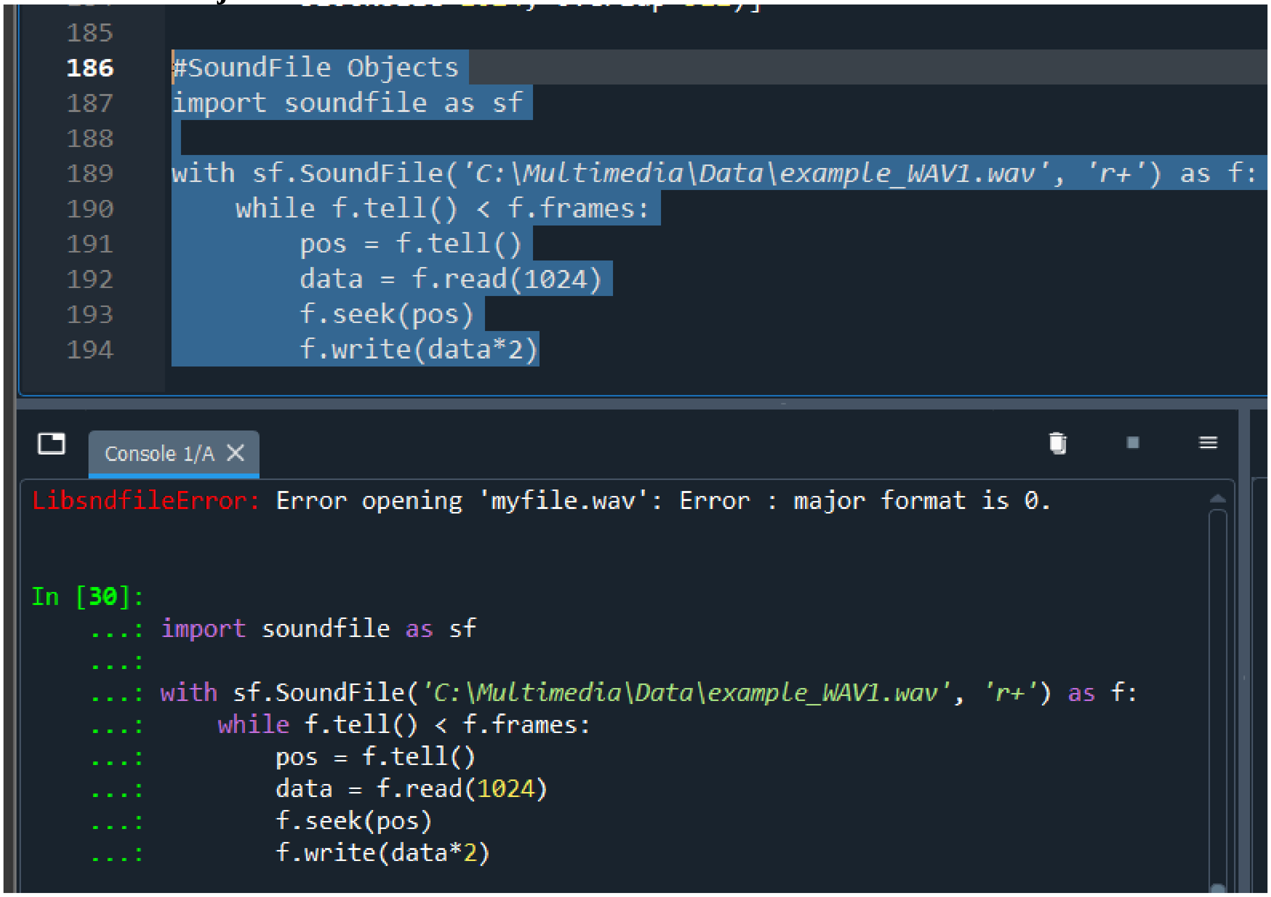
#Block Processing



```
#Block Processing
import numpy as np
import soundfile as sf

rms = [np.sqrt(np.mean(block**2)) for block in
       sf.blocks('C:\Multimedia\Data\example_WAV1.wav',
        blocksize=1024, overlap=512)]
```

```
In [28]:
   ...: import numpy as np
   ...: import soundfile as sf
   ...:
   ...: rms = [np.sqrt(np.mean(block**2)) for block in
   ...:        sf.blocks('C:\Multimedia\Data\example_WAV1.wav',
   ...:         blocksize=1024, overlap=512)]

In [29]:
```

