

Sine wave(소리)

→ theta 공식 + 시간(원래는 시간 없이도 파형은 그릴 수 있으나, 그러나 소리 형성 불가능)

```
ipd.Audio(c.real, rate=sr)
```

말고 소리 내는 다른 방법

!pip install sounddevice <- (sounddevice library가 깔려있지 않는 경우, 우선 설치한다)

```
import sounddevice as sd
```

```
sd.play(c.real, sr)
```

→ run하면 관련 소리 발생

Sine wave 그릴 때 어떻게 **amp** 사용?

s = amp*np.sin(theta) (<- phasor 앞에다가 amp를 넣어서 곱)

e.g. amp = 2 면? y축이 -2~2로 바뀐다

3차원에서 실시? 원 반지름 = amp, 소리 역시도 증가(amplitude 가 음세기이므로)

Sr, freq는 연관성이 있다 (둘다 Hz이기도 하고)

e.g. Sr = 100hz 가정(표현할 수 있는 숫자가 100개라는 의미)

freq = 1hz 표현 가능(1초에 1개의 sine wave)

freq = 2hz 표현 가능(n분의 1[여기서 n은 freq의 hz] 해서 2개의 sine wave)

freq = 10000hz 표현 불가능 (10000개의 sine wave를 100개의 숫자로 표현x, 숫자가 너무 적음)

sr= 10hz 일 때, 최대 그릴 수 있는 freq는 5번

→ 표현가능한 주파수(freq)는 sr의 반 이하

Nyquist frequency = sr/2 (sr의 반 되는 freq)

F0 = 100; (이것의 harmonics는 어디가 끝인가? sr의 반까지가 끝);

Fend(F의 끝) = int(sr/2[=Nyquist freq]) -> int로 반올림 정수 처리;

s = np.zeros(len(t)); (s는 가장 초기값이다 라는 것을 표현,

t 벡터의 개수 만큼 0만들어 줘라)

for freq in range(F0, Fend+1, F0): (F0부터 끝까지 짝짝 올라간다,

+1은 제일 마지막 꺼까지 포함해서)

/ 100, 200, 300, ... 5000Hz (50번 돈다)

theta = t * 2*np.pi * freq

tmp = amp * np.sin(theta)

s = s + tmp (계속 끝까지 더하고 더해라 -> 수시 업뎃)

[or s = s + amp * np.sin(theta) 라고 표현해도 됨]

fig = plt.figure()

ax = fig.add_subplot(111)

ax.plot(t[0:1000], s[0:1000]);

ax.set_xlabel('time (msec)')

import sounddevice as sd (내 컴퓨터를 위해서)

sd.play(c.real, sr)

➔ 선 하나에 0000, 선 하나에 0000...

1~5000까지 다 더했을 때 나온 그래프

Wave(Sine) + 여러가지 sine wave 합쳐짐 -> 모두 더하면? Pulse Train의 모양

X축 time, Y축 value / Waveform

X축 time, Y축 frequency /Spectrogram

한 슬라이스 절단, 어떤 frequency 성분이 많은가?

➔ X축 frequency, y축 amplitude / spectrum

100hz~5000hz까지 있을 수 있음(정수배로)

Formant / Peaks and valleys 가 있는 모양, 성대 이후 조음되어서 filtering 된 상황

Parameter setting

amp = 1 / sr = 10000

dur = 0.5 / freq = 440.0

[모든 필요한 library들을 import했다는 전제로]

t = np.arange(1, sr * dur+1)/ sr

theta = t* 2*np.pi * freq

s = amp*np.sin(theta)

fig = plt. figure()

ax = fig.add_subplot(111)

ax.set_xlabel('time(s)')

ax.set_ylabel('real')

➔ 440hz짜리 sine wave 형성

/ freq 변환 -> 880hz로 -> play하면 440hz와 똑같이 "라" 소리, 음높이의 차이뿐

(옥타브로 올릴려면 배수로 곱하면 된다)

s = amp*np.cos(theta) <- cosine 으로 변환

freq 440hz으로 다시 바꾸고

소리는? Cos, sin의 "라" 소리 변함이 없음

Cos -> sin으로 옮기려면 몇 초 정도? $\pi/2$ (90도) 차이

$\pi/8$ 옮겨도 소리차이? Nope! 소리 차이는 존재하지 x

phase에는 전혀 sensitive하지 x, 인식하지 못함

Pulse Train

Complex tone -> 우선 gradually decreasing 모양(성대 모양) -> 후에 산맥처럼 깎기(formants)