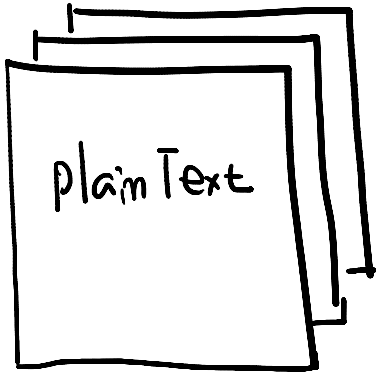
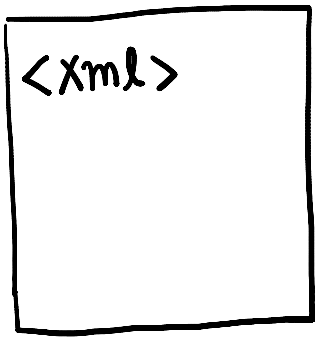
파이썬 자연어 처리

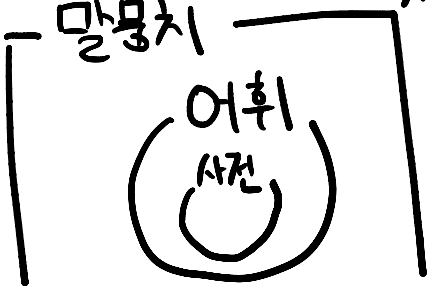
이성주
seongjoo@codebasic.io

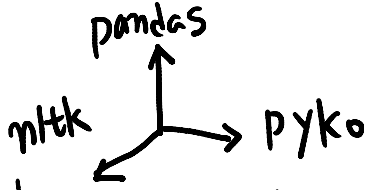
Day 3

말뭉치









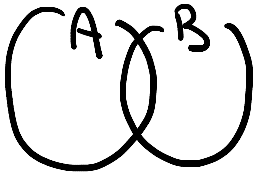
nltk.CorpusReader

X Pyko

Sejong CorpusReader

Plaintext ← gutenberg

Categorized



Tagged Word/tag → 아버지, 가, 방

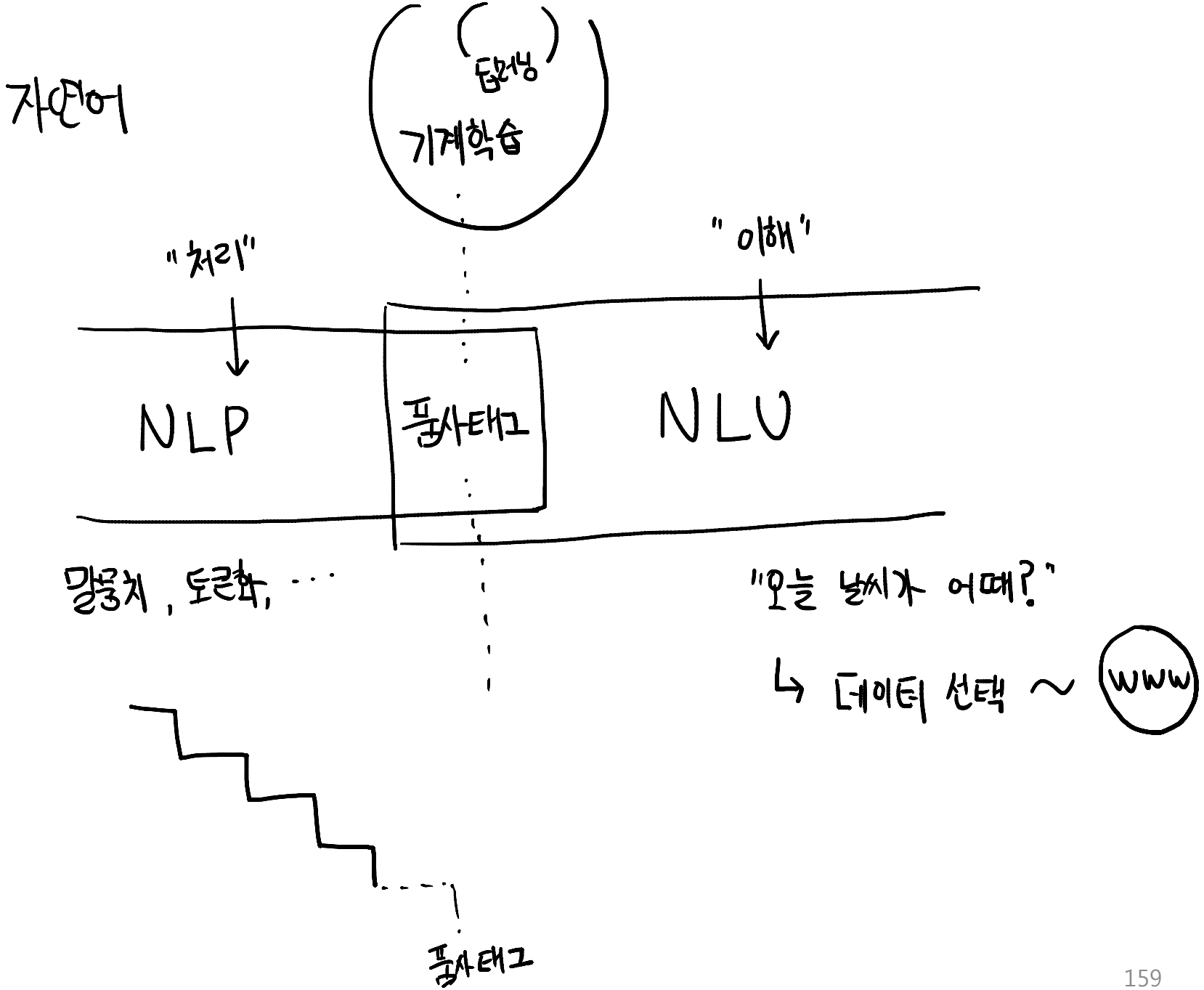
XML

e.g. BNC

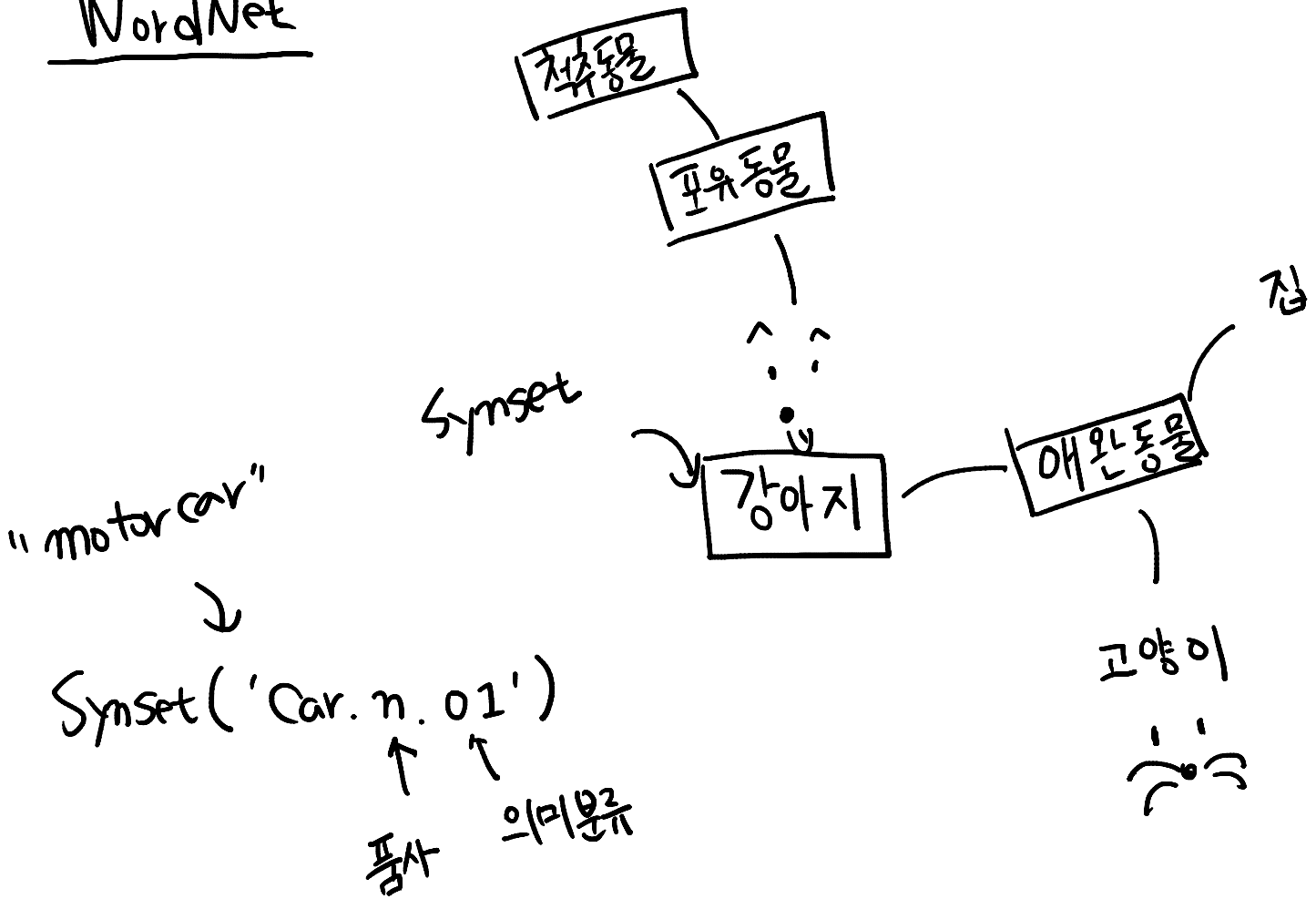
```
<p>
  <s>
    <w pos="NOUN">
```

processor

.tokenize( )


