

12기 음성 프로젝트

ToBig's 12기 박진혁

# Sound Data Analysis

# Contents

---

Unit 01 | 소리의 표현

---

Unit 02 | 소리를 합치고 분리해보자 – 소리 처리

---

Unit 03 | 푸리에 변환 (중요!!)

---

Unit 04 | 푸리에 변환을 통한 소리 데이터의 처리

---

## Unit 01 | 소리의 표현

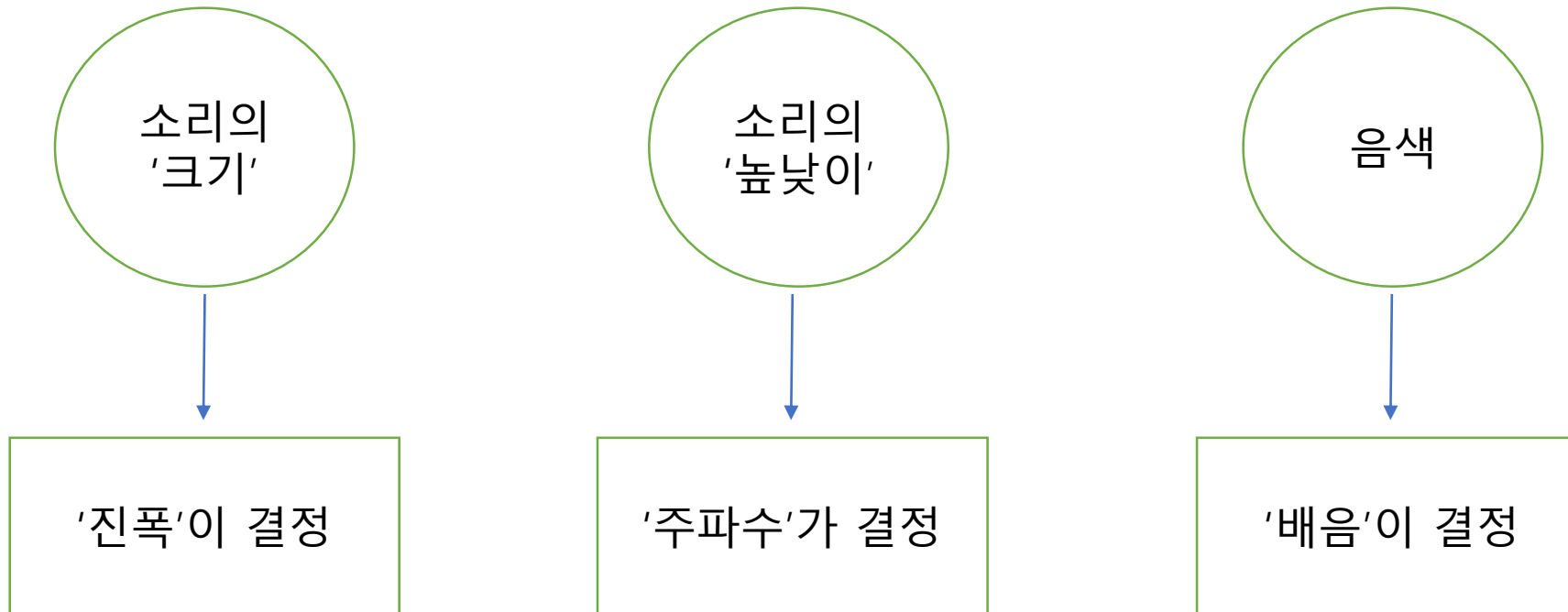
- 우선 소리를 그래프로 나타내면 다음과 같은 그림이 떠오를 것이다. 이러한 그림에서 가로축은 '시간'이고 세로축은 '진폭'이다.



뒤에서 자세하게 설명할 것이므로 일단 가로축이 '시간' 세로축이 '진폭'인것만 확실히 알아두자!!

## Unit 01 | 소리의 표현

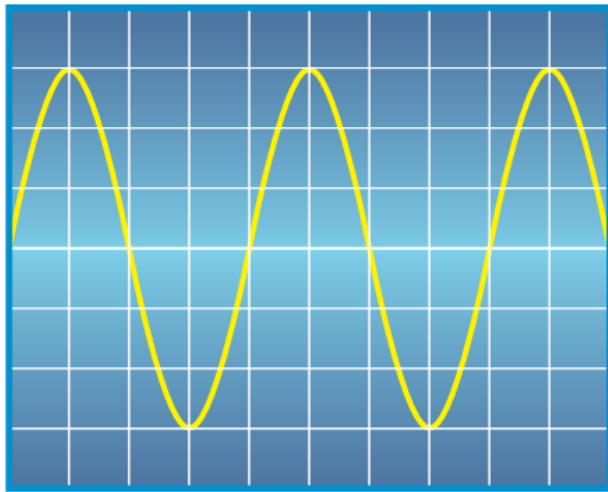
다른 소리를 구별하는 3가지 요소는 크기, 높낮이, 음색이 있고  
이 것은 다음과 같은 요소에 의해 결정된다. 하나하나씩 알아보자!



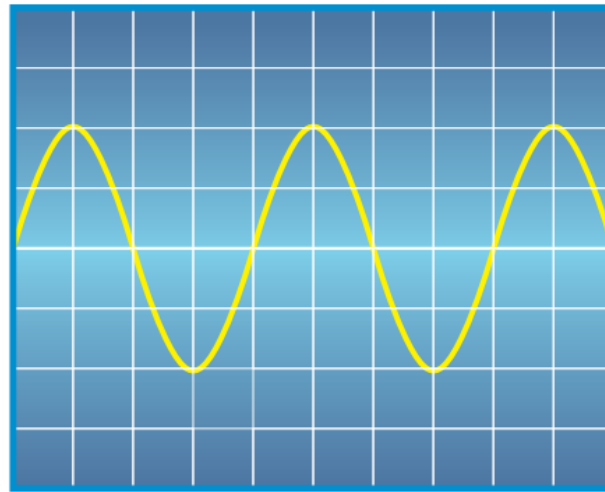
## Unit 01 | 소리의 표현

## ■ 소리를 구별하는 요소1. 소리의 크기

소리의 크기는 단순히 진폭에 영향을 받는다. 즉 저 y축 값이 크면 클수록 **큰 소리**라고 할 수 있다. 단위는 우리가 흔히 아는 db로 표현한다.



