

9.3~5

<Praat>

Hz가 높을수록 고주파, 무지개 중 보라색 쪽, 빠른 빨간색, 느린 검은색 부분이 많을수록 성분이 많다.

9.17

<English consonants and vowels>

(ex) gap에서 자음은? 철자와 소리의 구분 year 은 ear에 자음y를 붙인 것

shy she, vision → pair를 이룸

shy she - 혀 닿으면 안됨 --- 무성

vision - 혀가 닿으면 안됨 --- 유성

혀가 닿았다면 jive 와 유사

j(y) yearn youte? year 여 소리 발음기호 j라고 쓸 때도 여 소리

year과 ear이 같으면 안됨 이얼, 여 는 자음이다

jive 소리에서 떨림을 빼면, cheap, chime → cheap chime 은 목이 안떨리지만 jive는 목이 떨린다

(group 으로 묶을 수 있다)

목이 떨고 안떨고

목이 떠는 것 - 유성음 voiced sound, 목이 떨지 않는 것 - voiceless

모음은 모두 voiced

자음 중 voiced → b d g m n ŋ v z vision l w r j chime

코로 나오는 소리 → m n ŋ

입술 두 개 쓰는 것 → p, b

아랫입술과 윗니 → f, v

혀와 윗니 → thigh thy

(모음)

단모음과 복모음

(phonetics)

인지적인 과정이 phonology - 상위, 고정되어 있는

phonetics- 물리적이고, 늘 차이가 있는 physical

→ **a study on speech - speech는 사람이 하는 말에 대한 모든 연구**

a study on sound system - phonology

연속적인 소리 - phonetics

성대의 떨림의 정도에 따라 - 소리의 높이

소리가 바뀌는 것 - 입모양(혀의 위치, 턱) → 턱의 높낮이 때문에 소리가 바뀌는 것 - 주된

요인은 아닐 수 있다

한글은 음절의 반복, 영어는 stress를 기본으로 해서 반복이 된다

한국어는 턱이 말할 때에 맞추어서 간다. 한국어는 턱을 많이 쓰는 language
턱과 반대로 가는 것이 혀, 반면에 stress를 rhythm으로 생각하면 혀를 더 자유롭게 쓸 때

1. 위 내용까지 articulation

2.공기를 타고 가는 과정을 acoustic - 공기가 어떻게 움직여가는지
공기와 소리가 어떻게 되는가 - 물리, 사람이 수반되지 않은 물리

3.auditory - 공기를 타고 오는 소리를 귀로 듣는 것
귓바퀴는 소리를 증폭시키기 위해
고막이 움직이는 것까지 물리 - 청각세포가 신경으로 전달

<Articulation>

The vocal tract

upper structure은 고정

인강(목젖부터 후두까지 있는 긴 관), 후두

palate 입천장

alveolar, hard palate, soft palate(velum), uvula(목젖), pharynx(인두) wall, larynx(후두)
알아야함

lower - lip blade tip front center back root, epiglottis

tract는 관 - 식도와 기도 - epi(덮개) glottis - 기도로 가는 문을 막는 과정

음(소리) oral tract은 막히고, nasal tract은 열려있는 상태

아(소리) nasal tract을 안쓴다 - 달아 놓았다 - velum은 lower or raised 두가지 밖에 없
는데 올라가면 막히고 내려가면 열린다

시험문제 → velum이 raised 됐습니다 nasal tract은 열렸을까요 닫혔을까요? - 막혔다 -
소리의 예는 모든 모음, 자음 중에는 비음을 뺀 모든 자음들 음 은 응 빼고는 음은 응 은
velum이 lower되서 소리가 난다

숨을 쉴 때 velum이 raised 될까 lower가 될까 - lower가 된다, nasal tract은 열린다

nasal sound와 그렇지 않은 sound를 구분하는 - oro-nasal process

(phonation process)

모든 영어의 소리는 유성음과 무성음으로

유성음은 모든 모음과 일부 자음

(phonation process in larynx)

phonation은 voiceless와 voiced 구분 짓는

* 정리 → articulation 유성 무성(성대) , velum에서 lower and raised, 혀 부분에

9.19

larynx = voicebox

voiced : can feel vibration

lips- p

tongue tip - t

tongue body - k

constrictor - lips, tongue tip, tongue body

constriction location : where exactly 앞뒤

constriction degree : how much 상하

→ 각각의 constrictor은 location과 degree에 의해 더 자세하게 specify 된다

<constriction location>

y 에 해당하는 소리가 tongue body를 쓴다 tongue tip 안쓴다

y, g - tongue body쓴다는 관점에서는 똑같지만, location에서는 미세조정된다

(ex)tongue tip - 영어에서 4개정도 - th - tongue tip- location관점에서는 윗니를 hit - 조금 뒤로 오면 alveolar (d t n) - 조금 더 뒤로오면 she - r 도 tongue tip을 쓴다

tip 2개(bilabial, labiodental)

tongue body 2개 (palatal, velar)

tongue tip 4개 정도(dental, alveolar, palato-alveolar, retroflex)

<constriction degree>

upper part vs lower part

d- constriction location: tongue tips -alveolar - constriction degree : stops

p t k - stops - b d g

s z f v th sh- fricatives -소리가 계속 유지된다

r, l, w, j(y) - approximants

- vowels (모음은 자음보다 degree가 더 작다)

모든 자음은 stops(폐쇄형), fricatives(마찰), approximants

(ex) m n 응은 어디에 해당? stops - 코가 열렸을 뿐이지 막았으므로)

velum raised, larynx의 틈을 glottis open tongue tip, alveolar, stop - t

모든 모음은 constrictor로서 tongue body 만 쓴다

모음과 같은 constrictor를 쓰는 자음의 예시 - k, + velum lower되면 응 소리

응 소리는 glottis가 closed가 되어 된다

<acoustic in praat>

duration, pitch, intensity(pitch와는 완전히 독립적으로 갈 수 있다), spectrogram(프리즈 음로 빛을 분산시켜서 frequency관점에서 분석하는) 검은 띠? -first, second formant 줄여

서 f1, f2 - f1이 뭐냐 f2가 뭐냐에 따라 모음이 무엇인지를 결정한다
pitch setting에서 range를 남자는 65~200 , 여자는 150~300 로 측정해야 한다.
show formant :점을 따라 빨강게 보이는 부분, check를 안하면 안나온다

<vowel acoustics - 한학기 동안 배울 theme>

어떻게 만들어지는가 mechanism 이 뭐가?

praat

wave

129Hz - 1초에 성대가 129번 떨린다- 1 나누기 0.007852 = 127.358/s

-new - create sound as pure tone - 자신의 pitch를 똑같이 적어서
소리작으면 modify에서 100정도 증폭하면 됨 --- 소리 높이가 똑같다
신호처리 DSP (digital signal processing) - 디지털 신호처리

9.24

1Lips - p BL STOP

tip

Tongue body

2velum - raised

3larynx - open

b - 는 p와 같은데 larynx가 closed

d - tongue tip(alveolar), stop

z - tongue tip(alveolar), fricatives

n - tongue tip(alveolar), stop, velum(lowered)

n 상태에서 larynx가 open 되면 숨쉬는 것과 같다

CL CD

LIPS

TT

TB

VELUM

LARYNX

tip 2개(bilabial, labiodental)

tongue body 2개 (palatal, velar)

tongue tip 4개 정도(dental, alveolar, palato-alveolar, retroflex)

p t k - stops - b d g n

s z f v th sh- fricatives -소리가 계속 유지된다

r, l, w, j(y) - approximants

Praat(시험)

the number of occurrences of a repeating event per second : Hz

mega는 백만, 1억 2천7백만 - 127Hz (ex) sine wave(반복되는 주파수) - 두 가지 성분을 가지고 있다 frequency(반복되는 것) + 크기에 의해서 결정된다

(ex) 파도

repeating event = vibration of vocal folds (떨리는 것 한쪽을 vocal fold) 두 개이므로 folds

sound quality는 다르다, 높이는 똑같겠지만

입모양에 의해서 바뀌는 것이지 성대에서는 높낮이 정도만 다르다

sine wave -가장 기본적인 형태다 (frequency 와 magnitude)에 의해 sine wave가 결정된다

이 세상에 존재하는 모든 signal은 (소리를 포함한) 여러 다르게 생긴 sine wave의 결합으로 표현된다 - 중요한 발견이고, 여러 신호와 관련된 엔지니어링을 완전히 바꾸어 놓은

모든 신호는 조금씩 조금씩 다른 sine wave의 합으로 표현될 수 있다

→ complex한 세상을 simple하게 간단하게 (푸리에의 발견)

첫 번째 (magnitude 크지만, frequency는 작다) frequency가 작으면 소리가 저음

두 번째는 첫 번째 보다 frequency가 높다 첫 번째보다 두 배 빠르다,

세 번째는 첫 번째보다 세 배 빠르다

가로축 시간 세로축 value (숫자 값) - x축은 frequency y축은 amplitude로 변환을 한 것이다

각각의 sine wave는 변환가능하다

x축이 frequency y축이 amplitude로 되는 것을 spectrum

주변에서 듣는 spectrum은 complex tone

equalize

위에서 밑으로 sythesis 밑에서 어떤 결로 이루어졌는지를 보는 것을 analysis

magnitude는 두 번째가 가장 작다, 첫 번째가 가장 크다

마지막은 세 그림의 합으로 나타내어진다

sine wave의 합은 복잡한 신호로 만들 수 있다 ↔ 복잡한 신호는 단순한 sine wave들의 합으로 표현될 수 있다

마지막 꺼 1초에 100번 반복되는

simplex tone 은 sine wave complex tone은 sine wave 아니다

amplitude=magnitude

떨림은 주기와 같다

아 (simplex 톤의 합으로 이루어진다, 제일 slow한 sine wave의 frequency 가 우리말의

pitch와 동일하다) 진동수 = frequency (1초에 몇 번 반복되는가)(Hz)

pure tone은 simplex tone

spectrogram은 spectrum을 시간축으로 계속 올려놓은 것이다

나의pitch와 같다 133 (젤 처음 나온)

simplex tone을 여러 개 해놓으면 그래도 반복되는 게 있는데

가장 처음 것과 일치

성대에서 떨리는 소리를 그대로 capture - source (larynx에서)

