-English consonants and vowels

p	pie	pea		lowercase *p*
t	tie	tea		lowercase *t*
k	kye	key		lowercase *k*
b	by	bee		lowercase *b*
d	dye	D		lowercase *d*
g	guy			lowercase *g*
m	my	me	ra*m*	lowercase *m*
n	nigh	knee	ra*n*	lowercase *n*
ŋ			ra*ng*	eng (or angm
f	fie	fee		lowercase *f*
v	vie	V		lowercase *v*
θ	thigh			theta
ð	thy	thee		eth
s	sigh	sea	listen	lowercase *s*
z		Z	mizzen	lowercase *z*
ʃ (š)	shy	she	mission	esh (or long *s*
ʒ (ž)			vision	long *z* (or yog
l	lie	lee		lowercase *l*
w	why	we		lowercase *w*
r	rye			lowercase *r*
j (y)		ye		lowercase *j*
h	high	he		lowercase *h*

Note also the following:

tʃ (tš)	chi(me)	chea(p)
dʒ (dž)	ji(ve)	G

1	2						
i	i	heed	he	bead	heat	keyed	lowercase *i*
ɪ	ɪ	hid		bid	hit	kid	small capital *I*
eɪ	eɪ	hayed	hay	bayed	hate	Cade	lowercase *e*
ɛ	ɛ	head		bed			epsilon
æ	æ	had		bad	hat	cad	ash
ɑ	ɑ	hard		bard	heart	card	script *a*
ɑ	ɒ	hod		bod	hot	cod	turned script *a*
ɔ	ɔ	hawed	haw	bawd		cawed	open *o*
ʊ	ʊ	hood				could	upsilon
oʊ	əʊ	hoed	hoe	bode		code	lowercase *o*
u	u	who'd	who	booed	hoot	cooed	lowercase *u*
ʌ	ʌ	Hudd		bud	hut	cud	turned *v*
ɝ	ɜ	herd	her	bird	hurt	curd	reversed epsilon
aɪ	aɪ	hide	high	bide	height		lowercase a (+I)
aʊ	aʊ		how	bowed		cowed	(as noted above)
ɔɪ	ɔɪ		(a)hoy	Boyd			(as noted above)
ɪr	ɪə		here	beard			(as noted above)
ɛr	ɛə		hair	bared		cared	(as noted above)
aɪr	aə	hired	hire				(as noted above)
Note also:							
ju	ju	hued	hue	Bude		cued	(as noted above)

-Phonetics: a study on speech

1. Articulatory phonetics(from mouth): how to produce speech
2. Acoustic phonetics(through air): how to transmit speech
3. Auditory phonetics(to ear):how to hear speech

-upper vocal tract: Lip, teeth, alveolar ridge, hard palate, soft palate(velum), uvular, pharynx wall, larynx

-lower vocal tract: lip, tongue tip, blade, front, center, back, root, epiglottis(음식이 기도로 가는것을 막음)

-larynx

1. Voiced: vibration- 성대가 붙어있을 때
2. Voiceless:no vibration-성대가 떨어져 있을 때

-velum lowered: nasal, breathing

-constriction

1.Constriction location

1) Lips-bilabial, labiodental
2) tongue body- palatal, velar cf)모든 모음은 tongue body를 사용
3) tongue tip- (inter)dental, alveolar, retroflex, palato-alveolar

