텍스트 마이닝이란?

- 1. 텍스트안에 숨어있는 경향이나 패턴을 찾아서 양질의 정보를 만들어 내기 위함이다.
- 1. 일반적으로는 통계적인 방법을 사용한다. ex. 회귀분석
- 1. 비정형 데이터를 분석 가능한 데이터로 변형하는 과정

텍스트 마이닝 방법

- 자연어 처리 (NLP)
- 통계학, 선형대수
- 머신러닝
- 딥러닝

텍스트 마이닝 단계

- 일반적으로는 한 개의 문장을 최소 단위로 지정한다
- 1. 단어 단위로 쪼개는 것 -〉 tokenize
- 1. 표준화, 영단어의 과거형 미래형등을 원형으로 돌리는 것 -〉 normalize
- 1. 순서가 의미있는 단어들의 시퀀스로 만든다.

- 사용하고자 하는 것에 따라 방법론이 달라진다
- 1. FIXED SIZE VECTOR (순서 무시) EX. BOW, TFIDF
- 1. FIXED SIZE VECTOR (순서 존중) DOC2VEC
- 1. SERIES OF WORD EMBEDDING (순서 정보 유지) (리스트 안의 단어들이 어떤 VECTOR로 변형된다)

파이썬을 이용한 텍스트 마이닝 도구

- 1. NLTK (NATURAL LANGUAGE TOOL KIT)
- -〉 가장 많이 알려진 영어 기준의 라이브러리
- 1. SCIKIT LEARN
- -〉 머신러닝 라이브러리, 기본적인 NLP, 다양한 텍스트 마이닝 도구 지원
- 1. GENSIM
- -〉 WORD2VEC 으로 유명해졌다
- 1. KERAS
- -〉 이 외에 가장 요즘 많이 사용하는 것은 PYTORCH

텍스트 마이닝의 기본 도구

- 집합이라는 단어는 틀리고, 시퀀스라는 말을 사용하여 중복됨을 허용한다.
- 워드로 시퀀스 된 애들을 벡터로 변환한다.
- 값을 가진 애들이 희소하다고 하면 SPARSE라고한다.

- TOKENIZE : 대상이 되는 문서와 문장을 최소 단위로 쪼갠다
- TEXT NORMALIZATION : 최소 단위를 표준화 시키는 것
- POS - TAGGING : 나눠진 값들에 품사를 부착시키는 것
- CHUNKING : 이전 단계에서 품사를 붙힌 후, 다시 하나의 문장으로 결합하는 과정
- BOW, TFIDF : TOKENIZE를 이용하여, 문서의 VECTOR를 표현하기 위함

Tokenize

- 다큐먼트 -〉 문장 , 문장 -〉 단어 세분화 시키는 과정
- split함수를 쓰듯이 분리 가능 (영어에서만 가능하다)
- 한글은 영어처리에 비해 어렵다

Text Normalization

- 단어를 원형으로 돌리는 것 (went -〉 go)

- Stemming (어간 추출) -〉 다양한 형태를 원형으로 바꾸는 것, 의미는 무시되고 규칙에의해서만 변화
- Lemmatization (표제어 추출) -〉 라이브러리의 사전을 이용하여 추출한다. 품사를 고려하고 wordnet lemmatizer 이용

KoNLPy를 이용한 실습

In [1]:

```
from konlpy.corpus import kolaw
c = kolaw.open('constitution.txt').read()
```

- 가져온 데이터의 type을 확인
- 데이터 type은 string이다.

In [2]:

```
print(type(c))
print(len(c))
print(c[:600])
```

<class 'str'>
18884
대한민국헌법

유구한 역사와 전통에 빛나는 우리 대한국민은 3·1운동으로 건립된 대한민국임시정부의 법통과 불의에 항거한 4·19민주이념을 계승하고, 조국의 민주개혁과 평화적 통일의 사명에 입각하여 정의·인도와 동포애로써 민족의 단결을 공고히 하고, 모든 사회적 폐습과 불의를 타파하며, 자율과 조화를 바탕으로 자유민주적 기본질서를 더욱 확고히 하여 정치·경제·사회·문화의 모든 영역에 있어서 각인의 기회를 균등히 하고, 능력을 최고도로 발휘하게 하며, 자유와 권리에 따르는 책임과 의무를 완수하게 하여, 안으로는 국민생활의 균등한 향상을 기하고 밖으로는 항구적인 세계평화와 인류공영에 이바지함으로써 우리들과 우리들의 자손의 안전과 자유와 행복을 영원히 확보할 것을 다짐하면서 1948년 7월 12일에 제정되고 8차에 걸쳐 개정된 헌법을 이제 국회의 의결을 거쳐 국민투표에 의하여 개정한다.