자연어
딥러닝
기계학습
"처리"
"이해"
NLP
품사태그
NLU
말뭉치, 토큰화, ···
"오늘 날씨가 어때?"
↳ 데이터 선택 ~
WWW
품사태그

WordNet


척추동물
포유동물
Synset
강아지
애완동물
집
고양이
"motorcar"
Synset('car.n.01')
품사
의미분류

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

In [5]:
```
synset = wordnet.synset('car.n.01')
```
.synsets(검색키워드)

색인

In [6]:
```
synset.definition()
```

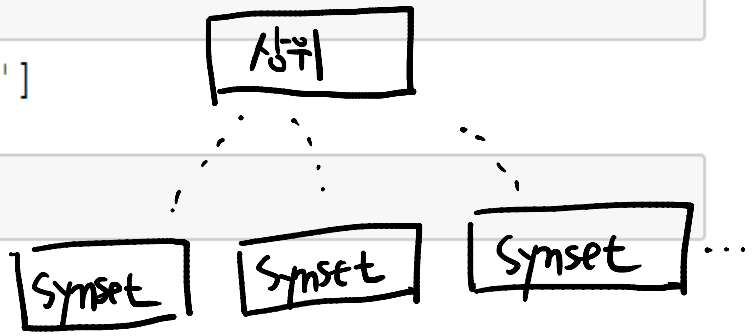
Out[6]:
```
'a motor vehicle with four wheels; usually propelled by an interna
l combustion engine'
```

In [7]:
```
synset.examples()
```

Out[7]:
```
['he needs a car to get to work']
```

In [8]:
```
synset.lemmas()
```
동의어

Out[8]:
```
[Lemma('car.n.01.car'),
 Lemma('car.n.01.auto'),
 Lemma('car.n.01.automobile'),
 Lemma('car.n.01.machine'),
 Lemma('car.n.01.motorcar')]
```



In [ ]:

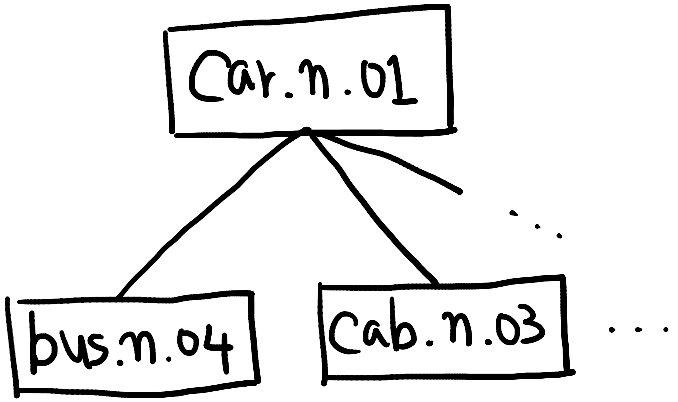
Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

In [11]:
```
motorcar = wordnet.synset('car.n.01')
```

In [12]:
```
motorcar.hyponyms()
```
"하위어"

Out[12]:
```
[Synset('ambulance.n.01'),
 Synset('beach_wagon.n.01'),
 Synset('bus.n.04'),
 Synset('cab.n.03'),
 Synset('compact.n.03'),
 Synset('convertible.n.01'),
 Synset('coupe.n.01'),
 Synset('cruiser.n.01'),
 Synset('electric.n.01'),
 Synset('gas_guzzler.n.01'),
 Synset('hardtop.n.01'),
 Synset('hatchback.n.01'),
 Synset('horseless_carriage.n.01'),
 Synset('hot_rod.n.01'),
 Synset('jeep.n.01'),
 Synset('limousine.n.01'),
 Synset('loaner.n.02'),
 Synset('minicar.n.01'),
 Synset('minivan.n.01'),
 Synset('model_t.n.01'),
 Synset('pace_car.n.01'),
```



Entity ← 물리적실체 ← 물체

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

...

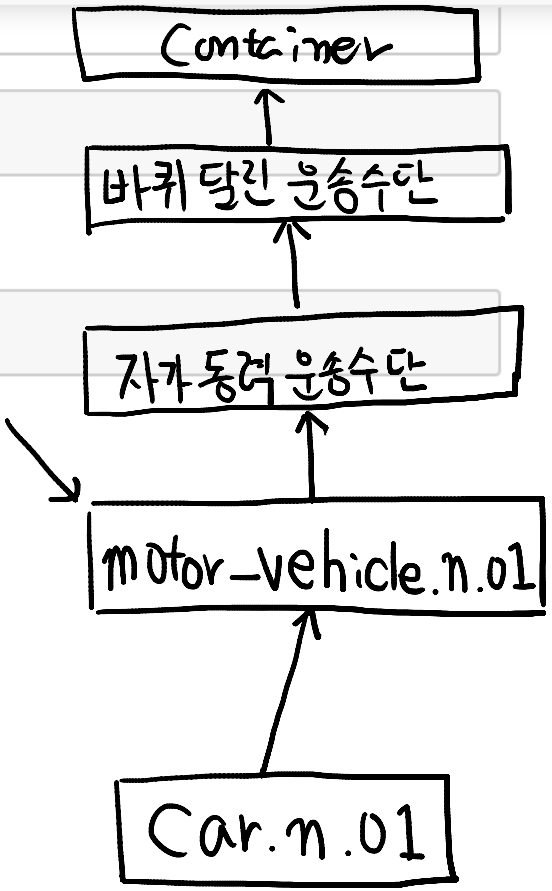
In [13]: motorcar.hypernyms() "상위어"

Out[13]: [Synset('motor_vehicle.n.01')]

In [14]: motorcar.hypernym_paths() "상위어 경로"

Out[14]: ① [[Synset('entity.n.01'),
Synset('physical_entity.n.01'),
Synset('object.n.01'),
Synset('whole.n.02'),
Synset('artifact.n.01'),
Synset('instrumentality.n.03'),
Synset('container.n.01'),
Synset('wheeled_vehicle.n.01'),
Synset('self-propelled_vehicle.n.01'),
Synset('motor_vehicle.n.01'),
Synset('car.n.01')],

② [Synset('entity.n.01'),
Synset('physical_entity.n.01'),
Synset('object.n.01'),
Synset('whole.n.02'),
Synset('artifact.n.01'),
Synset('instrumentality.n.03'),
Synset('conveyance.n.03'),



Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

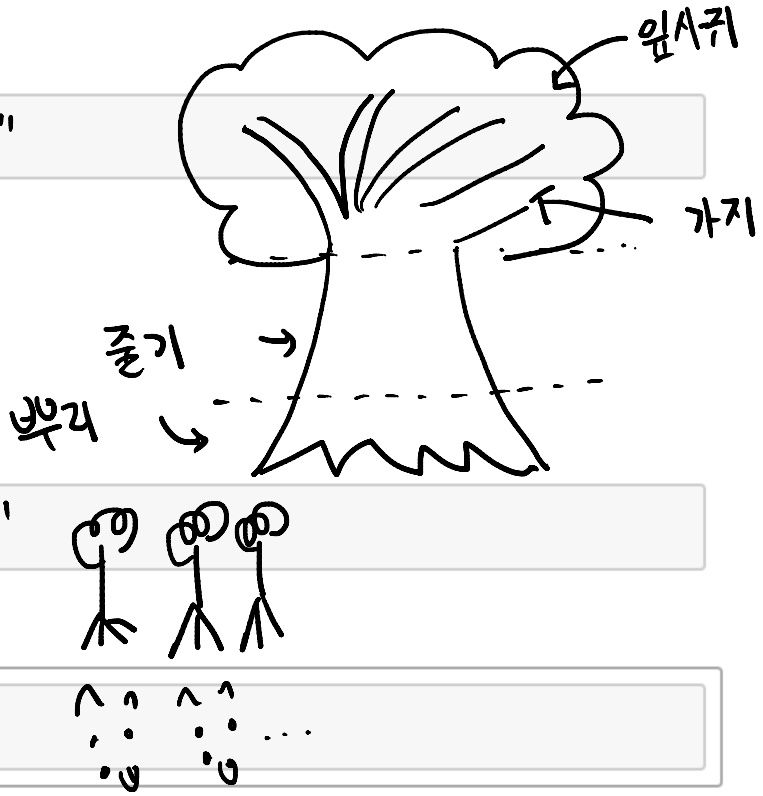
In [15]:
```
tree = wordnet.synset('tree.n.01')
```