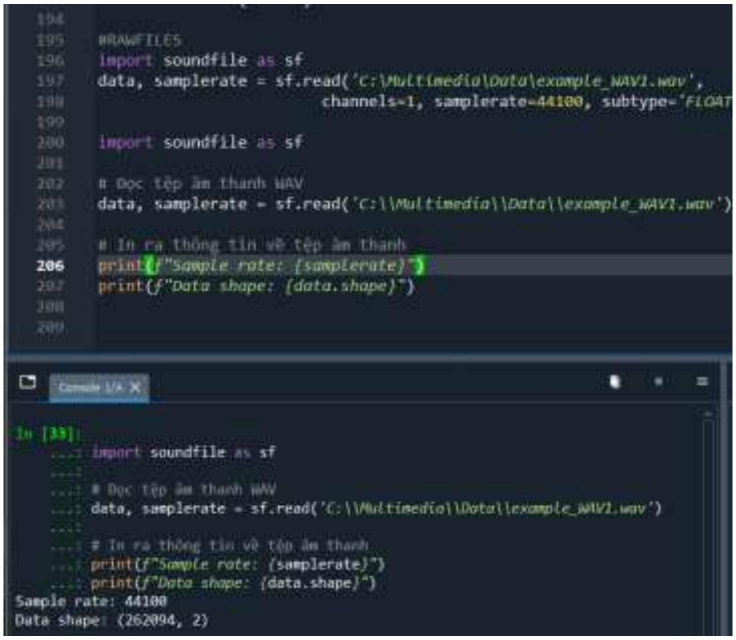
SoundFile Objects



```
#SoundFile Objects
import soundfile as sf

with sf.SoundFile('C:\Multimedia\Data\example_WAV1.wav', 'r+') as f:
    while f.tell() < f.frames:
        pos = f.tell()
        data = f.read(1024)
        f.seek(pos)
        f.write(data*2)
```

```
LibsndfileError: Error opening 'myfile.wav': Error : major format is 0.


In [30]:
    ...: import soundfile as sf
    ...:
    ...: with sf.SoundFile('C:\Multimedia\Data\example_WAV1.wav', 'r+') as f:
    ...:     while f.tell() < f.frames:
    ...:         pos = f.tell()
    ...:         data = f.read(1024)
    ...:         f.seek(pos)
    ...:         f.write(data*2)
```

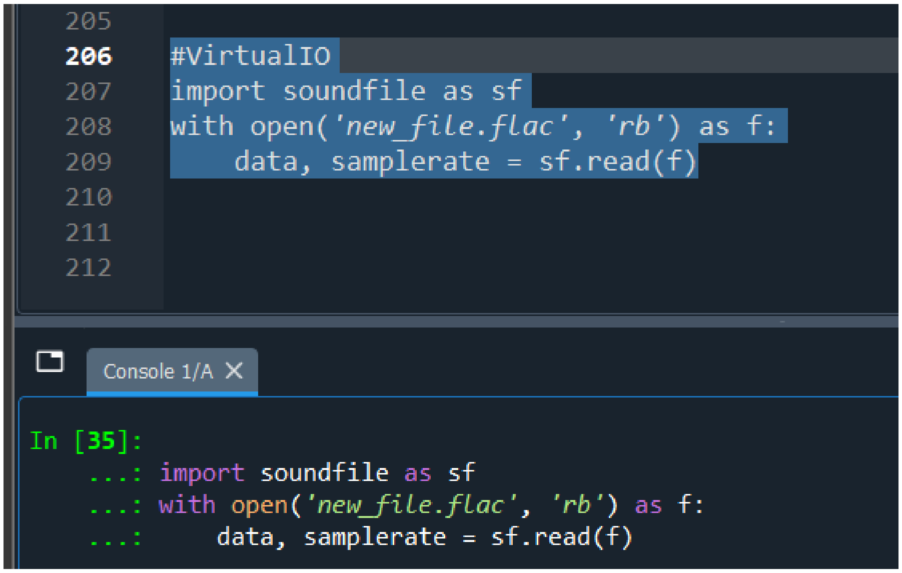
#RAW FILES



```
#RAWFILES
import soundfile as sf
data, samplerate = sf.read('C:\Multimedia\Data\example_WAV1.wav',
                           channels=1, samplerate=44100, subtype='FLOAT

import soundfile as sf

# Đọc tệp âm thanh WAV
data, samplerate = sf.read('C:\\Multimedia\\Data\\example_WAV1.wav')

# In ra thông tin về tệp âm thanh
print(f"Sample rate: {samplerate}")
print(f"Data shape: {data.shape}")
```

```
In [33]:
    ...: import soundfile as sf
    ...:
    ...: # Đọc tệp âm thanh WAV
    ...: data, samplerate = sf.read('C:\\Multimedia\\Data\\example_WAV1.wav')
    ...:
    ...: # In ra thông tin về tệp âm thanh
    ...: print(f"Sample rate: {samplerate}")
    ...: print(f"Data shape: {data.shape}")
Sample rate: 44100
Data shape: (262094, 2)
```

