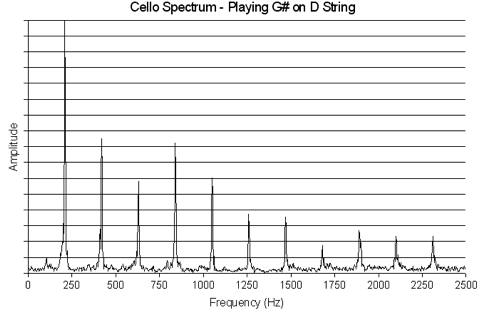
In this tutorial, we will implement a pitch shifting using DFT.

The relation between pitch and frequency is not straightforward: while the frequency is an objective value that can be measured, the pitch is a subjective perception of sound. However, a *definite* pitch, i.e. a pitch that can be easily discerned, has a harmonic frequency spectra: what we hear is a *fundamental* frequency, and all other frequencies present in the signal are *partials*, being multiples of the fundamental one, and serve to give a particular timbre. Here is an example of a G# played on cello:



Luckily, we’re only interested in shifting the definite pitches of an input signal, as they are the only ones we can perceive. In a speech signal, definite pitches are vowels and voiced consonants, so ideally we would like to shift their pitches and nothing else. However, it would be too complicated to extract them from the input signal; it is easier to shift the pitch of every phonetic one by one, as shifting the pitch of an unvoiced consonant wouldn’t change much. Instead of parsing the input speech to phonetics, we simply separate the signal into intervals small enough: this way, every processed part of signal contains no more than one consonant or vowel.

To keep the definite pitch characteristics, the shifted pitch sould still have a fundamental frequency and its partials. It means that we can’t simply shift all frequencies of the pitch, but we should strecth or compress them, so that the partials are still multiples of the fundamental frequency.