성대로부터 입술까지(튜브)가 어떻게 달라지는 가 Filter

모든 사람의 source의 패턴은 조금씩 줄어든다

음의 높낮이는 젤 첫 번째 나오는 것과 같음 fundamental 하다 f_0 fundamental frequency

= f_0 = pitch = the number of vocal folds vibration in a second

fundamental에 해당하는 것은 amplitude가 크다

harmonics (배를 이룬다) x_2 x_3 x_4

여성은 더 듬성듬성 생긴다 첫 시작이 크고

남자가 갖는 배음의 수가 더 많다

배음의 구조는 동일한데 amplitude의 패턴이 깨졌다 원래는 smoothly decreased된 게 (filter)

spectrogram 도 x축이 time, y축이 frequency

검은 부분이 크기가 세다(amplitude) = 위쪽으로 갈수록 열어진다

no frequency쪽으로 가면서 값이 커진다

9.26

complex wave 분석하면 어떤 성분들이 얼마나 에너지를 가지고 있는지 분석가능하다

맨 마지막 그림이 spectrum 이라고 한다

waveform은 x축이 시간, y축은 그냥 value

sine wave는 소리

한 슬라이스가 스펙트럼

spectrogram에서 x축은 시간, y축은 frequency (100% 시험문제)

spectral analysis

sinewave의 x_0 가 정해지고, 그것들의 배음의 합으로 우리들의 소리가 정해진다

pitch - 첫 번째 frequency와 일치 - f_0 = pitch - 단위는 Hz, 115Hz, 230Hz --- 배음으로 된다

등간격으로, gradually decreasing 되는데 amplitude

이정도 크기의 sinewave가 하나있고, 좀 작은 sinewave가 있고, 좀 더 작은 sinewave가 있고, 그러나 frequency는 점점 빨리지는 2배, 3배, 4배

저주파에서만 높고, 고주파로 갈수록 에너지가 약해지는

harmonics는 sine wave들의 배음으로 이루어져 있다

검은 곳 peak, 흰색 회색으로 갈수록 valley - 이런것들을 만들어주는게 입모양(filter)의 역할

filter

EGG - 성대에서 그대로 녹음한 것

it is filtered by the vocal tract(VT)

peaks(mountains:frequencies VT likes = formants_

valleys: frequencies VT does not like

모든 사람들의 아 의 산맥 패턴이 똑같이 나타난다 - 매우 중요

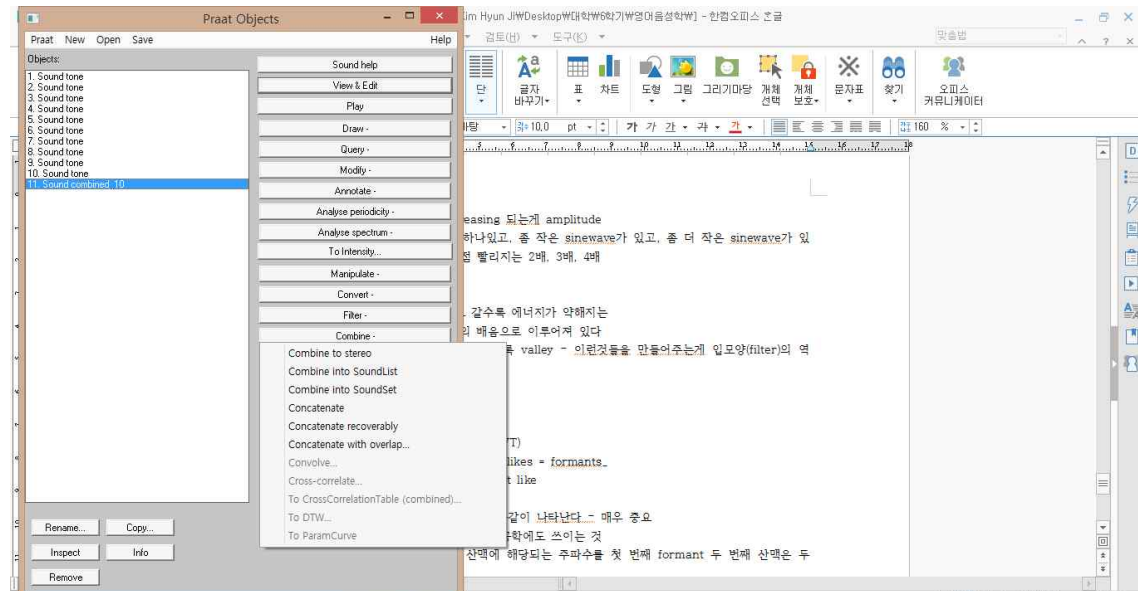
이런패턴을 가지고 음성인식, 다른 공학에도 쓰이는 것

제일 저주파로부터 첫 번째 산맥에 해당되는 주파수를 첫 번째 formant 두 번째 산맥은 두 번째

f0는 켈 첫 번째 harmonics ↔ source에서 첫 번째 산맥이 만들어지는게 formant

shape에 따라 어떤 소리를 좋아할지 결정되어있다 (ex)콜라병이 어떤 소리와 딱 맞게 맞춰지냐

입도 콜라병이랑 똑같다 - 어떤 특정한 음의 높이를, 입모양을 좋아할 수 밖에 없는데, 이게

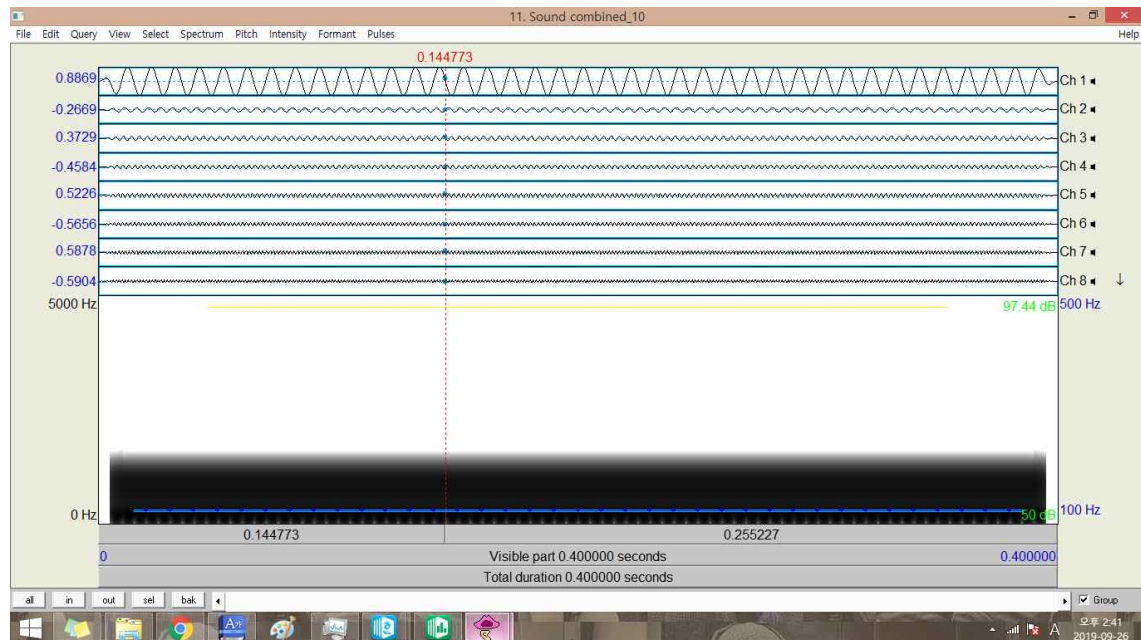


formant와 일치한다 - 산맥이라고 했을 때 아 라는 입모양이 좋아하는 산이있고, 어떤 소리의 높이들은 싫어하는게 있는데 이것이 valley이다