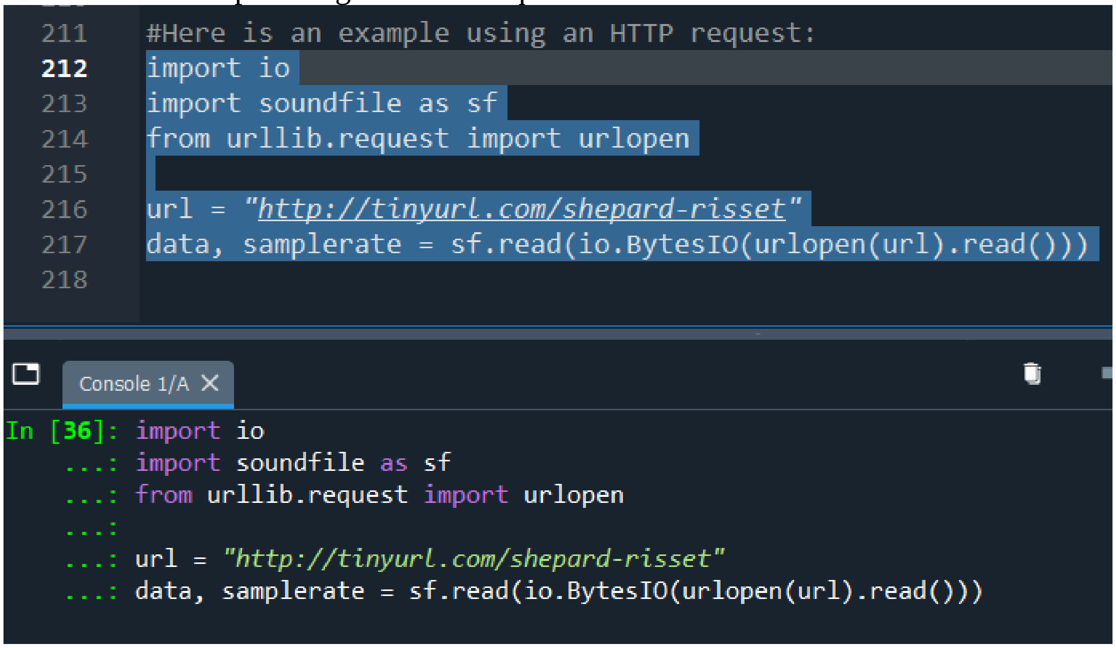
Virtual IO



```
#VirtualIO
import soundfile as sf
with open('new_file.flac', 'rb') as f:
    data, samplerate = sf.read(f)
```

```
In [35]:
   ...: import soundfile as sf
   ...: with open('new_file.flac', 'rb') as f:
   ...:     data, samplerate = sf.read(f)
```

