

영어음성학 정리 - 9월 3~6주차

영어의 자모는 크게 voiced 와 voiceless로 구분된다. 이 둘은 성대의 떨림을 기준으로 한다. 성대가 떨리면 voiced, 그렇지 않으면 voiceless이다.

자음은 b, d, g, j, l, m, n, ng, r, th, v, w, y, z가 voiced consonants이고,

Ch, f, k, p, s, sh, t, th는 voiceless consonants이다.

모음은 모두 voiceless이다.

모음은 크게 두종류로 나뉜다. 단모음(monophthongs)와 복모음(diphthongs)가 있다.

Phonetics는 ‘말(speech)’를 연구하는 학문이다.

Phonetics는 다시 세 종류로 나뉘는데, Articulatory phonetics, Acoustic phonetics, Auditory phonetics가 그 예시이다.

Articulatory phonetics는 우리의 구강구조에서 어떻게 말을 생산하는가에 대해 연구한다.

Acoustic phonetics는 어떻게 말이 전달되는지 파동의 개념으로 본다.

Auditory phonetics는 어떻게 말이 우리의 귀로 전달되어 인식하는가를 배운다.

조음(Articulation)

말소리가 이동하는 통로: 이 - 귀(ear), 비 - 코(nose), 인 - 인두(pharynx), 후 - 후두(larynx)

Vocal tract(upper):

입술(lip): v, p와 같은 소리의 조음 장소이다.

치아(teeth): th와 같은 치음 소리의 조음 장소이다.

Alveolar ridge: 치아 바로 뒤 잇몸부분이다. d와 같은 치경음의 조음장소이다.

경구개(hard palate): 입천장 중에 앞부분으로, 상대적으로 딱딱한 부분이다.

연구개(soft palate): 입천장 중 뒷부분으로, 상대적으로 부드러운 부분이다. Velum이라 부르기도 한다.

목젖(uvula): 목젖이다.

인두벽(pharynx wall): 인두 부분을 말한다.

Vocal tract(lower):

아랫부분은 크게 3군데로 나뉜다. 입술, 혀, 후두개(epiglottis)

그중에 혀는 혀끝(tip), 혀날(blade), 앞, 중간, 뒤, 혀뿌리(root)가 있다.

epiglottis는 후두에 있는 후두개를 말한다.

Phonation process in larynx

폐에서 나오는 공기는 larynx에서 phonation process 거친다. 이때 성대가 매우 빠른 속도로 진동하여 소리를 만든다. 보통 소리의 높낮이와 진동수는 비례한다. 후두에서 진동이 있느냐 없느냐에 따라 유성음 무성음으로 나뉜다. 영어의 모든 모음은 유성음이다.

Oro-nasal process in velum

m, n, ng와 같이 흔히 비음이라 말해지는 것들은 코로 공기가 감을 통해서 이루어진다.

이때, velum이 주역할을 한다. Velum이 열리면 비음이 되고, 다른 음성일때는 닫혀 있다.

Control of constrictors

constriction이 어디에 이루어지느냐에 따라 영어 자모음이 달라진다.

입술로 이루어지는 소리들, p와 같은 소리들이 대표적이다.

Tongue body가 palatal과 velar 어디에 있느냐에 따라 소리가 달라진다.

velar는 k, palatal의 경우 g가 대표적이다.

Tongue tip의 경우 dental부터 치아 뒷부분인 alveolar, 더 뒷부분인 palato-alveolar부분(sh), 그리고 그 부분을 지나가면서 발성하는 retroflex음이 있다.

한편 세기에 따라 달라지기도 한다.

완전히 막고 세게하면 stop, 조금 틈을 열면 fricative, 모음보다 조금 더 열면 approximants, 그 다음은 모음이다.

이러한 소리들은 조음 기관의 조합으로 이루어진다. 이러한 소리들로 단어가 구성된다. 이때 구성되는 하나하나의 소리를 phoneme이라 한다.

Praat에서는 intensity를 알 수 있고, pitch를 통해 음의 높낮이를, 모음구분할 수 있는 formant 또한 확인할 수 있다.

음성을 나타내는 방법은 크게 두가지이다. Waveform과 spectrogram이 그것이다. waveform의 경우 x축과 y축 각각 time과 value를 나타낸다. Sine wave로 나타내진다. 폭이 넓을수록 value가 낮아지고 이는 낮은 소리를 나타낸다. 즉, sine wave의 폭이 소리의 높낮이와 비례한다.

다른 표기 방법으로 spectrum은 각각 frequency와 amplitude를 가진다. spectrum을 보면 진한 부분이 있고 그렇지 않은 부분이 있다. 이는 spectrum에서 amplitude가 높은 부분이 진하게 보인다.

인간의 목소리 근원을 source라 하며, harmonics를 기반으로 한다. harmonics란 praat에서 Pure tone으로 볼 수 있는데, 일정한 주파수가 있고 그 주파수 대역에 있는 특성, 성분들을 나타낸다. Wave form의 경우 일정한 모습을 보이며, spectrogram으로 볼때는 frequency의 pitch 부분 당 amplitude가 gradually decreasing 하는 모습을 볼 수 있다. 여기서 가장 낮은 frequency를 fundamental frequency, F0라 부른다.