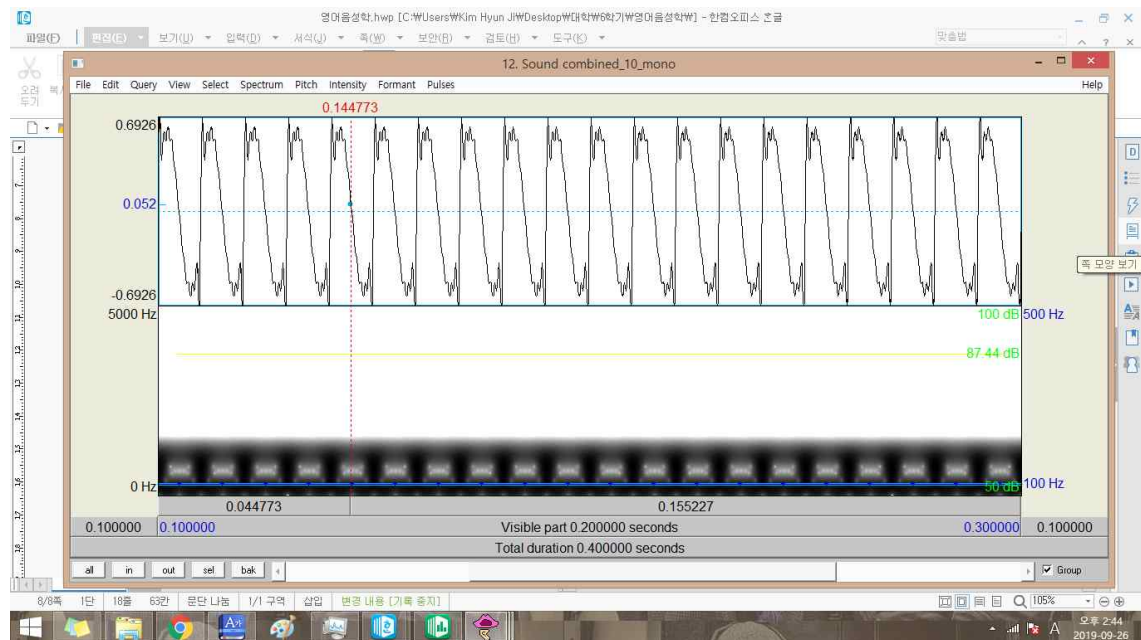
harmonics는 굵은 소리, 두드리는 소리는 안됨

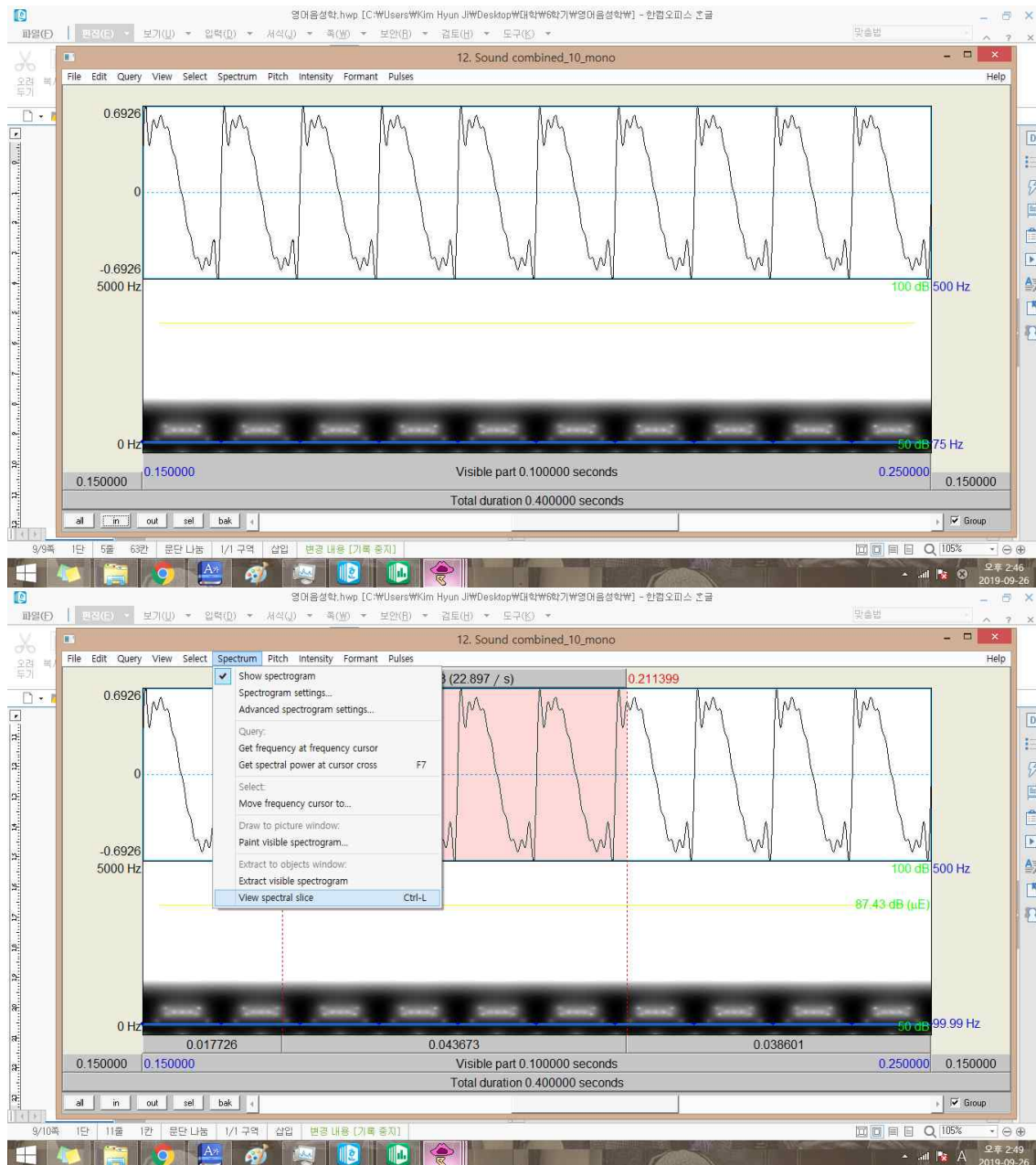
guitar소리가 우리의 입에서 나오는 소리(harmonics)와 비슷하다

기타 소리는 complex tone - fundamental 만 따서 만들면 pure tone으로 만들 수 있다



simple tone을 모아놓은 것 - complex tone아님



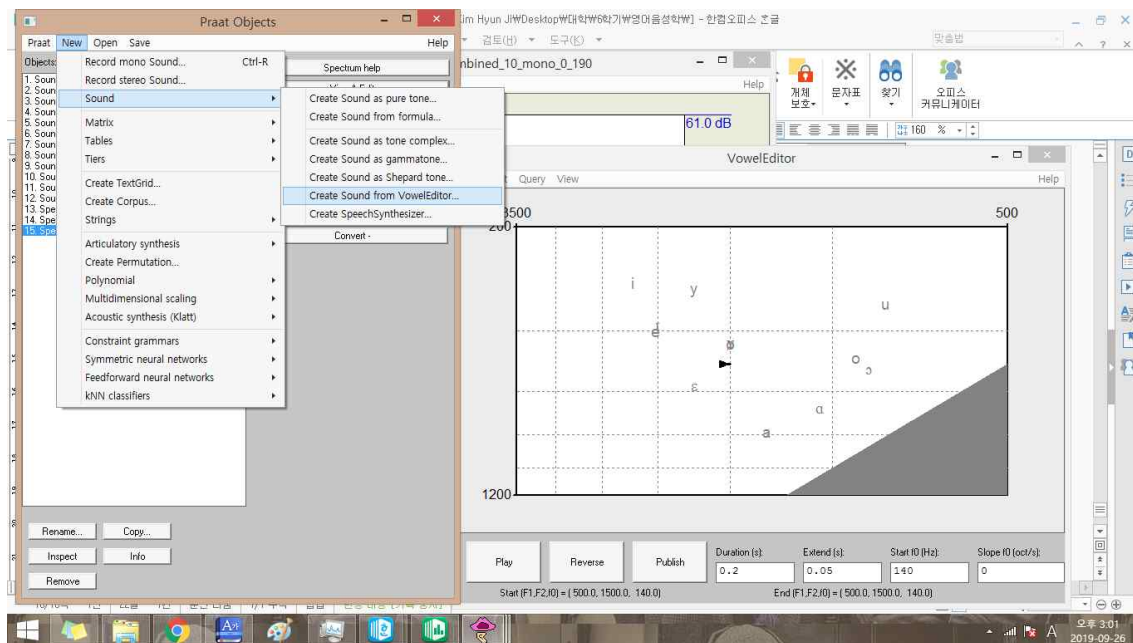


반복주기 - 10개의 sine wave 중에서 f_0 의 sine wave와 일치한다
 인지심리학적으로 100Hz소리와 같게 들린다
 무한대로 가면, peak 하나 000000 peak하나 0000 - perse train
 10개의 peak가 나와야 한다

