

1주차

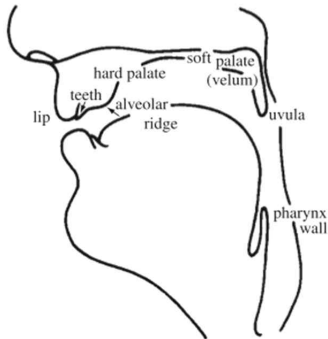
<English consonants & vowels>

A. Phonetics(음성학)

1. articulatory phonetics - 어떻게 조음되는지
2. acoustic phonetics - 어떻게 공기 중으로 전달되는지
3. auditory phonetics

B. Articulation

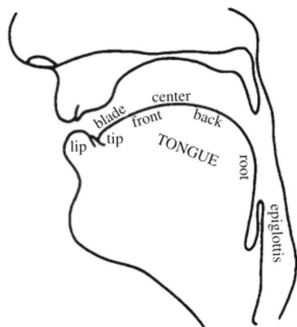
<Vocal Tract>



(Upper vocal tract)

-soft palate에는 뼈가 없음. 혀로 치아 뒤를 따라가다가 부드러워지는 부분이 soft palate=velum

-larynx부터 uvula까지를 pharynx라고 부름



(Lower vocal tract)

-혀는 blade, tip, front, center, back, root로 세분화됨.

-epiglottis에서 epi는 '뚜껑'의 의미. 침이 기도로 가는 길을 막는 역할을 한다.

-성대 밑으로는 허파가 있다.

5 speech organs (=constrictors)

-soft palate(velum), larynx, lips, tongue tip, tongue body  
각각 articulation의 세 가지 process에 연관됨.

#### 1. oro-nasal process

-velum이 오르내리며 관여, velum이 올라가면 nasal tract으로 가는 길이 막히고, 내려가면 열림.  
-즉 코로 숨을 쉴 때 (코에서 바람이 나오므로) velum은 lowered  
-nasal sound에는 [m], [n], [ng]가 있음

#### 2. phonototation process

-무성음과 유성음  
-larynx에서 관여, 성대의 떨림  
-유성음(voiced)= vibration O, 무성음(voiceless)= vibration X

#### 3. articulatory process

-lips, tongue tip, tongue body가 관여  
-CL(constriction location), CD(constriction degree)  
-CL은 bilabial(양 입술), labiodental(입술과 윗니)/ palatal, velar(tongue body)/ alveolar(tongue tip)으로 구분됨  
-CD는 stops> fricatives> approximants> vowels 순으로 구분지음. (constriction degree가 강한 정도)  
-CD: 얼마나 upper part를 치는지. 예를 들어 stops는 upper part를 완전 치고, fricatives는 사이 틈을 조금 남김.  
-자음은 stops, fricatives, approximants 셋 중 하나  
-approximants(접근음): 조음기관이 서로 근접하지만 직접 닿지는 않으면서 내는 음.  
[r], [l], [w], [y]  
-CL, CD, velum이 lowered 됐는지의 여부를 보면 그게 어떤 phoneme인지 알 수 있게

### C. Phonemes

: individual sounds that form words (음소) 개별적인 소리  
-phoneme은 이전에 배운 speech organs의 결합  
-모든 모음은 tongue body를 사용한다.  
-모든 phonemes를 specify할 수 있어야 함.

### D. Acoustics

-praat 쓰는 방법  
: object를 select해야 하는데, object를 뜨게 하는 방법 2가지 (녹음 혹은 open해서 불러옴)  
이후 view&edit 누름  
-자음, 모음 발음되는 길이가 어느 정도인지 duration 알 수 있음  
-pitch(높낮이): 파란 선, intensity: 노란 선  
-Praat으로 measure할 수 있는 것- duration, pitch, intensity

- 빨간 띠 = formant (제일 밑에 있는 게 first formant=F1)
- F1, F2가 뭐냐에 따라서 모음이 결정됨. 모음을 구별하는 수치적인 지표로서 formant가 쓰임
- pitch setting - pitch range에서 남자 목소리, 여자 목소리 설정 다르게 해야 측정이 잘 됨.
- formant: show formants 체크를 해야 함.

#### <Vowel acoustics>

Praat 켜고 실습.

-시각화된 wave에서 제일 큰 부분: larynx의 떨림과 같음.

