

Part1. 워드클라우드란 개요

1. 워드 클라우드(Word Cloud), 태그 클라우드(tag cloud)란?

워드 클라우드란 문서의 키워드, 개념 등을 직관적으로 파악할 수 있도록 핵심 단어를 시각화하는 기법이다. 예를 들면 많이 언급될수록 단어를 크게 표현해 한눈에 들어올 수 있게 하는 기법 등이 있다. 주로 빅데이터(big data)를 분석할 때 데이터의 특징을 도출하기 위해 활용한다.

wordcloud

클라우드(영어: tag cloud) 또는 워드 클라우드(word cloud)는 메타 데이터에서 얻어진 태그들을 분석하여 중요도나 인기도 등을 고려하여 시각적으로 늘어 놓아 웹 사이트에 표시하는 것

```
tmp='강아지 산책 강아지 목욕 강아지 미용 강아지 쇼핑 친구와 저녁 먹음 가족과 점심 먹음 혼자 저녁 먹음 친구와 쇼핑'
tmp
tmp.count('강아지')
```

원문읽기

- ★ txt, csv 파일 읽기
- ★ wordcloud의 자료는 여러개의 문단이 아니라 한개의 문단이어야 함.

단어분리

- ★ 띄어쓰기등 구분자로 분리
- ★ KoNLPy패키지를 이용하여 분리

단어빈도수계산

- ★ 가장 많이 나온 단어를 1로 세팅하여
- ★ 나머지 단어 백분율을 계산

시각화

- ★ 단어를 비율에 맞추어 시각화

원분	증복제거	빈도수 (카운트)	전체비율	강아지를1로했을때 비율
강아지		4	21%	1.00
산책	먹음	3	16%	0.75
강아지	쇼핑	2	11%	0.50
목욕	친구와	2	11%	0.50
강아지	저녁	2	11%	0.50
미용	산책	1	5%	0.25
강아지	목욕	1	5%	0.25
쇼핑	미용	1	5%	0.25
친구와	가족과	1	5%	0.25
저녁	점심	1	5%	0.25
먹음	혼자	1	5%	0.25
가족과				
점심				
먹음				
혼자				
저녁				
먹음				
친구와				
쇼핑				

2. wordCloud 시각화



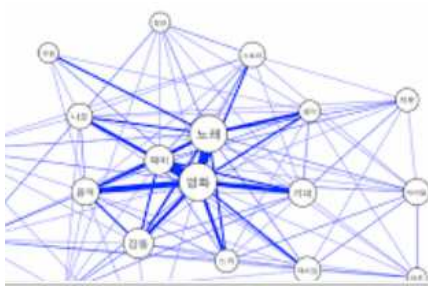
클라우드 쉽게 만들기 : 네이버 블로그



빅데이터! 워드 클라우드(Word Cloud)로 손쉽게 표..



박근혜 대통령 간담회 전문 단어구름 ...



3. 종류(위키백과사전)

태그 구름은 외적인 모습보다는 그 의미적인 면에서 두 가지로 구분된다.

어떤 하나의 연결에 연관된 태그들이 얼마나 많으며 어떤 종류인지 보여주는 것이다. 이것은 어떤 내용에 민주적으로 투표된 것과 마찬가지로 여러 사용자에게 의해 그것이 어떤 태그와 연결되는 것이 알맞은지를 보여줄 수 있다.

예를 들어 어떤 음악가의 음악이 어떤 장르의 음악인지 보여주는 Last.fm의 경우에 볼 수 있다.

가장 대표적인 경우로 각 태그들이 얼마나 인기도가 높은지를 보여주는 표시법으로 사용되는 경우이다. 이때 태그들의 글자 크기나, 색상, 형태들이 인기도에 따라 변화되며 이때 인기도는 사용자들의 선택에 의해 자동적으로 갱신되게 된다.