큰 소리



작은 소리

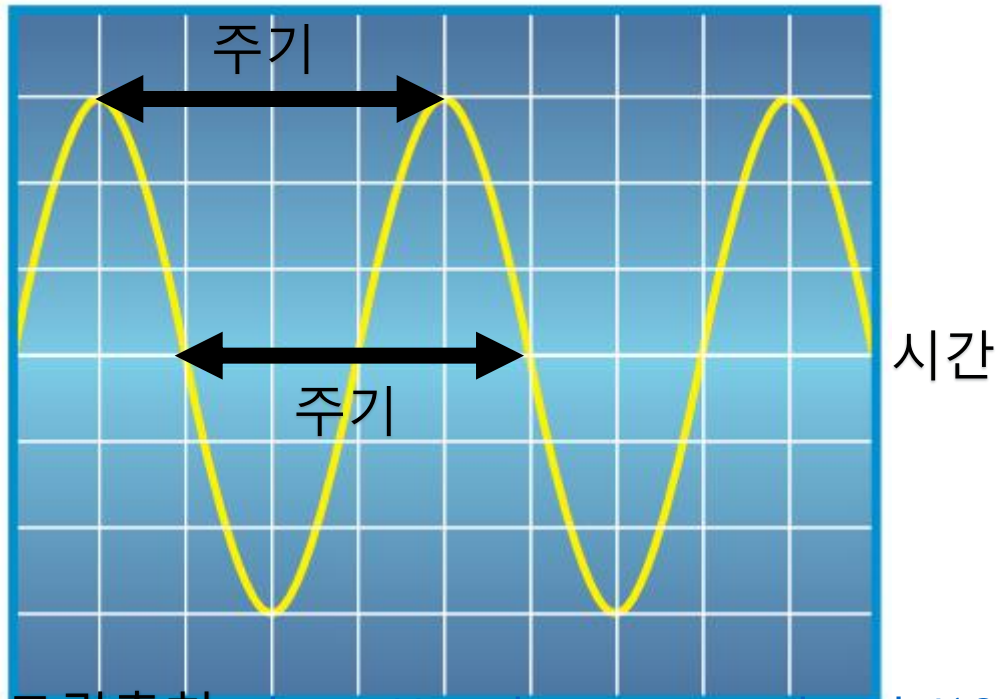
그림출처 : <http://study.zum.com/book/12653>

## Unit 01 | 소리의 표현

## ■ 소리를 구별하는 요소2. 음의 높낮이 (음정)

우리가 흔히 말하는 고음 저음을 말하는 음정은 소리의 '주파수'에 영향을 받는다. 이 주파수는 소리 분석의 핵심이 될 것이다! 다음과 같은 음파가 있다고 해보자.

진폭



시간

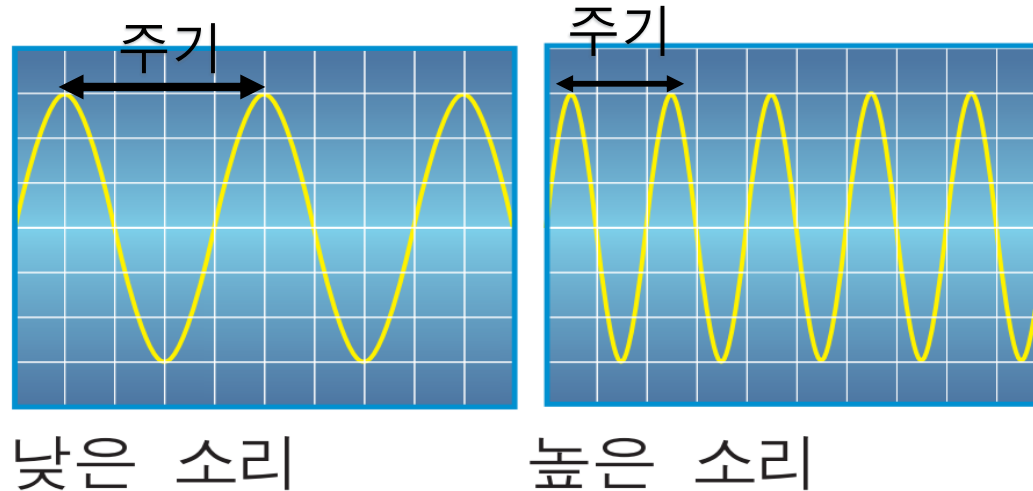
**주기** : 소리는 진폭의 변화라고 했는데, 진폭이 커졌다 작아졌다 하면서 반복되는데 걸리는 시간이 '주기' 이다. 옆에 그림에서 가로축이 시간이라고 생각한다면 주기가 무엇인지 바로 이해가 될 것이다.

**주파수** : 주파수는 1초동안 총 몇주기가 도는지를 나타내는 값이며 Hz가 단위이다. 즉 4초마다 한 주기가 돈다면, 1초에 0.25번 도는 것이므로 0.25Hz, 0.2초마다 한 주기가 돈다면 5Hz가 되는 것이다. 즉 주파수 =  $1/\text{주기}$  이다.

그림출처 : <http://study.zum.com/book/12653>

## Unit 01 | 소리의 표현

- 소리를 구별하는 요소2. 음의 높낮이 (음정)



그림출처 : <http://study.zum.com/book/12653>

즉 옆에 그림에서보면, 왼쪽 그림은 원래 위치로 돌아오는데 시간이 오래 걸려서 오른쪽 그림보다 주기가 길다. 따라서 왼쪽 그림이 오른쪽보다 주파수가 낮을 것이고, (같은시간동안 훨씬 더 적게 반복되므로) 왼쪽 소리가 더 낮은 소리가 된다!

