English Consonants & Vowels

- Phonetics: a study on speech

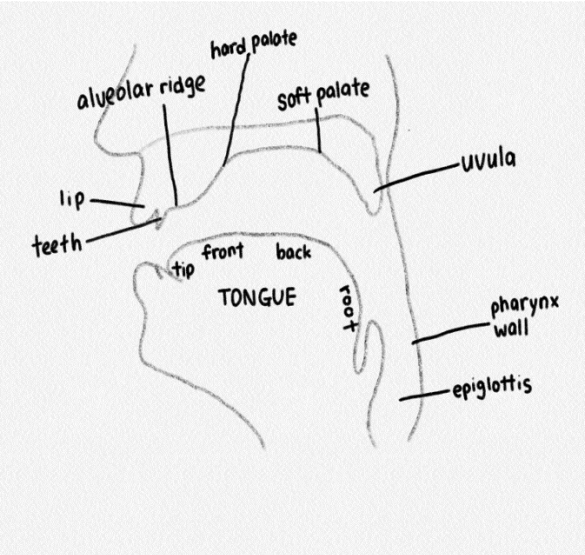
  articulatory phonetics (from mouth) → how to produce speech

  acoustic phonetics (through air) → how to transmit speech

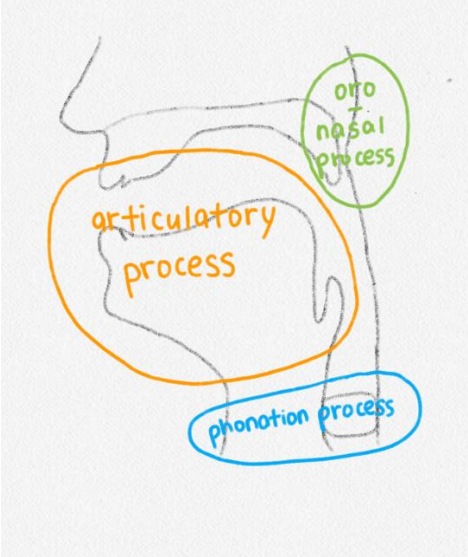
  auditory phonetics (to ear) → how to hear speech

Articulation

- Vocal tract:



- 5 speech organs = constrictors = articulators



Phonation Process in Larynx

- larynx = voicebox: voiced → can feel vibration

  ex. v, z, l, m, a, i

  voiceless → can't feel vibration

  ex. f, s, k, p, h

Oro-nasal Process in Velum

- nasal: when velum lowered

  ex. m, n, ng

Articulatory Process

- lips / tongue tip / tongue body

Control of Constrictors(Articulators)

- Each constrictor needs to be more specific in geometry

  constriction location(CL) / constriction degree(CD)

- Constriction location: Lips → bilabial / labiodental

  Tongue body → palatal / velar

  Tongue tip → dental / alveolar / retroflex / palate-alveolar

- Constriction degree: stops > fricatives > approximants > vowels

How to Produce English Consonants and Vowels

- constrictors(lips, tongue tip, tongue body) / CD / CL / velum / larynx
  ex) /p/: lips / bilabial / stop / velum raised / larynx open
  /b/: lips / bilabial / stop / velum raised / larynx closed
  /d/: tongue tip / alveolar / stop / velum raised / larynx closed
  /z/: tongue tip / alveolar / fricative / velum raised / larynx closed
  /n/: tongue tip / alveolar / stop / velum lowered / larynx closed
- Phonemes: individual sounds that form words
  a combination of speech organs' actions