2.Constriction degree

Stops>fricative>approximants>vowel

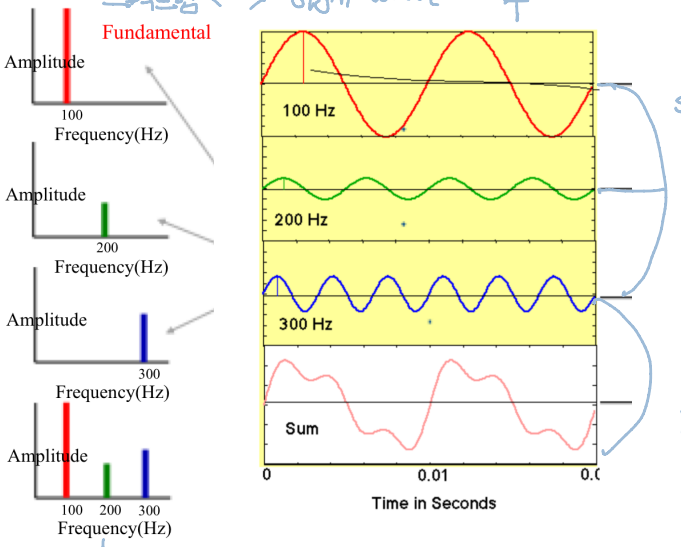
-phonemes: individual sounds that form words

* Spectrum

모든 소리는 spectrum으로 이루어져 있음.

time x value (frequency)

스펙트럼 ↔ sign wave



simplex(Pure) tone

complex tone

: simplex tone을 합성한 결과.

frequency x amplitude (magnitude)

(주파수 x 그 주파수의 크기)

cf) Praat 기준용

spectrum : 숫자 입력.

또는 : sign wave

* Source (목청에서만 나온소리)

- Human voice source consists of *harmonics* 배음 증가
- A complex tone = sum of pure tones at integer multiples of the lowest pure tone
- the lowest pure tone
  - *Fundamental frequency (F0)* = Pitch
  - rate of vibration of the larynx
  - the number of opening-closing cycles of the larynx per second (초당 진동수)
- Amplitude of pure tones gradually decreases

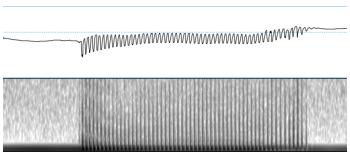
cf) 여 f0 ↑ → 등잔등성
남 f0 ↓ → 백백.

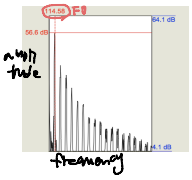
* Filter

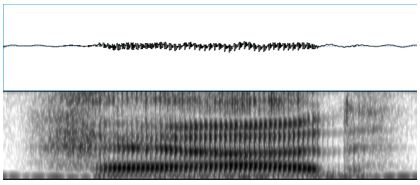
- Compare spectrums between audio and EGG
- EGG: gradual decreasing
- audio: peaks/mountains and valleys harmonics인걸 고려하나, amplitude가 gradual decrease가 아님. → Curved.

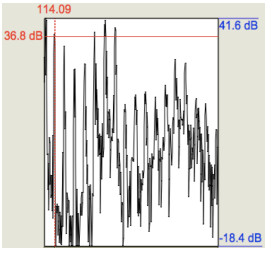
source즉시는 기각.

- Because it is filtered by the vocal tract (VT)
  - peaks/mountains: frequencies VT likes = formants
  - valleys: frequencies VT does NOT like





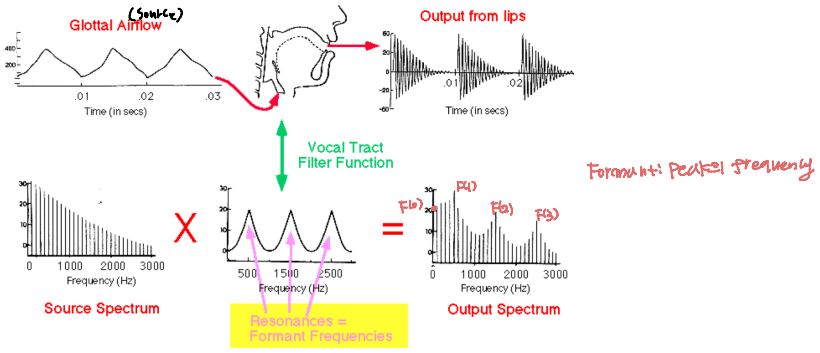




* Synthesizing Source

- convert to mono → 첫번째 sign wave와 주기일치, 높이도 첫번째 거와 비슷.
- 무한대로 가면, pulse train.