도장찍는 역할을 입모양이 한다
 첫 번째 튀어나온 것이 first formant 라고 한다.
 f_0 -젤 첫 번째 나오는 frequency

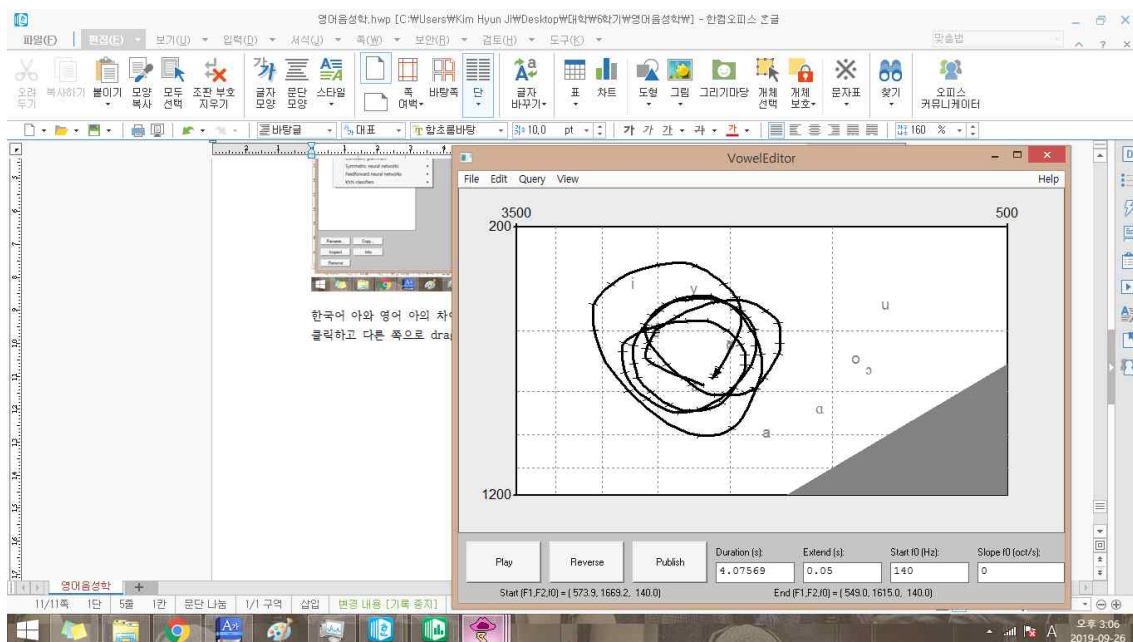
f1과 f2로만 모든 모음을 만들어낼 수 있다

입의 위치와 같다 - f1은 높낮이(혀의) height, f2는 front or back을 결정



한국어 아와 영어 아의 차이? 영어가 더 back, low

클릭하고 다른 쪽으로 drag하고 release 하면 - 이중모음



10.1

코딩 - 자동화

ex) praat

자동화를 하는 이유? 똑같은 것이 계속반복되게

sine wave는 숫자들(하나의 선이 아닌)- 하나하나 더할수도 있다

반복되는 것의 자동화

컴퓨터에서 쓰는 모든 것은 코딩으로 되어 있다

태블릿pc, 스마트폰- 모두 컴퓨터, 반복되는 것을 시킬 때 코딩해서 넣어야 한다

program language

사람의 언어의 공통점? 단어, 단어를 어떻게 combine하느냐

단어가 있어야 하고, 단어가 뭘까? 단어는 그 속에 의미(정보)가 들어있다. 단어는 정보를 담는 그릇

그릇이 하나가 있는데, 정보로서의 사과를 담으면 단어는 사과가 되고, 물을 담으면 물이 되고, 그 그릇에 여러 가지가 바뀔 수 있다

어떤 단어를 선택할 것인가,

단어에 해당하는 부분이 변수 variable

정보를 담는 그릇이 변수로써 필요한 것이다

정보를 가지고 뭔가를 기계한테 무슨 이야기를 할 때, 기계에게도 문법이 필요하다

기계와의 문법은 그렇게 어렵지 않다

1. 변수에 정보를 넣는 것

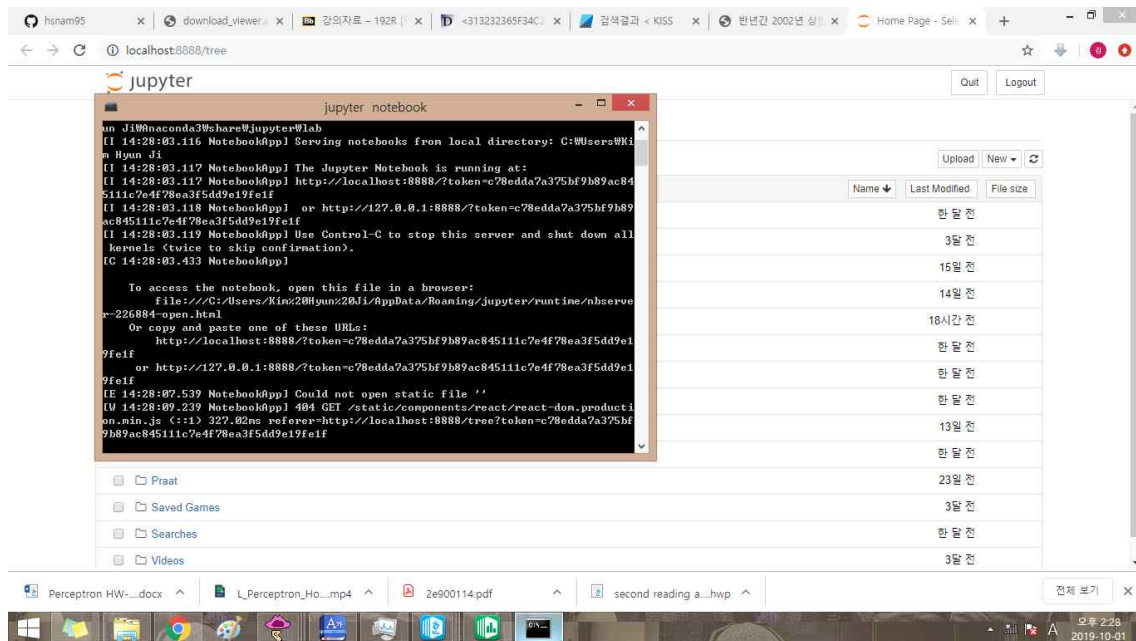
2. if 라는 문법 if 조건문

3. 여러번 반복 for

4. 제일 중요 - 함수의 정의 - 중간에 뭐가 돌아가는지는 모르겠지만, 입력을 넣으면 내가 원하는 출력이 나와야 한다

함수 - packaging -- 여러 가지 명령에 따라

함수는 재사용도 가능하다



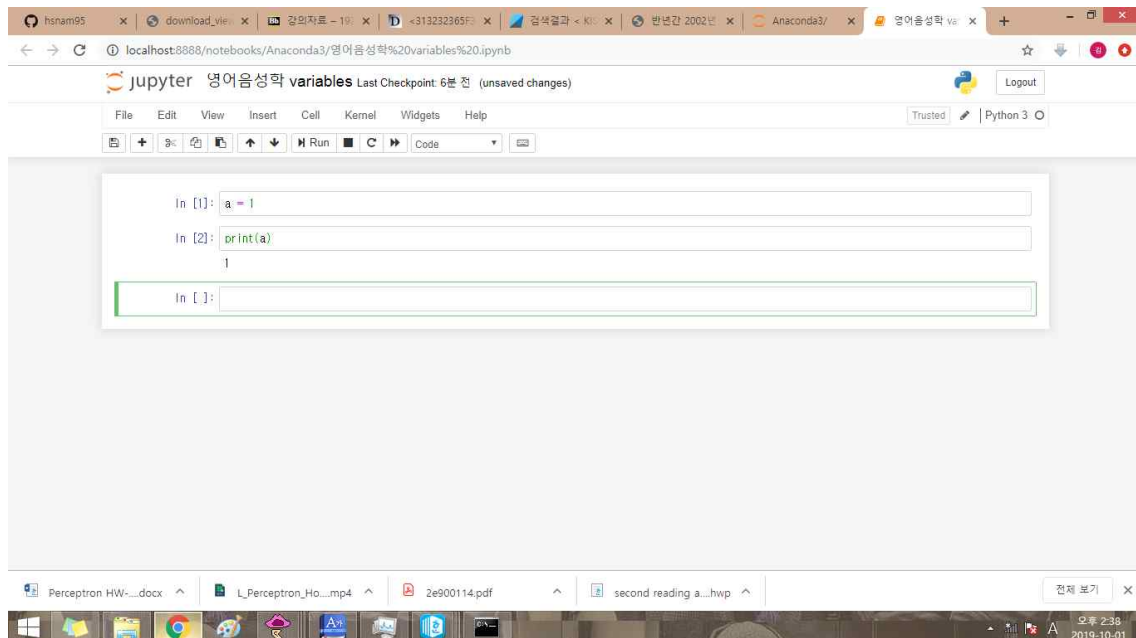
jupyter notebook

숫자, 문자 -

오른쪽 정보를 왼쪽의 variable로 assign한다 (=)

a = 1, 1이 정보, a는 variable, 순서도 바뀌면 안됨

1이라는 정보를 a라는 variable에 넣는다



함수는 이미 만들어져 있어야 하거나, 우리가 스스로 만들 수도 있다

print도 함수역할을 한다

함수이름을 치고 괄호안에다가 입력

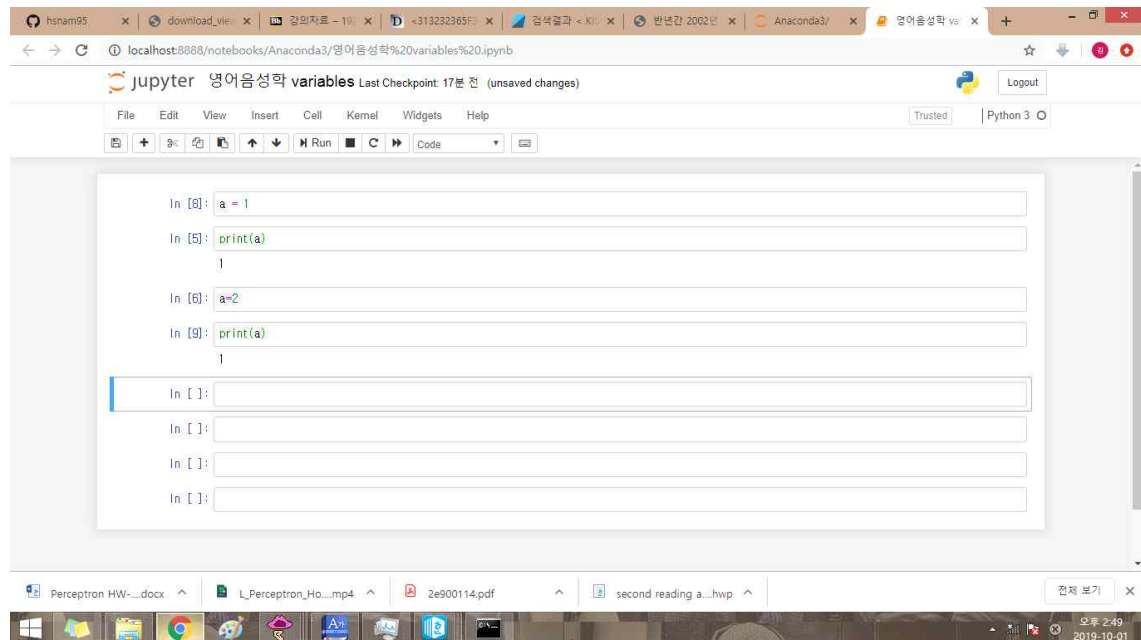
print라는 함수의 입력은? a , 입력이라고 표시해주는 것이 괄호이다