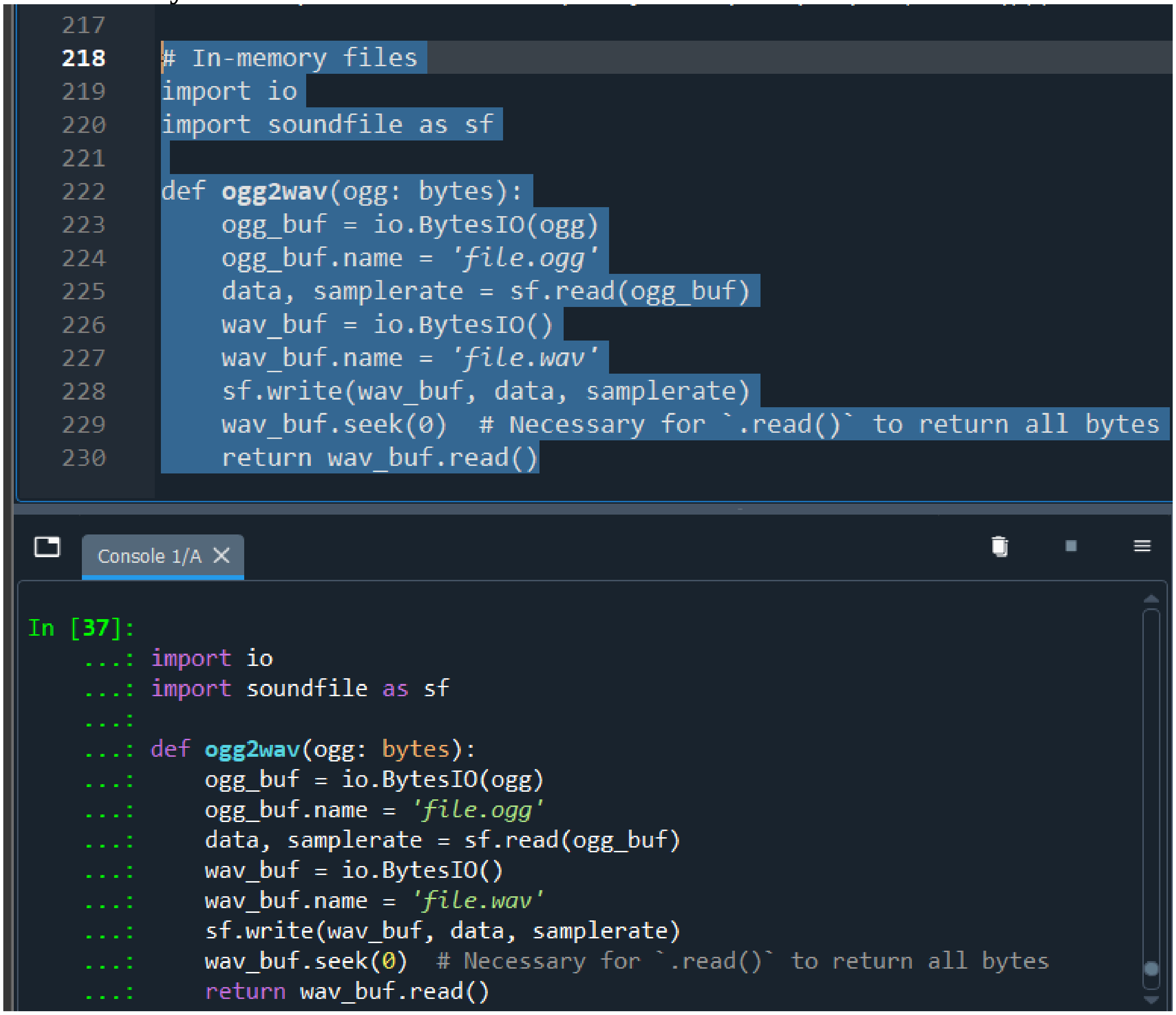
Here is an example using an HTTP request:



```
#Here is an example using an HTTP request:
import io
import soundfile as sf
from urllib.request import urlopen

url = "http://tinyurl.com/shepard-risset"
data, samplerate = sf.read(io.BytesIO(urlopen(url).read()))
```

```
In [36]: import io
    ...: import soundfile as sf
    ...: from urllib.request import urlopen
    ...:
    ...: url = "http://tinyurl.com/shepard-risset"
    ...: data, samplerate = sf.read(io.BytesIO(urlopen(url).read()))
```