## Unit 01 | 소리의 표현

## ■ 소리를 구별하는 요소 2. 음의 높낮이 (음정)

T.M.I) 우리가 일반적으로 생각하는 2옥타브 '라'는 440Hz이다.

T.M.I 2) 소리의 높낮이는 log스케일이라고 생각하면 되는데, 옥타브가 하나 올라가면 주파수가 두배 높아진다. 즉 1옥타브 '라'는 220Hz이고 3옥타브 '라'는 880Hz이다. 그리고 검은건반을 포함하면 한 옥타브에 음은 12개가 있는데 한 음간의 차이는  $2^{(1/12)}$ 배만큼 차이가 난다!

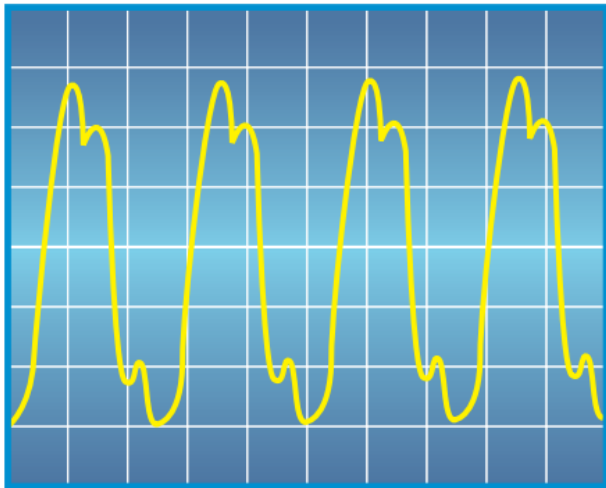
즉 '도'와 '파#'은 6음차이가 나므로 '파#'이 '도'보다  $2^{(6/12)}$ , 즉 루트2배의 진동수를 가지고 있다고 할 수 있다.



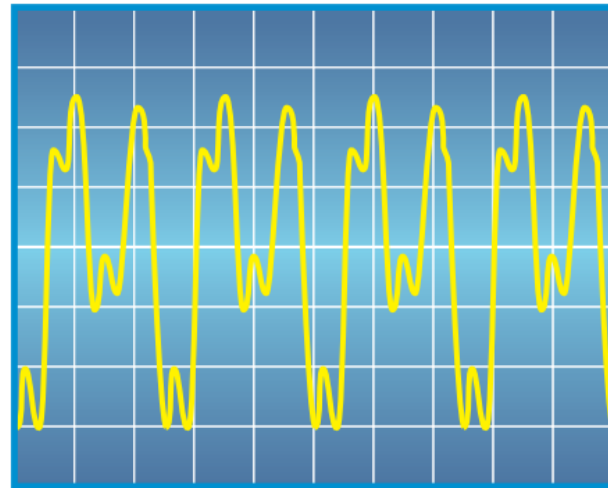
## Unit 01 | 소리의 표현

## ■ 소리를 구별하는 요소3. 음색

그런데 피아노로 내는 '도' 랑 바이올린으로 내는 '도'는 분명 느낌이 다르다. 차이는 바로 음색이다. 같은 주파수라도 이 '파형'이 다르면 음색이 다른데, 이 부분과 주파수는 굉장히 중요한 역할을 하므로 뒤에 푸리에 변환 등에서 더 자세히 설명하겠다.



피아노 소리



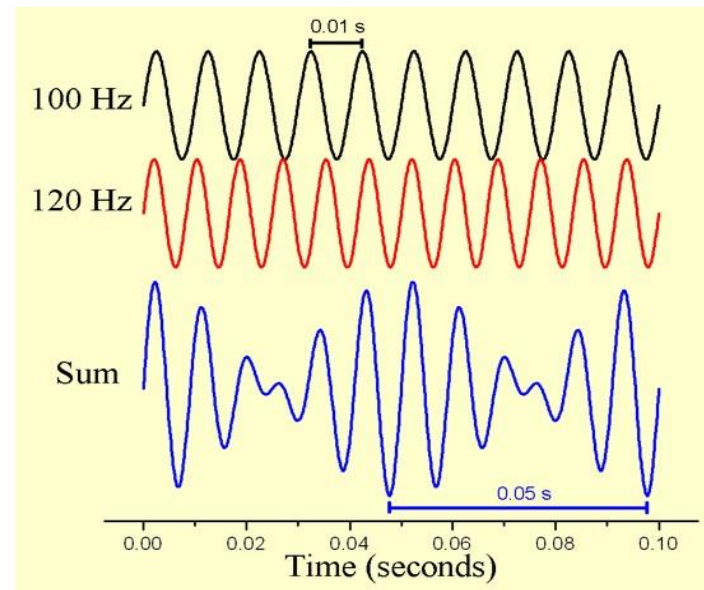
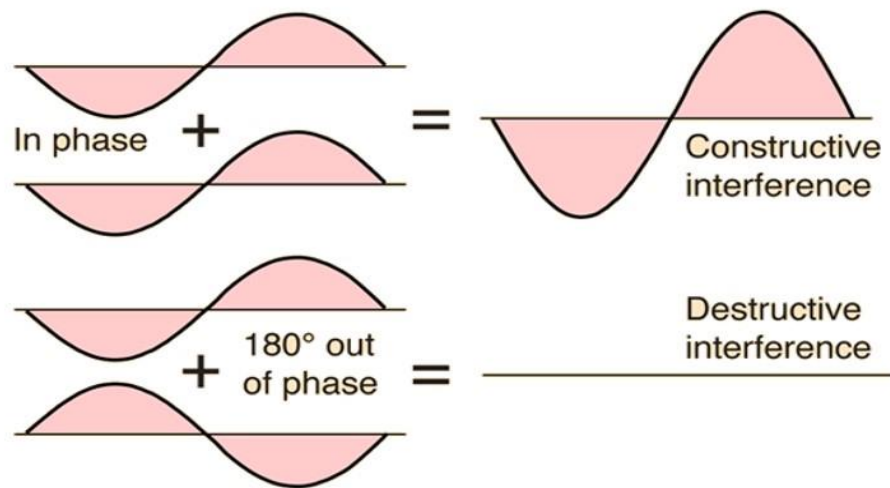
바이올린 소리