a - 셀 위에 추가

b - 셀 밑에 추가

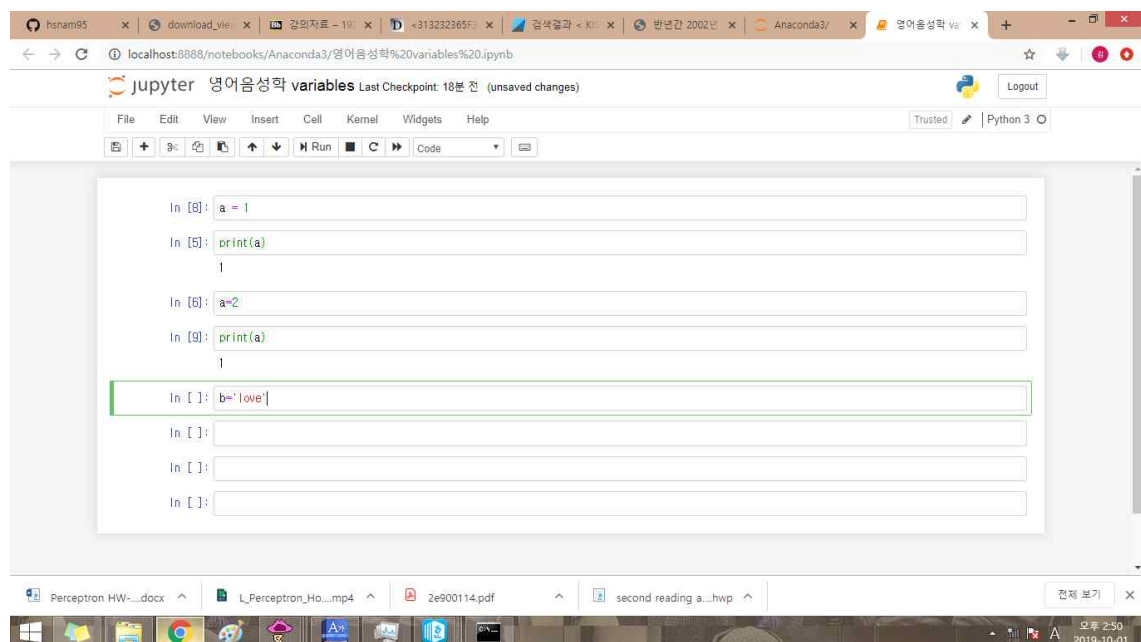
x - 셀 삭제

처음에 a=1, 그 후에 a=2 넣으면, overwrite해서, a=1은 없어진다



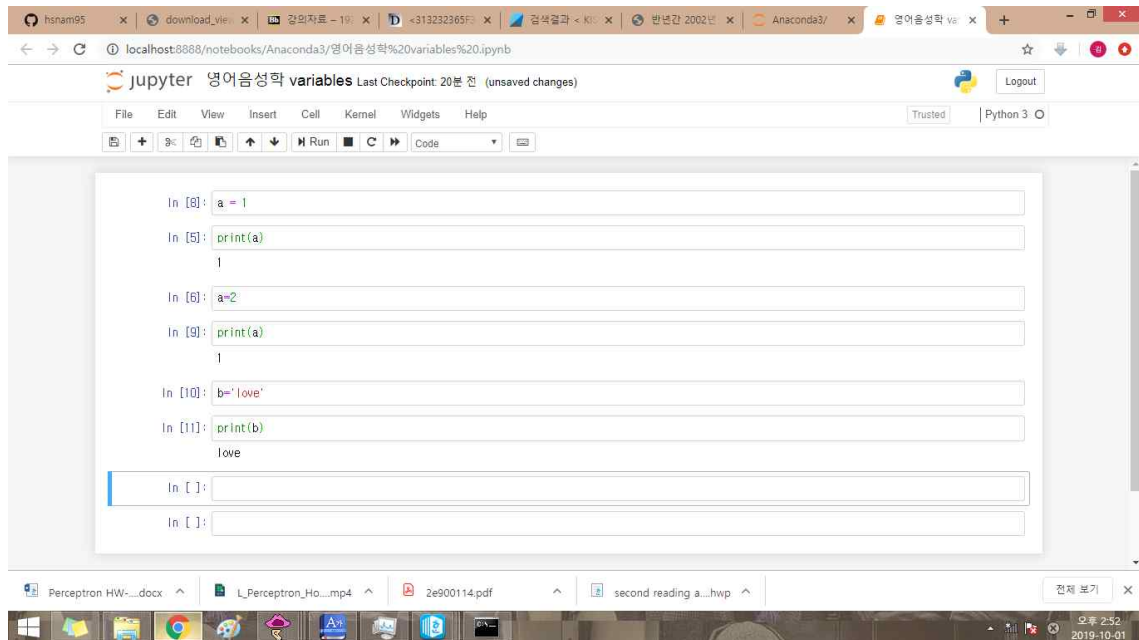
```
In [8]: a = 1
In [5]: print(a)
1
In [6]: a=2
In [9]: print(a)
1
In [ ]:
In [ ]:
In [ ]:
In [ ]:
```

셀 단위로 계산이 된다



```
In [8]: a = 1
In [5]: print(a)
1
In [6]: a=2
In [9]: print(a)
1
In [ ]: b='love'
In [ ]:
In [ ]:
In [ ]:
```

문자는 ' '해줘야 함, 아니면 변수로 인식, shift + Enter = 실행버튼

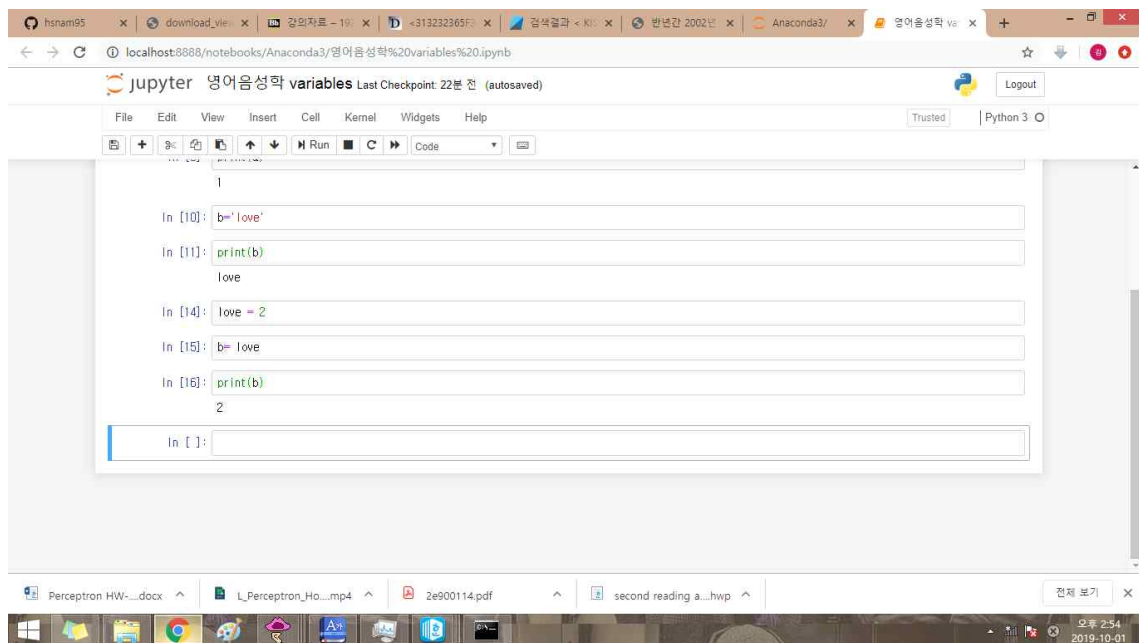


A screenshot of a Jupyter Notebook interface. The browser tabs at the top include 'hnam95', 'download_vie...', '강의자료 - 1P...', '<313232365F...', '검색결과 < K...', '반년간 2002년', 'Anaconda3/', and '영어음성학 variables'. The notebook title is 'variables Last Checkpoint: 20분 전 (unsaved changes)'. The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The toolbar shows icons for new, open, save, run, and other actions. The code area contains the following cells:

```
In [8]: a = 1
In [5]: print(a)
1
In [6]: a=2
In [9]: print(a)
1
In [10]: b='love'
In [11]: print(b)
love
In [ ]:
In [ ]:
```

The bottom of the window shows a Windows taskbar with various application icons and a system clock indicating '오후 2:52 2019-10-01'.

quote가 안되어 있으면 변수로 인식,

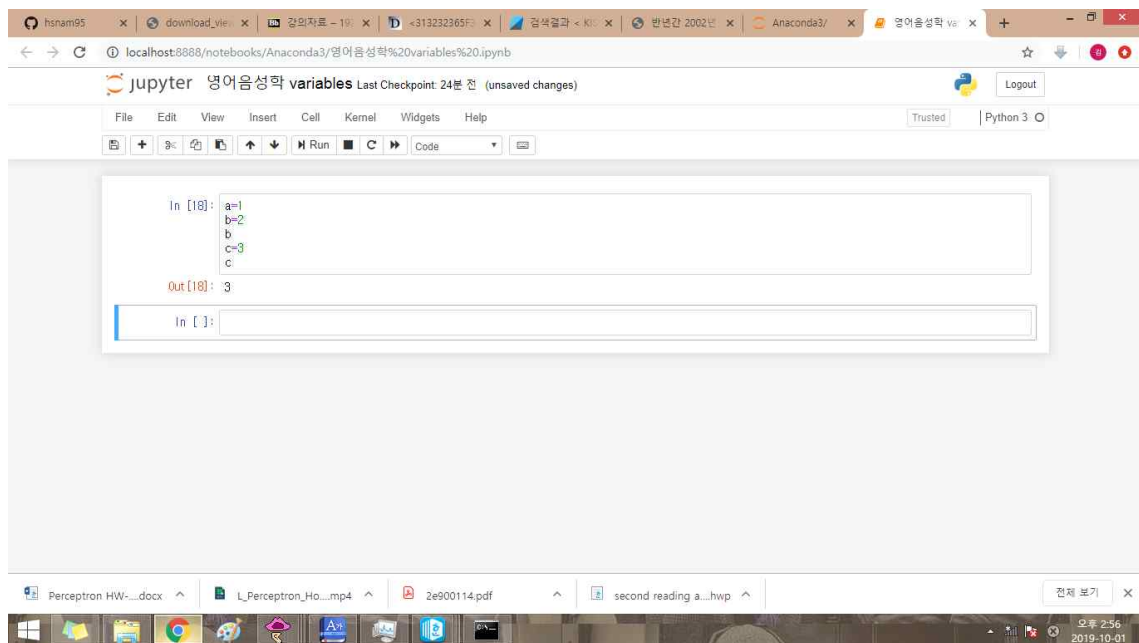


A screenshot of a Jupyter Notebook interface. The browser tabs at the top include 'hnam95', 'download_vie...', '강의자료 - 1P...', '<313232365F...', '검색결과 < K...', '반년간 2002년', 'Anaconda3/', and '영어음성학 variables'. The notebook title is 'variables Last Checkpoint: 22분 전 (autosaved)'. The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The toolbar shows icons for new, open, save, run, and other actions. The code area contains the following cells:

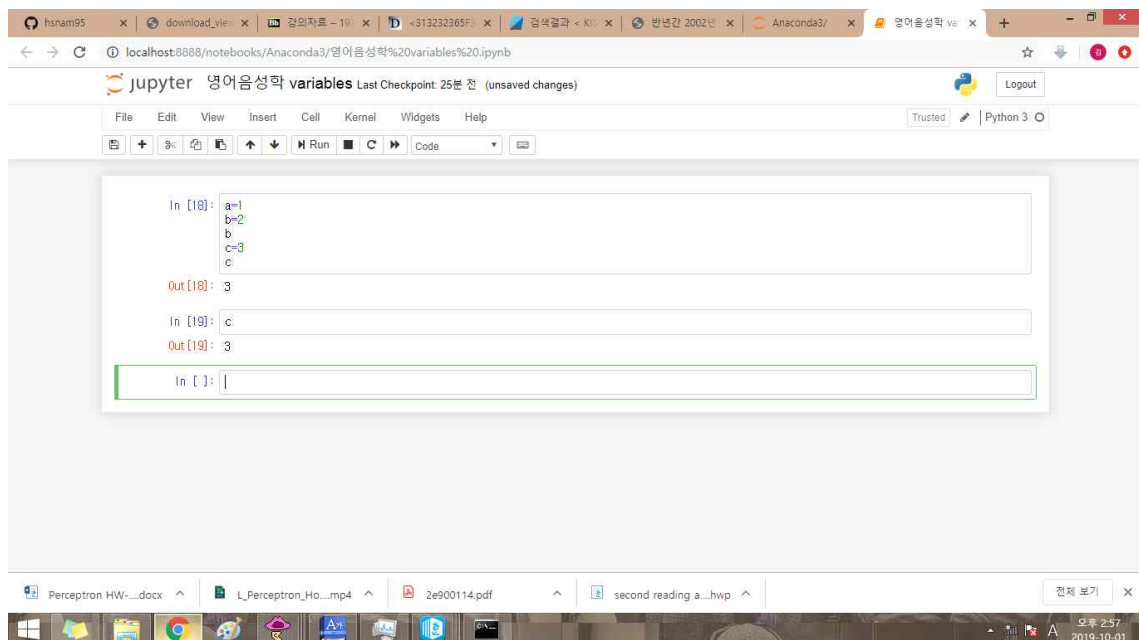
```
1
In [10]: b='love'
In [11]: print(b)
love
In [14]: love = 2
In [15]: b= love
In [16]: print(b)
2
In [ ]:
```