4. 워드클라우드 만들기 사이트 참조

- ▶ <http://wordcloud.kr/>
- ▶ <https://juem.tistory.com/10>
- ▶ <https://www.wordclouds.com/>
- ▶ <https://worditout.com/>
- ▶ 크롬 프로그램, Drive Word Cloud
- ▶ (영문만 가능함)구글드라이브의 - 구글문서 - 부가기능 - word cloud generator
=> 다음사이트의 가장 하단의 내용 참조
<https://ichi.pro/ko/tableau-python-mich-google-word-cloud-generatorleul-sayonghan-word-cloud-129412610808060>
- ▶ 빅카인즈의 [뉴스분석-형태소개체명분석]

뉴스 분석	기획 분석	뉴스 보기	빅카인즈 활용	보
뉴스검색 분석	형태소개체명 분석	분석결과 시각화	시각화보고서 만들기 ▼	

Part2. 파이썬 문장분석 기본

2-1. 문장(string)내에서 단어를 찾아 출현빈도수 계산

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt
# %matplotlib inline
plt.rcParams['font.family'] = 'NanumGothic'
```

```
txt='강아지 산책 강아지 목욕 강아지 미용 강아지 쇼핑 친구와 저녁 먹음 가족과 점심 먹음 혼자 저녁 먹음 친구와 쇼핑'
find_word='강아지'
cnt=txt.count(find_word)    # count 글자위에서 shift+tab키를 눌러서 도움말을 확인
print('▶선택하신 %s 단어의 출현빈도수는 => %d 번입니다.' %(find_word,cnt))

plt.bar(find_word,cnt)
plt.ylim(0,10)
```

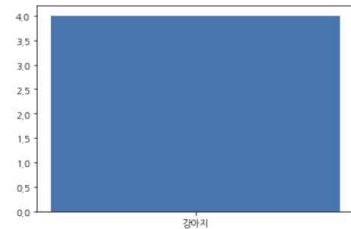
▶count함수의 특징

- 대소문자를 구별함문자를 구분합니다.
- 문자열의 개수는 1개이상임
- count 옵션값을 넣지 않으면 문자열의 처음에서 마지막까지 탐색함.

```
1 find_word='강아지'
2 cnt=txt.count(find_word) # count 글자위에서 shift+tab키를 눌러서
3 print('▶선택하신 %s 단어의 출현빈도수는 => %d 번입니다.' %(find_wo
```

Docstring:
S.count(sub[, start[, end]]) -> int
Return the number of non-overlapping occurrences of substring sub in

<BarContainer object of 1 artists>



1 10

```
1 txt='강아지 산책 강아지 목욕 강아지 미용 강아지 쇼핑 친구와 저녁 먹음 가족과 점심 먹음 혼자 저녁 먹음 친구와 쇼핑'
2 find_word='강아지'
3 cnt=txt.count(find_word,0,10)
4 cnt
```

2

- ▶ 두 개이상의 string를 갖는 리스트구조에서는 한 개의 문자열로 합친후 작업해야함.

```
txtList=['강아지 산책 강아지 목욕 강아지 미용 강아지 쇼핑',
        '친구와 저녁 먹음 가족과 점심 먹음 혼자 저녁 먹음 친구와 쇼핑']

# txtList[0]+txtList[1] 또는 for 구문을 이용해야 하나 파이썬에서는 join함수로 쉽게 결합할수 있음.
txt=''.join(txtList)
find_word='강아지'
cnt=txt.count(find_word)
cnt
```

** 참고 count함수는 이렇게 만들어 집니다.

인덱스		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
txt		강	아	지		산	책		강	아	지		목	욕		강	아

찾는글자 '강아지'	0,1,2	강	아	지													
	1,2,3		아	지													
	2,3,4			지		산											
	3,4,5				산	책											
	4,5,6				산	책											
	5,6,7					책		강									
	6,7,8							강	아								
	7,8,9							강	아	지							
	8,9,10								아	지							
	9,10,11									지		목					
	10,11,12										목	욕					
	11,12,13										목	욕					
	12,13,14											욕		강			
	13,14,15													강	아		