제1장 총강
제1조 ① 대한민국은 민주공화국이다.
②대한민국의 주권은 국민에게 있고, 모든 권력은 국민으로부터 나온다.
제2조 ① 대한민국의 국민이 되는 요건은 법률로 정한다.
②국가는 법률이 정하는 바에 의하여 재외국민을 보호할 의무를 진다.
제3조 대한민

NLTK (Natural Language Tool Kit을 이용한 자연어 처리)

In [3]:

```
import nltk
nltk.download('punkt')
from nltk.tokenize import sent_tokenize
c_sent = sent_tokenize(c)
print(len(c_sent))
print(c_sent[:5])
```

357
['대한민국헌법₩n₩n유구한 역사와 전통에 빛나는 우리 대한국민은 3·1운동으로 건립된 대한민국임시정부의 법통과 불의에 항거한 4·19민주이념을 계승하고, 조국의 민주개혁과 평화적 통일의 사명에 입각하여 정의·인도와 동포애로써 민족의 단결을 공고히 하고, 모든 사회적 폐습과 불의를 타파하며, 자율과 조화를 바탕으로 자유민주적 기본질서를 더욱 확고히 하여 정치·경제·사회·문화의 모든 영역에 있어서 각인의 기회를 균등히 하고, 능력을 최고도로 발휘하게 하며, 자유와 권리에 따르는 책임과 의무를 완수하게 하여, 안으로는 국민생활의 균등한 향상을 기하고 밖으로는 항구적인 세계평화와 인류공영에 이바지함으로써 우리들과 우리들의 자손의 안전과 자유와 행복을 영원히 확보할 것을 다짐하면서 1948년 7월 12일에 제정되고 8차에 걸쳐 개정된 헌법을 이제 국회의 의결을 거쳐 국민투표에 의하여 개정한다.', '제1장 총강₩n 제1조 ① 대한민국은 민주공화국이다.', '②대한민국의 주권은 국민에게 있고, 모든 권력은 국민으로부터 나온다.', '제2조 ① 대한민국의 국민이 되는 요건은 법률로 정한다.', '②국가는 법률이 정하는 바에 의하여 재외국민을 보호할 의무를 진다.']

[nltk_data] Downloading package punkt to
[nltk_data] C:₩Users₩USER₩AppData₩Roaming₩nltk_data...
[nltk_data] Package punkt is already up-to-date!

- 단어 하나 하나로 쪼개는 것을 진행한다.

In [4]:

```
from nltk.tokenize import word_tokenize
c_words = word_tokenize(c)
len(c_words)
```

Out[4]:

4640

- 0부터 50개까지만 카운트해서 나타낸다.

In [5]:

```
" ".join(c_words[:50])
```

Out[5]:

'대한민국헌법 유구한 역사와 전통에 빛나는 우리 대한국민은 3·1운동으로 건립된 대한민국임시정부의 법통과 불의에 항거한 4·19민주이념을 계승하고 , 조국의 민주개혁과 평화적 통일의 사명에 입각하여 정의·인도와 동포애로써 민족의 단결을 공고히 하고 , 모든 사회적 폐습과 불의를 타파하며 , 자율과 조화를 바탕으로 자유민주적 기본질서를 더욱 확고히 하여 정치·경제·사회·문화의 모든 영역에 있어서 각인의 기회를 균등히'

- 형태소 단위로 tokenize를 진행한다

In [6]:

```
from konlpy.tag import Okt
okt = Okt()
tokens_c = okt.morphs(c)
```

- tokens_c의 길이와 type을 확인한다.

In [7]:

```
len(tokens_c)
```

Out[7]:

8796

In [8]:

```
type(tokens_c)
```

Out[8]:

list

In [9]:

```
" ".join(tokens_c[:50])
```

Out[9]:

'대한민국 헌법 ₩n₩n 유구 한 역사 와 전통 에 빛나는 우리 대 한 국민 은 3 · 1 운동 으로 건립 된 대한민국 임시정부 의 법 통과 불의 에 항거 한 4 · 19 민주 이념 을 계승 하고 , 조국 의 민주 개혁 과 평화 적 통일 의 사명'

- nltk의 Text class를 이용하여 다양한 기능을 수행
- tokens_c 대신 c를 사용

In [10]:

```
import nltk
c_nltk_text = nltk.Text(tokens_c, name = "대한민국헌법")
c_nltk_text
```

Out[10]:

<Text: 대한민국헌법>

- 모든 단어의 수
- 서로 다른 단어의 수

In [11]:

```
print(len(c_nltk_text.tokens))
print(len(set(c_nltk_text.tokens)))
```

8796
1364

- 단어와 단어의 빈도를 dictionary 형태로 표현

In [12]:

```
c_nltk_text.vocab()
```

Out[12]:

FreqDist({'의': 380, '.': 357, '에': 282, '을': 211, '₩n': 195, '은': 179, '제': 178, '이': 176, '한다': 155, '·': 145, ...})