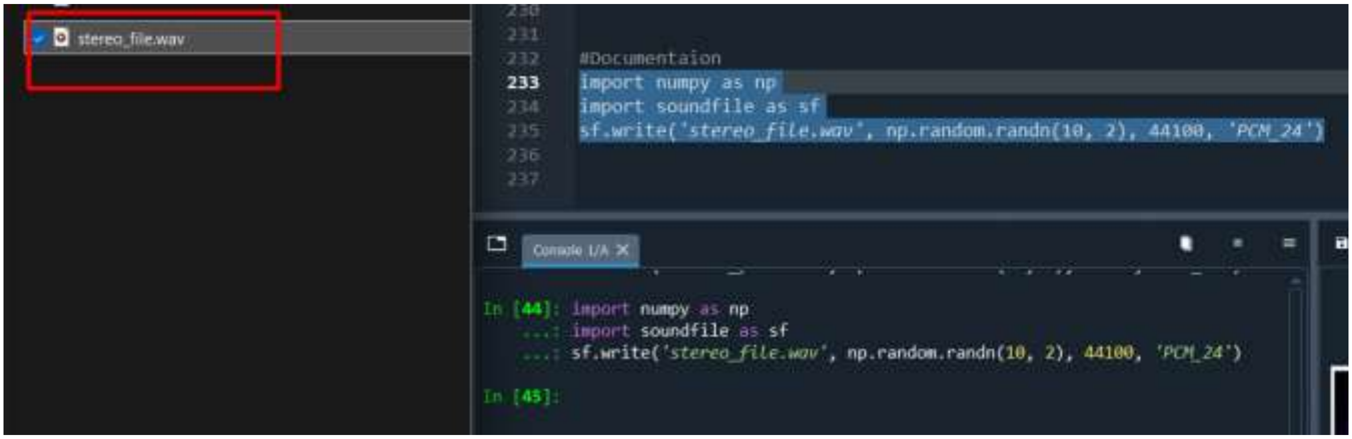
In-memory files



```
# In-memory files
import io
import soundfile as sf

def ogg2wav(ogg: bytes):
    ogg_buf = io.BytesIO(ogg)
    ogg_buf.name = 'file.ogg'
    data, samplerate = sf.read(ogg_buf)
    wav_buf = io.BytesIO()
    wav_buf.name = 'file.wav'
    sf.write(wav_buf, data, samplerate)
    wav_buf.seek(0)  # Necessary for `.read()` to return all bytes
    return wav_buf.read()
```

```
In [37]:
    ...: import io
    ...: import soundfile as sf
    ...:
    ...: def ogg2wav(ogg: bytes):
    ...:     ogg_buf = io.BytesIO(ogg)
    ...:     ogg_buf.name = 'file.ogg'
    ...:     data, samplerate = sf.read(ogg_buf)
    ...:     wav_buf = io.BytesIO()
    ...:     wav_buf.name = 'file.wav'
    ...:     sf.write(wav_buf, data, samplerate)
    ...:     wav_buf.seek(0)  # Necessary for `.read()` to return all bytes
    ...:     return wav_buf.read()
```

#Write 10 frames of random data to a new file:

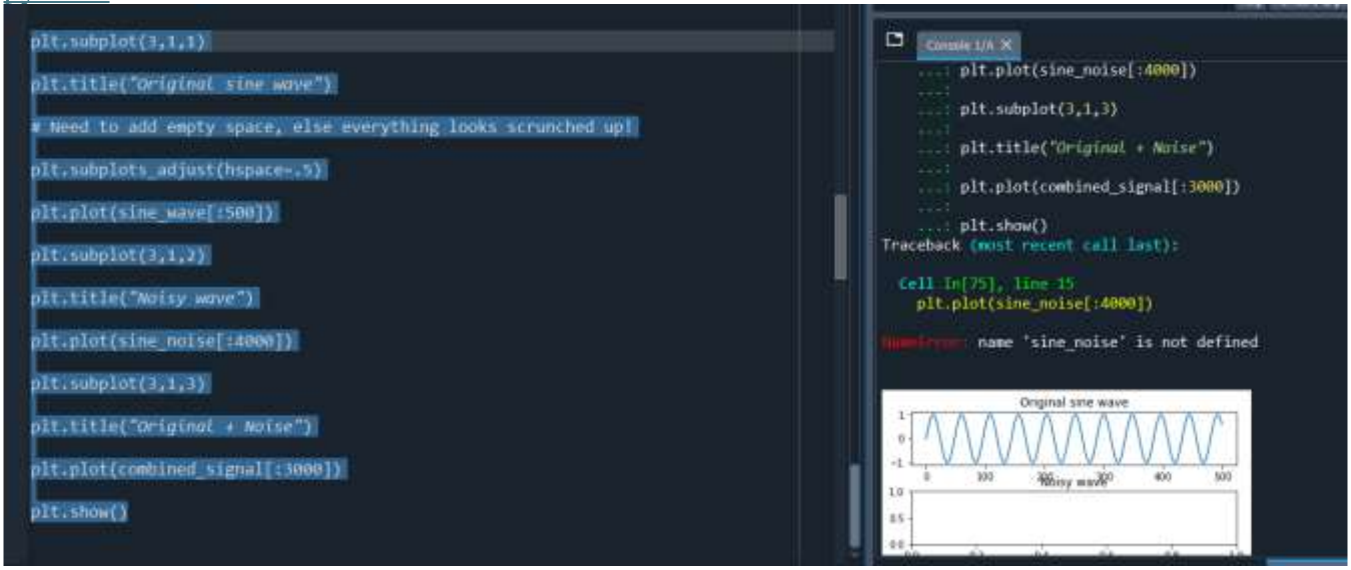


```
#Documentaion
import numpy as np
import soundfile as sf
sf.write('stereo_file.wav', np.random.randn(10, 2), 44100, 'PCM_24')
```

```
In [44]: import numpy as np
    ...: import soundfile as sf
    ...: sf.write('stereo_file.wav', np.random.randn(10, 2), 44100, 'PCM_24')

In [45]:
```