```

찾는글자길이=len(find_word)
cnt=0
for i in range(len(txt)-찾는글자길이+1):
    start=i; end=start+찾는글자길이
    자른문자열=txt[start:end]
    if find_word==자른문자열:
        cnt+=1
print(cnt)

```

```

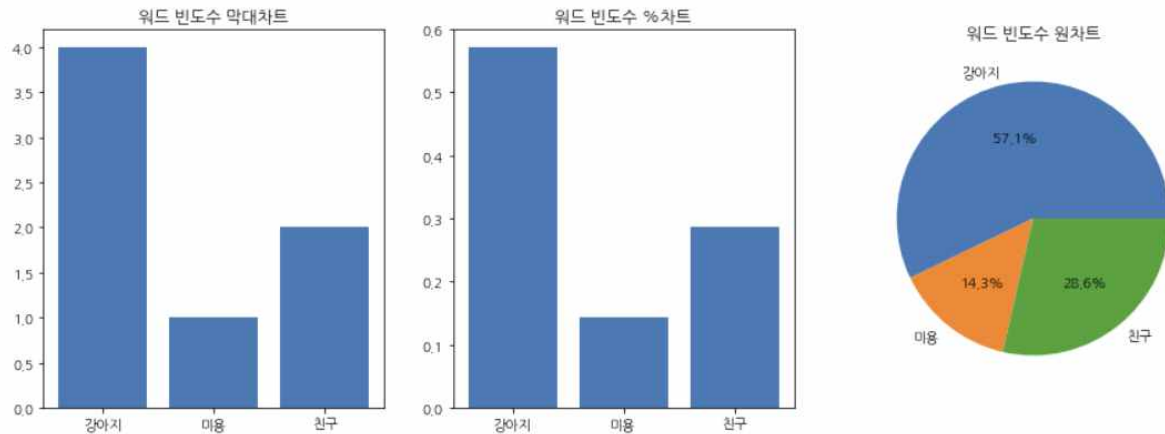
1 # Count 함수 제작
2 def mycount(txt,find_word):
3     찾는글자길이=len(find_word)
4     cnt=0
5
6     for i in range(len(txt)-찾는글자길이+1):
7
8         start=i ;end=start+찾는글자길이
9         자른문자열=txt[start:end]
10        if find_word==자른문자열:
11            cnt+=1
12    return cnt
13
14 txtList=['강아지 산책 강아지 목욕 강아지 미용 강아지 쇼핑',
15          '친구와 저녁 먹음 가족과 점심 먹음 혼자 저녁 먹음 친구와 쇼핑']
16 txt=''.join(txtList)
17
18
19 find_word='강아지'
20 mycount(txt,find_word)
21

```

2-2. 단어빈도수 차트제작

‘강아지’, ‘미용’, ‘친구’ 관련한 자료만 단어별 빈도수를 집계하고, 막대차트와 원차트 제작

['강아지', '미용', '친구'] [4, 1, 2] [57.14285714 14.28571429 28.57142857]



```
import matplotlib.pyplot as plt
import numpy as np
from wordcloud import WordCloud
plt.rcParams['font.family'] = 'NanumGothic'

txtList=['강아지 산책 강아지 목욕 강아지 미용 강아지 쇼핑',
        '친구와 저녁 먹음 가족과 점심 먹음 혼자 저녁 먹음 친구와 쇼핑']
txt=''.join(txtList)

find_word=['강아지','미용','친구']    ## find_word=np.unique(txt.split(' '))
word=[];cnt=[]

for i in find_word:
    word.append(i)
    cnt.append(txt.count(i))
percent=np.array(cnt)/sum(cnt)
print(word,cnt,percent*100)