- 그림 그리기 전 폰트를 한글로 설정하는 과정

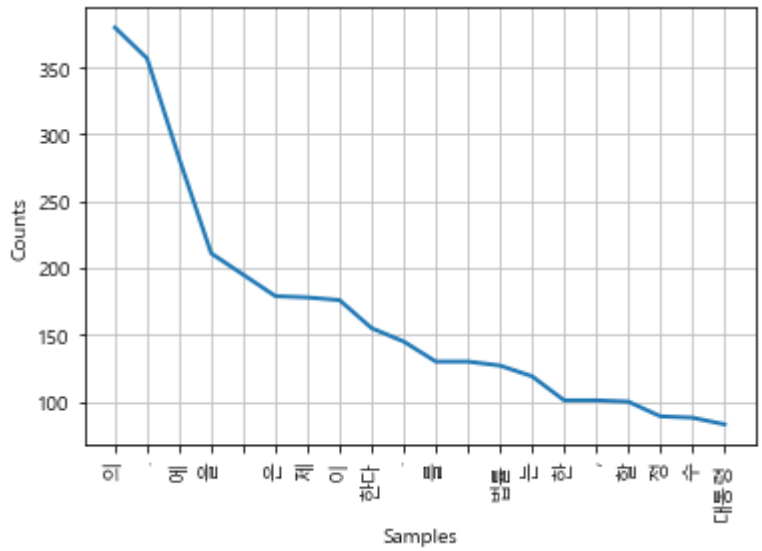
In [13]:

```
from matplotlib import font_manager, rc
font_name = font_manager.FontProperties(fname="c:/Windows/Fonts/malgun.ttf").get_name()
rc('font', family=font_name)
```

- 높은 빈도를 띈 단어들이 그래프로 출력된다.
- 의미 없는 단어나 특수문자가 많다

In [14]:

```
%matplotlib inline
c_nltk_text.plot(20)
```



- 띄어쓰기를 제외하고 나머지 문자열의 길이가 "둘" "이상"인 "단어만 포함"

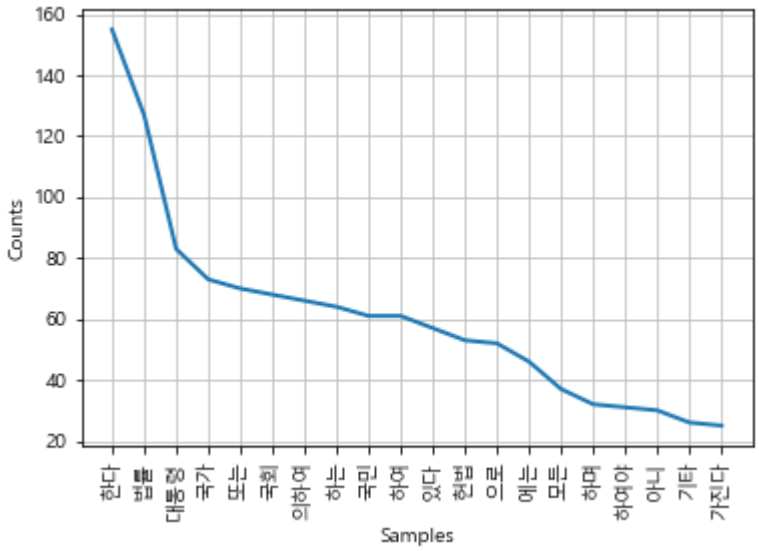
In [15]:

```
tokens_new = []
for token in tokens_c:
    if len(token.strip()) > 1:
        tokens_new.append(token.strip())
print(len(tokens_new))
```

4520

In [16]:

```
c_nltk_text = nltk.Text(tokens_new, name = "대한민국헌법")
c_nltk_text.plot(20)
```



- 특정한 단어의 빈도를 알고 싶다면 "c_nltk_text.count"를 사용한다

In [17]:

```
c_nltk_text.count('대한민국')
c_nltk_text.count('헌법')
c_nltk_text.count('대통령')
c_nltk_text.count('국민')
```

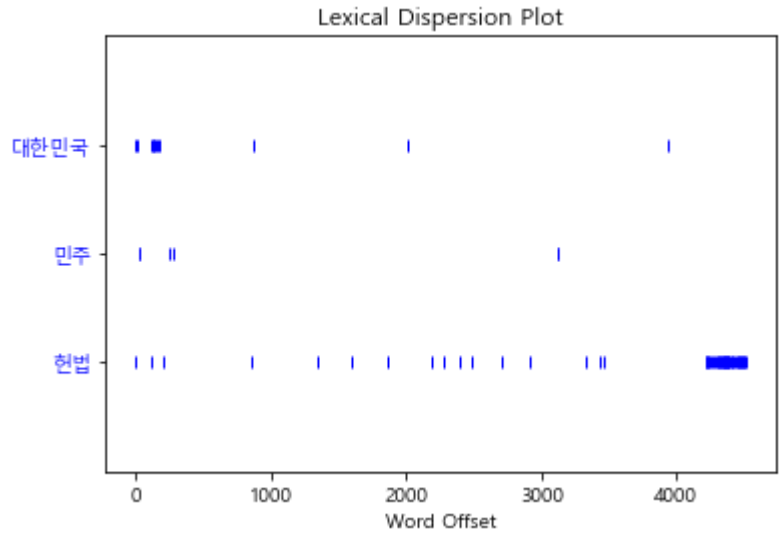
Out[17]:

61

- 여러 단어들 사이에 주어진 단어들의 위치를 표시한다

In [18]:

```
c_nltk_text.dispersion_plot(['대한민국', '민주', '헌법'])
```



- Context를 보는 방법
- concordance(): 주어진 단어를 중심으로 앞과 뒤의 단어들을 나타낸다.