-헤르츠(Hz): 1초에 성대가 몇 번 떨리는지

-1 나누기 (한 wave의 duration) = 헤르츠

view-sound->create sound and pure tone->tone frequency에 본인의 pitch 입력하면 내 소리와 높이가 똑같은 음이 난다. (wave가 같아서)

## 2주차

-specify 해보기

ex) [p]

(lips) constriction location- bilabial

constriction degree- stop

velum- raised (lowered 되는 게 nasal sound임)

larynx- open 무성음 (closed 되는 건 유성음)

영어의 모든 소리는 저 방식으로 specify가 가능하다!!

#### <vowel acoustics>

pitch: the number of occurrences of a repeating event per second (Hz)

repeating event- sine wave로도 표시됨-성대의 떨림(vibration of vocal folds)과도 일치

SiSwati의 두 번째 소리: 성대에서 나는 소리를 바로 녹음한 것.

즉, 어떤 소리가 만들어지는가는 입에서 결정되지 성대에서 결정되는 것은 아님

sine wave: 가장 기본적인 형태. pitch, frequency, magnitude에 의해 형태가 결정됨  
이 세상에 존재하는 모든 signal은 여러 다르게 생긴 sine wave의 결합으로 표현된다.

complex한 세계를 단순한 sine wave로 표현할 수 있다는 발견.

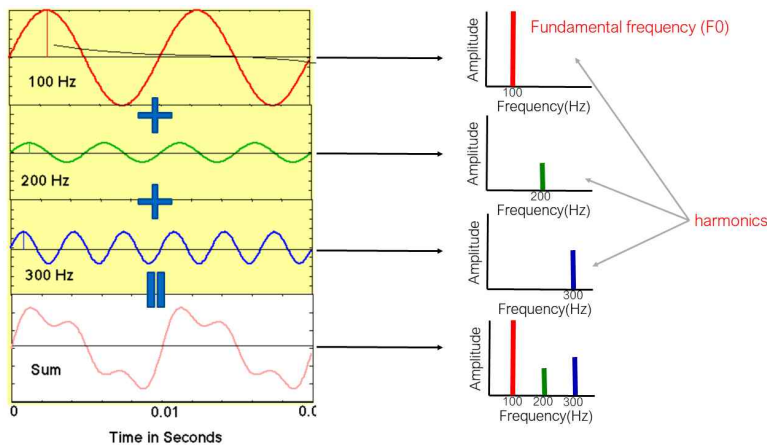
-100Hz: magnitude 큼, frequency 작음 (적게 반복)

300Hz: 첫 번째보다 frequency 높음

200Hz는 100Hz보다 두 배 빠름. magnitude는 첫 번째가 제일 크고 두 번째가 제일 작음.

-100, 200, 300 합친다면? 맨 밑의 그림처럼 됨.

-sine wave의 합은 sine wave가 아닌 복잡한 신호 = 복잡한 신호는 다양한 sine wave의 합으로 이루어진다



- 복잡한 신호에도 반복되는 주기가 있는데, 이게 100Hz랑 똑같은 양상
- 왼쪽을 더 단순하게 표현하는 방법이 오른쪽
- 오른쪽의 x축 frequency, y축 amplitude(진폭)
- 왼쪽의 x축 시간, y축 value값 (단순한 숫자값)
- time-value 그래프를 frequency-amplitude 그래프로 변환할 수 있어야 함.
- 오른쪽 그래프=spectrum
- 우리 주변의 소리들은 complex tone임 (spectrum에서 x축이 완전 채워져서 나옴)
- 소리를 합쳐서 소리를 만들어 내는 것 합성 synthesis, 소리를 다시 쪼개는 것 analysis

-praat 켜고 실습

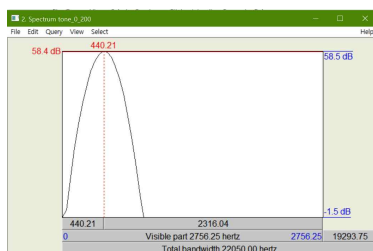
pure tone: 440Hz, amplitude 1

최저값 -1, 최고값 1 (amplitude를 1로 설정해놨기 때문)

duration은 440분의 1 (440Hz이기 때문)

밀의 spectrogram: 스펙트럼에 시간 개념을 적용해서 늘어놓음

view spectral slice-> x축이 frequency y축이 amplitude인 그림이 나옴



(이렇게)

‘아’ 녹음해서 분석-> 여러 sine wave들은 등간격임. (Hz가 배수 사이라는 얘기)

처음에 나왔던 frequency는 나의 pitch와 일치.

이게 아까 봤던 100Hz 200Hz 300Hz의 합에서의 반복되는 패턴이 100Hz와 일치하는 것을 얘기함.