## -----차트 출력
plt.figure(figsize=(15,5))
plt.subplot(1,3,1) ; plt.bar(word, cnt) ; plt.title('워드 빈도수 막대차트')

plt.subplot(1,3,2) ; plt.bar(word, percent) ; plt.title('워드 빈도수 %차트')

plt.subplot(1,3,3) ; plt.pie(cnt, labels=word, autopct='%1.1f%%', shadow=False)
plt.title('워드 빈도수 원차트')
plt.show()
```

2-3. 네이버 영화 주요 정보에서 단어빈도수 차트제작

The screenshot shows a web browser displaying the Naver movie page for 'The Great Escape' (대탈출). The page includes a navigation bar with '주요정보' (Main Information) highlighted. The main content area shows the movie's title, cast, and a synopsis. A code editor is overlaid on the page, showing the following Python code:

```
import requests
from bs4 import BeautifulSoup

url = 'https://movie.naver.com/movie/bi/mi/basic.naver?code=190991'

response = requests.get(url)

if response.status_code == 200:
    html = response.text
    soup = BeautifulSoup(html, 'html.parser')
else:
    print(response.status_code)

txt_List=[]
for i in soup.find_all('p', 'con_tx'):
    txt_List.append(i.get_text())

txt=''.join(txt_List)

print(txt)
```

The code is designed to scrape the movie's description from the Naver movie page. It uses the requests library to get the page content and BeautifulSoup to parse the HTML. The code then extracts all paragraphs with the class 'con_tx' and joins them into a single string. The final output is printed to the console.

<pre>import requests from bs4 import BeautifulSoup import numpy as np import matplotlib.pyplot as plt plt.rcParams['font.family'] = 'NanumGothic' url = 'https://movie.naver.com/movie/bi/mi/basic.naver?code=190991' response = requests.get(url) if response.status_code == 200: html = response.text soup = BeautifulSoup(html, 'html.parser') else: print(response.status_code)</pre>	<pre>txt_List=[] for i in soup.find_all('p', 'con_tx'): txt_List.append(i.get_text()) txt=''.join(txt_List) find_word=['수학', '과학', '학문'] word=[];cnt=[] for i in find_word: word.append(i) cnt.append(txt.count(i)) percent=np.array(cnt)/sum(cnt) print(word,cnt,percent*100) ## -----차트 출력 plt.figure(figsize=(15,5)) plt.subplot(1,3,1) plt.bar(word, cnt) plt.title('워드 빈도수 막대차트') plt.subplot(1,3,2) plt.bar(word, percent) plt.title('워드 빈도수 %차트') plt.subplot(1,3,3) plt.pie(cnt, labels=word, autopct='%1.1f%%', shadow=False) plt.title('워드 빈도수 원차트') plt.show()</pre>
---	--

2-4. 외부자료(csv 또는 txt 파일) 의 전처리 된 단어를 이용한 countv

[1] 빅카인즈에서 회원가입후 뉴스기사를 검색하여서 엑셀로 자료를 다운로드 한뒤 본문을 특성추출하여서 가중치 상위50개를 분석한 자료로 워드 카운트를 하고자함.

‘빅카인즈샘플.xlsx’ 자료를 읽어서 16번째열의 본문과 15번째열의 특성추출열만 별도로 분리

본문: 문장으로 되어 있음	단어별로 쉼표(.)로 나누어놓음.
본문	특성추출(가중치순 상위 50개)
0 울산 남구 선암호수공원이 봄맞이 준비를 완료했다. \n\n23일 남구에 따르면 선암...	선암호수공원,남구,코로나바이러스,봄맞이,선암,울산,선암호수,봄맞이놀꽃,꽃말을,가우라...
1 울산북구의회가 공동주택 입주민의 삶의 질을 높이고 공동주택 공동체의 활성화를 위...	공동주택,공동체,조례안,본회의,공동주택관리법,임시회,울산시,관리법,우수관리단,지,울산
2 23일 울산시의회 프레스센터에서는 국민의힘 소속 전현직 울산시의원들의 6·1 지...	울주,시의원,중구,울산,울산시의원,5대,울산시,울주군수,국민의힘,울산시의회,울주군,...
울산 울주군이 2022년 고고보도 첫다 출카도 기반 디지털 형식 커성리 대사기과	출카도 울주군 지역구 노인 구보세 오카기과가 여객서 하구저보하지효의 울사 자

이자료에서 단어별로 쉼표(,)로 나누어 놓은 자료는 2차원 리스트임.

[[선암호수공원,남구,코로나바이러스],

[공동주택, 조례안, 본회의]]

=> 이 자료를

선암호수공원, 남구, 코로나바이러스, 공동주택, 조례안, 본회의 또는

선암호수공원 남구 코로나바이러스 공동주택 조례안 본회의 의 한문장으로 만들어야함.

단어와 단어사이의 구분자는 join함수에서 결정할수 있음.

[2] 빅카인즈에서 회원가입후 뉴스 기사를 검색하여서 엑셀로 자료를 다운로드 한뒤 본문을 특성추출하여서 가중치 상위50개를 분석한 자료로 워드 카운트를 하고자함.

*텍스트자료는 입을 때 read로 읽으면 한 개의 문자열로 자료를 읽음

*readlines() 로 읽으면 리스트로 읽음.