그림출처 : <http://study.zum.com/book/12653>

## Unit 02 | 소리를 합치고 분리해보자!

## ■ 그렇다면 잡음을 제거하고, 소리를 합치고 분리는 어떻게 할까?

사실 이 부분은 굉장히 쉽다. 그냥 진폭끼리 더하면 새로운 소리가 되며, 섞인 소리에서 빼고자 하는 소리의 음파의 진폭을 빼주면 소리가 빠진다.



사진출처  
: [http://fluorf.net/lectures/lectures3\\_2.htm](http://fluorf.net/lectures/lectures3_2.htm)

위 그림을 보면, 같은 방향이면 오히려 소리가 두 배 커지고 다른 방향이면, 소리가 상쇄돼서 사라지는 것을 볼 수 있다. 오른쪽 그림을 보면 파란색에서 빨간색을 빼면 다시 검은 소리를 얻을 수 있다. 즉 음원소리 에서 MR 소리를 단순히 빼면 보컬 소리만 남길 수 있는 것이다

## Unit 02 | 소리를 합치고 분리해보자!

- 그렇다면 잡음을 제거하고, 소리를 합치고 분리하는 어떻게할까?

TMI) 노이즈 캔슬링도 이러한 원리를 사용하는데, 들어오는 진폭의 반대 진폭을 만들어서 상쇄시켜서 소음을 제거시키는 원리이다!

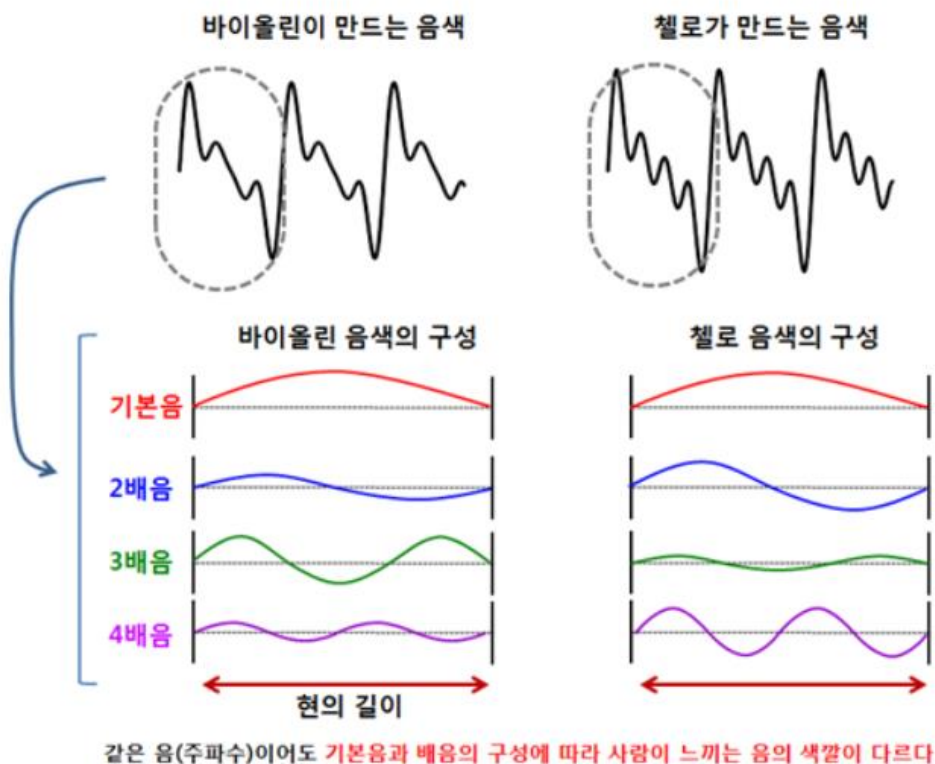
## Unit 03 | 푸리에 변환

- 그. 령. 다. 면 소리는 어떻게 처리할 수 있을까?

보통의 소리의 분리나 음역대별로 소리를 나눌 때에는 **주파수를 기준으로** 나눈다. 아까도 얘기했듯, 주파수는 '음정'을 결정하는 요소이기도 해서 굉장히 중요한 요소라고 할 수 있으며 소리 분석은 기존에 생각하던 음파가 아닌 주파수를 기준으로 분석하고 쪼개고 보정한다.

## Unit 03 | 푸리에 변환

- 그. 령. 다. 면 소리는 어떻게 처리할 수 있을까?