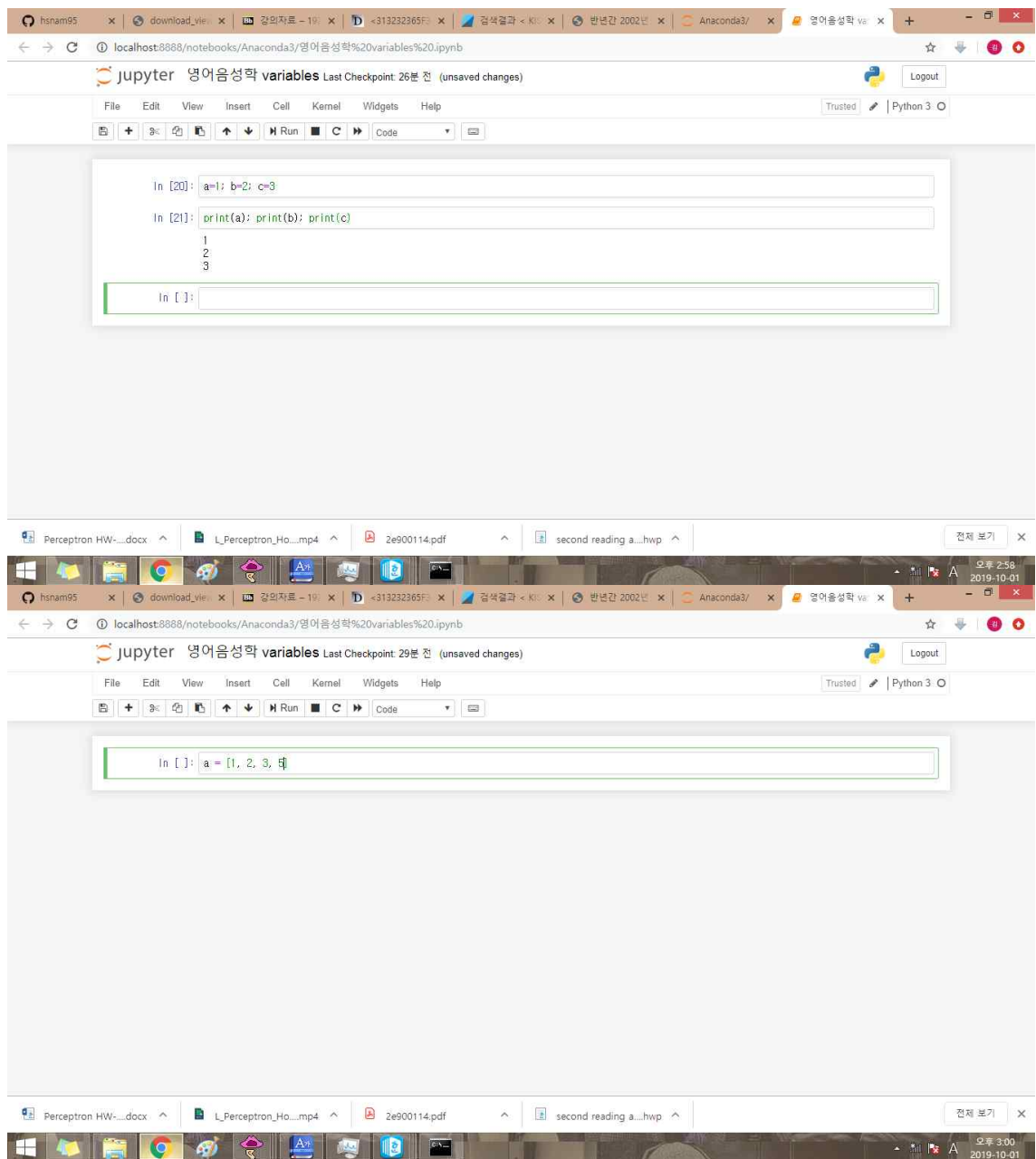
The bottom of the window shows a Windows taskbar with various application icons and a system clock indicating '오후 2:54 2019-10-01'.

```
love = 2
b = love
print(b) = 2
```