c)https://www.pythonforengineers.com/audio-and-digital-signal-processingdsp-in-python/ {thực hành}

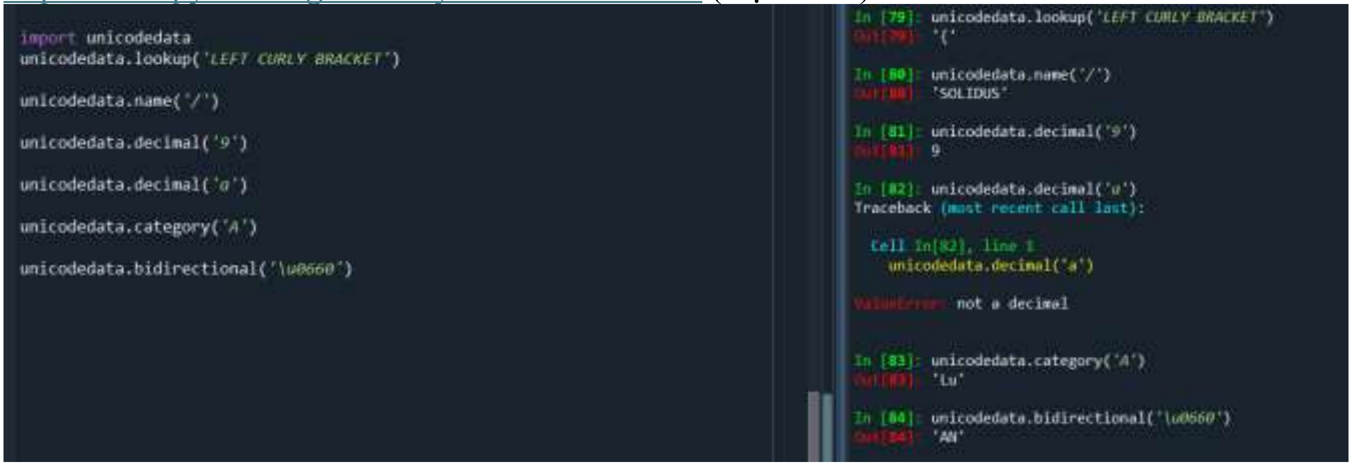
Link: https://new.pythonforengineers.com/blog/audio-and-digital-signal-processingdsp-in-python/



```
plt.subplot(3,1,1)

plt.title("Original sine wave")

# Need to add empty space, else everything looks scrunched up!

plt.subplots_adjust(hspace=.5)

plt.plot(sine_wave[:500])

plt.subplot(3,1,2)

plt.title("Noisy wave")

plt.plot(sine_noise[:4000])

plt.subplot(3,1,3)

plt.title("Original + Noise")

plt.plot(combined_signal[:3000])

plt.show()
```

```
...: plt.plot(sine_noise[:4000])
...:
...: plt.subplot(3,1,3)
...:
...: plt.title("Original + Noise")
...:
...: plt.plot(combined_signal[:3000])
...:
...: plt.show()
Traceback (most recent call last):

  Cell In[75], line 15
    plt.plot(sine_noise[:4000])

NameError: name 'sine_noise' is not defined
```