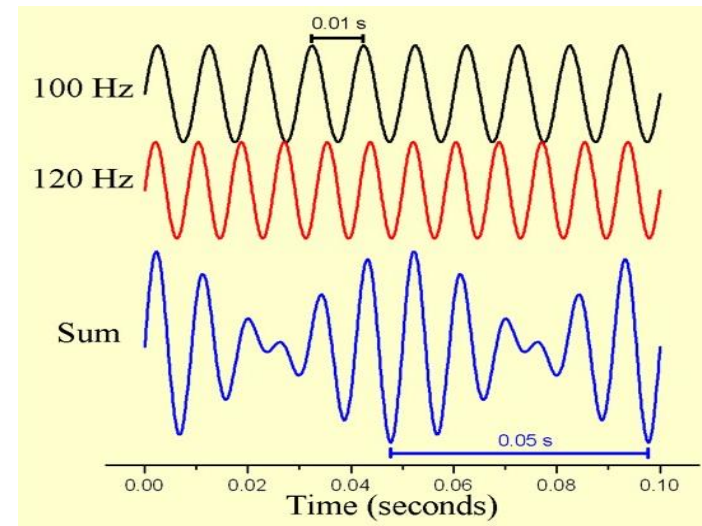
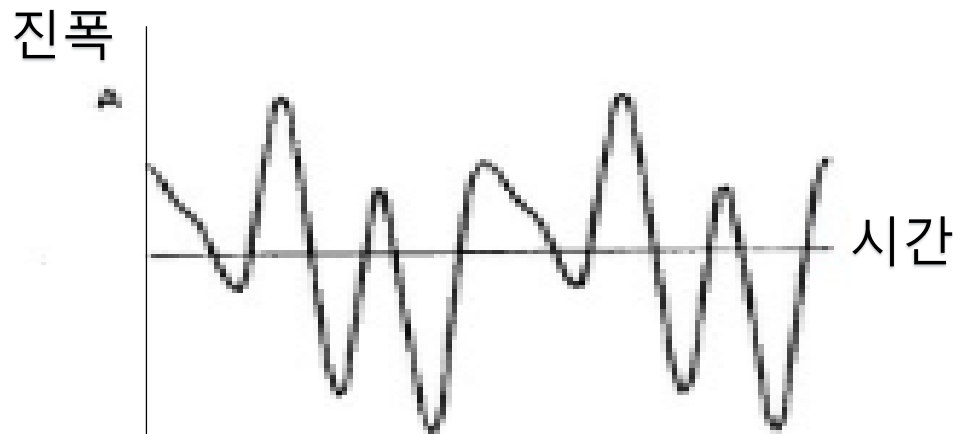


아까 음색부분에서 나왔는데, 같은 주파수라도 소리가 다르게 들리는 이유는 그 주파수의 두배음, 세배음 등의 값의 세기가 미묘하게 다르기 때문이다. 즉 우리는 소리를 주파수로 분리해야 처리할 수 있다.

\* 어떤 신호의 주파수의 두배 세배 네배(정수배)의 주파수가 더해지더라도 주파수는 제일 작은 기본 주파수와 같아서 음의 높낮이는 같다.

## Unit 03 | 푸리에 변환

- 그. 령. 다. 면 소리는 어떻게 처리할 수 있을까?



그런데 이 그림을 보고 주파수로 쪼갤 수 있을까? 아까 말했듯 소리는 섞이면 그냥 파형이 더해진다. 오른쪽을 보면 더하는 건 쉬워보이는데 분리하는 엄청 어려워 보인다.

## Unit 03 | 푸리에 변환

- 그. 령. 다. 면 소리는 어떻게 처리할 수 있을까?

### 그래서 우리는 Fourier Transforamtion을 한다!!!!

푸리에 변환? 세상에 있는 모든(심지어 규칙성이 아예없는) 함수 그래프도 주기가 다른 사인함수의 합으로 표현할 수 있고, 그렇게 바꾸는 것을 푸리에 변환이라고 한다. 푸리에 변환된 함수의 경우 가로축이 '시간'이 아니라 '주파수'가 된다. 뒤에서 그림을 보면서 설명을 보면 이해가 아주 쉬울 것이다.

수식적으로 보면 복잡하고, 계산량이 많다. 수업시간에 직접 변환해봤는데 정말 쉬워보이는 함수도 변환하는 거 손으로 하면 A4한페이지의 계산량이 나온다. 하지만 이것은 파이썬이 다 해줄 것이다.

## Unit 03 | 푸리에 변환

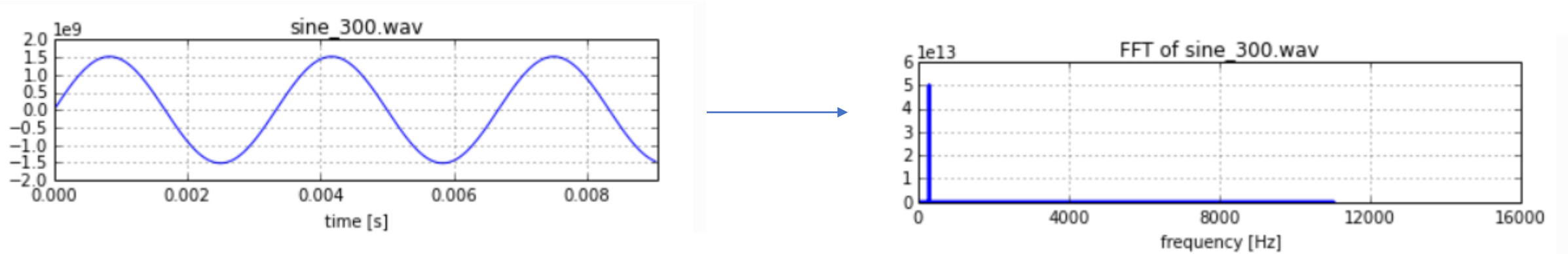
## ■ 보충

<https://www.youtube.com/watch?v=spUNpyF58BY>

개인적으로 시간이 남는다면 이 영상을 보는 것을 추천한다. 음향에서 주파수를 분해하는 푸리에 변환을 시각적으로 표현해준 영상인데 정말 쉽게 잘 설명돼있으며 푸리에 변환이라는 공대에서나 하는 것에 조회수가 300만일 정도로 훌륭한 영상이며 심지어 한글 자막이다... 푸리에 변환에 대해 직관적으로 이해할 수 있을 것이다



## Unit 03 | 푸리에 변환



출처 : <https://opencv-python.readthedocs.io/en/latest/doc/23.imageFourierTransform/imageFourierTransform.html>

자 왼쪽 그래프를 보면, 주파수가 300Hz인 Sin 신호이다. 이것 푸리에 변환을 하면 주파수가 300일 때만 값을 가지고 나머지는 다 0인 함수가 되는 것이다! 따라서, 오른쪽 그래프를 역푸리에 변환을 하면 왼쪽 사인함수가 되는 것이다. 잘 보면 왼쪽에는 그래프의 가로축이 시간이지만 오른쪽은 주파수이다.