```
1 f=open('빅카인즈_코로나.txt','r',encoding='utf-8')
2 txt=f.read()
3 f.close
4 txt=txt[:1000]
5 txt
```

·Wueff소상공인 파주시 매출액 지원금 100만 종사자 코로나19 끝자리 사업자 사업장 3억 최종환
학습지도사 동의서Wn코로나 미국 사망자 확진자 1a 코로나19 30만 겨울철 존스홉킨스대학 35만Wn
부산시 감사소 9개월 부산환경공단 종사자 서경민 입소자 코호트 임시선별검사소 요양병원 파랑새
자 종사자 소상공인 창원 100만 목욕장업 재산세 임대료 재난지원금 노동자 사업장 사용자 중고기
Wn미세먼지 중국 국립환경과학원 코로나19 계절관리제 과학원 연평균 김영우Wn북미 금지령 참석자
영산편지 이은경 김형근 취재기자 과태료 칠레 열매열시 코로나 산티아고 7명 독서부 200여 2주

Part3. 파이썬 워드클라우드 모듈 설치 -

[방법1] 주피터 노트북 또는 콘다 cmd에서 설치

설치장소	설치명령
주피터노트북	!pip install wordcloud
콘다 프롬프트	conda install wordcloud

[방법2] whl(wheel) 자료를 다운받아 직접 설치

whl(wheel) 파일은 파이썬 패키지를 Windows 환경에서 설치하기 위한 패키지 설치파일임

2-1. 파이썬 버전을 확인한다

