영어음성학 summary

2018130819 영어영문학과
정해민

Phonetics

- Articulatory phonetics (from mouth) 조음
- Acoustic phonetics (through air) 음향
- Auditory phonetics (to ear) 청각

Articulation

5 speech organs

1. Articulatory process

   Lips / tongue tip / tongue body: 가장 중요한 부분

2. Oro-nasal process

   Velum: nasals (m, n, ng, …)

3. Phonation process

   Larynx: voiced/voiceless (larynx=voicebox)

Control of constrictors

- 3 major constrictors: lips, tongue tip, tongue body
- Constriction Location (CL)

  Lips: bilabial / labiodental

  Tongue body: palatal / velar

  Tongue tip: dental / alveolar / retroflex / palate-alveolar
- Constriction Degree (CD)

  (upper) stops / fricatives / approximants / vowels (lower)

➔ By specifying constrictors, CD, and CL, we produce English consonants & vowels.

코딩은 '자동화'

자동화를 하는 이유: 초기 에너지를 투자하면 반복하는 데 필요한 불필요한 에너지를 절약할 수 있다.

모든 언어는 '단어'가 있고, 이를 '결합'하는 방법이 있다.

단어는 '정보'를 담는 그릇.

컴퓨터 언어에서 '단어'에 해당하는 부분이 '변수'

기계와의 '문법'은 생각보다 어렵지 않다.

1. 변수라는 그릇에 정보를 넣는 것 variable assigning
2. 자동화 If conditioning
3. 여러 번 반복 for route
4. 함수: 어떤 입력을 넣으면 출력이 나오도록 하는 것

A=1

등호 오른쪽이 정보(1) 왼쪽이 변수(a)

A: variable / 1: information

Run을 해야 입력이 됨!

python에서 모든 함수는 누가 만들어 놔야 함(직접 만들 수도 있음)

정보의 종류: 숫자, 문자

셀 지우기: x

셀 만들기: b (위로 만들기 a)

실행(run) 단축키: shift+enter

a=[1,2] ; b=[3,4] ;c=a[0]+b[0]

c

4

A[0]+b[0]에서 대괄호 안에 숫자는 '순서'를 의미 (컴퓨터에서는 1이 아니라 0부터 시작)

어떤 variable의 정보로 들어갈 때 대괄호 안에 index를 적으면 내부 정보를 데려온다.
a='123';print(type(a));print(a[1])

Dictionary: pair에서 앞 부분을 index로 쓴다. (그냥 list에서는 0,1,2,3 등 숫자를 index로 썼음) a={"a":"apple","b":"orange","c":2014} ; print(type(a)) ; print(a["a"])

String과 list는 정보의 측면에서 거의 비슷

S='abcdef' N=[100,200,300]

<len 함수>

Len(s) 6

Len(n) 3

*** this는 문장에 두개 들어 있는데 첫번째 것만 찾아주는 함수 (find)

*** rindex는 오른쪽에서 카운트하는 것

<split 함수> 아주 유용: 잡 스러운 것 지워줌

s=s.strip()

s -> 'this is a house built this year.'

<문장을 list로 만들 때: split 함수>

s.split(' ') -> space를 기준으로 잘라라 (작은 따옴표 안에 들어간 거 기준으로 잘라라)

numpy라는 package library 속에는 또 작은 package들이 들어 있을 수 있고, 그 안에 또, 그 안에 또 ...

numpy.A.B.@ 의 경우 numpy안의 A안의 B안의 @라는 함수를 뜻함 -> 이게 너무 복잡해서 쓰는 방법들이 몇 가지 있음.

from numpy import A : numpy에 있는 A를 불러오자. -> A.D.@ 로 접근할 수 있음

from numpy import A.D : (한 마디로 import를 크게도 할 수 있고 from을 사용해서 작게도 할 수 있음)

import랑 from을 잘 쓰면 됨

#numpy가 필요한 이유? numpy는 list와 아주 비슷한데 수학적인 계산 가능 앞으로 쓰게 될 모든 데이터 처리는 list가 아니라 numpy 처리

#empty는 함수 (함수는 괄호로 그 뒤에 input을 받는다고 했으니까 empty는 함수)

#numpy라는 제일 큰 library 속에 최상단에 있는 empty라는 함수

#2x3로 (옆으로 뚱뚱한 모양) 내부적으로 들어간 data type을 int(eger)로 하나 만들어라 (숫자는 random)

#float도 종류가 있음 float64의 경우 소수점 64번째자리까지 -> 아주 정밀하게, 오차를 허용하고 싶지 않을 때 사용

#단점은 많은 용량 차지

data=np.random.normal(0,1,100)

print(data)

%plt.hist(data,bins=10)

%plt.show()

#normal이라는 함수는 normal distribution을 만들어주는 함수 normal distribution은 '정규분포'를 뜻함

#-> shape을 만들어주는 게 아니라 data를 만들어 주는 것

#plt는 matplotlib라는 library 속의 pyplot을 부름.

#plt속의 hist라는 함수를 쓸 것. hist는 '히스토그램' 히스토그램의 option에는 늘 bins라고 해서 바구니를 총 몇개를 할 건가를 정함

#x축에 바구니를 10개 만드는 것 (바구니 5개 다음 0(mean)나오고 5개 나오고)

#range속에 들어가는 위의 값들이 y값. y값에 들어가는 것들은 절대 소수값이 나올 수 없음. 값들을 다 더하면 100개