

```

In [ ]: 1 # all imports
2 from io import BytesIO
3 from base64 import b64decode
4 from google.colab import output
5 from IPython.display import Javascript
6
7 RECORD = ""
8 const sleep = time => new Promise(resolve => setTimeout(resolve, time))
9 const b2text = blob => new Promise(resolve => {
10   const reader = new FileReader()
11   reader.onloadend = e => resolve(e.srcElement.result)
12   reader.readAsDataURL(blob)
13 })
14 var record = time => new Promise(async resolve => {
15   stream = await navigator.mediaDevices.getUserMedia({ audio: true })
16   recorder = new MediaRecorder(stream)
17   chunks = []
18   recorder.ondataavailable = e => chunks.push(e.data)
19   recorder.start()
20   await sleep(time)
21   recorder.onstop = async ()=>{
22     blob = new Blob(chunks)
23     text = await b2text(blob)
24     resolve(text)
25   }
26   recorder.stop()
27 })
28 ""
29
30 def record(sec=3):
31   print("Speak Now...")
32   display(Javascript(RECORD))
33   sec += 1
34   s = output.eval_js('record(%d)' % (sec*1000))
35   print("Done Recording !")
36   b = b64decode(s.split(',')[1])
37   return b #byte stream

```

```

In [ ]: 1 voice_bytes = record()

```

Speak Now...

<IPython.core.display.Javascript object>

Done Recording !

```

In [ ]: 1 voice_bytes

```

```

84\xe4U\xd3\xc3B\x00yb\xt7\xa5j=\xd7\x90\xb3P\x1b\xca\x00&\xee-/ak2\xt5\x06\x8t\x81\xea\x9e)?<\x8929\x8a+dx0e\xbt\x12\\\xet
\x22^\x94<Mvx\x19\x9bi\x0fw\xa3A0\x81\x00;\x80\xfb\x83^`R\xeb\x81)\xb5\xd5\xd4\xb7\x0b\x1a\x92\xfb\xef\x1a^\xa2\xebI\xb0\xee
\x89\x18\xe3\x17\xbb*\x0f,\xdf\xcd\xa1z%e&\xa8\x12\xb2\x8aY\x06\x97;:A\x9c\xeb\xfb07\xa2\xb2\x00\x96\xd9\x89\xad\n`_s\xc5\xa7
\xbb\xbf\xe4\x96\x89\xbf\x91\x1c\xb2o\x906?\xceP\xd97\xff\xfb\xab\x9a` \xed\xe2\x19^R\xa4\x0f\xdeRA\xbf\x81\xa5D\xbe55.\xc0\x
0c\x02\xe0\xdb\x8c\xbf\x18rShl\xeb\xcb\x1c\xg\xcd\x8d\xcd\x1\x05\x00v{f\x06@\xf5x%\xac\x010ft\xe5N\xa2\x8d\x15\x05Nni\x1f\x86\t
\xb7\xd5\xe3\xa8\xe7\xc8\x07\x8f\xb2o\x07\x02\xde\x02dm\x08z'\x13\xfd'Ly\xe8\xa2\x13K\x7fF\xfb\xfb\xdd$_\x01\xbd(Q\xe2\x0B
\xee\xe4)Ft\x15\x93f\x06\x16\x89\xfb\x06\xfa\x95\x1f\x02T\xb6o\x7f\x8b\xbb]\x86\x99\x9bB\xfa\x06\x0b\xbe"\xe3\xa6\xb3\xfb\xdb
\xcb70\xc85\xfb\xcf\x8cBQ\xfb6\x07\x07\x9fD\x1c\xa9\xfb2\x87.\xe9\xac|\rN\xcd@MX\x8c\xa9I\xeaUA\xaa\x043Xo\x15\xcf\x15K\x832\x9
3\xee\xfb4Y\xa5\xb4a\x03Z\x9b\xfb\x05}iB\xbd\xfb5\xfb6\xa3A\x83\x81\x00x\x80\xfb\x03P\xd3\xde\x03\xcb\xa0?Y\xec\x05\xcb\xcf\x0b1
\tp\xda\xfbf\x80\x89\xbb\x87M\x0fgG=\xect\x09\x199m=\xa3\xef\xa6\x89\x0b2L#\xd6\x02'\xf2K0d\xa4nX\xe6\xa3\x01\xdb<\xe4f[o0\x
a7\x7f\x7fgU1\x07\xfb9g\x0bj\x09e\x0d\xa5\xbe\x03V\x00*\x95\x82\x91\x96B\xac\x19j\xa8=(1C\x0dG\xd47\x0411\x0e\x02\x07\xfb2@\x9
6W:\xd8\xeb<\n\x05\xcfMh4\xfb1dn\x06Sn'\@\x05\x03\x83\xfb2.0U#4+\x08\x82T3G6\xa2\xfb1z\xa7\x18Bw]\x03\x03\xfb7\x1bz\x81^K\xfb1Ii\x
90\xb8\xb6\xee\x1f%g\x8e\x89\x03\xab\xfb8\x19\xe1\x9c\x90\x8f+\xed\x05\x06\xcdby\xa38\xfb9\t\x02\x07Ce\xfbf~\x0d\x1f\xfb6%\xe6=V
\xfb43\x0b\x8eft\xae\x0b\xa4\x08!\xc7\x04{\x03\x84>d \xe3\x03\x83\x02\x0b\x06^Zb\x05hu\x01\x04\x01\xae\x17\xeb\x80W\xae\x90\x0c
8^o8w\x06\x10r\x81DJ[#\xf0\n\x1d\x06v\xed\xa0\x95\xe9\xfb3\x03wi\xa3\xec\x0b9\x06\xe8{\xecwt\x07\xfb4\xfb\x05\xaa\x9a\x06\x1e\x
a9\xae\xcc\x02\x83A\xfb.[*\x80\x9a\x0b\x98e\x01\x08\xfb\x9d\xdd\x0b\x01\x0c\x9b\xacK\xfb0Y\x02\x06\x9d\xa7%0\x18\x84T\x06\xaa
v\xa3\x0d9#\xe1\xebq[\x91\x0d\x03\x00r\xcb>\xa1\xa51\xe0\x06\x07\xfb7\xfb\x03\x06\xcd\x0b9\xbbp\xcb\x02\xddG\xcc\xae\x9f\xabTg
\xfb7\x07\x91\x08\x0eT\xfb8/\xab\x85\xfb*\xfb\xa3A\x83\x81\x00\x0b3\x00\xfb\x03;\x94&\xba\x96A\xfbf\x19\xa3 \xe1\xa4^5\x0bd%\x06c#
+11'\xb4\xa0\xa4\xfb8\xdb\xfbf\x83\x06U\x00\x1e p|Qg\xa0/\xef\x0d\x1c5\xef\xed\xfb9Q _m\xbd\xfb1\xa15\xcePR/B\x10\x0e\xac\xcf\x0d
9\x00]\xeb\x07\xbd\x044Q\x9d(x\xcdJ\x06\xac\x1e\xfd\xa7{!8zd\x08i\x1f\x02|\x12\x03DP\xfc\xfb7F5J9\x03\x0e1\x91T|\xc3\x06\x0b\x

```

```

In [ ]: 1 import numpy as np

```

```

In [ ]: 1 import matplotlib.pyplot as plt
2 %matplotlib inline

```

```

In [ ]: 1 len(voice_bytes)

```

Out[30]: 24956

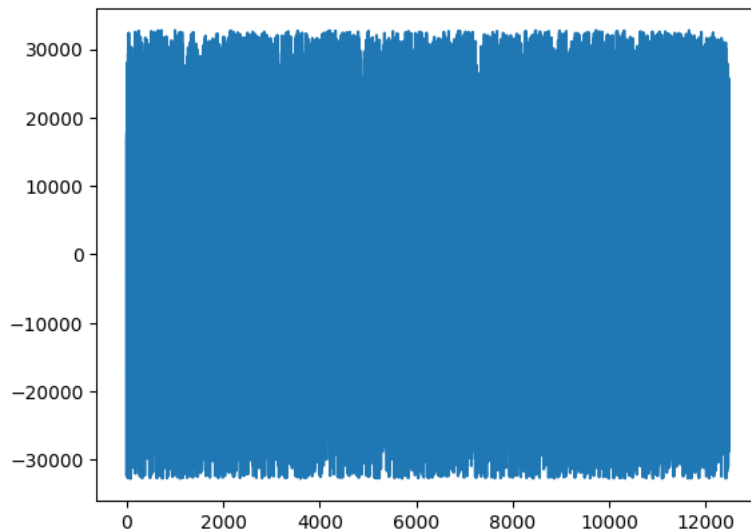
```
In [ ]: 1 np.frombuffer(voice_bytes, dtype=np.int16)
```

```
Out[28]: array([ 17690, -23585, 17055, ..., 25736, 1075, 14092], dtype=int16)
```

```
In [ ]: 1 len(np.frombuffer(voice_bytes, dtype=np.int16))
```

```
Out[36]: 12478
```

```
In [ ]: 1 plt.plot(np.frombuffer(voice_bytes, dtype=np.int16))  
2 plt.show()
```



```
In [ ]: 1 import wave
```

```
In [ ]: 1 obj = wave.open('rec.wav', 'wb')
```

```
In [ ]: 1 framerate = len(np.frombuffer(voice_bytes, dtype=np.int16))/3
```

```
In [ ]: 1 obj.setsampwidth(2)  
2 obj.setnchannels(1)  
3 obj.setnframes(len(np.frombuffer(voice_bytes, dtype=np.int16)))  
4 obj.setframerate(framerate)
```

```
In [ ]: 1 obj.writeframes(voice_bytes)
```

```
In [ ]: 1 obj.close()
```

```
In [ ]: 1 !pip install ffmpeg-python
```

```
Looking in indexes: https://pypi.org/simple, (https://pypi.org/simple,) https://us-python.pkg.dev/colab-wheels/public/simple/  
(https://us-python.pkg.dev/colab-wheels/public/simple/)  
Collecting ffmpeg-python  
  Downloading ffmpeg_python-0.2.0-py3-none-any.whl (25 kB)  
Requirement already satisfied: future in /usr/local/lib/python3.10/dist-packages (from ffmpeg-python) (0.18.3)  
Installing collected packages: ffmpeg-python  
Successfully installed ffmpeg-python-0.2.0
```



```

In [ ]: 1 from IPython.display import HTML, Audio
        2 from google.colab.output import eval_js
        3 from base64 import b64decode
        4 import numpy as np
        5 from scipy.io.wavfile import read as wav_read
        6 import io
        7 import ffmpeg
        8
        9 AUDIO_HTML = """
10 <script>
11 var my_div = document.createElement("DIV");
12 var my_p = document.createElement("P");
13 var my_btn = document.createElement("BUTTON");
14 var t = document.createTextNode("Press to start recording");
15
16 my_btn.appendChild(t);
17 //my_p.appendChild(my_btn);
18 my_div.appendChild(my_btn);
19 document.body.appendChild(my_div);
20
21 var base64data = 0;
22 var reader;
23 var recorder, gumStream;
24 var recordButton = my_btn;
25
26 var handleSuccess = function(stream) {
27     gumStream = stream;
28     var options = {
29         //bitsPerSecond: 8000, //chrome seems to ignore, always 48k
30         mimeType : 'audio/webm;codecs=opus'
31         //mimeType : 'audio/webm;codecs=pcm'
32     };
33     //recorder = new MediaRecorder(stream, options);
34     recorder = new MediaRecorder(stream);
35     recorder.ondataavailable = function(e) {
36         var url = URL.createObjectURL(e.data);
37         var preview = document.createElement('audio');
38         preview.controls = true;
39         preview.src = url;
40         document.body.appendChild(preview);
41
42         reader = new FileReader();
43         reader.readAsDataURL(e.data);
44         reader.onloadend = function() {
45             base64data = reader.result;
46             //console.log("Inside FileReader:" + base64data);
47         }
48     };
49     recorder.start();
50 };
51
52 recordButton.innerText = "Recording... press to stop";
53
54 navigator.mediaDevices.getUserMedia({audio: true}).then(handleSuccess);
55
56
57 function toggleRecording() {
58     if (recorder && recorder.state == "recording") {
59         recorder.stop();
60         gumStream.getAudioTracks()[0].stop();
61         recordButton.innerText = "Saving the recording... pls wait!"
62     }
63 }
64
65 // https://stackoverflow.com/a/951057
66 function sleep(ms) {
67     return new Promise(resolve => setTimeout(resolve, ms));
68 }
69
70 var data = new Promise(resolve=>{
71     //recordButton.addEventListener("click", toggleRecording);
72     recordButton.onclick = ()=>{
73         toggleRecording()
74
75         sleep(2000).then(() => {
76             // wait 2000ms for the data to be available...
77             // ideally this should use something like await...
78             //console.log("Inside data:" + base64data)
79             resolve(base64data.toString())
80         });
81     });
82 }
83 }
84 });
85
86 </script>