이러한 harmonics는 우리의 vocal tract를 지나며 달라진다. 구강구조로 인해 공기가 저항받아 성대의 떨림으로 일어나던 일정한 harmonics가 바뀌는 것이다. 이를 vocal tract에 의해 filtered 된다고 한다. Vocal tract를 지나가면서 filtered 된 voice는 음성적으로 peak 혹은 mountain을 가진다. harmonics와 다르게 일정하지 않다. 여기서 mountain 부분은 스펙트럼에서 진하게 보이고, 이는 소리가 구분되는 formant를 나타내기도 한다. 여기서 첫번째 mountain을 F1이라 부른다.

반복되는 부분을 pulse, 그것이 이어진다 해서 pulse train이라 부른다. 반복주기가 가장 낮은 주파수의 웨이브와 일치한다.

5주차

프로그래밍 언어와 언어:

프로그래밍 언어와 언어의 공통점은 단어와 문법이 있다는 것이다. 우리가 언어를 사용할때, 혹은 언어를 공부할때 흔히 단어와 문법을 가지고 공부하고 글과 말을 구성한다. 인간의 언어처럼, 프로그래밍 언어도 마찬가지다. 프로그래밍 언어도 단어와 문법을 기반으로 구성된다. 우리 언어가 단어와 문법을 어떻게 결합하느냐에 따라 의미가 달라지듯이, 프로그래밍 언어도 그렇다. 여기서 언어의 단어에 해당하는 것이 프로그래밍 언어의 variable, 변수이다. 단어가 뜻, 즉 정보를 담는 그릇이듯이, 변수도 정보를 담는 그릇이다.

우리가 프로그래밍을 공부하면서 알아야할 것들이 있다.

1. 변수에 정보를 넣는 것
2. 조건에 대한 것 -> if문
3. 반복 -> for문
4. 변수와 명령을 packaging하는 함수

우리가 프로그래밍하면 생각나는 것이 바로 자동화, 기계화이다. 인간이 하기 귀찮은 것들을 대신 하거나 반복되는 일들을 자동화하는 것이 바로 프로그래밍의 역할이다. 이 점을 생각했을때, 어떨때 어떻게 하라, 즉 조건에 대해 생각할 수 있다. 이것을 구현하는 것이 if문이다. 그리고 반복하는 것은 for문이다. 하지만 조건과 반복, 그리고 변수가 하나밖에 없을리가 없고, 또 여러개가 필요하기도 하다. 이러한 것들을 묶어 편리하게 사용할 수 있게 하는 것이 함수이다. 입력을 하면 함수를 거쳐 출력형태가 나오는 것이 기본 형태이다.

우리가 변수를 설정할때 오른쪽에 있는 정보를 왼쪽 변수에 assign한다. 따옴표를 통해 문자열이라는 자료형을 표현하고, print 함수를 통해 출력을 이끌어낸다. 대괄호를 통해 리스트라는 자료형을 표현하고, 이는 여러 자료형이 리스트 하나에 있다. 이와 비슷한 것이 튜플인데, 리스트와 달리 정보의 변경이 힘들다는 특징이 있다. '{}'로 딕셔너리라는 자료형을 나타낸다. 키값과 밸류값을 통해 연결되어 있다. type함수를 통해 자료형이 무엇인지 알 수 있다.

6주차

numpy를 사용해서 array를 구성할 수 있다.

```
import numpy as np
a = [(1,2,3),(3,8,0)]
b = np.array(a)
type(b)
```

index를 사용해서 자료형 속 일부를 나타낼 수 있다. A = [1,2]라 할때, A[0]이라 하면 1이 나온다. Variable[index]의 형식으로 쓰인다. D = {1:'key'}라 할때, 1이 key, 'key'가 value 이다. 문자열도 해당된다. s = 'abcd'라 할때, s[1]은 'b'이다.

:을 통하여 범위를 지정할 수 있다. s[1:3]은 'bc'이다.

find는 해당 문자열을 찾아 시작점의 인덱스를 알려주는 함수 이다.

'my house'.find('house')라 하면 'house'의 시작 인덱스인 3을 아웃풋으로 내놓는다.

rindex는 오른쪽 부터 해당 문자열을 찾아 해당 문자열의 시작 인덱스를 알려준다.

'home, my home'.rindex('home') -> 9

range함수를 사용하여 일정한 범위를 지정할 수 있다.

list(range(4)) 라 하면 [0,1,2,3]이 나온다.

len() function은 해당 자료형의 길이나 개수를 나타낸다.

일정 범위의 숫자를 출력하는 함수는 for 문을 사용하면 된다.

```
a = list(range(1,8))
for w in range(len(a)):
    print(a[w])
```

enumerate함수는 인덱스와 값을 연결시킨다. a = ['red', 'green', 'blue', 'purple']가 있으면 0 - 'red'와 같은 방식으로 구성된다. 이것을 응용하면 다음과 같이 출력할 수 있다.

```
for i, s in enumerate(a):
    print('{}: {}'.format(s,b[i]*100))
```

format이라는 함수를 통해 일정한 형식으로 출력하게끔 할 수 있다.

```
red: 20.0%
green: 30.0%
blue: 10.0%
purple: 40.0%
```

zip이라는 함수를 사용하면 두 리스트를 묶을 수 있다. [1,2,3] , ['a','b','c']를 묶어서 리스트로 만들면 [(1,'a') ...]형식으로 도출된다.

파이썬에서 같다는 ==이다. 같지 않다는 !=이다. for문안에 또 쓸 수 있으며, 조건문도 마찬가지이다. for문이 두개가 있다면 우선 하위 반복문에서 먼저 반복이 이루어진 뒤에 다시 상위 반복문으로 넘어간다.