In [16]:
```
tree.definition()
```

Out[16]:
```
'a tall perennial woody plant having a main trunk and branches for
ming a distinct elevated crown; includes both gymnosperms and angi
osperms'
```

In [17]:
```
tree.part_meronyms()
```
"부분어"

Out[17]:
```
[Synset('burl.n.02'),
 Synset('crown.n.07'),
 Synset('limb.n.02'),
 Synset('stump.n.01'),
 Synset('trunk.n.01')]
```



In [18]:
```
tree.member_holonyms()
```
"전체어"

Out[18]:
```
[Synset('forest.n.01')]
```

In [ ]:

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

In [23]:
```
강아지 = wordnet.synset('dog.n.01')
강아지.definition()
```

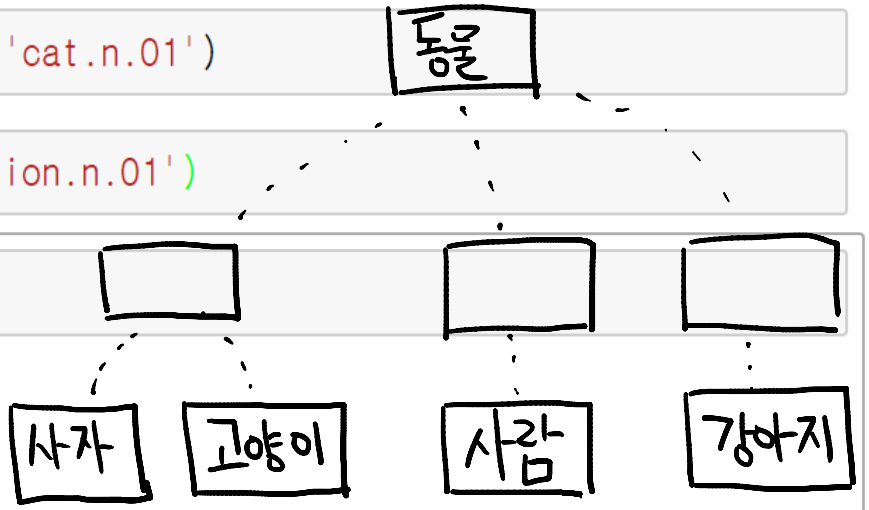
Out[23]:
```
'a member of the genus Canis (probably descended from the common wolf) that has been domesticated by man since prehistoric times; occurs in many breeds'
```

In [26]:
```
고양이 = wordnet.synset('cat.n.01')
```

In [35]:
```
사자 = wordnet.synset('lion.n.01')
```

In [27]:
```
wordnet.synsets('human')
```

Out[27]:
```
[Synset('homo.n.02'),
 Synset('human.a.01'),
 Synset('human.a.02'),
 Synset('human.a.03')]
```



In [32]:
```
인간 = wordnet.synset('homo.n.02')
```

In [ ]:

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

```
Out[27]: [Synset('homo.n.02'),
          Synset('human.a.01'),
          Synset('human.a.02'),
          Synset('human.a.03')]
```

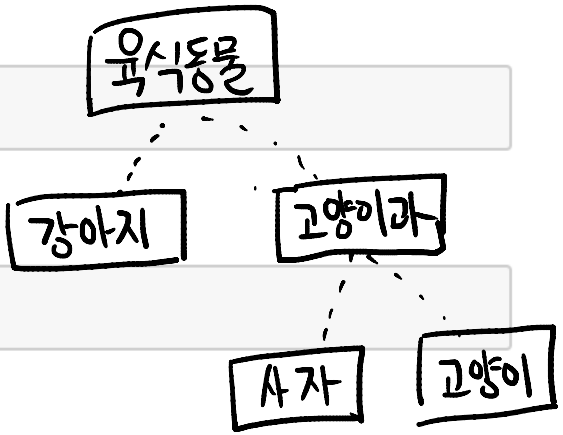
```
In [32]: 인간 = wordnet.synset('homo.n.02')
```

```
In [36]: 강아지.lowest_common_hypernyms(인간)
Out[36]: [Synset('placental.n.01')]
```

```
In [37]: 강아지.lowest_common_hypernyms(고양이)
Out[37]: [Synset('carnivore.n.01')]
```

```
In [38]: 고양이.lowest_common_hypernyms(사자)
Out[38]: [Synset('feline.n.01')]
```

```
In [ ]:
```

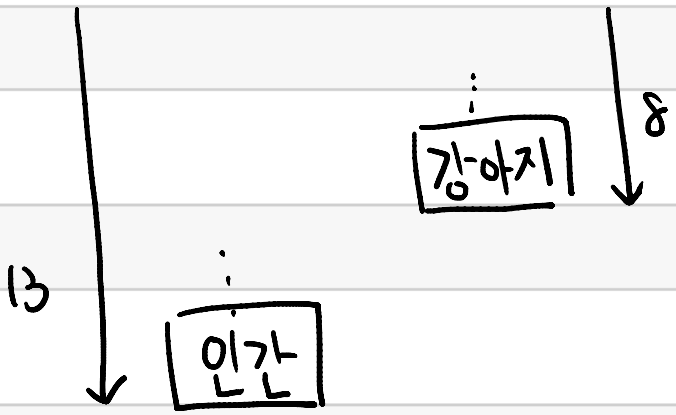


Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

In [39]:
```
인간.min_depth()
```
Out[39]: 13

In [40]:
```
강아지.min_depth()
```
Out[40]: 8

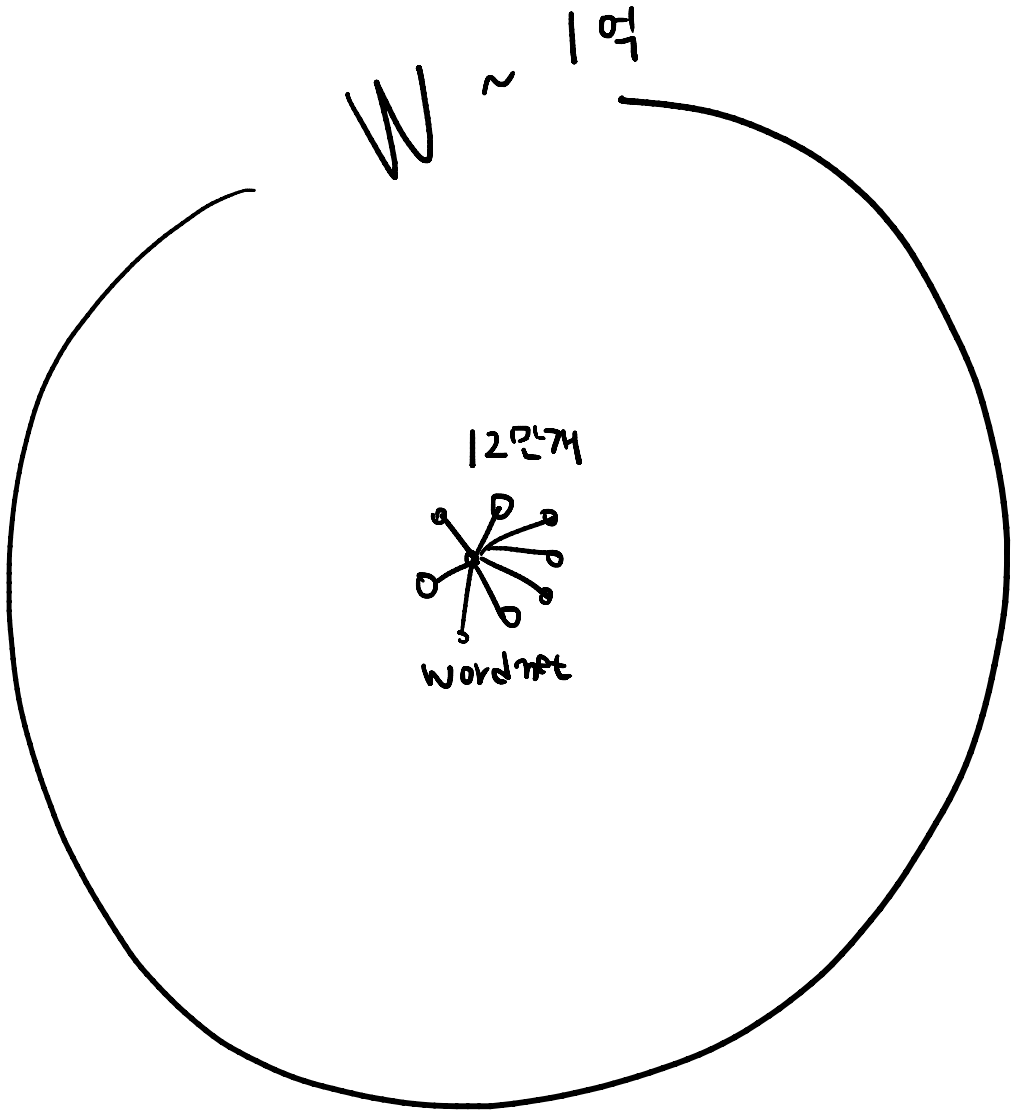


In [41]:
```
인간.path_similarity(강아지)
```
Out[41]: 0.14285714285714285 14%

In [42]:
```
인간.path_similarity(고양이)
```
Out[42]: 0.14285714285714285 14%

In [43]:
```
인간.path_similarity(사자)
```
Out[43]: 0.125 12%

In [ ]:


W ~ 1억
12만개
wordnet

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

정규표현식

In [45]:
```
text = """연락주세요 010-1234-5678. 혹시 안 되면,
010.2345.7890으로 부탁드립니다."""
```

STD ↘ ✓ Regular ✓ Expression 010-1234-5678

In [44]:
```
import re
```

In [46]:
```
전화번호패턴 = r'\d{3}-\d{4}-\d{4}'
```

"숫자3개" - "숫자4개" - "숫자4개"

↓

r '\d{3}-\d{4}-\d{4}'

\w\n ↵

In [47]:
```
re.findall(전화번호패턴, text)
```

Out[47]:
```
['010-1234-5678']
```

In [ ]:

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

정규표현식

In [45]:
```
text = """연락주세요 010-1234-5678. 혹시 안 되면,
010.2345.7890으로 부탁드립니다."""
```

In [44]:
```
import re
```

In [48]:
```
전화번호패턴 = r'₩d{3}[.-]₩d{4}[.-]₩d{4}'
```

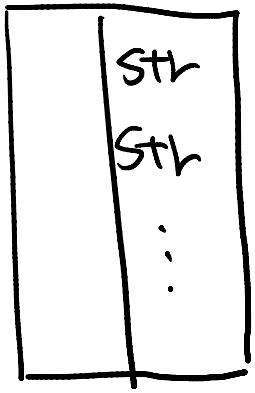
[.-] "또는"

In [49]:
```
re.findall(전화번호패턴, text)
```

Out[49]:
```
['010-1234-5678', '010.2345.7890']
```

In [ ]:

Series .str.contains(패턴)



True
False
⋮

Str ... ed

re.match(패턴, Str)

```
for text in [Str, Str, ...]:
    re.match(패턴, Str)
```

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

"익명함수"

In [56]:
```
패턴선택 = lambda 패턴: 영단어목록[영단어목록.str.contains(패턴)]
```

In [57]:
```
패턴선택('ed$').sample(10)
```

```
211647        unbeaued
224795        unwailed
186601      spiculated
211326      unarticled
218226    uninfuriated
12263      aralkylated
109978          magged
30437           carded
212767      uncolleged
219662       unmounted
dtype: object
```

def 패턴선택(패턴):

return ...

"여덟 글자이면서, 세 번째 글자가 j, 여섯 번째 글자가 t인 경우"

In [ ]:
```
패턴 = '^..j..t..$'
```

In [ ]:

정규표현식 META 문자

$ "…로 끝나는"

^ "…로 시작"

. "아무 글자"

[ ] 글자범위 ; 범위표현 e.g. 0-9, a-z 가-힣

+ 1개 이상

* 0개 이상

[^ab] a 또는 b가 아닌 것

{S,E} 개수 지정

ab|cd ab 또는 cd

( ) 그룹

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

```
3       aal
4     aalii
dtype: object
```

In [55]:
```
영단어목록[영단어목록.str.contains('ed$')].sample(10)
```

Out[55]:
```
212726      uncoddled
223724     unsurnamed
141201         peeled
27888      busybodied
40429     componented
220589      unplained
219531       unmissed
169041       revulsed
218693    unjointured
219346      unmedaled
dtype: object
```

정규표현식 META 문자

$ "…로 끝나는 패턴"

In [ ]:

```
dtype: object
```

"여덟 글자이면서, 세 번째 글자가 j, 여섯 번째 글자가 t인 경우"

In [61]:
```
패턴선택('^..j..t..$').sample(10)
```

Out[61]:
```
110321    majestic
94828     injector
2620      adjuster
50274     dejectly
129006    objectee
166286    rejecter
166291    rejector
276       abjectly
218683    unjilted
218736    unjustly
dtype: object
```

^..j..t..$

In [ ]:
```

```

```
dtype: object
```

"여덟 글자이면서, 세 번째 글자가 j, 여섯 번째 글자가 t인 경우"

In [63]:
```
패턴선택('^..j..t..$').sample(10)
```

...

In [62]:
```
패턴선택('..j..t..').sample(10)
```

Out[62]:
```
210848     unadjectived
218675       unjesting
163758      readjuster
129029    objectization
156693     projectedly
223544     unsubjected
129016     objectioner
129004     objectation
41385    conjecturably
2619       adjustation
dtype: object
```

In [ ]:

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

"여덟 글자이면서, 세 번째 글자가 j, 여섯 번째 글자가 t인 경우"

In [63]:
```
패턴선택('^..j..t..$').sample(10)
```

```
...
```

In [62]:
```
패턴선택('..j..t..').sample(10)
```

```
...
```

In [74]:
```
패턴선택('^[ghi][mno][jlk][def]$')
```

Out[74]:
```
78596    gold
78655    golf
86476    hold
86492    hole
dtype: object
```

^[ghi][mno]...

^

$

[ab] a 또는 b

In [ ]:

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

In [84]:
```
chat_words.sample(5)
```

Out[84]:
```
39677           dd
6735     beautiful
557             ca
8615      removing
8903           U58
dtype: object
```

"1개 이상"

m+ m
m m
m m m ...

^m+i+n+e+$

In [85]:
```
패턴선택(chat_words, '^m+i+n+e+$')
```

Out[85]:
```
3222                                          mine
29128                  miiiiiinnnnnnnnnneeeeeeee
29198    miiiiiiiiiiiinnnnnnnnnnneeeeeeeeee
29204     mmmmmmmmiiiiiiiinnnnnnnneeeeeeee
dtype: object
```

heellooo ...

In [ ]:

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

In [89]:
```
패턴선택(chat_words, '^m*i*n*e*$')
```

m * "이게 아닌"

^m*i*n*e*$

Out[89]:

36	m
45	me
69	i
275	in
1867	min
2576	mmm
3222	mine
3315	n
6946	mm
15069	mmmm
18067	mmmmm
19375	
29128	miiiiiinnnnnnnnnneeeeeeee
29198	miiiiiiiiiiiinnnnnnnnnneeeeeeeee
29204	mmmmmmmmiiiiiiiiinnnnnnnneeeeeeee
30010	mmmmmmmmmmmm
30586	e
31203	ne
32129	meeeeeeeeeeee
32158	mmmmmmmmmmmm
34375	mi
36378	mmmmmmmmm
36982	mmmmm

dtype: object

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

In [89]:
```
패턴선택(chat_words, '^m*i*n*e*$')
```

```
...
```

In [91]:
```
패턴선택(chat_words, '^[^aeiouAEIOU]+$').sample(10)
```

Out[91]:
```
9250                                =)
20654    ]]]]]]]]]]]]]]]]]]]]
4305                                ++
13471                 ))))))))))))
10484                               wz
37502                             ;-(
6132                           <<<<<<
13121                        (((((((
15016                     ..........
43105                               fl
dtype: object
```

시작 ↓ 부정 ←

^[^aeiouAEIOU]+$

In [ ]:

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

In [93]:
```
신문단어 = Series(t for t in 신문.words()).drop_duplicates()
```

In [94]:
```
신문단어.sample(10)
```

...

In [95]:
```
패턴선택(신문단어, '^[0123456789]+.[0123456789]+$')
```

Out[95]:

159	1956
346	1953
348	1955
350	9.8
668	1997
683	160
869	400
888	8.45
891	8.47
1075	8.04
1078	7.90
1106	1.5
1116	352.7
1167	9.37
1176	9.45
1212	8.12
1217	8.14
1228	8.19

목표 : 3.14 와 같은 소수점을 포함한 숫자

^[0-9]+\.[0-9]+$

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

...

In [96]:
```
패턴선택(신문단어, '^[0-9]+₩.[0-9]+$')
```

Out[96]:

350	9.8
888	8.45
891	8.47
1075	8.04
1078	7.90
1106	1.5
1116	352.7
1167	9.37
1176	9.45
1212	8.12
1217	8.14
1228	8.19
1231	8.22
1242	8.53
1245	8.56
1263	83.4
1487	2.80
1492	2.87
1527	3.1
2167	5.29
2173	0.7
2187	5.39

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

In [96]: 패턴선택(신문단어, '^[0-9]+$')

...