## Unit 03 | 푸리에 변환

- 푸리에 변환이 되면..

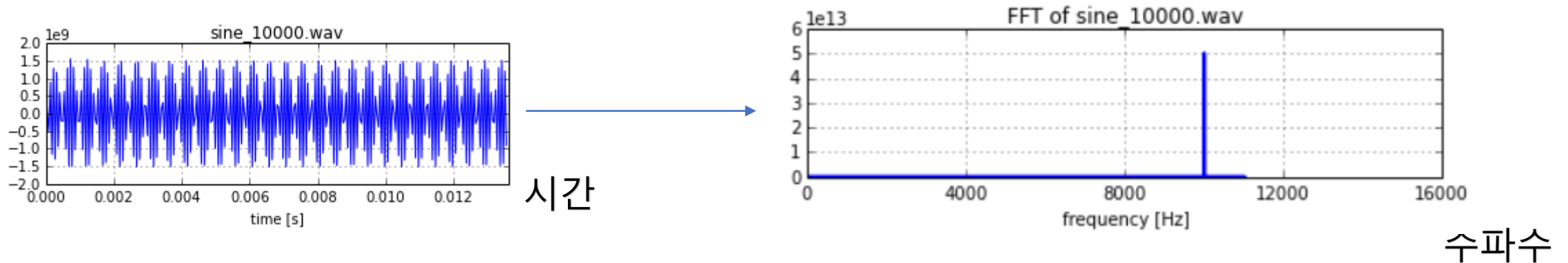
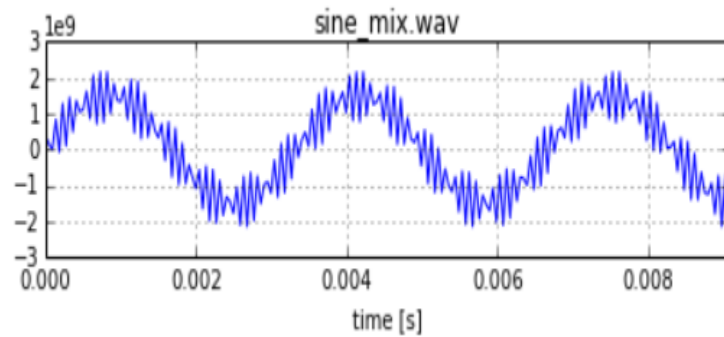


사진 출처 : <http://www.incodom.kr/%ED%91%B8%EB%A6%AC%EC%97%90%EB%B3%80%ED%99%98>

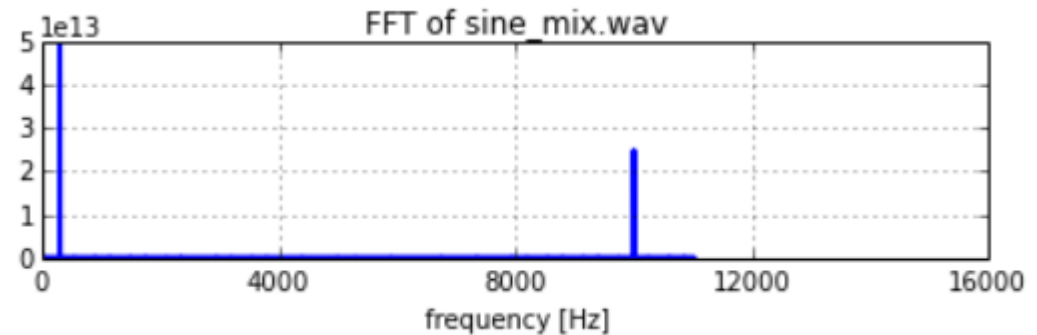
왼쪽 사진은 주파수가 10,000인 사인 그래프이고, 이것을 푸리에 변환하면 주파수가 10000일때만 값을 가지고 나머지는 0을 가지는 함수가 된다!

## Unit 03 | 푸리에 변환

- 푸리에 변환이 되면..