우리의 pitch를 안다면 그 배수들의 sine wave를 무한대로 만들어서 합하고, 그러면 “아”라는 소리를 얻을 수 있음.

“아”라고 얘기하면 여러 다른 simplex tone의 합으로 이루어지는데, 그중 가장 앞의 sine

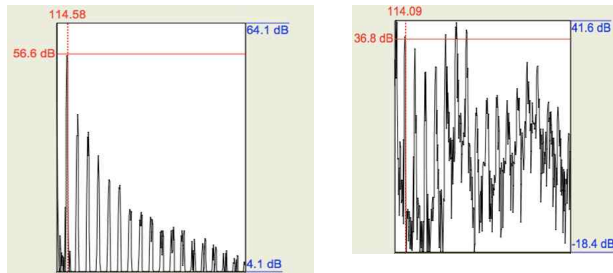
wave frequency가 나의 pitch와 같다.

<source&filter>

성대에서 나는 소리를 그대로 캡처한 것=source

source: larynx에서 나는 소리 (다 똑같음)

tube가 어떻게 작동하느냐-> filter(조음기관을 거치면서 소리가 달라짐)



(원) human voice source: 첫 번째 sine wave를 F0, fundamental frequency라고 부름.

배음 (harmonyx): F0의 배수들로 이루어진 것.

여성의 목소리는 더 듬성듬성 나타남.

1-10000 frequency를 잘랐을 때 남자가 갖는 배음의 수가 더 많다.

(오) filtered by vocal tract: 배음의 구조는 그대로 유지, amplitude의 패턴이 깨짐 (smoothly decreased X)

spectrogram: x축은 시간, y축은 frequency

까만 부분이 에너지가 강한 곳(peak)

source의 스펙트로그램은 low frequency에서 제일 진한 색 (에너지 강함)

filtered by vocal tract의 spectrogram은 뒤죽박죽임.

<source & filter>

filter에서, 고주파로 갈수록 전반적으로 약해지긴 하나, 에너지는 고르지 않게 분포됨.

source: complex tone = sum of pure tones

fundamental frequency = F0 = 1초에 성대가 떨리는 횟수

filter: peak가 있다는 점이 중요. source에는 산맥이 없음.

입모양에 따라, 어디에 산맥이 나타나는가가 달라짐.

산맥의 주파수를 formants라고 함. 첫 번째 formants는 F1

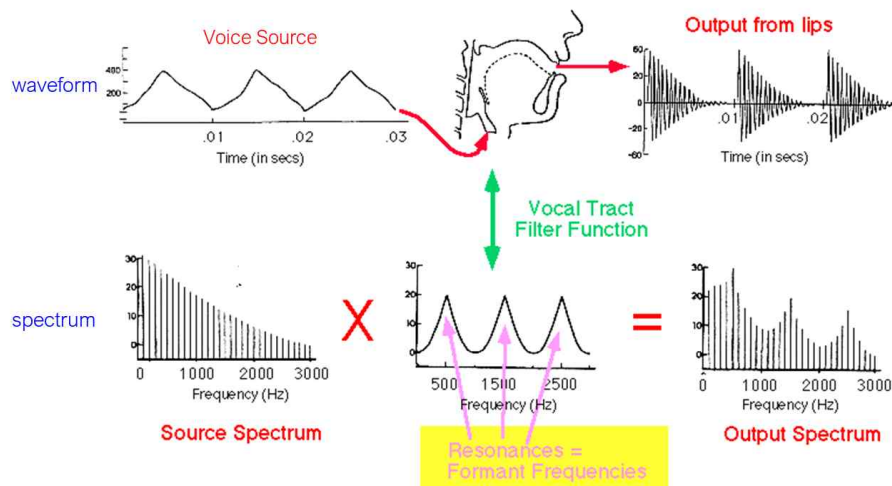
<guitar plucking>

기타 소리는 우리 목에서의 하모닉스와 비슷함, complex tone

praat 켜고 실습: 100Hz부터 1000Hz까지 만들어서 합침.

convert to mono 하면 높이도 주기도 100Hz와 비슷함, 그래서 첫 번째를 fundamental frequency라고 하는 것임.

최종 요약 <source-filter theory>



wave form: time-value

spectrum: frequency-amplitude

vocal tract(tube)가 들어간 후의 output

spectrum을 보면, source spectrum위에 뾰족한 산맥같은 도장을 찍은 느낌(vocal tract의 filtering)-> output spectrum

산맥에서의 첫 번째 튀어나온 부분 = F1(위의 그림에서는 약 600정도)

F1, F2만 있으면 모든 모음을 구분 가능하다. (spectrogram에서 어두운 부분이 formants임)  
서로 다른 모음은 서로 다른 입모양을 가짐.

