## **Audio Formats**

- mp3
- flac
- wav







## **Audio Signal Parameters**

- number of channels
- sample width (Bytes)
- framerate / sample rate (44100 KHz)
- · number of frames
- · value of frames

```
In [2]:
         1 import wave
         1 obj = wave.open('test.wav','rb')
In [3]:
In [4]:
         1 obj
Out[4]: <wave.Wave_read at 0x7fb402147550>
In [6]:
        1 obj.getnchannels()
Out[6]: 2
        1 obj.getsampwidth()
In [7]:
Out[7]: 2
In [ ]:
In [8]:
        1 obj.getframerate()
Out[8]: 44100
In [9]: 1 obj.getnframes()
Out[9]: 308700
```

```
In [10]:
  1 obj.getparams()
Out[10]:
  _wave_params(nchannels=2, sampwidth=2, framerate=44100, nframes=308700, comptype='NONE', compname='not compressed')
In [11]:
  1 time audio = obj.getnframes()/obj.getframerate()
In [12]:
  1 time_audio
Out[12]: 7.0
In [14]:
  1 frames = obj.readframes(-1)
In [15]:
  1 frames
In [16]:
  1 len(frames)
Out[16]: 1234800
In [19]:
  1 (len(frames)/obj.getsampwidth())/obj.getnchannels()
Out[19]: 308700.0
In [20]:
  1 obj.getnframes()
Out[20]: 308700
In [21]:
  1 type(frames)
Out[21]: bytes
In [22]:
  1 type(frames[0])
Out[22]: int
In [23]:
  1 frames[0]
Out[23]: 0
In [25]:
  1 frames[40000]
Out[25]: 230
In [26]:
   new_obj = wave.open('new_audio.wav','wb')
  1
   new_obj.setframerate(44100)
   new_obj.setsampwidth(2)
   new_obj.setnchannels(2)
   new obj.writeframes(frames)
In [27]:
  1 new_obj.close()
```

```
In [29]:
          1 new_obj2 = wave.open('new_audio2.wav','wb')
             new_obj2.setframerate(32000)
           3
             new_obj2.setsampwidth(2)
           4 new_obj2.setnchannels(2)
             new_obj2.writeframes(frames)
             new_obj2.close()
In [30]:
          1 new_obj3 = wave.open('new_audio3.wav','wb')
             new_obj3.setframerate(64000)
             new_obj3.setsampwidth(2)
           4 new_obj3.setnchannels(2)
             new_obj3.writeframes(frames)
             new_obj3.close()
In [31]:
          1 new_obj4 = wave.open('new_audio4.wav','wb')
             new_obj4.setframerate(44100)
             new_obj4.setsampwidth(2)
           4 new_obj4.setnchannels(1)
           5 new_obj4.writeframes(frames)
           6 new_obj4.close()
In [32]:
          1 new_obj5 = wave.open('new_audio5.wav','wb')
           2 new_obj5.setframerate(44100)
             new_obj5.setsampwidth(1)
           4 new_obj5.setnchannels(2)
             new_obj5.writeframes(frames)
             new_obj5.close()
In [42]:
           1 import numpy as np
In [43]:
           1 signal_array = np.frombuffer(frames,dtype=np.int16)
In [44]:
          1 signal_array
Out[44]: array([
                               0, ..., 1421, 1475, 1457], dtype=int16)
In [45]:
          1 signal_array.shape
Out[45]: (617400,)
In [55]:
          1 import matplotlib.pyplot as plt
           2 %matplotlib inline
          1 plt.plot(signal_array)
In [56]:
Out[56]: [<matplotlib.lines.Line2D at 0x7fb3eb502bf0>]
            15000
            10000
             5000
                0
            -5000
```

300000

400000

500000

600000

-10000

-15000

0

100000

200000

```
In [57]: 1 plt.plot(signal_array[::2])
Out[57]: [<matplotlib.lines.Line2D at 0x7fb3eb3ab9d0>]
```

```
15000 -

10000 -

5000 -

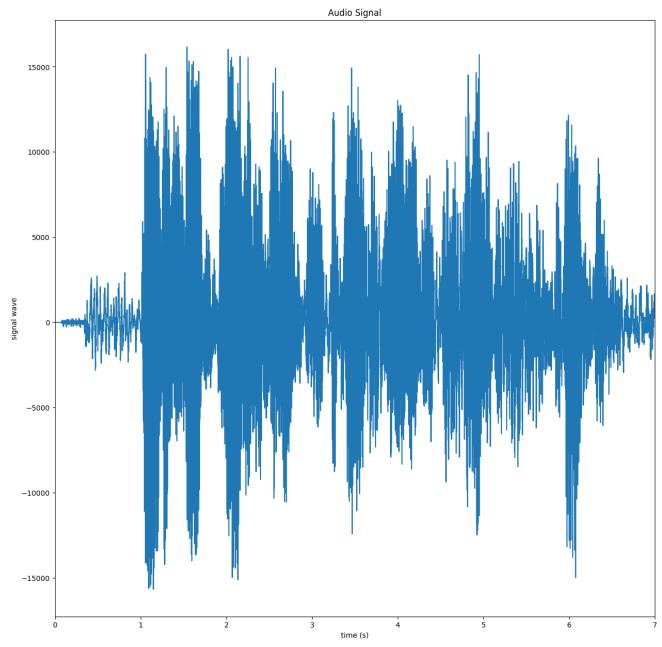
-5000 -

-15000 -

0 50000 100000 150000 200000 250000 300000
```

```
In [ ]:
          1
In [58]:
           1
             new_obj6 = wave.open('new_audio6.wav','wb')
             new_obj6.setframerate(44100)
             new_obj6.setsampwidth(2)
           3
             new_obj6.setnchannels(1)
           5 new_obj6.writeframes(signal_array[::2].tobytes())
           6 new_obj6.close()
In [59]:
           1 new_obj7 = wave.open('new_audio7.wav','wb')
           2 new_obj7.setframerate(22050)
             new_obj7.setsampwidth(2)
           4 new_obj7.setnchannels(1)
             new_obj7.writeframes(signal_array[::4].tobytes())
           6 new_obj7.close()
```

```
In [60]: 1     data = signal_array[::2]
2     times = np.linspace(0,time_audio,num=obj.getnframes())
3     plt.figure(figsize=(15,15))
4     plt.plot(times,data)
5     plt.title('Audio Signal')
6     plt.ylabel('signal wave')
7     plt.xlabel('time (s)')
8     plt.xlim(0,time_audio)
9     plt.show()
```



In [ ]: