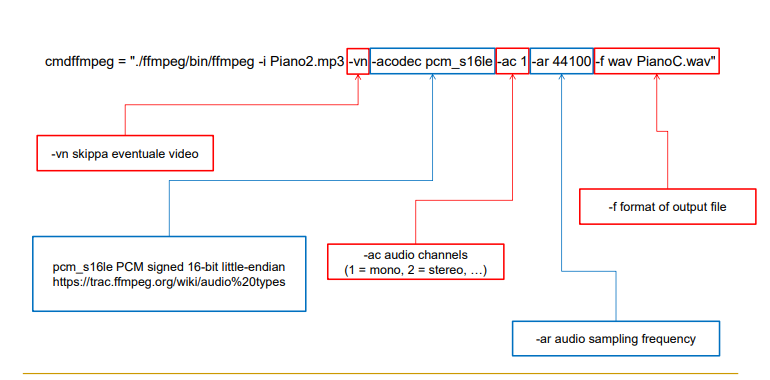
# Audio Processing SCRIPT

## Conversione da MP3 a WAV



Questa stringa da dare in input al nostro script permette di convertire il nostro mp3 in wav

* **-vn** skippa il video
* **-adonec pcm\_s16le** indica di usare una codifica PCM con segno little endian
* **-ac 1** indica di usare un solo canale audio
* **-ar 44100** indica di impostare la frequenza di campionamento a 44100 Hz
* **-f wav “nome.wav”** indica di usare come output un wav ( e si specifica il nome)

**IL FLAG USATO PER INDICARE IL NUMERO DI CANALI DI UN FILE AUDIO DA CONVERTIRE E’ -ac**

**IL FLAG USATO PER INDICARE LA FREQUENZA DI CAMPIONAMENTO DI UN FILE AUDIO DA CONVERTIRE E’ -ar**

Questa stringa la usiamo con una chiamata disponibile grazie alla libreria subprocess  
**LA LIBRERIA SUBPROCESS PERMETTE DI LANCIARE COMANDI SU SHELL  
LA FUNZIONE CHE USIAMO PER LANCIARE IL COMANDO cmdffmpeg E’ sp.call(cmdffmpeg)**

Potremmo aver bisogno di gestire path e in generale interagire col file system.  
**LA LIBRERIA OS PERMETTE DI GESTIRE IL FILE SYSTEM**

## Lettura di un wav

**LA LIBRERIA CHE SERVE A LEGGERE I FILE WAV E’ scipy.io  
nella domanda viene data come possibilità wave che è errata. La risposta corretta alla domanda è “nessuna delle precedenti”**samplerate, data = wavfile.read("PianoC.wav")  
**OTTENIAMO IL SAMPLERATE E I DATI.**  
**LA FUNZIONE CHE CI PERMETTE DI LEGGERE I FILE WAV E’ wavfile.read(“nome,wav”)**

## Fast Fourier Transform

from scipy.fftpack import fft,fftfreq # Libreria per calcolare la FFT

datafft = fft(data) # Restituisce un numero complesso (parte reale e immaginaria)

fftabs = abs(datafft) # Calcoliamo la magnitudine = sqrt(real+imag)

freqs = fftfreq(data.shape[0],1./samplerate)

**fftabs contiene le intensità (y).  
Le frequenze (x) stanno nella variabile freqs**

**LA LIBRERIA DA IMPORTARE E’ scypy.fftpack**

## VISUALIZZAZIONE

**LA LIBRERIA CHE USIAMO PER I GRAFICI E’ matplotlib.pyplot**

## ELABORAZIONE TRACCE

**LA LIBRERIA PER FARE ELABORAZIONE TRACCE E’ scipy.fftpack**

**(Da cui importiamo ifft)**