In [19]:

```
c_nltk_text.concordance('대통령')
```

```
Displaying 25 of 83 matches:
조 국회 정기회 법률 하는 의하여 매년 1회 집회 되며 국회 임시회 대통령 또는 국회 재적 의원 4분 이상 요
구 의하여 집회 된다 정기회 기는
하여 집회 된다 정기회 기는 100일 임시회 기는 30일 초과 없다 대통령 임시회 집회 요구 에는 기간 집회
요구 이유 명시 하여야 한다 48
있다 53조 국회 에서 의결 법률 안은 정부 이송 되어 15일 이내 대통령 공포 법률 이의 있을 에는 대통령 항
의 기간 이의 붙여 국회 환부
안은 정부 이송 되어 15일 이내 대통령 공포 법률 이의 있을 에는 대통령 항의 기간 이의 붙여 국회 환부 하
고 재의 요구 있다 국회 폐회 에
의 붙여 국회 환부 하고 재의 요구 있다 국회 폐회 에도 또한 같다 대통령 법률 일부 대하 또는 법률 수정 하
여 재의 요구 없다 재의 요구 있
상 찬성 으로 전과 같은 의결 하면 법률 안은 법률 로서 확정 된다 대통령 항의 기간 공포 재의 요구 하지 아
니한 에도 법률 안은 법률 로서
 공포 재의 요구 하지 아니한 에도 법률 안은 법률 로서 확정 된다 대통령 항의 규정 의하여 확정 법률 지체
없이 공포 하여야 한다 의하여 법
 한다 의하여 법률 확정 또는 의한 확정 법률 정부 이송 5일 이내 대통령 공포 하지 아니 에는 국회의장 이를
공포 법률 특별한 규정 없는 공
여금 출석 답변 하게 있다 63조 국회 국무총리 또는 국무위원 해임 대통령 에게 건의 있다 항의 해임 건의
국회 재적 의원 3분 이상 발의 의
3분 이상 찬성 있어야 한다 항의 처분 대하 법원 제소 없다 65조 대통령 국무총리 국무위원 행정각부 헌법
재판소 재판관 법관 중앙 선거 관리
의 있어야 하며 의결 국회 재적 의원 과반수 찬성 있어야 한다 다만 대통령 대한 탄핵 국회 재적 의원 과반
수 발의 국회 재적 의원 3분 이상
러나 의하여 민사 이나 형사 상의 책임 면제 되지는 아니 한다 정부 대통령 66조 대통령 국가 원수 이며 외
국 대하 국가 대표 한다 대통령 국
사 이나 형사 상의 책임 면제 되지는 아니 한다 정부 대통령 66조 대통령 국가 원수 이며 외국 대하 국가 대
표 한다 대통령 국가 독립 영토
부 대통령 66조 대통령 국가 원수 이며 외국 대하 국가 대표 한다 대통령 국가 독립 영토 보전 국가 계속 성
과 헌법 수호 책무 진다 대통령
 대통령 국가 독립 영토 보전 국가 계속 성과 헌법 수호 책무 진다 대통령 조국 평화 통일 성실한 의무 진다
행정권 대통령 수반 으로 하는 정
법 수호 책무 진다 대통령 조국 평화 통일 성실한 의무 진다 행정권 대통령 수반 으로 하는 정부 한다 67조
대통령 국민 보통 평등 직접 비밀
 성실한 의무 진다 행정권 대통령 수반 으로 하는 정부 한다 67조 대통령 국민 보통 평등 직접 비밀선거 의
하여 선출 한다 항의 선거 있어서
재적 의원 과반수 출석 공개 회의 에서 다수 얻은 자를 당선자 한다 대통령 후보자 인일 에는 득표 수가 선
거권 총수 3분 이상 아니면 대통령
 대통령 후보자 인일 에는 득표 수가 선거권 총수 3분 이상 아니면 대통령 으로 당선 없다 대통령 으로 선거
있는 자는 국회의원 피선거권 있고
는 득표 수가 선거권 총수 3분 이상 아니면 대통령 으로 당선 없다 대통령 으로 선거 있는 자는 국회의원 피
선거권 있고 선거일 현재 40 하여
거 있는 자는 국회의원 피선거권 있고 선거일 현재 40 하여야 한다 대통령 선거 사항 법률 한다 68조 대통
령 임기 만료 되는 에는 임기 만료
 선거일 현재 40 하여야 한다 대통령 선거 사항 법률 한다 68조 대통령 임기 만료 되는 에는 임기 만료 70
일 내지 40일 전에 후임 선거
만료 되는 에는 임기 만료 70일 내지 40일 전에 후임 선거 한다 대통령 궐위 또는 대통령 당선자 사망 거나
판결 기타 사유 자격 상실한 에
기 만료 70일 내지 40일 전에 후임 선거 한다 대통령 궐위 또는 대통령 당선자 사망 거나 판결 기타 사유 자
격 상실한 에는 60일 이내 후
결 기타 사유 자격 상실한 에는 60일 이내 후임 선거 한다 69조 대통령 취임 즈음 하여 다음 선서 한다 헌법
준수 하고 국가 보위 하며 조
```

- similar(): 주어진 단어와 비슷한 context에서 사용된 단어들을 반환

In [20]:

```
c_nltk_text.similar('대통령')
```

국회 국회의원 대법원 국가 국무위원 모든 의하여 한다 국군 정당 필요한 헌법재판소 법관 다만 항의 국회
의장 중앙 위원 주재
감사원

- 함께 많이 나타난 단어들을 출력

In [21]:

```
nltk.download('stopwords')
c_nltk_text.collocations()
```

의하지 아니하고는; 국무총리 국무위원; 그러하지 아니하다; 단결권 단체교섭권; 헌법재판소 재판관; 단체
교섭권 단체행동권;
인하여 불이익; 대법원 대법관; 단체행동권 가진다; 비밀선거 의하여; 대통령 국무총리; 국무위원 행정각
부; 의하여 공무원

[nltk_data] Downloading package stopwords to
[nltk_data] C:₩Users₩USER₩AppData₩Roaming₩nltk_data...
[nltk_data] Package stopwords is already up-to-date!