* Source-filter theory

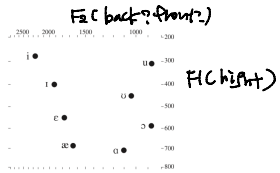


Formant: peak의 frequency

* Spectrogram

- Dark band : mountains = Formants

* F1, F2 만으로 모음구분 可



코딩 = 자동화 (ex. 콘에있는 프로그램들)

언어必

사람의언어 ≒ 컴퓨터언어
- 사람의언어: 단어 : 정보 (단어는 정보를 담는그릇) / 문법
- 컴퓨터언어: 변수: 정보 ex) 숫자, 문자... / 문법
  1. 변수에 정보를 assign
  2. 조건 (if)
  3. 반복 (for)
  ★ 4. 함수 : 입력 → 출력

x = y ← y를 x로 assign. ex) a=1 / Run (단축키: Shift + enter)

들어갔는지확인 : print(a)

셀 왼쪽을 클릭하고 파란색으로 (a : 위에 셀생김 / b : 아래에 〃) / select ⊕ X = delete.

문자입력 : ' ' 必

- love = 2 b = love print(b) = 2

- a=1
  b=1
  c=3
  c
  ||
  a=1; b=1; c=3; → run → 3 (마지막하나는 print해줌.)

a=1; b=1; c=3;

- 한 변수에 여러개 넣는법 → 대괄호 ex) a = [1, 2, 3, 5]
  (list) 괄호바꿔도 ok (소괄호) tuple

- type(변수) → 어떤 type 인지 ex) int (숫자), list, float (소수점있는 숫자), str (문자)

- a = [1, 'love', [1, 'bye']]

- a = {'a' : 'apple', 'b' : 'banana'} 표제어 : 설명.
  ↳ type : dictionary

A=1, b=1 c=a+b
C=2

A= [1,2]
B=[3,4]
C=a[0]+b[0]
C
-4(1+3)

C=a[1]+b[1]
C
-6

A=1.2 a=int(a) print(a) -1
A= '123' print(a[1]) -1. Q. ???
A= 123 print(a[1]) - error ← 이건 index도, 몰라서 숫자에만 씀

Index: 내부적인부분

Dict에있는 정보를 가져올때는 pair중의 앞부분을 index의 수단으로 쓴다. 그러면 뒷부분이 나옴.

Type(a[0])

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

Print s[0] 제일 첫 번재 꺼 .....
s[-1] 제일 마지막 꺼 .....
S[1:3] = 첫번째에서 세번째의 직전까지 / b,c
1: = 첫번째에서 끝까지
:3 = 첫번째에서 세번째 의 직전까지

Len[ ] = 길이

S.upper( ) - 대문자 됨

R index ??

Strip: 쓸데없는거 없애주는
Split: ' ' -스페이스를 사용해서 나눠라

a='123' print(a[1]) → 2 (그동 음자에서도, 두번째꺼)
(two가아니라 이전 문자처럼.

( )로 묶으면 tuple, [ ]로 묶으면 list.

dic → {"a" : "apple"}

print(a ["a"])

S="abcdef"
0번부터 1에서 끝까지 0에서 3직전까지
print(s[1:3], s[1:], s[:3], s[:])
bc bcdef abc abcdef.

s.upper()
result = s.find( ) : 띄어쓰기포함. 몇번째에서? ※ 0부터시작 print(result)
result = s.rindex(?)
s = s.strip() ← print(s)
빈칸을 쪼개서
tokens = s.split(' ') ← print(tokens)
s = ' '.join(tokens)