시간



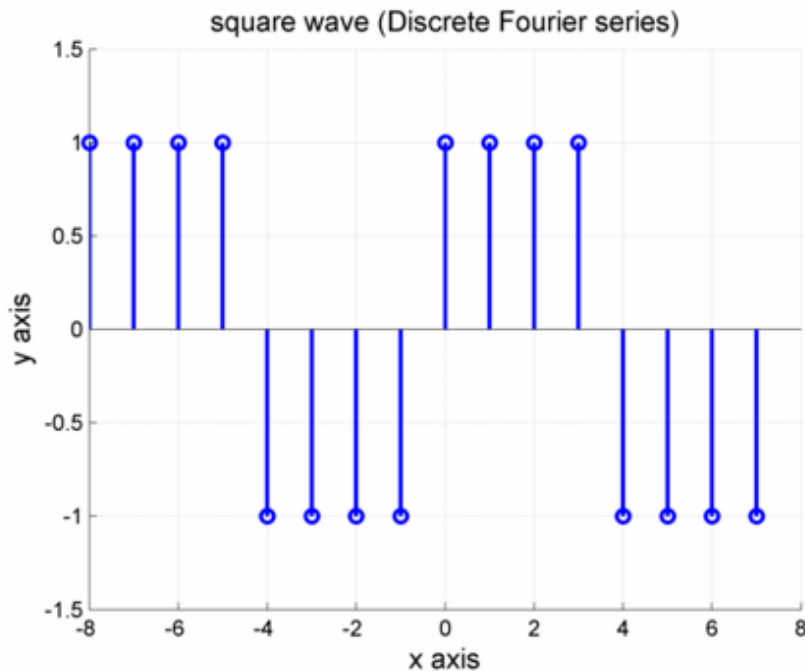
주파수

사진 출처 : <http://www.incodom.kr/%ED%91%B8%EB%A6%AC%EC%97%90 %EB%B3%80%ED%99%98>

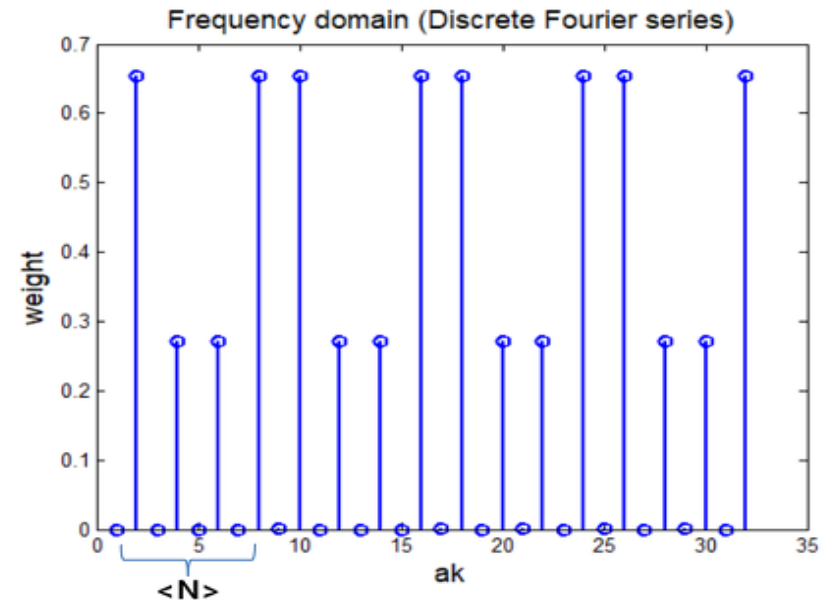
왼쪽사진은 앞 두 슬라이드에 있던 300Hz, 10000Hz Sin함수를 더한 것이다. 이것을 푸리에 변환하면, 300Hz, 10000Hz일때만 값을 가지고 나머지는 0인 함수가 되는 것이다.

## Unit 03 | 푸리에 변환

- TMI) 일단은 몰라도 될 것 같지만, 나중에 필요할 것 같음.



시간



주파수

사진 출처 : <https://twlab.tistory.com/60>

이렇게 원래 주기함수인 경우는 푸리에 변환을 했을 때 discrete한 신호(디지털처럼 0.001초단위로 값을 측정하거나 위의 그림처럼 뚝뚝 끊어져 있는 신호)가 나오며, 원래 연속적이지 않은 신호를 푸리에 변환하면 주기함수가 된다.

## Unit 03 | 푸리에 변환

## ■ 역 푸리에 변환

말 그대로 푸리에 변환의 반대이다. 아래처럼 주파수로 표현돼 있는 신호를 다시 시간에 따른 진폭 값으로 바꿔주는 것이다. 즉 어떤 신호가 주파수 단위로 표현돼 있다면, 시간단위로 다시 바꿔주는 것이다 !

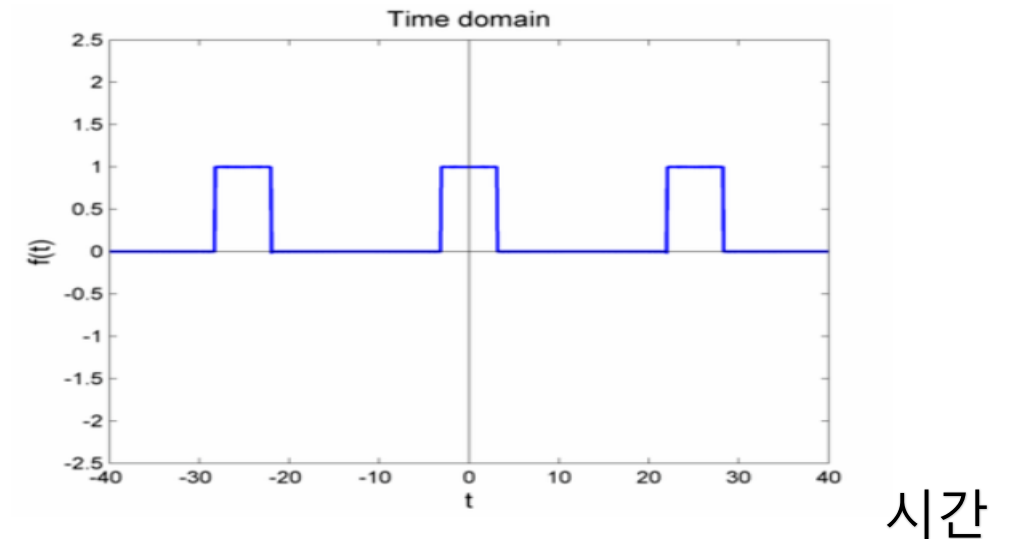
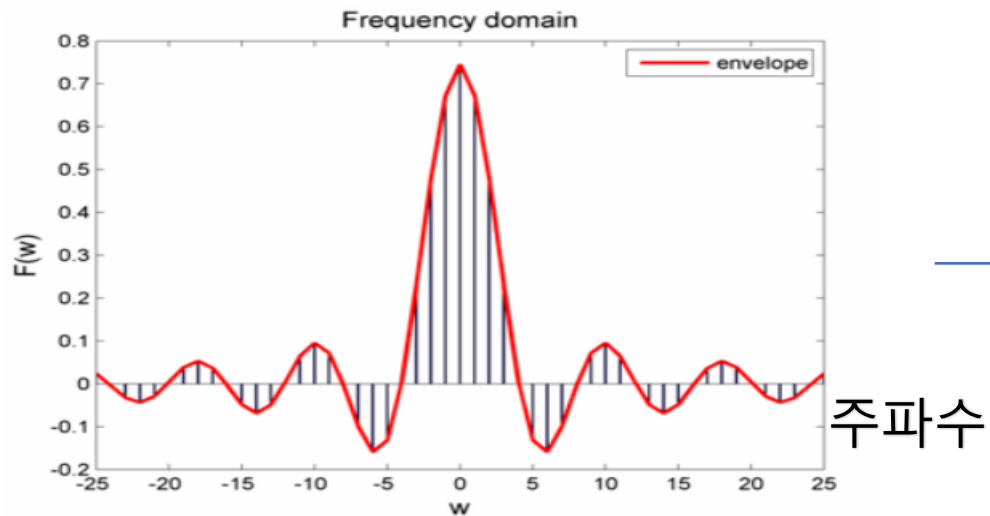
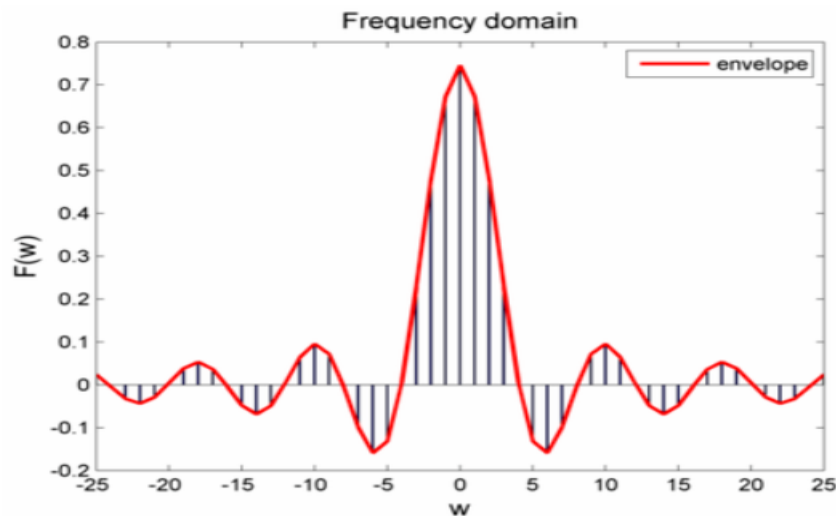