```
import sys  
sys.version
```

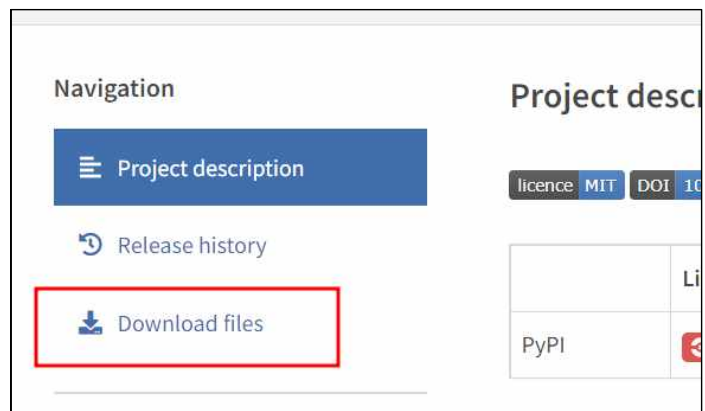
```
1 # 아래 내용이 실행되면서 에러가 나면  
2 #!pip install wordcloud  
3 import sys  
4 print(sys.version)
```

3.9.7 (default, Sep 16 2021, 16:59:28) [MSC v.1916 64 bit (AMD64)]

2-2. whl 다운로드 사이트로 이동

구글검색 'wordcloud whl download'

<https://pypi.org/project/wordcloud/>



2-3. 파이썬 버전과 맞는 워드클라우드 whl 파일을 다운로드 받음.

이 안내서에서는

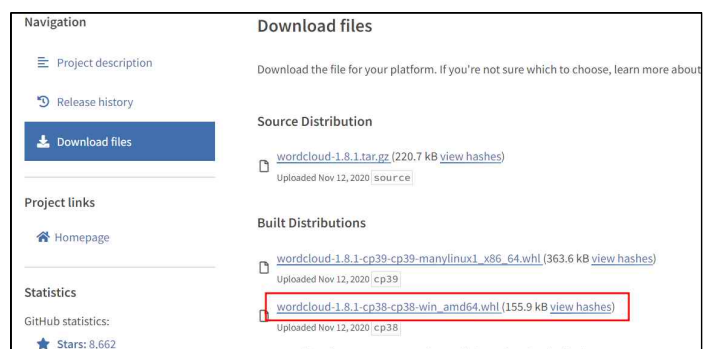
'파이썬 - 3.9

Windows10 환경에

64비트 운영체제'

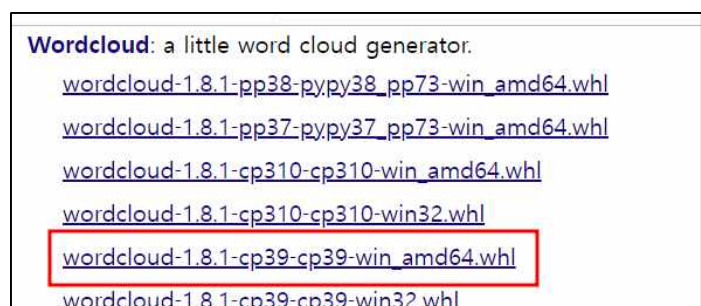
임.

현재 3.8 버전까지만 있음으로



2-4. 이 사이트에서 3.9버전을 다운로드함.

<https://www.lfd.uci.edu/~gohlke/pythonlibs/#wordcloud>



2-5. cmd를 실행한뒤

cd downloads

로 다운로드 폴더로 이동함

C:\Users\BSS>cd downloads

C:\Users\BSS\Downloads>dir *.whl

2-6. 폴더목록에서 whl 파일이 있는지 확인

dir *.whl

2-7. whl 파일 설치

pip install whl파일명