https://docs.python.org/3/library/mm.html (tham khảo)
https://docs.python.org/3/library/unicodedata.html (thực hành)



```
import unicodedata
unicodedata.lookup('LEFT CURLY BRACKET')

unicodedata.name('/')

unicodedata.decimal('9')

unicodedata.decimal('a')

unicodedata.category('A')

unicodedata.bidirectional('\u0660')
```

```
In [79]: unicodedata.lookup('LEFT CURLY BRACKET')
Out[79]: '{'

In [80]: unicodedata.name('/')
Out[80]: 'SOLIDUS'

In [81]: unicodedata.decimal('9')
Out[81]: 9

In [82]: unicodedata.decimal('a')
Traceback (most recent call last):

  Cell In[82], line 1
    unicodedata.decimal('a')

ValueError: not a decimal


In [83]: unicodedata.category('A')
Out[83]: 'Lu'

In [84]: unicodedata.bidirectional('\u0660')
Out[84]: 'AN'
```

https://matplotlib.org/tutorials/introductory/pyplot.html {thư viện tham khảo}
pip install scikits.audiolab==0.11.0

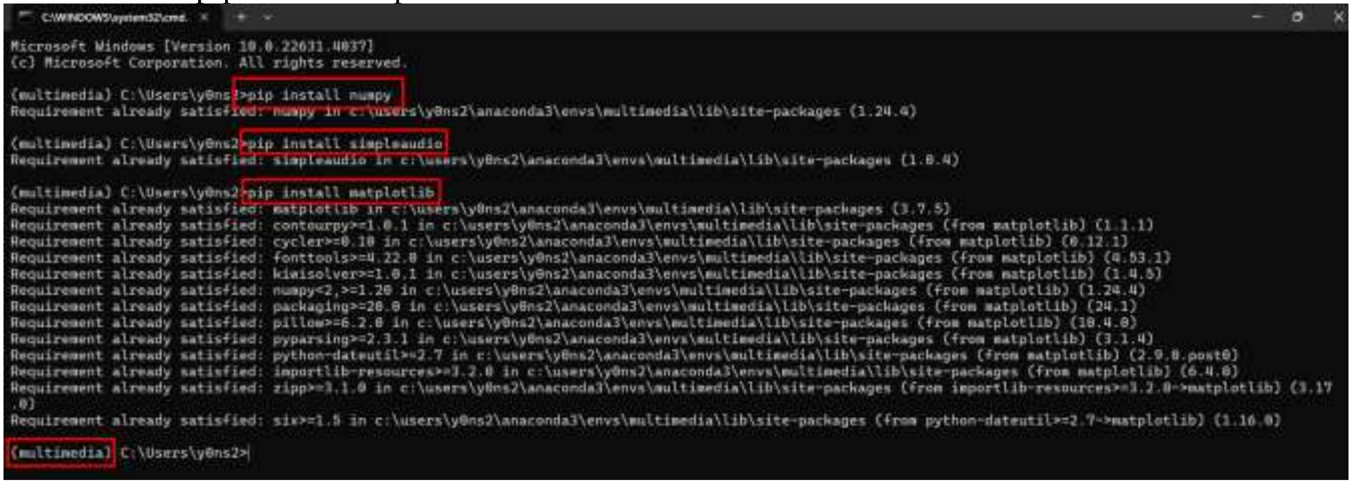
3) Công cụ : Python programming language

a)Python IDE hoặc PYTHON command line (trên LAPTOP)

b)Cài đặt các thư viện hỗ trợ :