In [97]: 패턴선택(신문단어, '^[0-9]{4}$')

Out[97]:

159	1956
346	1953
348	1955
668	1997
2208	1986
2325	1990
2559	1989
3627	1988
3716	1985
4767	1991
6225	1977
6420	1975
6445	1981
6786	1987
8315	1929
8791	1992
9206	1917
9211	1934
12226	1979
12644	1901
12704	1972

^[0-9][0-9][0-9][0-9]$

[0-9] * "0개 이상"

+ "1개 이상"

{n} "n개"

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

...

In [97]:
```
패턴선택(신문단어, '^[0-9]{4}$')
```

...

In [98]:
```
패턴선택(신문단어, '^[0-9]+-[a-z]{3,5}$')
```

Out[98]:

1221	30-day
1919	10-lap
7290	52-week
17616	50-state
18521	14-hour
20851	69-point
27263	10-year
34158	30-year
34191	36-day
39975	150-point
39986	20-point
40135	12-point
40214	30-point
42909	500-stock
46378	10-day
47858	90-day
47888	15-day
48892	100-share

^[0-9]+-[a-z]{3,5}$

{n} "n개"

{n,m} "n≤ ≤m"

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

```
In [97]: 패턴선택(신문단어, '^[0-9]{4}$')
```

```
...
```

```
In [98]: 패턴선택(신문단어, '^[0-9]+-[a-z]{3,5}$')
```

```
...
```

```
In [100]: 패턴선택(신문단어, '^[a-z]{5,}-[a-z]{2,3}-[a-z]{,6}$')
```

```
Out[100]: 12948        black-and-white
          39511      savings-and-loan
          44106         father-in-law
          45462    machine-gun-toting
          94123      bread-and-butter
          dtype: object
```

{n} n개

{s,e} s≤ ≤e

{s,} s개 이상

{,e} 6개 이하

```
In [ ]:
```

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

In [101]:
```
패턴선택(신문단어, '(ed|ing)$')
```

```
C:₩ProgramData₩Anaconda3₩lib₩site-packages₩ipykernel_launcher.py:
1: UserWarning: This pattern has match groups. To actually get the
groups, use str.extract.
  """Entry point for launching an IPython kernel.
```

Out[101]:

28	publishing
41	Consolidated
47	named
63	used
72	caused
84	exposed
95	reported
120	causing
141	York-based
150	stopped
151	using
191	bring
212	talking
221	having
243	studied
286	led
324	making
334	replaced
372	worked

"ed, ing 끝나는"

[ed ing]

(ed|ing)$

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

In [103]:
```
패턴선택(신문단어, '(ed|ing)$')
```

ed$ "또는" ing$

...

In [104]:
```
패턴선택(신문단어, 'ed|ing$')
```

ed "또는" ing$

Out[104]:

28	publishing
41	Consolidated
47	named
63	used
72	caused
84	exposed
95	reported
120	causing
141	York-based
150	stopped
151	using
184	Medicine
191	bring
212	talking
221	having
243	studied
286	led
298	medical
324	making

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

```
45462     machine-gun-toting
94123       bread-and-butter
dtype: object
```

In [103]:
```
패턴선택(신문단어, '(ed|ing)$')
```

```
...
```

In [104]:
```
패턴선택(신문단어, 'ed|ing$')
```

```
...
```

1) 패턴확인 → 010-1234-5678

2) ( ) "그룹생성"

In [114]:
```
re.findall(
    r'(\d{3})-(\d{3,4})-(\d{4})',
    '내 전번은 010-1234-5678입니다.')
```

Out[114]:
```
[('010', '1234', '5678')]
```

In [ ]:

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

In [104]:
```
패턴선택(신문단어, 'ed|ing$')
```

```
...
```

In [114]:
```
re.findall(
    r'(\d{3})-(\d{3,4})-(\d{4})',
    '내 전번은 010-1234-5678입니다.')
```

Out[114]:
```
[('010', '1234', '5678')]
```

In [115]:
```
re.sub(    1         2       3
    r'(\d{3})-(\d{3,4})-(\d{4})',
    r'\1-****-\3',
    '내 전번은 010-1234-5678입니다.'
)
```

Out[115]:
```
'내 전번은 010-****-5678입니다.'
```

In [ ]:

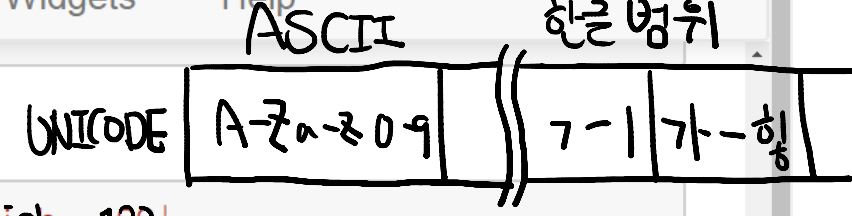
"교체" substitute

re.sub(패턴, 대체, 대상)

\1 "1번그룹"

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help




```
Out[115]: '내 전번은 010-****-5678입니다.'

In [117]: text = '알흠다운_우리말 ^0^ Enlgish, 123'

In [120]: re.findall('[^ ㄱ-ㅣ가-힣]+', text)
Out[120]: ['^0^', 'Enlgish,', '123']

In [118]: re.sub('[^ㄱ-ㅣ가-힣]+', '', text)
Out[118]: '알흠다운우리말'

In [119]: re.sub('[^ ㄱ-ㅣ가-힣]+', '', text)
Out[119]: '알흠다운 우리말   '

In [ ]:
```

Out[120]: ['^0^', 'Enlgish,', '123']

In [118]:
```
re.sub('[^ㄱ-ㅣ가-힣]+', '', text)
```

Out[118]: '알흠다운우리말'

In [119]:
```
re.sub('[^ ㄱ-ㅣ가-힣]+', '', text)
```

Out[119]: '알흠다운 우리말 '

In [124]:
```
re.sub('[ㄱ-ㅣ]+', '', 'ㅋㅋㅋ, ㅎㅎㅎ 이런 것들을 제거합니다.')
```

Out[124]: ', 이런 것들을 제거합니다.'

In [125]:
```
tokens = Series([
    'processing',
    'processed',
    'processes',
    'process'
])
```

In [128]:
```
tokens.str.findall('(.+)(ing|ed|es)$')
```

Out[128]:
```
0    [(process, ing)]
1     [(process, ed)]
```

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

Out[124]: ', 이런 것들을 제거합니다.'

In [125]:
```
tokens = Series([
    'processing',
    'processed',
    'processes',
    'process'
])
```

(.+)(ing|ed|es)$

do did done

In [128]:
```
tokens.str.findall('(.+)(ing|ed|es)$')
```

Out[128]:
```
0    [(process, ing)]
1     [(process, ed)]
2     [(process, es)]
3                  []
dtype: object
```

타다
타고
타면
탔다 타+ㅆ다

In [ ]:

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

In [131]:
```
영화리뷰 = pd.read_csv('data/nsmc/ratings.txt', sep='₩t')
```

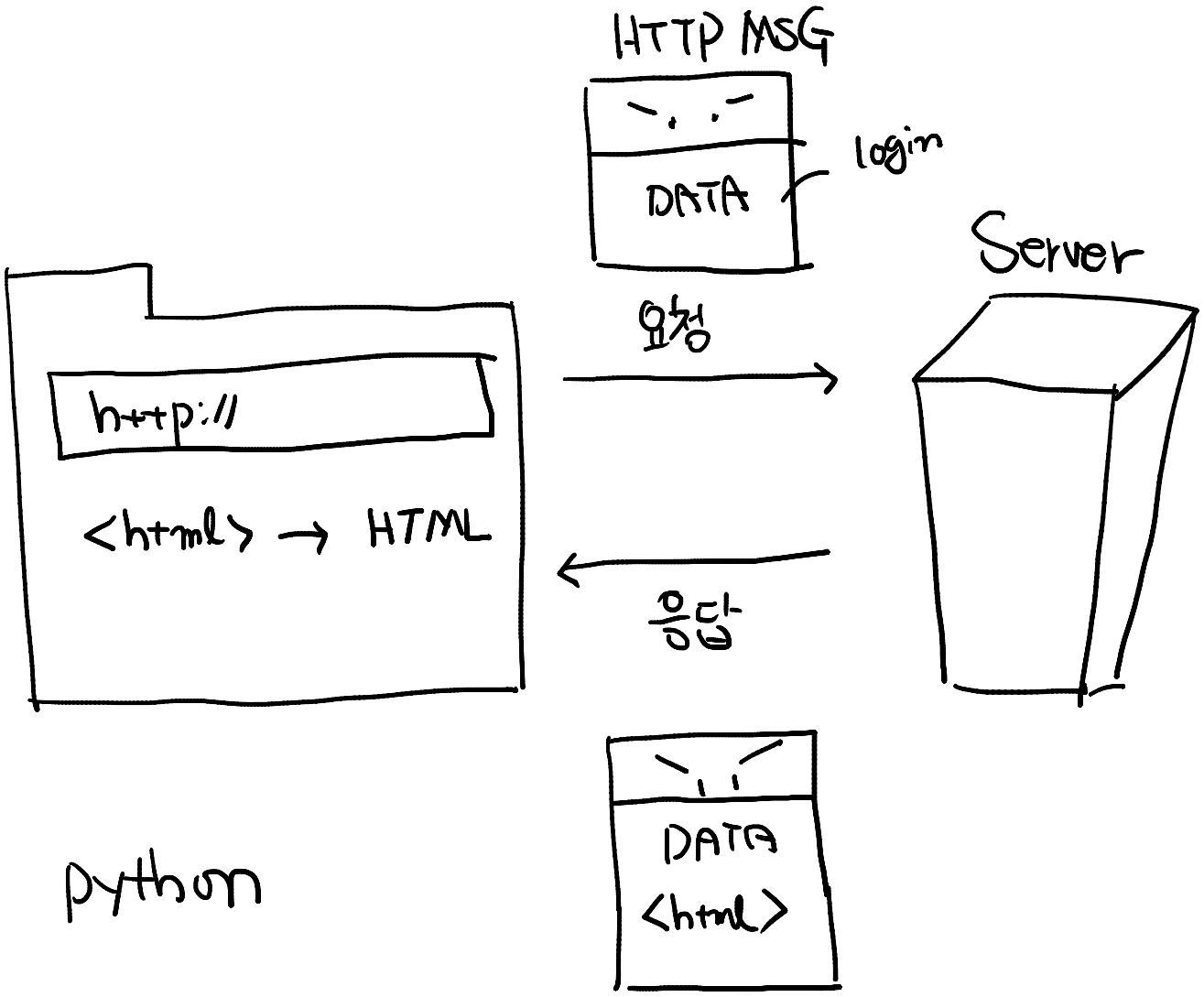
In [132]:
```
영화리뷰['document'].sample(10)
```