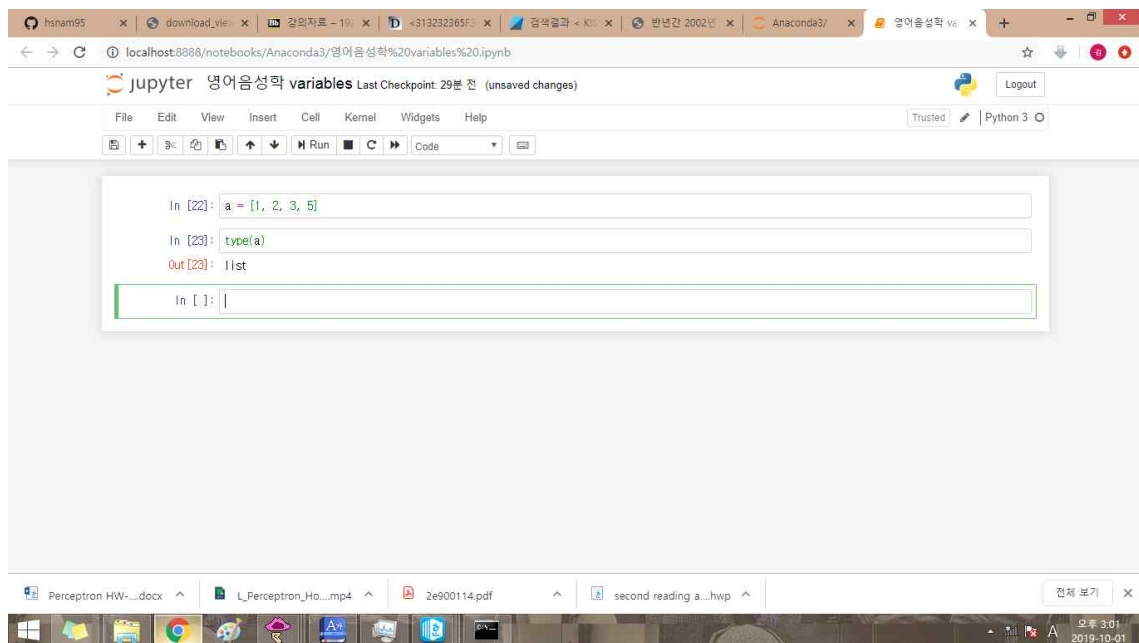
젤 마지막에 변수이름을 치면 print out 해준다



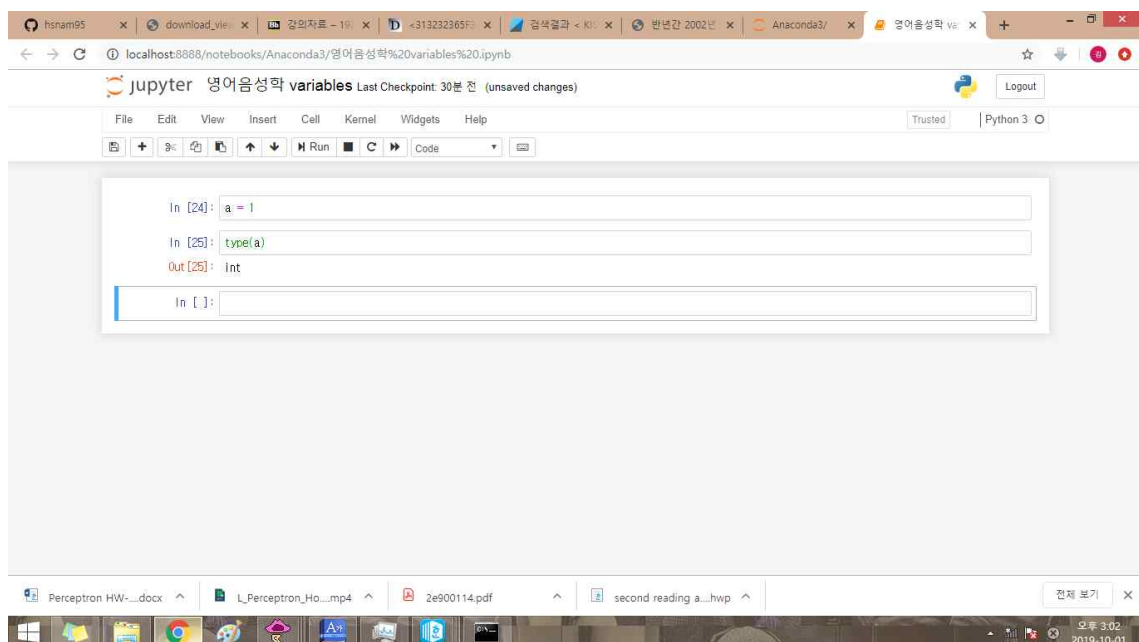


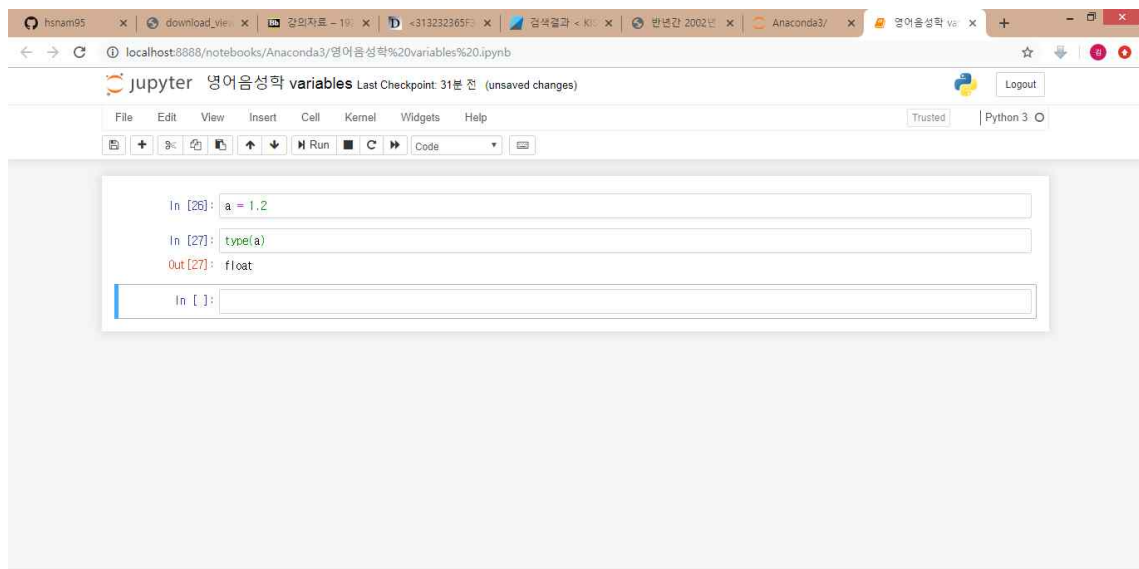
한꺼번에 넣어주는 것이 list

```
list = []
```

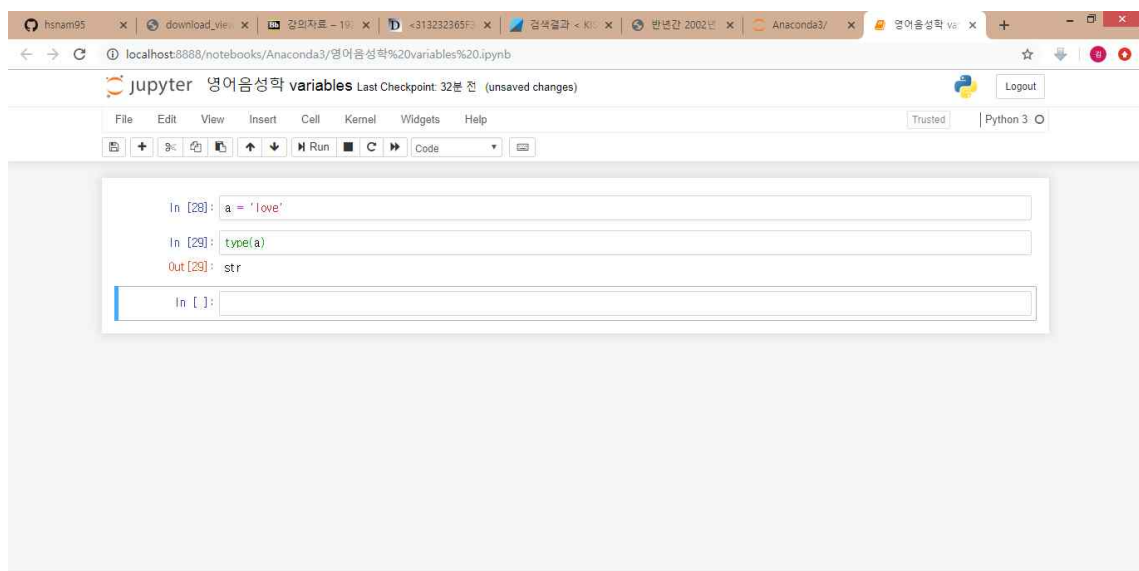



type() 괄호안에는 변수만 들어갈 수 있다

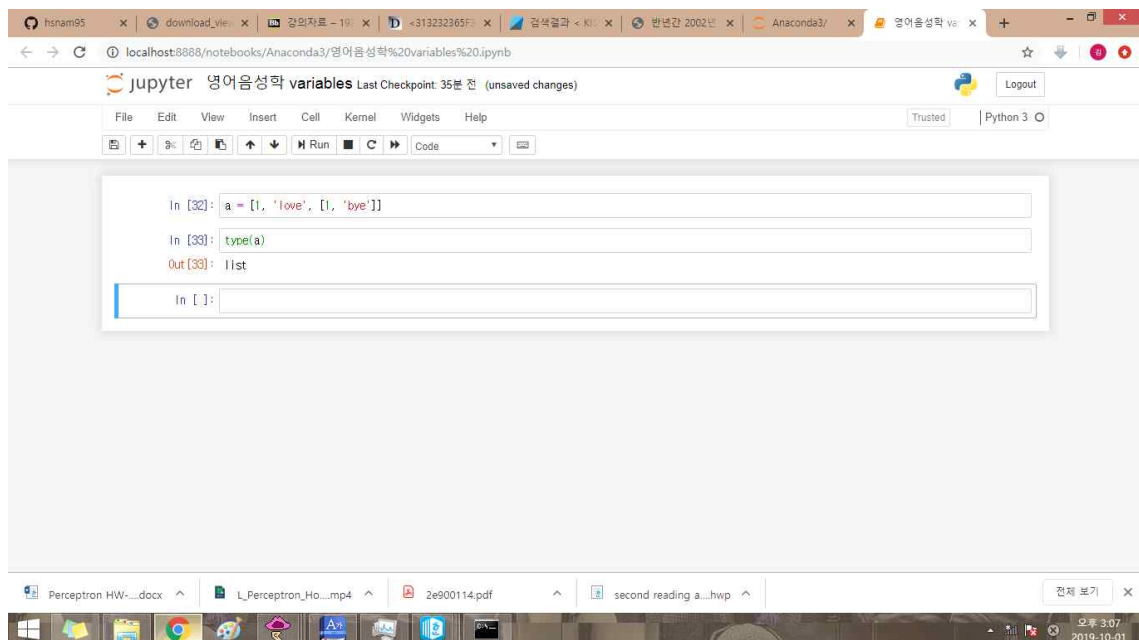
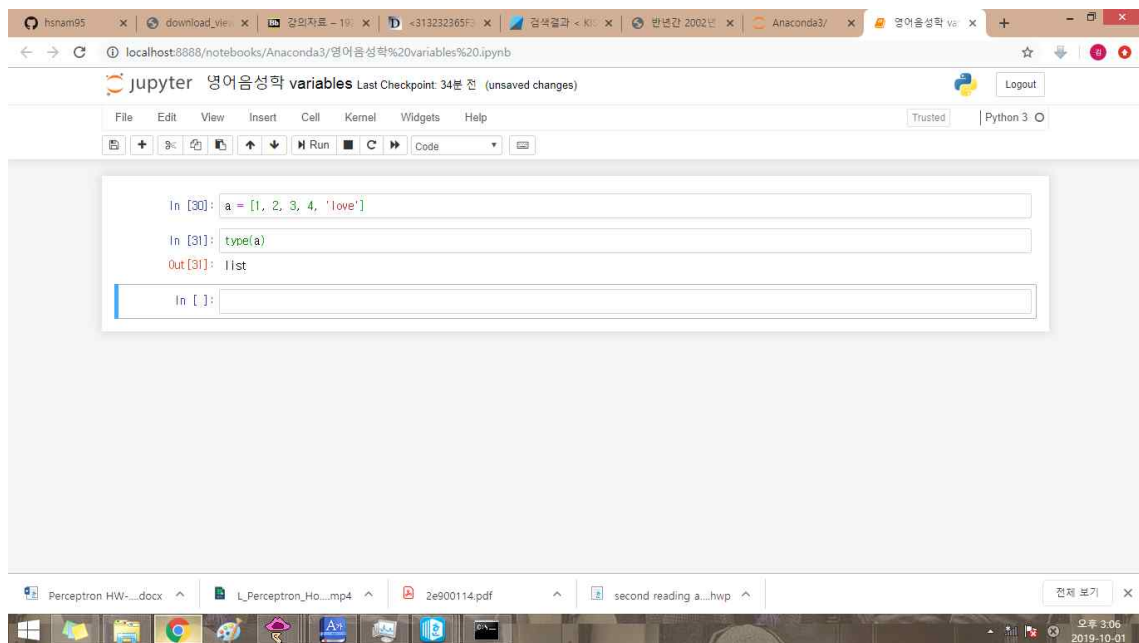