Acoustics

- Praat: duration > select(click and drag on waveform or spectrogram) →
  read a value (sec.) on the top → zoom in (if not visible)
  intensity > show intensity → click on green → read a value (dB) on the right
  pitch > show pitch → pitch setting – pitch range (65-200Hz male / 145-276Hz female)
  → click on blue → read a value (Hz) on the right
  formant > show formants → place the cursor on one of the trajectories
  → read a value (Hz) on the left
- the number of occurrences of a repeating event per second (frequency, Hz)
  repeating event = vibration of vocal folds / repeating > sine wave = pure wave
  * sine wave: frequency + magnitude(amplitude) (x 축 시간 / y 축 value, voltage)
- 복잡한 신호는 단순한 sine wave 들의 합으로 표현된다. (synthesis)
  simplex tone: 단순한 sine wave / complex tone: 복잡한 신호
  spectrum: x 축 frequency / y 축 magnitude(amplitude)
  spectrogram: spectrum 을 시간으로 visualize 한 것 (x 축 시간 / y 축 frequency)
  sine wave→spectrum: spectral analysis
- pure tone→spectral analysis: frequency 가 같은 sine wave 한 개
  complex tone→spectral analysis: 일정한 간격의 sine wave 여러 개 (간격=pitch)
  (Praat: Spectrum > View Spectrum Slice)
- source: 성대에서 나는 소리
  human voice source consists of harmonics
  a complex tone = sum of pure tones at integer multiples of the lowest pure tone
  the lowest pure tone = fundamental frequency(F0) = rate of vibration of the larynx
  = the number of opening-closing cycles of the larynx per second
  amplitude of pure tones gradually decreases
- filter: vocal tract 에 의해서 달라지는 소리
  filter 의 spectrum → jigjagging with peaks and valleys (amplitude 의 패턴이 사라짐)
  peaks/mountains: frequencies VT likes (formants)
  valleys: frequencies VT does not like

- Synthesize Source: New > Sound > Create Sound as Pure Tone
  > Tone frequency 100~1000Hz / Amplitude 1.0~0.55Pa
  Combine > Combine to Stereo
  Convert > Convert to Mono
  반복 주기: frequency 100Hz / Amplitude 1.0Pa 와 일치
  음: frequency 100Hz / Amplitude 1.0Pa 와 일치
  Spectrum > View Spectral Slice: gradually decrease / 10 개 / 100Hz
- F1: 모음의 height / F2: 모음의 frontness(backness)
  F1 and F2 are enough to disambiguate vowels.
  (Praat: New > Sound > Create Sound as VowelEditor)

Coding

- 코딩: 자동화 > 똑같은 과정을 쉽게 반복할 수 있기 위해서
- 모든 language 는 공통적으로 단어와 문법으로 이루어짐
  단어: 정보를 담는 그릇
- Computer Language 의 단어: 변수(variable)
  Computer Language 의 문법: 1. variable assigning
  2. 'if' conditioning
  3. 'for' loop
  함수: 어떤 입력을 넣어야 자신이 원하는 출력이 나오는지
  ex. Praat 입력: 마우스로 구간 설정 / Praat 출력: 소리
- Anaconda Prompt > 'Jupyter Notebook' 입력
  원하는 디렉토리 > New > Python 3
- cell 생성: cell 선택 후 b(아래쪽에 생성) / a(위에 생성) / x(삭제)
- =: 오른쪽에 있는 정보를 왼쪽에 있는 variable 로 assign 한다
  ex. a=1 > 정보: 1 / variable: a
  Print: 어떤 변수를 넣으면 그 값을 출력함
- In [1] : a = 1 입력 > Run > In [2] : print (a) 입력 > Run > 1
  In [3] : a = 2 입력 > Run > In [1](a = 1) 함수가 사라짐 > In [4] : print (1) 입력 > Run > 2
  In [1] 선택 > Run > In [4] 선택 > Run > 1
- In [1] b = 'love' 입력 > Run > In [2] print (b) 입력 > Run > love
  In [3] : love = 2 입력 > Run > In [4] : b = love 입력 > Run > In [5] : print (b) > Run > 2
  * 문자 입력의 경우, ' ' 가 없을 때는 변수
- In [1] : a = 1; b = 2; c = 3 > Run > In [2] : print (a); print (b); print (c) > Run > 1; 2; 3
- In [1] : a = [1, 2, 3, 5] > Run > In [2] : type (a) > Run > list
  In [1] : a = 1 > Run > In [2] : type (a) > Run > int
  In [1] : a = 1.2 > Run > In [2] : type (a) > Run > float
  In [1] : a = 'love' > Run > In [2] : type (a) > Run > str
  In [1] : a = [1, 2, 3, 5, 'love'] > Run > In [2] type (a) > Run > list

In [1] : a = [1, 'love', [1, 'bye']] > Run > In [2] : type (a) > Run > list

In [1] : a = (1, 'love', [1, 'bye']) > Run > In [2] : type (a) > Run > tuple

* list = tuple (list 보다 보안이 강함)

In [1] : a = {'a': 'apple', 'b': 'banana'} > Run > In [2] type (a) > Run > dict

* 표제어: 설명