Out[132]:
```
116340                                          줄거리에 내용을 써놨네
ㅋㅋㅋ
169853                                          여자 존나 짜증나네ㅎ
ㅋㅋ
87978                                           티아 레오니가 너무
예쁨
90039                                             너무 보고싶어요...
26085     금욜 삼시세끼 꼭 챙겨봅니다. 첨엔 일단 나pd님. 이서진씨
믿고본거긴한데.. 크 ...
113712              피터잭슨이 이런 000기영화를 만들다니 ㅋㅋㅋ 러블
리본즈에 이어 또실망..
190524                                  여명은 한국에 눈이 멀었던
게지..
126063                                                  안타까움
34649                                 평점 너무 약하다....견자단 완전
멋짐!!!
11666     단순한 킬링타임용 호러물이 절대 아닙니다.. 완성도도 그렇
지만 무엇보다 에밀리 로즈...
Name: document, dtype: object
```

1) 정규화

2) 토큰화

처음에는


HTTP MSG
DATA
login
Server
요청
http://
<html> → HTML
응답
DATA
<html>
python

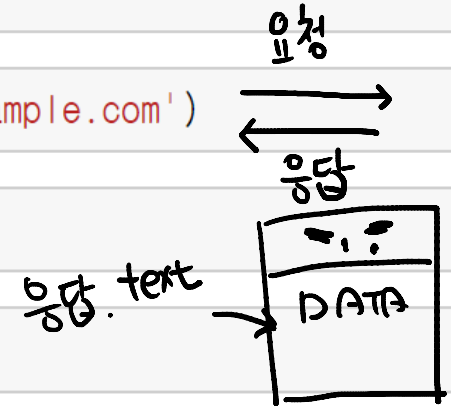
Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

In [133]:
```
import requests
```

In [134]:
```
응답 = requests.get('http://www.example.com')
```

In [135]:
```
data = 응답.text
```



In [136]:
```
print(data)
```

```
<!doctype html>
<html>
<head>
    <title>Example Domain</title>

    <meta charset="utf-8" />
    <meta http-equiv="Content-type" content="text/html; charset=ut
f-8" />
    <meta name="viewport" content="width=device-width, initial-sca
le=1" />
    <style type="text/css">
    body {
        background-color: #f0f0f2;
        margin: 0;
        padding: 0;
        font-family: "Open Sans", "Helvetica Neue", Helvetica, Ari
al, sans-serif;
```

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

In [149]:
```
html = BeautifulSoup(data, 'lxml')
```

In [150]:
```
html.title
```

Out[150]:
```
<title>NAVER</title>
```

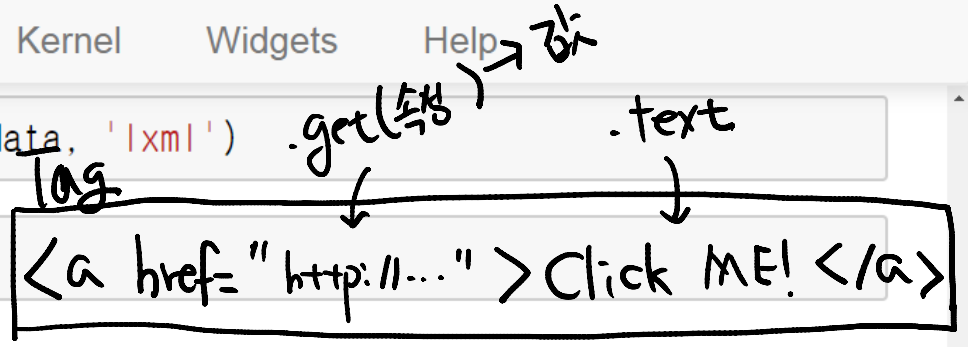
In [151]:
```
html.title.text
```

Out[151]:
```
'NAVER'
```

In [154]:
```
links = html.find_all('a')
```

In [155]:
```
for link in links:
    print(link.text, link.get('href'))
```

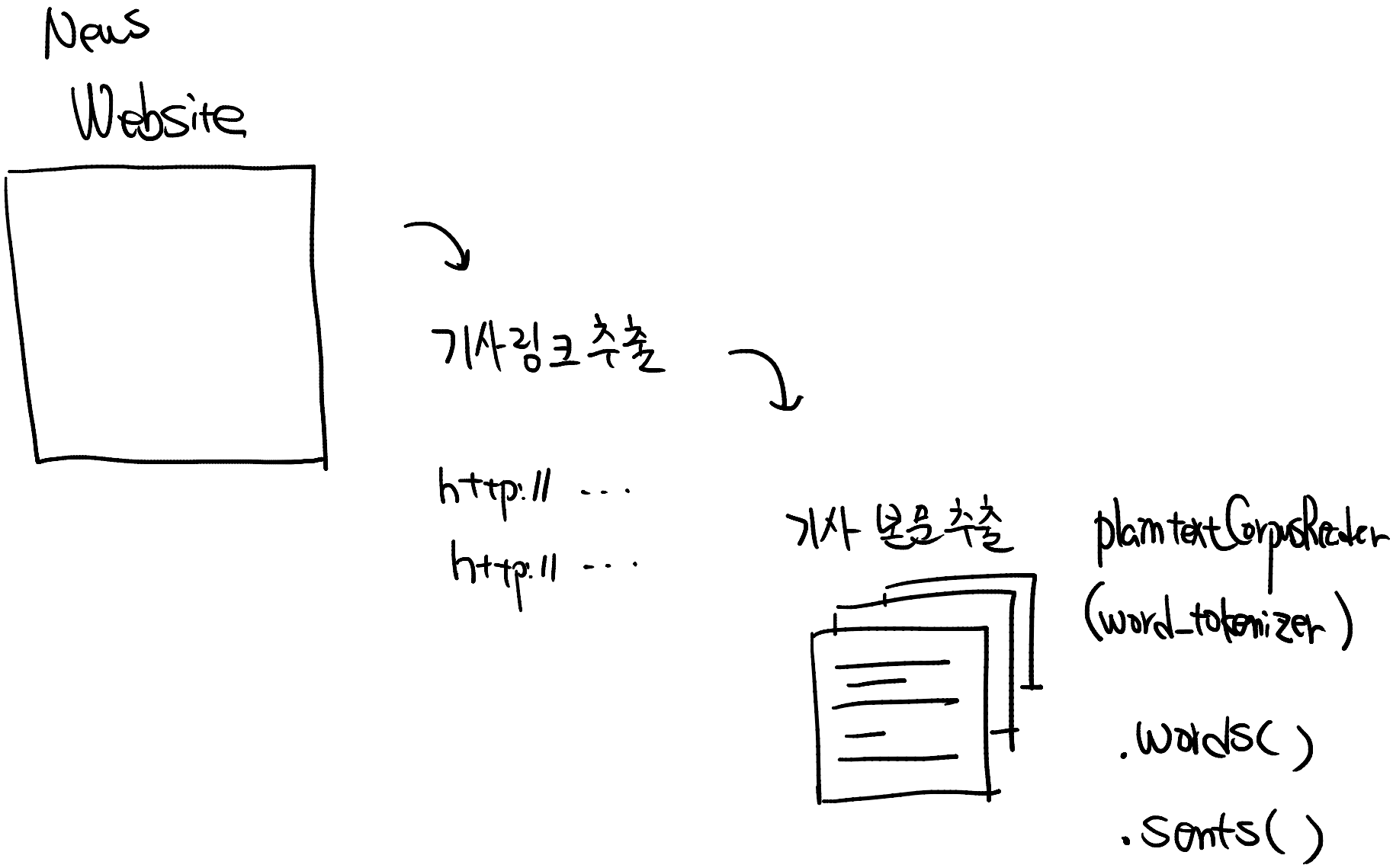
```
뉴스스탠드 바로가기 #news_cast
주제별캐스트 바로가기 #themecast
타임스퀘어 바로가기 #time_square
쇼핑캐스트 바로가기 #shop_cast
로그인 바로가기 #account
네이버 /
네이버를 시작페이지로 http://help.naver.com/support/alias/contents2/naverhome/naverhome_1.naver
쥬니어네이버 http://jr.naver.com
해피빈 http://happybean.naver.com/main/SectionMain.nhn
```



html.a

.find_all('a')