float 는 실수

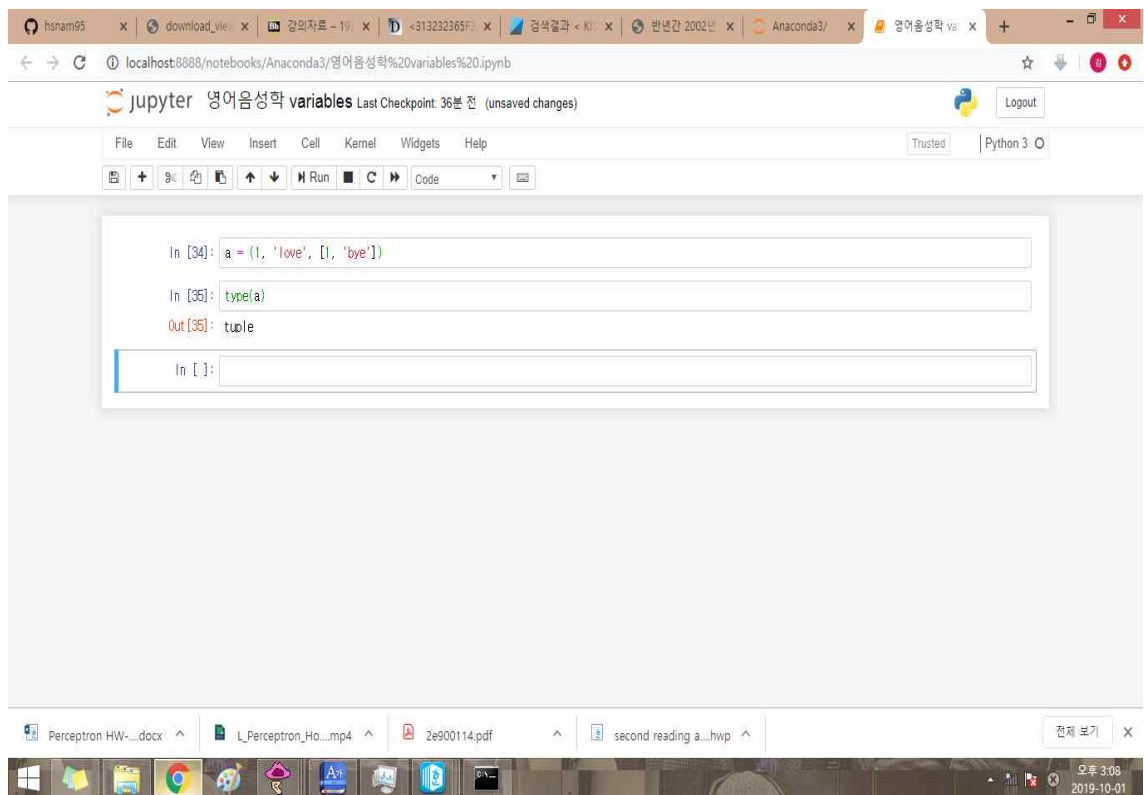


문자는 string

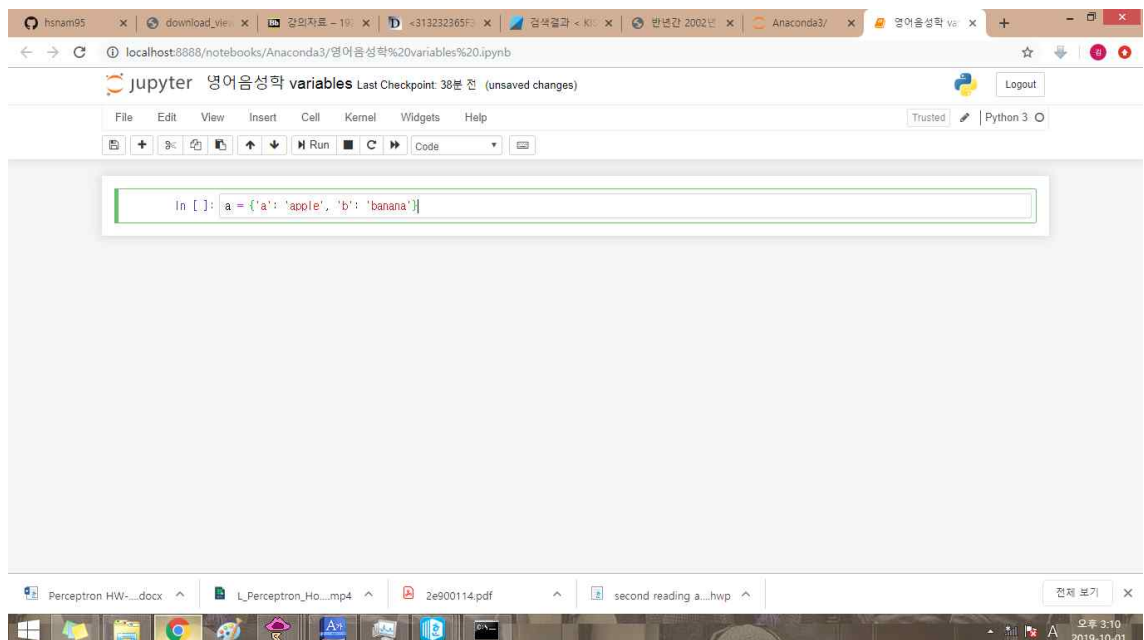


list속에 있는 item은 세 개로 구성되어 있다

1 - int, 'love' - str, [1, 'bye'] - list



괄호를 쓰면 tuple, []를 쓰면 list
tuple은 보안에 강하다, 바꾸기가 힘들다



dictionary

표제어와 설명

comma로 구분되고, 쌍을 묶어주는 것은 콜론

hsnam95 x download_view x 강의자료 - 19 x <313232365F x 검색결과 < K x 반년간 2002년 x Anaconda3/ x 영어음성학 ver x + - x

localhost:8888/notebooks/Anaconda3/영어음성학%20variables%20.ipynb

jupyter 영어음성학 variables Last Checkpoint: 40분 전 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Notebook saved Trusted Python 3

```
In [36]: a = {'a': 'apple', 'b': 'banana'}
In [37]: type(a)
Out[37]: dict
In [ ]:
```

Perceptron HW...docx L_Perceptron_Ho...mp4 2e900114.pdf second reading a...hwp 전체 보기 x

오후 3:12 2019-10-01

Papago x 61집-4KoPub_41.hwp x Anaconda3/ x Untitled - Jupyter Notebook x -Copy1 - Jupyter Notebook x + - x

localhost:8888/notebooks/Anaconda3/-Copy1.ipynb#

jupyter -Copy1 Last Checkpoint: 하루 전 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

10/10

```
In [2]: a = {'a': 'apple', 'b': 'banana'}
In [3]: type(a)
Out[3]: dict

In [1]: a= {'a':"apple", "b":"orange", "c":2014}
print(type(a))
print(a['a'])
<class 'dict'>
apple

In [2]: a=[(1,2,3),(3,8,0)]
print(type(a))
a
<class 'list'>

Out[2]: [(1, 2, 3), (3, 8, 0)]
```

-Copy1.ipynb -Copy1.html ipynb.ipynb phonetics2019 (4).pptx ipynb.html 전체 보기 x

오후 10:37 2019-10-13

Browser tabs: Papago, 61집-4KoPub_41.hwp, Anaconda3/, Untitled - Jupyter Notebook, -Copy1 - Jupyter Notebook

Address bar: localhost:8888/notebooks/Anaconda3/-Copy1.ipynb#

Jupyter Notebook Interface: -Copy1 Last Checkpoint: 하루 전 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run button: Run

```
In [5]: a=[1,2]
        b=[3,4]
        c=a[1]*b[1]
        c

Out[5]: 6

In [6]: a=1 ; print(type(a))
        <class 'int'>

In [7]: a=1 ; a=float(a) ; print(type(a))
        <class 'float'>

In [8]: a=1.2 ; a=int(a) ; print(type(a))
        <class 'int'>

In [9]: a='123' ; print(type(a)) ; print(a[1])
        <class 'str'>
        2

In [10]: a=1.2 ; a=int(a) ; print(a)
```

Taskbar: -Copy1.ipynb, -Copy1.html, ipynb.ipynb, phonetics2019 (4).pptx, ipynb.html, 전제 보기

System tray: 오후 10:38 2019-10-13

Browser tabs: Papago, 61집-4KoPub_41.hwp, Anaconda3/, Untitled - Jupyter Notebook, -Copy1 - Jupyter Notebook

Address bar: localhost:8888/notebooks/Anaconda3/-Copy1.ipynb#

Jupyter Notebook Interface: -Copy1 Last Checkpoint: 하루 전 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run button: Run

```
In [10]: a=1.2 ; a=int(a) ; print(a)
        1

In [ ]:

In [11]: a= 123 ; print(type(a)) ; print(a[1])
        <class 'int'>

        Traceback (most recent call last):
      <ipython-input-11-7141cf7f2275> in <module>
----> 1 a= 123 ; print(type(a)) ; print(a[1])

        TypeError: 'int' object is not subscriptable

In [12]: a= {'1':'apple'}
        print(a[1])
        apple

In [14]: a=[(1,2,3),(3,0,0)]
```

Taskbar: -Copy1.ipynb, -Copy1.html, ipynb.ipynb, phonetics2019 (4).pptx, ipynb.html, 전제 보기

System tray: 오후 10:38 2019-10-13

Browser tabs: Papago, 61집-4KoPub_41.hwp, Anaconda3/, Untitled - Jupyter Notebook, -Copy1 - Jupyter Notebook

Address bar: localhost:8888/notebooks/Anaconda3/-Copy1.ipynb#

Jupyter Notebook Interface: -Copy1 Last Checkpoint: 하루 전 (autosaved) | Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

Run | Markdown

```
In [14]: a=[(1,2,3),(3,8,0)]
         print(a[0])
         type(a[0])
Out[14]: (1, 2, 3)
         tuple

In [18]: a=[(1,2,3),(3,8,0)]
         print(a[0])
         type(a[0])
Out[18]: (1, 2, 3)
         tuple

In [2]: s='abcdef'
        print(s[0],s[5],s[-1],s[~-6])
        print(s[1:3], s[1:], s[:3], s[:])
Out[2]: a f f a
        bc bcdef abc abcdef

In [22]: n=[100,200,300]
```

Browser tabs: -Copy1.ipynb, -Copy1.html, ipynb.ipynb, phonetics2019 (4).pptx, ipynb.html

Address bar: localhost:8888/notebooks/Anaconda3/-Copy1.ipynb#

Jupyter Notebook Interface: -Copy1 Last Checkpoint: 하루 전 (autosaved) | Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

Run | Markdown

```
In [22]: n=[100,200,300]

In [20]: len(s)
Out[20]: 6

In [23]: len(n)
Out[23]: 3

In [3]: s[1]
Out[3]: 'b'

In [16]: s= ' this is a house built this year. \n'
         s
Out[16]: ' this is a house built this year. \n'

In [17]: result = s.find('house')
         result
Out[17]: 11
```

Browser tabs: Papago, 61집-4KoPub_41.hwp, Anaconda3/, Untitled - Jupyter Notebook, -Copy1 - Jupyter Notebook

Address bar: localhost:8888/notebooks/Anaconda3/-Copy1.ipynb#

Jupyter Notebook Interface:

- Menu: File, Edit, View, Insert, Cell, Kernel, Widgets, Help
- Buttons: Run, Stop, Restart, Clear, Markdown
- Language: Python 3

Code cells:

```
Out [17]: 11
```

```
In [18]: tokens=s.split(' ') # ' ' 안에 아무것도 없으면 공백을 기준으로, 두연가가 있으면 그것을 기준으로
tokens
```

```
Out [18]: ['', 'this', 'is', 'a', 'house', 'built', 'this', 'year.', '\n']
```

```
In [19]: tokens=s.split() # ' ' 안에 아무것도 없으면 공백을 기준으로, 두연가가 있으면 그것을 기준으로
tokens
```

```
Out [19]: ['this', 'is', 'a', 'house', 'built', 'this', 'year.']
```

```
In [20]: s=' '.join(tokens)
s
```

```
Out [20]: 'this is a house built this year.'
```

```
In [21]: s=', '.join(tokens)
s
```

```
Out [21]: 'this, is, a, house, built, this, year.'
```

Taskbar: -Copy1.ipynb, -Copy1.html, ipynb.ipynb, phonetics2019 (4).pptx, ipynb.html, 10:40 2019-10-13