s = s.replace('this', 'that')

```
a= [1,2,3,4]
for i in a:          # for in : 라는 문법은 in 뒤에 있는 것을 i에 할당하는 것
    print(i)         # for과 if 다음에는 indent를 하는 것이 매우 중요하다.
1234

a=[1,2,3,4]
for i in range(4):   #range는 index를 만들어주는 것 range(4): 0부터 4개 (0,1,2,3)
    print(a[i])      #range(len(a)) 라고 일반적으로 함.
                     # len(a): 그냥 몇갠지
1234
a=['red','green','blue','purple']
for s in a:
    print(s)
red green blue purple

a=['red','green','blue','purple']
for s in range(len(a)):
    print(a[s])
red green blue purple

a=['red','green','blue','purple']
for i,s in enumerate(a) :                    #enumerate: 숫자를 부여
    print(I,s)
0 red 1 green 2 blue 3 purple

a=['red','green','blue','purple']
b=[0.2, 0.3, 0.1, 0.4]
for i, s in enumerate(a) : #앞에있는게 번호, 뒤에있는게 element. enumerate 없으면 변수 2개 못씀
    print("{}:{}%". format(s, b[i]*100))
 red: 20.0% green:30.0% blue:10.0% purple:40.0%

a=['red','green','blue','purple']
b=[0.2, 0.3, 0.1, 0.4]
for s,i in zip(a, b):
   print("{}:{}%". format(s, i*100))
red: 20.0% green:30.0% blue:10.0% purple:40.0%
```

```
a=0
if a==0:
    print("yay!")
```
yay!

```
if a!=0:
    print("yay!")
else:
    print("no")
```
no

```
#시험문제
for i in range(1,3):
    for j in range(3,5):
        print(i*j)
```
3468

```
for i in range(1,3):
    print(i)
    for j in range(3,5):
        print(i*j)
```
1 3 4 2 6 8
순서: 1: 1을프린트, 1을 3,4와 곱함

```
for i in range(1,3):
    for j in range(3,5):
        if j>=4:
            print(i*j)
```
4 8
```
for i in range(1,3):
    if i >= 3:
        for j in range(3,5):
            print(i*j)
```
아무것도안나옴

Numpy

```
import numpy as np
import matplotlib.pyplot as plt
```

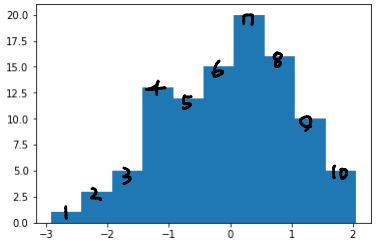
```
np.empty([2,3], dtype='int')
np.zeros([2,3])
np.arange(0,10,2, dtype='float64')
np.linspace(0,10,6, dtype=float)
```

array([0., 2., 4., 6., 8.])

array([0., 2., 4., 6., 8., 10.])

```
X = np.array([[1,2,3],[4,5,6]])
X.astype(np.float64)
np.zeros_like(X)
```

```
data = np.random.normal(0,1, 100)
print(data)
plt.hist(data, bins=10)
plt.show()
```




```
X = np.ones([2, 3, 4])
Y = X.reshape(-1, 3, 2)
np.allclose(X.reshape(-1, 3, 2), Y)
```

```
a = np.random.randint(0, 10, [2, 3])
b = np.random.random([2, 3])
np.savez("test", a, b)
```

```
del a, b
%who  # Print all interactive variables
```

```
npzfiles = np.load("test.npz")
npzfiles.files
```

['arr_0', 'arr_1']

```
npzfiles['arr_0']
```

```
np.savetxt("regression_saved.csv", data, delimiter=",")
!ls -al regression_saved.csv
```

```
arr = np.random.random([5,2,3])
```

```
print(type(arr))
print(len(arr))
print(arr.shape)
print(arr.ndim)
print(arr.size)
print(arr.dtype)
```

```
<class 'numpy.ndarray'>
5
(5, 2, 3)
3
30
float64
```

```
a = np.arange(1, 5)
b = np.arange(9, 5, -1)
print(a - b)
print(a * b)
```

```
[-8 -6 -4 -2]
[ 9 16 21 24]
```

```
a = np.arange(1, 10).reshape(3,3)
b = np.arange(9, 0, -1).reshape(3,3)
```

```
a.sum(), np.sum(a)
a.sum(axis=0), np.sum(a, axis=0)
a.sum(axis=1), np.sum(a, axis=1)
```