상대	절대
	http://


News
Website
기사링크추출
http:// ...
http:// ...
기사 본문추출
plaintextCorpusReader
(word_tokenizer)
.words()
.sents()

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

In [179]:
```
응답 = requests.get(url)
```

In [181]:
```
응답.encoding = 'utf-8'
```

In [183]:
```
웹페이지 = BeautifulSoup(응답.text, 'lxml')
```

In [185]:
```
웹페이지.title.text
```

Out[185]: '방남 北응원단 "기존에 없던 것 보여줄 생각…보시면 압네다"(종합)'

In [188]:
```
본문 = '₩n'.join(
    문단.text for 문단 in 웹페이지.select('.article p'))
```

CSS 선택자

<div class="article>

 …

In [189]:
```
print(본문)
```

(도라산 · 서울=연합뉴스) 공동취재단 이정진 김정은 기자 = 평창동계올림픽에서 남북 선수들을 응원할 북한 응원단과 태권도시범단 등이 7일 방남했다.

버스 9대에 나눠타고 이날 오전 9시 28분 경기 파주의 도라산 남북출입사무소(CIQ)에 도착한 이들은 10시 13분부터 기자단, 응원단, 태권도시범단, 민족올림픽위원회(NOC) 관계자 순으로 차례차례 남측 출구로 나왔다.

남성들은 검은색 코트에 털모자, 여성들은 붉은 코트에 검은색 털

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

In [200]:
```
with open(filename,'w', encoding='utf-8') as file:
    file.write(내용)
```

In [201]:
```
from nltk.corpus import PlaintextCorpusReader
```

In [202]:
```
processor = pyko.OpenKoreanTextProcessor()
```

In [203]:
```
reader = PlaintextCorpusReader(
    root='.',
    fileids=['연합뉴스-0505000000AKR20180207084551014.txt'],
    word_tokenizer=processor
)
```

In [204]:
```
tokens = reader.words()
```

In [205]:
```
tokens
```

Out[205]:
```
['[', '방남', '北', '응원단', '"', '기존', '에', '없던', '것',
...]
```

In [206]:
```
nltk.Text(tokens).collocations()
```

```
올림픽 위원회; NOC 관계자
```

모찌

w_{-2}	w_{-1}	w_0	w_{+1}
예쁜	우리	강아지	사랑해
t_{-2}	t_{-1}	t_0	t_1
형용사	대명사	명사	동사

?

현실:

→ 내 새 꾸 러브러브

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

In [212]:
```
sejong.sents()
```

Out[212]:
```
[['뭐', '타고', '가?'], ['지하철.'], ['기차?'], ['아침에', '몇',
'시에', '타고', '가는데?'], ['아침에', '한', '일찍', '가면은', '일
곱시', '이십분', ...], ['음.'], ...]
```

In [210]:
```
tagged_sents = sejong.sents(tagged=True)
```

In [211]:
```
tagged_sents
```

Out[211]:
```
[[('뭐', 'NP'), ('타', 'VV'), ('고', 'EC'), ('가', 'VV'), ('ㅏ',
'EF'), ('?', 'SF')], [('지하철', 'NNG'), ('.', 'SF')], [('기차',
'NNG'), ('?', 'SF')], [('아침', 'NNG'), ('에', 'JKB'), ('몇', 'M
M'), ('시', 'NNB'), ('에', 'JKB'), ('타', 'VV'), ...], [('아침',
'NNG'), ('에', 'JKB'), ('한', 'MM'), ('일찍', 'MAG'), ('가', 'V
V'), ('면은', 'EC'), ...], [('음', 'IC'), ('.', 'SF')], ...]
```

In [214]:
```
pd.read_csv('data/sejong_tagset.txt', index_col='Tag')
```

Out[214]:

Tag	Description
NNG	일반 명사

BNC

<w pos="NOUN">dog</w>

. tagged_words( )

→ (dog, NOUN)

Sejong

<w>

강아지 강아지/NNG

. words(tagged=True)

→ (강아지, NNG)

품사 분류 (tagset)

영어

Class	POS
Open	Noun
	Verb
	Adjective
	Adverb
	Interjection
Close	Pronoun
	Preposition
	Conjunction
	Article

한국어

5언	9품사
체언	명사
	대명사
	수사
용언	동사
	형용사
수식언	관형사
	부사
관계언	조사
독립언	감탄사

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

In [214]:
```
pd.read_csv('data/sejong_tagset.txt', index_col='Tag')
```

Out[214]:

Tag	Description
NNG	일반 명사
NNP	고유 명사
NNB	의존 명사
NR	수사
NP	대명사
VV	동사
VA	형용사
VX	보조 용언
VCP	긍정 지정사
VCN	부정 지정사
MM	관형사
MAG	일반 부사

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

곱시', '이십분', ...], ['음.'], ...]

In [210]:
```
tagged_sents = sejong.sents(tagged=True)
```

In [211]:
```
tagged_sents
```
→ [NP, VV, EC, VV, EF, ···]

Out[211]:
```
[[('뭐', 'NP'), ('타', 'VV'), ('고', 'EC'), ('가', 'VV'), ('ㅏ',
'EF'), ('?', 'SF')], [('지하철', 'NNG'), ('.', 'SF')], [('기차',
'NNG'), ('?', 'SF')], [('아침', 'NNG'), ('에', 'JKB'), ('몇', 'M
M'), ('시', 'NNB'), ('에', 'JKB'), ('타', 'VV'), ...], [('아침',
'NNG'), ('에', 'JKB'), ('한', 'MM'), ('일찍', 'MAG'), ('가', 'V
V'), ('면은', 'EC'), ...], [('음', 'IC'), ('.', 'SF')], ...]
```

In [214]:
```
pd.read_csv('data/sejong_tagset.txt', index_col='Tag')
```

...

In [ ]:

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

In [215]:
```
tags = []
for sent in tagged_sents:
    for token in sent:
        tag = token[1]
        tags.append(tag)
```

쌍 만들기

In [219]:
```
tag_pairs = list(nltk.bigrams(tags))
```

In [220]:
```
tag_pairs[:10]
```

Out[220]:
```
[('NP', 'VV'),
 ('VV', 'EC'),
 ('EC', 'VV'),
 ('VV', 'EF'),
 ('EF', 'SF'),
 ('SF', 'NNG'),
 ('NNG', 'SF'),
 ('SF', 'NNG'),
 ('NNG', 'SF'),
 ('SF', 'NNG')]
```

NP, VV, EC

(NP, VV), (VV, EC)

(? , NN.)

In [ ]:

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

In [220]:
```
tag_pairs[:10]
```

...

In [221]:
```
target_tags = []
for t1, t2 in tag_pairs:
    if t2.startswith('NN'):
        target_tags.append(t1)
```

^NN

In [224]:
```
Series(target_tags).value_counts()[:10]
```

Out[224]:
```
ETM     53694
SP      34334
MM      34006
NNG     32204
SF      22829
MAG     21477
IC      11725
EC      10918
NR       9261
JKB      8964
dtype: int64
```

명사

나, 너, 우리

우리 강아지

In [ ]:

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

(뭐,NP),(타,VV), ⇄ [뭐,타,…] [NP,VV,…]

In [227]:
```
words, tags = zip(*(token for sent in tagged_sents for token in ser
```

In [228]:
```
tags = Series(tags, index=words)
```

In [229]:
```
tags[:10]
```

Out[229]:
```
뭐          NP
타          VV
고          EC
가          VV
냐          EF
?          SF
지하철       NNG
.          SF
기차         NNG
?          SF
dtype: object
```

In [ ]:

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

In [237]:
```
기본태그기.tag(sample)
```

Out[237]:
```
[('뭐', 'NNG'),
 ('타', 'NNG'),
 ('고', 'NNG'),
 ('가', 'NNG'),
 ('ㅏ', 'NNG'),
 ('?', 'NNG')]
```

In [239]:
```
tagged_sents[0]
```

Out[239]:
```
[('뭐', 'NP'), ('타', 'VV'), ('고', 'EC'), ('가', 'VV'), ('ㅏ', 'EF'), ('?', 'SF')]
```

In [238]:
```
기본태그기.evaluate(tagged_sents[:1000])
```

Out[238]:
```
0.11324162845630123
```

In [ ]:
```

```

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

조건부 도수분계

In [240]:
```
단어별품사빈도 = nltk.ConditionalFreqDist(
    token for sent in tagged_sents for token in sent
)
```

In [241]:
```
단어별품사빈도
```

Out[241]:
```
ConditionalFreqDist(nltk.probability.FreqDist,
                    {'뭐': FreqDist({'IC': 7321,
                                     'MAG': 1,
                                     'NNG': 11,
                                     'NNP': 1,
                                     'NP': 5284,
                                     'UNT': 1}),
                     '타': FreqDist({'MM': 3,
                                     'NNB': 21,
                                     'NNG': 11,
                                     'UNT': 3,
                                     'VV': 570,
                                     'XPN': 1}),
                     '고': FreqDist({'EC': 19104,
                                     'EF': 518,
                                     'ETM': 1,
                                     'IC': 9,
                                     'JC': 2,
```

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

...