사진 출처 : <https://twlab.tistory.com/60>

## Unit 04 | 푸리에 변환을 이용한 소리 처리

- 그러면 푸리에 변환으로 뭘 어찌자고.

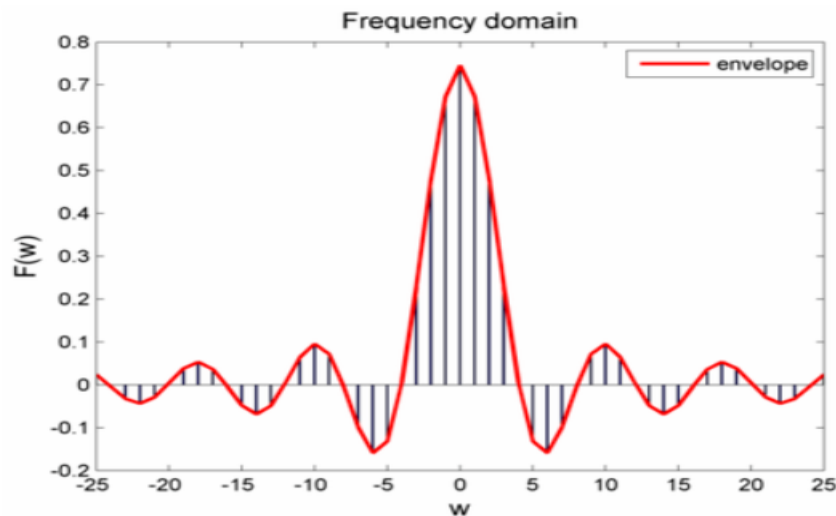
어떤 소리에서 저음역만 걸러내고 싶으면?? 푸리에 변환한 함수에서 주파수가 특정값 이하인것 빼고 다 0으로 바꾼다음에 역푸리에 변환을 하면 고음이 싹 사라진다! 고음을 강조하고 싶으면? 고음 역대값을 2배 3배해주고 역푸리에 변환하면 고음 부분이 커진다!



## Unit 04 | 푸리에 변환을 이용한 소리 처리

- 그러면 푸리에 변환으로 뭘 어찌자고.

만약 보컬만 뽑고 싶으면? 보컬에 해당하는 주파수만 찾아서(아마 이런걸 Deep Learning으로 할 것 같다) 그 부분만 제거하거나 Transfer하고 역변환하면 될 것이다!!



## 추가! MP3 WAV 소리 파일의 구성

- 보통 wav나 mp3파일을 많이 할텐데, 보면 32bit 16bit 192kb이런 값들이 있다. 이것이 음질의 지표인데 무엇을 나타낼까?

우리는 컴퓨터로 신호처리를 할 때 '디지털'로 처리한다. 즉 연속적이고 모든 값을 표현한 것 같아 보여도, 사실은 0.000001초 단위로, 또 진폭을 000000000.1db단위로 쪼개서 표현하는 것이다.



## 추가! MP3 WAV 소리 파일의 구성

- 보통 wav나 mp3파일을 많이 할텐데, 보면 32bit 16bit 192kb이런 값들이 있다. 이것이 음질의 지표인데 무엇을 나타낼까?

Bit : 16bit, 32bit 등은 소리의 세기 즉 진폭을  $2^n$ 개 단위로 쪼개는 것이다. 32bit는  $2^{32}$ 개로 쪼개는 것이고 16bit는  $2^{16}$ 개로 진폭을 쪼개는 것이다. 당연히 bit가 크고 소리의 진폭을 세심하게 쪼개면 음질이 좋아질 것이다.

Kb : K가 1000이므로 192kb는 Sampling rate이  $192 \times 1000$ 이라는 뜻이다. 즉, 1/192000초마다 소리값을 측정하는 것이다. 이 역시 측정 간격이 짧을 수록 음질이 좋아지는 것이다. 아마 이 kb와 Bit가 어떤 것을 쓰느냐에 따라 연상량이 아주 달라질 것 같은데 FLAC같은것만 아니면 무난해 보인다.

Q & A

암거나 카톡으로 물어보세요