```

```

87 """
88
89 def get_audio():
90     display(HTML(AUDIO_HTML))
91     data = eval_js("data")
92     binary = b64decode(data.split(',')[1])
93
94     process = (ffmpeg
95         .input('pipe:0')
96         .output('pipe:1', format='wav')
97         .run_async(pipe_stdin=True, pipe_stdout=True, pipe_stderr=True, quiet=True, overwrite_output=True)
98     )
99     output, err = process.communicate(input=binary)
100
101     riff_chunk_size = len(output) - 8
102     # Break up the chunk size into four bytes, held in b.
103     q = riff_chunk_size
104     b = []
105     for i in range(4):
106         q, r = divmod(q, 256)
107         b.append(r)
108
109     # Replace bytes 4:8 in proc.stdout with the actual size of the RIFF chunk.
110     riff = output[:4] + bytes(b) + output[8:]
111
112     sr, audio = wav_read(io.BytesIO(riff))
113
114     return audio, sr

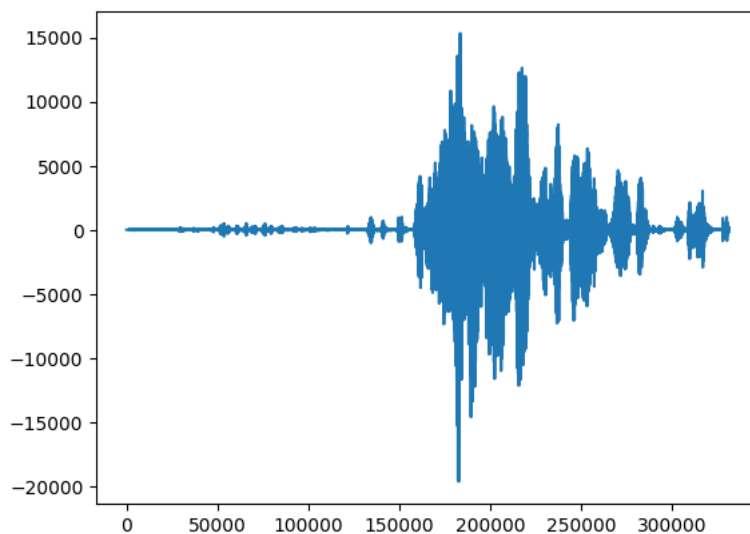
```

```
In [ ]: 1 audio, sr = get_audio()
```

```
In [ ]: 1 audio
```

```
Out[44]: array([0, 0, 0, ..., 0, 2, 1], dtype=int16)
```

```
In [ ]: 1 plt.plot(audio)
2 plt.show()
```



```
In [ ]: 1 sr
```

```
Out[45]: 48000
```

```
In [ ]: 1
```

```
In [ ]: 1 import wave
```

```
In [ ]: 1 obj = wave.open('rec.wav', 'wb')
```

```
In [ ]: 1 framerate = sr
```

```
In [ ]: 1 obj.setsampwidth(2)
        2 obj.setnchannels(1)
        3 obj.setnframes(len(audio))
        4 obj.setframerate(framerate)
```

```
In [ ]: 1 obj.writeframes(audio.tobytes())
```

```
In [ ]: 1 obj.close()
```

MP3

```
In [ ]: 1 !pip install pydub
```

Looking in indexes: <https://pypi.org/simple>, (<https://pypi.org/simple>,) <https://us-python.pkg.dev/colab-wheels/public/simple/> (<https://us-python.pkg.dev/colab-wheels/public/simple/>)
Collecting pydub
 Downloading pydub-0.25.1-py2.py3-none-any.whl (32 kB)
Installing collected packages: pydub
Successfully installed pydub-0.25.1

```
In [ ]: 1 from pydub import AudioSegment
```

```
In [ ]: 1 audio = AudioSegment.from_wav('rec.wav')
```

```
In [ ]: 1 audio = audio +6 #dB
```

```
In [ ]: 1 audio.fade_in(2000)
```

Out[73]:
0:00 / 0:00

```
In [ ]: 1 audio.export('rec.mp3', format='mp3')
```

Out[74]: <_io.BufferedReader name='rec.mp3'>

```
In [ ]: 1 audio2 = AudioSegment.from_mp3('rec.mp3')
```

```
In [ ]: 1 audio2
```

Out[76]:
0:00 / 0:00

```
In [ ]: 1 len(audio2)
```

Out[78]: 6900

```
In [ ]: 1
```