Câu lệnh cài đặt : pip install <gói cài đặt>

Vd : pip install numpy
pip install simpleaudio
pip install matplotlib

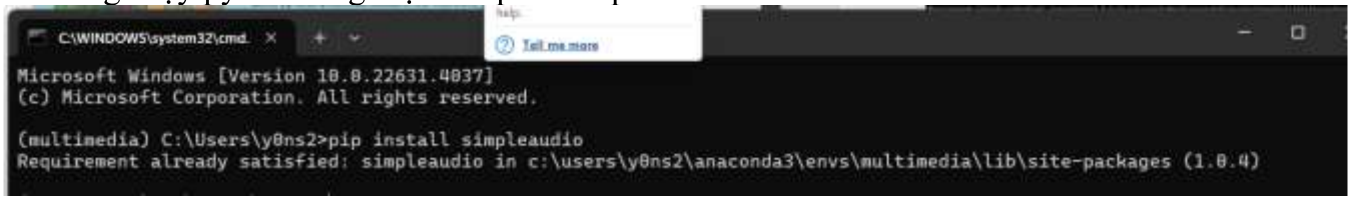


```
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

(multimedia) C:\Users\y0ns2>pip install numpy
Requirement already satisfied: numpy in c:\users\y0ns2\anaconda3\envs\multimedia\lib\site-packages (1.24.4)

(multimedia) C:\Users\y0ns2>pip install simpleaudio
Requirement already satisfied: simpleaudio in c:\users\y0ns2\anaconda3\envs\multimedia\lib\site-packages (1.0.4)

(multimedia) C:\Users\y0ns2>pip install matplotlib
Requirement already satisfied: matplotlib in c:\users\y0ns2\anaconda3\envs\multimedia\lib\site-packages (3.7.5)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\y0ns2\anaconda3\envs\multimedia\lib\site-packages (from matplotlib) (1.1.1)
Requirement already satisfied: cycler>=0.10 in c:\users\y0ns2\anaconda3\envs\multimedia\lib\site-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\y0ns2\anaconda3\envs\multimedia\lib\site-packages (from matplotlib) (4.53.1)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\y0ns2\anaconda3\envs\multimedia\lib\site-packages (from matplotlib) (1.4.5)
Requirement already satisfied: numpy<2,>=1.20 in c:\users\y0ns2\anaconda3\envs\multimedia\lib\site-packages (from matplotlib) (1.24.4)
Requirement already satisfied: packaging>=20.0 in c:\users\y0ns2\anaconda3\envs\multimedia\lib\site-packages (from matplotlib) (24.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\y0ns2\anaconda3\envs\multimedia\lib\site-packages (from matplotlib) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\y0ns2\anaconda3\envs\multimedia\lib\site-packages (from matplotlib) (3.1.4)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\y0ns2\anaconda3\envs\multimedia\lib\site-packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: importlib-resources>=3.2.0 in c:\users\y0ns2\anaconda3\envs\multimedia\lib\site-packages (from matplotlib) (6.4.0)
Requirement already satisfied: zipp>=3.1.0 in c:\users\y0ns2\anaconda3\envs\multimedia\lib\site-packages (from importlib-resources>=3.2.0->matplotlib) (3.17
.0)
Requirement already satisfied: six>=1.5 in c:\users\y0ns2\anaconda3\envs\multimedia\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)

(multimedia) C:\Users\y0ns2>
```

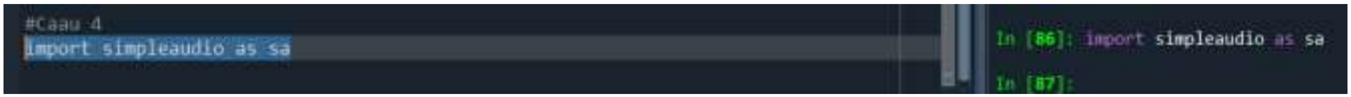
4)Thực hiện: theo hướng dẫn trong LINK

Vào trang https://realpython.com/playing-and-recording-sound-python/#playsound

- Cài đặt thư viện (nếu chưa có): mở CMD trên Windows, gõ pip install simpleaudio, cài xong chạy python -> gõ lệnh import simpleaudio as sa kiểm tra lỗi…



```
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

(multimedia) C:\Users\y0ns2>pip install simpleaudio
Requirement already satisfied: simpleaudio in c:\users\y0ns2\anaconda3\envs\multimedia\lib\site-packages (1.0.4)
```



```
#Caau 4
import simpleaudio as sa
```

```
In [86]: import simpleaudio as sa

In [87]:
```

- Vào trang https://file-examples.com/index.php/sample-audio-files/ download file mẫu Wav file và lưu trong D:\
- Mở notepad viết chương trình *.py lưu trong một thư mục{thường là chung với thư mục của file Wav)
- Chạy thử code : >python baitap1.py
- Lưu các bài tập trong thư mục, nén và nộp (cuối giờ thực hành)