특정단어 가장높은 도수 품사

In [249]:
```
품사추측 = dict(
    (word, 단어별품사빈도[word].max()) for word in 단어도수.index)
```

LookUp

In [251]:
```
품사추측
```

(, , 'SP')

Out[251]:
```
{',': 'SP',
 '.': 'SF',
 '하': 'VV',
 '이': 'JKS',
 '는': 'ETM',
 '가': 'JKS',
 '?': 'SF',
 '에': 'JKB',
 '거': 'NNB',
 '어': 'IC',
 'ㄴ': 'ETM',
 '고': 'EC',
 '을': 'JKO',
 '있': 'VA',
 '(이)': 'VCP',
 '그': 'MM',
 '은': 'JX',
 '되': 'VV',
 '아': 'IC',
```

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

In [247]:
```
단어도수 = Series(t[0] for t in 토큰생성기()).value_counts()
```

In [248]:
```
단어도수[:10]
```

```
...
```

In [249]:
```
품사추측 = dict(
    (word, 단어별품사빈도[word].max()) for word in 단어도수.index)
```

In [251]:
```
품사추측
```

기계학습모델

```
...
```

In [252]:
```
baseline_tagger = nltk.UnigramTagger(model=품사추측)
```

In [*]:
```
baseline_tagger.evaluate(tagged_sents)
```

In [ ]:
```

```

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

...

In [252]:
```
baseline_tagger = nltk.UnigramTagger(model=품사추측)
```

In [253]:
```
baseline_tagger.evaluate(tagged_sents)
```

Out[253]: 0.8603526460812297 ← 정확도

In [254]:
```
sample = [token[0] for token in tagged_sents[3]]
sample
```

...

설정 예측

In [255]:
```
baseline_tagger.tag(sample)
```

...

In [256]:
```
tagged_sents[3]
```

...

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

Out[253]: 0.8603526460812297

In [254]:
```
sample = [token[0] for token in tagged_sents[3]]
sample
```

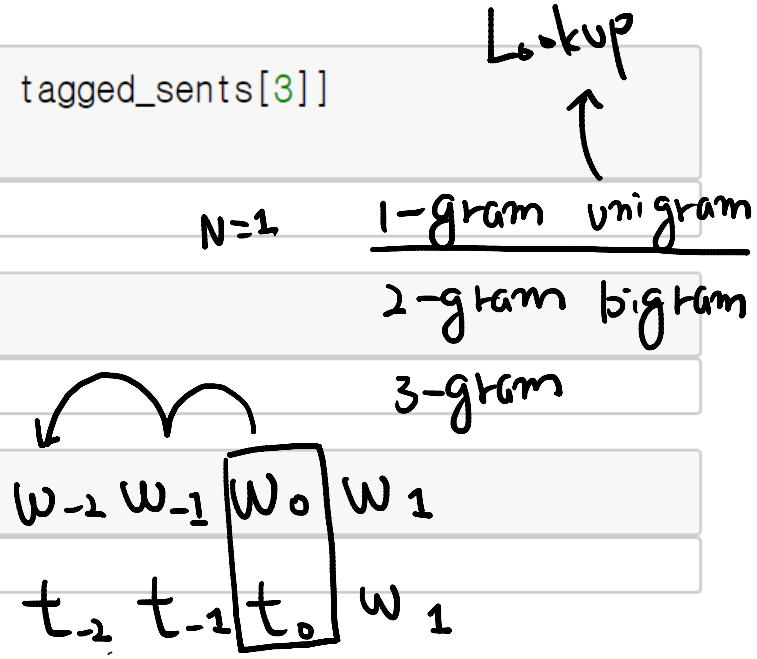
...

In [255]:
```
baseline_tagger.tag(sample)
```

...

In [256]:
```
tagged_sents[3]
```

...



N-Gram Tagging

In [ ]:

Trusted | Python 3

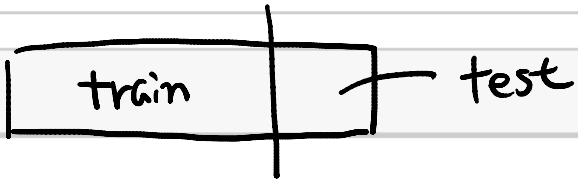
File Edit View Insert Cell Kernel Widgets Help

```
...
```

In [255]:
```
baseline_tagger.tag(sample)
```

```
...
```

In [256]:
```
tagged_sents[3]
```




```
...
```

N-Gram Tagging

"관전" ~ "학습"

In [257]:
```
unigram_tagger = nltk.UnigramTagger(tagged_sents)
```

In [*]:
```
unigram_tagger.evaluate(tagged_sents)
```

"평가"

In [ ]:
```

```

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

In [265]:
```
bigram_tagger.evaluate(test_sents)
```

Out[265]: 0.6856353591160221

t_{-1} t_0

In [266]:
```
t0 = nltk.DefaultTagger('NNG')
t1 = nltk.UnigramTagger(train_sents, backoff=t0)
t2 = nltk.BigramTagger(train_sents, backoff=t1)
```

None

None

In [267]:
```
t2.evaluate(test_sents)
```

Out[267]: 0.9252839165131983

In [268]:
```
t2.tag(sample)
```

Out[268]:
```
[('아침', 'NNG'),
 ('에', 'JKB'),
 ('몇', 'MM'),
 ('시', 'NNB'),
 ('에', 'JKB'),
 ('타', 'VV'),
 ('고', 'EC'),
 ('가', 'VV'),
 ('는데', 'EC'),
 ('?', 'SF')]
```

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

In [270]:
```
import pickle
```

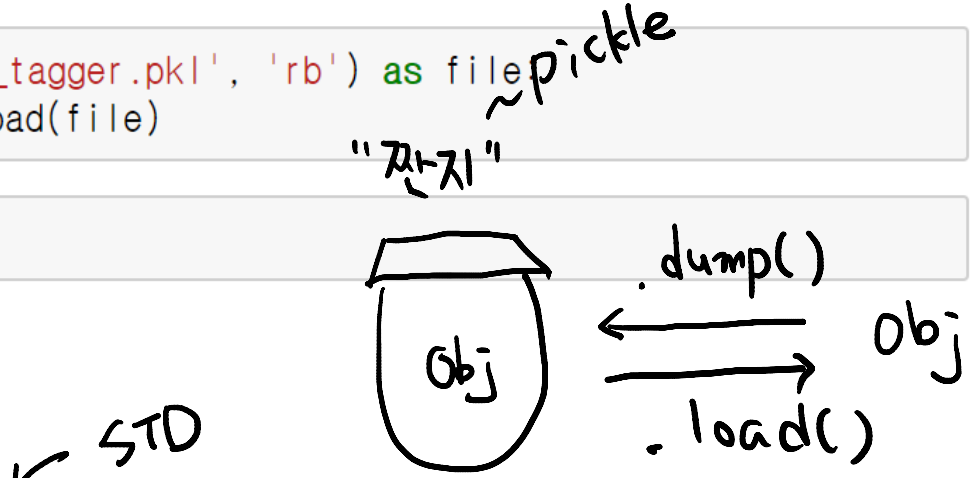
In [271]:
```
with open('sejong_tagger.pkl', 'wb') as file:
    pickle.dump(t2, file)
```

In [272]:
```
del t2
```

In [274]:
```
with open('sejong_tagger.pkl', 'rb') as file:
    t2 = pickle.load(file)
```

In [275]:
```
t2.tag(sample)
```

Out[275]:
```
[('아침', 'NNG'),
 ('에', 'JKB'),
 ('몇', 'MM'),
 ('시', 'NNB'),
 ('에', 'JKB'),
 ('타', 'VV'),
 ('고', 'EC'),
 ('가', 'VV'),
 ('는데', 'EC'),
 ('?', 'SF')]
```



STD