각 모음의 F1, F2를 따면 (F1을 y축, F2를 x축) 우리 입의 위치와 똑같아짐.

즉 F1은 그 모음의 높낮이(혀의 높낮이 height), F2는 front-back 여부를 결정  
(praat의 create sound from VowelEditor으로 확인할 수 있음)

-한국어보다 영어가 더 낮고, 뒤에서 소리남=우리가 가진 입의 구조를 더 크게 이용

### 3주차

코딩 본격적으로 배우기 시작.

코딩-반복되는 것을 자동화 (똑같은 게 계속 반복되니까 자동화하는 게 유리)

컴퓨터 언어도 C언어, 자바, 파이썬, R 등등 여러 가지가 있음.

각기 다른 언어의 공통점 - 문법, 단어가 있다는 것.

단어에는 정보가 있음. 단어는 정보를 담는 그릇이며, 컴퓨터 언어에서의 단어는 변수에 해당

됨. 즉, 변수가 정보를 담는 그릇임.

변수에 정보를 담고, 기계로 보내는 (인간과 컴퓨터의 소통을 가능하게 해주는) 방법 = 문법

① 빈 그릇에 정보 담기 (변수에다가 정보를 assign) variable assignment

② 조건에 대한 문법 (~할 땐 ~해라) if conditioning

③ 자동화 (여러 번 반복) for loop

④ 함수 이용- 문법 1,2,3을 모두 패키징해서 입력을 넣으면 출력이 나오도록 하는 것

ex) 자동차 작동원리=함수, 두 개의 자연수를 입력하면 그 사이 정수들의 총합을 구하는 함수

① Variable(변수)

variable-숫자 또는 글자

= 표시는 같다는 뜻이 아니라, 오른쪽에 있는 정보를 왼쪽의 variable로 assign 한다는 뜻  
(오른쪽, 왼쪽 잘 구별해야함)

print라는 함수->어떤 변수를 넣으면 그 정보를 스크린에 뿌려주는 역할

python에서 모든 함수는 누가 만들어놔야 하고, 혹은 직접 만들 수도 있다. (이런 함수들을 다 모아 놓은 게 아나콘다)

<Jupyter Notebook 기본적인 사용방법>

1. 한 칸을 셀이라고 부름

2. 셀 누르고 파란색으로 바뀌면 b(below) 치면 밑에 셀 생기고 a(above) 치면 위에 셀 생김

3. 셀을 지우고 싶으면 누르고 x 치면 됨

4. a에다가 1 넣었다가 2 넣으면?? variable은 unique하기 때문에 두 개 치면, 마지막에 친  
걸로 저장됨.

5. 함수 사용하는 방법: 함수이름을 치고 괄호 안에다 변수 넣어주면 됨 ex) print(a)

6. 문자는 반드시 ‘ ’ 따옴표 속에 넣어 줘야함.

7. 실행 단축키-shift+enter

8. b = 'love' : b에다가 love라는 정보를 assign

cf) love = 2, b = love : b에다가 2의 뜻을 가진 love를 assign

```
In [7]: love = 2
```

```
In [9]: b = love
```

```
In [10]: print(b)
```

2

9. 여러 개 한 번에 쓰려면 ; ex) print(a); print(b); print(c)

10. list = 한 변수에 여러 정보를 넣는 것, [] 사이에 넣어주면 됨. (.,로 나열)

11. 정보가 list인지, 문자인지 숫자인지 알고 싶다면 type 함수 씬. type(a)

숫자-int, float/ 리스트-list/ 문자-string/ dictionary-dict

```
In [24]: a = [1,2,3]
```

```
In [26]: type(a)
```

```
Out[26]: list
```

12. list에 반드시 숫자만 들어갈 필요는 X

13. list: 대괄호 대신 그냥 괄호 써도 되는데, 그냥 괄호 쓰면 type이 list 대신 tuple이라고 나옴. (tuple이 보안에 더 강하다)

14. dictionary: 몇 개인지 구분은 콤마로, {} 중괄호 사용. 표제어와 설명의 쌍으로 되어 있고, 그걸 묶어주는 건 :

```
In [41]: a={'love':142,'1':4,'8':26}
```

```
In [45]: a
```

```
Out[45]: {'love': 142, 1: 4}
```

```
In [46]: type(a)
```

```
Out[46]: dict
```