- 명사만 추출하여 처리

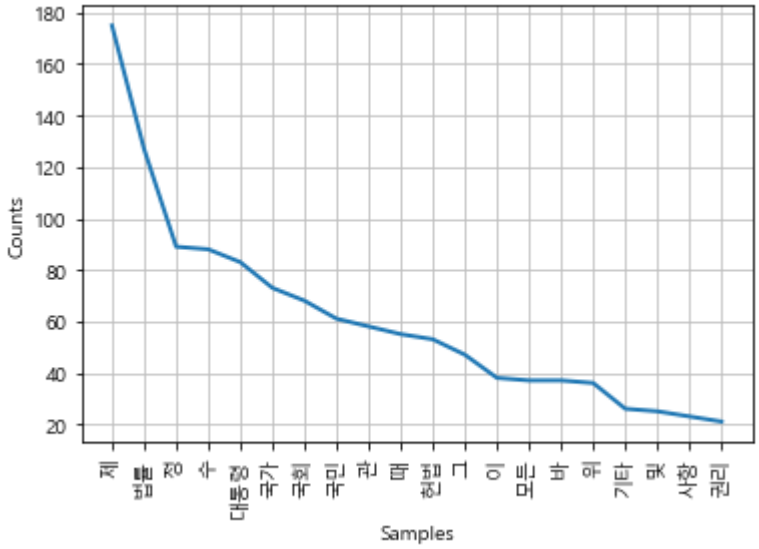
In [22]:

```
nc = okt.nouns(c)
print(len(nc))
print(" ".join(nc[:50]))
```

3882
대한민국 헌법 유구 역사 전통 우리 국민 운동 건립 대한민국 임시정부 법 통과 불의 항거 민주 이념 계승
조국 민주 개혁 평화 통일 사명 입 각하 정의 인도 동포 애 로써 민족 단결 공고 모든 사회 폐습 불의 타파 자
율 조화 바탕 자유민주 질서 더욱 정치 경제 사회 문화 모든

In [23]:

```
ncnt = nltk.Text(nc, name = "Okt명사")
ncnt.plot(20)
```



In [24]:

```
import nltk
nltk.download('gutenberg')
from nltk.corpus import gutenberg  # Docs from project gutenberg.org
files_en = gutenberg.fileids()     # Get file ids
doc_en = gutenberg.open('austen-emma.txt').read()
```

```
[nltk_data] Downloading package gutenberg to
[nltk_data]     C:₩Users₩USER₩AppData₩Roaming₩nltk_data...
[nltk_data]   Package gutenberg is already up-to-date!
```

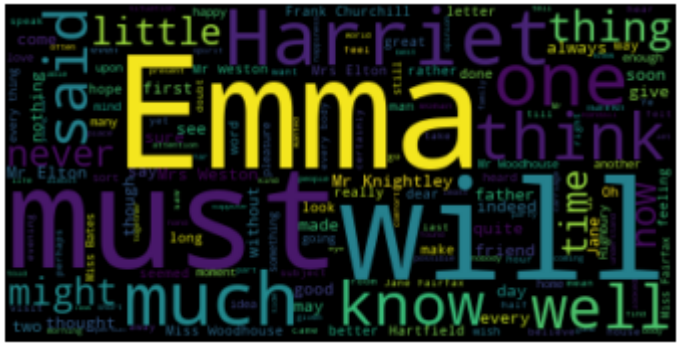
In [25]:

```
from wordcloud import WordCloud

# Generate a word cloud image
wordcloud = WordCloud().generate(doc_en)

# Display the generated image:
# the matplotlib way:
import matplotlib.pyplot as plt
%matplotlib inline

plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



In [26]:

```
# lower max_font_size
wordcloud = WordCloud(max_font_size=40).generate(doc_en)
plt.figure()
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.show()
```

한글을 사용한 Word Cloud

- 그냥 그리면 폰트가 깨지므로 한글 폰트를 아래와 같이 지정
- "text2" 파일로부터 읽어들여 워드클라우드 그리기

In [27]:

```
import matplotlib.pyplot as plt
from wordcloud import WordCloud
```

In [28]:

```
font_path = 'c:/Windows/Fonts/malgun.ttf' # 한글 폰트의 위치를 지정
wordcloud = WordCloud( #폰트 및 다른 속성들을 지정
   font_path = font_path,
   width = 800,
   height = 800,
   max_words=50,
   background_color='white' #영어와 달리 배경을 흰색으로
)

text=open('text2.txt', encoding = 'utf-8').read() #텍스트 파일을 읽음
wordcloud = wordcloud.generate(text) #워드 클라우드 생성

fig = plt.figure(figsize=(12,12)) #그림판 크기를 지정
plt.imshow(wordcloud)
plt.axis("off")
plt.show()
```

In [29]:

```
## 다음 한글 기사 읽어오기
from collections import Counter
import random
import webbrowser

from konlpy.tag import Hannanum
import sys
from bs4 import BeautifulSoup
import urllib.request as req


def get_bill_text_daum():
    url1 = "http://media.daum.net" #URL 변경됨
    res = req.urlopen(url1)
    soup = BeautifulSoup(res, "html.parser")

    text2=soup.select("strong.tit_g > a.link_txt")
    kk= [a.string for a in text2]
    corpus = ''
    for text in kk:
        text = text.replace('₩r','').replace('₩n','').replace('₩t','')
        corpus = corpus + ' ' +text

    return corpus

def get_tags(text, ntags=50, multiplier=10):
    h = Hannanum()
    nouns = h.nouns(text)
    count = Counter(nouns)
    return(nouns)

text = get_bill_text_daum()
tags = get_tags(text)
#print(tags)
#print(" ".join(tags))

### 워드클라우드 그리기
wordcloud = WordCloud( #폰트 및 다른 속성들을 지정
    font_path = font_path,
    width = 800,
    height = 800,
    # max_words=50,
    background_color='white' #영어와 달리 배경을 흰색으로
)

wordcloud = wordcloud.generate(" ".join(tags)) #위에서 만든 tags를 이용하여 word cloud 생성

fig = plt.figure()
fig = plt.figure(figsize=(12,12))
plt.imshow(wordcloud, interpolation="bilinear") #글자의 테두리를 매끄럽게... 위의 결과와 비교
plt.axis("off")
plt.show()
fig.savefig('wordcloud_without_axisoff.png') #이미지 파일로 저장
```

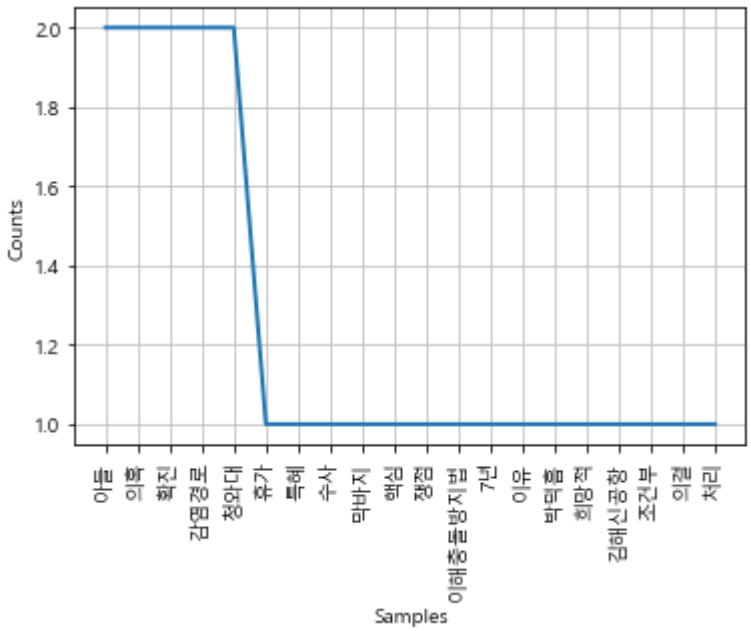
<Figure size 432x288 with 0 Axes>

- 동일한 데이터를 이용해 NLTK를 이용해 빈도 그래프 그리기

In [30]:

```
new_tags = []
for token in tags:
   if len(token.strip()) > 1: # 스페이스를 제외한 나머지 문자열의 길이가 둘 이상인 단어만 포함
      new_tags.append(token.strip())

cnt = nltk.Text(new_tags, name = "다음기사수집")
cnt.plot(20)
```



In [31]:

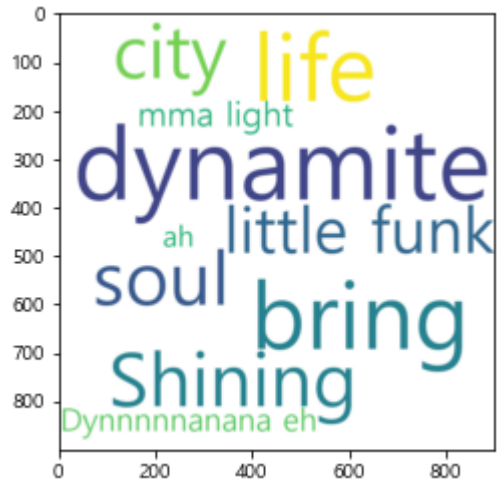
```
from wordcloud import WordCloud
import matplotlib.pyplot as plt
```

- 현재 가장 유명한 가수 "방탄소년단"의 노래 가사들로 진행
- 방탄소년단이 최근에 발매한 노래 "Dynamite"의 가사들로 진행

In [32]:

```
text = "Cos ah ah I'm in the stars tonight So watch me bring the fire and set the night alight Shoes on get u
p in the mornCup of milk let's rock and roll King Kong kick the drum rolling on like a rolling stone Sing song
 when I'm walking home Jump up to the top LeBron Ding dong call me on my phone Ice tea and a game of
 ping pong This is getting heavy Can you hear the bass boom, I'm ready Life is sweet as honey Yeah this be
at cha ching like money Disco overload I'm into that I'm good to go I'm diamond you know I glow up Hey, s
o let's go Cos ah ah I'm in the stars tonight So watch me bring the fire and set the night alight Shining thro
ugh the city with a little funk and soul So I'mma light it up like dynamite, woah Bring a friend join the crowd
Whoever wanna come along Word up talk the talk just move like we off the wall Day or night the sky's alig
ht So we dance to the break of dawn Ladies and gentlemen, I got the medicine so you should keep ya eyes
 on the ball, huh This is getting heavy Can you hear the bass boom, I'm ready Life is sweet as honey Yeah t
his beat cha ching like money Disco overload I'm into that I'm good to go I'm diamond you know I glow up L
et's go Cos ah ah I'm in the stars tonight So watch me bring the fire and set the night alight Shining throug
h the city with a little funk and soul So I'mma light it up like dynamite, woah Dynnnnnanana, life is dynamit
e Dynnnnnanana, life is dynamite Shining through the city with a little funk and soul So I'mma light it up lik
e dynamite, woah Dynnnnnanana eh Dynnnnnanana eh Dynnnnnanana eh Light it up like dynamite Dynnn
nnanana eh Dynnnnnanana eh Dynnnnnanana eh Light it up like dynamite Cos ah ah I'm in the stars tonigh
t So watch me bring the fire and set the night alight Shining through the city with a little funk and soul So
 I'mma light it up like dynamite Cos ah ah I'm in the stars tonight So watch me bring the fire and set the nig
ht alight Shining through the city with a little funk and soul So I'mma light it up like dynamite, woah Dynnn
nnanana, life is dynamite Dynnnnnanana, life is dynamite Shining through the city with a little funk and sou
l So I'mma light it up like dynamite, woah "

wordcloud = WordCloud(font_path = font_path, width = 900, height = 900,  max_words=10, background_c
olor='white'). generate(text)
plt.imshow(wordcloud, interpolation='bilinear')
fig = plt.figure(figsize=(12,12))
plt.show()
plt.savefig("dynamite.png")
```



<Figure size 864x864 with 0 Axes>

<Figure size 432x288 with 0 Axes>

외부의 csv 파일에서 데이터를 읽어, 필요한 부분만 Word Cloud 그리기

In [33]:

```
import pandas as pd
df = pd.read_csv('movie_data(1).csv', header=None, names=['review', 'rate', 'name'])
df.columns.tolist()
#".".join(df.review.tolist())
```

Out[33]:

['review', 'rate', 'name']

In [34]:

```
df.head() #파일의 내용 미리보기 (상위 5개)
```

Out[34]:

	review	rate	name
0	오~~ 종합선물셋트	9	인피니티 워
1	크레딧 올라올때부터 충격먹었어요 ㅋㅋㅋㅋ 나중에 더큰 활약기대하겠습니다! 와칸다포에...	10	인피니티 워
2	이제 남은건타노스 밖에	10	인피니티 워
3	타노스는 발암물질이다.	9	인피니티 워
4	정말재미있게봤습니다	10	인피니티 워

In [35]:

```
wordcloud = WordCloud(
   font_path = font_path,
   max_font_size = 50,
   width = 800,
   height = 800,
   background_color='white',
   max_words=50
)

wordcloud = wordcloud.generate(" ".join(df.review.tolist()))

fig = plt.figure()
fig = plt.figure(figsize=(12,12))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.show()
fig.savefig('wordcloud_without_axisoff.png')
```

<Figure size 432x288 with 0 Axes>

3주차 웹 크롤링 1 (Static Crawling)

1 urllib 사용

- urllib 라이브러리를 사용하여, URL를 다루는 방법
- urllib.request 모듈을 사용하여, 웹사이트 내에 다양한 요청과 처리를 진행

In [36]:

```
from urllib import request
```

1.1 urllib.request를 이용한 다운로드 진행

- urllib.request를 사용하여, urlretrieve()함수를 적용해, 그림 파일을 저장

In [37]:

```
url="http://uta.pw/shodou/img/28/214.png"
savename="test.png"
```

- urlretrieve를 이용해 저장되게 설정

In [38]:

```
request.urlretrieve(url, savename)
print("저장이되어버렸다")
```

```
저장이되어버렸다
```

- urlopen으로 파일에 저장하는 방법을 진행
- urlopen은 메모리에 데이터를 업로드한 후에 파일로 저장하는 방식
- 우선 저장할 파일의 url을 불러온다

1.2 urlopen으로 파일을 따로 저장하는 방법

In [39]:

```
url = "http://uta.pw/shodou/img/28/214.png"
savename = "test1.png"
```

- 불러온 파일을 따로 저장한다

In [40]:

```
memo = request.urlopen(url).read()
```

In [41]:

```
with open(savename, mode="wb") as f:
  f.write(memo)
  print("저장이되어버렸지뭐야")
```