```
C:\Users\BSS\Downloads>pip install wordcloud-1.8.1-cp39-cp39-win_amd64.whl
Processing c:\Users\BSS\downloads\wordcloud-1.8.1-cp39-cp39-win_amd64.whl
Requirement already satisfied: numpy>=1.6.1 in c:\Users\BSS\anaconda3\lib\site-packages (from wordcloud==1.8.1)
Requirement already satisfied: pillow in c:\Users\BSS\anaconda3\lib\site-packages (from wordcloud==1.8.1)
Requirement already satisfied: matplotlib in c:\Users\BSS\anaconda3\lib\site-packages (from wordcloud==1.8.1)
Requirement already satisfied: cycler>=0.10 in c:\Users\BSS\anaconda3\lib\site-packages (from wordcloud==1.8.1)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\Users\BSS\anaconda3\lib\site-packages (from wordcloud==1.8.1)
Requirement already satisfied: python-dateutil>=2.7 in c:\Users\BSS\anaconda3\lib\site-packages (from wordcloud==1.8.1)
Requirement already satisfied: pyparsing>=2.2.1 in c:\Users\BSS\anaconda3\lib\site-packages (from wordcloud==1.8.1)
Requirement already satisfied: six in c:\Users\BSS\anaconda3\lib\site-packages (from cycler>=0.10->wordcloud==1.8.1)
wordcloud is already installed with the same version as the provided wheel. Use --force-reinstall to force the wheel.
```

pip install wordcloud-1.8.1-cp39-cp39-win_amd64.whl

주피터노트북에서 워드클라우드를 실행하여 봄

직접 빈도수를 구해봄	워드클라우드 모듈을 이용해서 구해봄																																																												
<table><thead><tr><th></th><th>단어</th><th>빈도수</th><th>퍼센트</th><th>빈도수/빈도수중max값</th></tr></thead><tbody><tr><td>1</td><td>강아지</td><td>4</td><td>0.210526</td><td>1.00</td></tr><tr><td>2</td><td>먹음</td><td>3</td><td>0.157895</td><td>0.75</td></tr><tr><td>6</td><td>쇼핑</td><td>2</td><td>0.105263</td><td>0.50</td></tr><tr><td>7</td><td>저녁</td><td>2</td><td>0.105263</td><td>0.50</td></tr><tr><td>9</td><td>친구와</td><td>2</td><td>0.105263</td><td>0.50</td></tr><tr><td>0</td><td>가족과</td><td>1</td><td>0.052632</td><td>0.25</td></tr><tr><td>3</td><td>목욕</td><td>1</td><td>0.052632</td><td>0.25</td></tr><tr><td>4</td><td>미용</td><td>1</td><td>0.052632</td><td>0.25</td></tr><tr><td>5</td><td>산책</td><td>1</td><td>0.052632</td><td>0.25</td></tr><tr><td>8</td><td>점심</td><td>1</td><td>0.052632</td><td>0.25</td></tr><tr><td>10</td><td>혼자</td><td>1</td><td>0.052632</td><td>0.25</td></tr></tbody></table>		단어	빈도수	퍼센트	빈도수/빈도수중max값	1	강아지	4	0.210526	1.00	2	먹음	3	0.157895	0.75	6	쇼핑	2	0.105263	0.50	7	저녁	2	0.105263	0.50	9	친구와	2	0.105263	0.50	0	가족과	1	0.052632	0.25	3	목욕	1	0.052632	0.25	4	미용	1	0.052632	0.25	5	산책	1	0.052632	0.25	8	점심	1	0.052632	0.25	10	혼자	1	0.052632	0.25	<pre>wc=WordCloud(font_path='./NanumGothic.ttf', background_color='black', width=1000,height=1000, max_words=10, max_font_size=200) wordFre=wc.generate(tmp) print(wordFre) display(wordFre.words_)</pre> <pre>{'강아지': 1.0, '먹음': 0.75, '쇼핑': 0.5, '친구와': 0.5, '저녁': 0.5, '산책': 0.25, '목욕': 0.25, '미용': 0.25, '가족과': 0.25, '점심': 0.25}</pre>
	단어	빈도수	퍼센트	빈도수/빈도수중max값																																																									
1	강아지	4	0.210526	1.00																																																									
2	먹음	3	0.157895	0.75																																																									
6	쇼핑	2	0.105263	0.50																																																									
7	저녁	2	0.105263	0.50																																																									
9	친구와	2	0.105263	0.50																																																									
0	가족과	1	0.052632	0.25																																																									
3	목욕	1	0.052632	0.25																																																									
4	미용	1	0.052632	0.25																																																									
5	산책	1	0.052632	0.25																																																									
8	점심	1	0.052632	0.25																																																									
10	혼자	1	0.052632	0.25																																																									
<pre>plt.figure(figsize=(12,12)) plt.imshow(wordFre, interpolation='bilinear') plt.axis('off') plt.show()</pre>	