```
저장이되어버렸지뭐야
```

1.3 API 활용하기

- API는 코딩 진행자가 원하는 데이터를 웹상에서 가져올 수 있게 정보를 전달해주는 프로그램이다

- 데이터를 읽어들인다

In [42]:

```
url="http://api.aoikujira.com/ip/ini"
res=request.urlopen(url)
data=res.read()
```

- 바이너리를 문자열로 변환하는 과정
- 바이너리는 본래는 2진수로 표시되는 데이터의 의미지만, 일반적으로 바이너리라고 한다.
- 텍스트 동의 문자로서의 의미를 가진 데이터에 대하여 프로그램의 동작을 결정하는 것, 또는 일정한 포맷에 따라 기록되는 데이터를 말한다
- 출처: 네이버 백과사전

In [43]:

```
text=data.decode("utf-8")
print(text)
```

```
[ip]
API_URI=http://api.aoikujira.com/ip/get.php
REMOTE_ADDR=61.74.153.66
REMOTE_HOST=61.74.153.66
REMOTE_PORT=53708
HTTP_HOST=api.aoikujira.com
HTTP_USER_AGENT=Python-urllib/3.8
HTTP_ACCEPT_LANGUAGE=
HTTP_ACCEPT_CHARSET=
SERVER_PORT=80
FORMAT=ini
```

2. BeautifulSoup

- 파이썬을 이용해 웹사이트 내에 있는 데이터를 스크래핑 하기위해서 필요한 패키지이다.
- HTML / XML에서 정보를 가져올 수 있게 도와준다
- Anaconda Prompt에서 "pip install beautifulsoup4"를 타이핑하여 설치한다.

In [44]:

```
from bs4 import BeautifulSoup
```

In [45]:

```
html = """
<html><body>
 <h1>스크레이핑은 어떻게 하는걸까?</h1>
 <p>웹 페이지를 분석하는 것이다</p>
 <p>원하는 부분을 추출하는 것이다</p>
</body></html>
"""
```

2.1 기본 사용

- 우선적으로 라이브러리를 이용하여 웹 사이트내에 있는 HTML을 가져와 문자열로 만든다.
- 크게 h1, p1, p2로 나눠져있어서 구분지어야한다.

- HTML을 분석한다.

In [46]:

```
soup = BeautifulSoup(html, 'html.parser')
```

- 원하는 부분을 추출하기 위해, h1, p1, p2를 나눠서 대입한다.

In [47]:

```
h1 = soup.html.body.h1
p1 = soup.html.body.p
p2 = p1.next_sibling.next_sibling
```

In [48]:

```
print(f"h1 = {h1.string}")
print(f"p  = {p1.string}")
print(f"p  = {p2.string}")
```

```
h1 = 스크레이핑은 어떻게 하는걸까?
p  = 웹 페이지를 분석하는 것이다
p  = 원하는 부분을 추출하는 것이다
```

2.2 요소를 찾는 방법

- 단일 element 추출을 위해서는 find()라는 함수를 사용한다

In [49]:

```
soup = BeautifulSoup(html, 'html.parser')
```

- find()를 사용하여 원하는 부분만 따로 출력할 수 있다.
- 아래는 h1에 해당하는 입력 값을 추출하기 위한 코드 진행이다.

In [50]:

```
title = soup.find("h1")
body  = soup.find("p")
print(title)
```

<h1>스크레이핑은 어떻게 하는걸까? </h1>

- 텍스트 부분을 출력한다.
- 제목과 본문 부분을 나뉘어서 출력한다

In [51]:

```
print(f"#title = {title.string}" )
print(f"#body = {body.string}")
```

#title = 스크레이핑은 어떻게 하는걸까?
#body = 웹 페이지를 분석하는 것이다

- 이전에는 단일 element를 추출했다면, 이번에는 복수의 elements를 출력한다
- 여러개의 태그를 추출하는법을 알고있어야한다.

In [52]:

```
html = """
<html><body>
 <ul>
  <li><a href="http://www.naver.com">naver</a></li>
  <li><a href="http://www.daum.net">daum</a></li>
 </ul>
</body></html>
"""

soup = BeautifulSoup(html, 'html.parser')
```

- 단일 element와 같이 find()를 사용하여, 추출한다
- 입력시에는 www.naver.com과 www.daum.net를 나눠서 입력했지만
- 출력시에는 한 unit으로 나타난다

In [53]:

```
links = soup.find_all("a")
print(links, len(links))
```

[naver, daum] 2

- 두 웹사이트의 링크를 직관적으로 보기위해, 따로 빼서 출력한다.
- href는 영어로 hyperlink에 해당한다.
- 태그안에 속성을 넣는 이유는 함수에 추가할 수 있는 인자와도 같다.
- 태그가 기본으로 갖는 속성들을 사용자가 정할 수 있고, 정하지 않는다면 기본속성이 default값이 된다.

In [54]:

```
for a in links:
    href = a.attrs['href']
    text = a.string
    print(text, ">", href)
```

naver > http://www.naver.com
daum > http://www.daum.net

3. Css Selector

- Css Selector란 HTML 태그에 Css 스타일을 적용 시키기 위한 지시를 뜻한다.
- Css Selector는 하나의 태그를 설정하여 적용 시키는 것도 가능하고, 복수의 태그를 설정해 적용하는 것도 가능하다.

In [55]:

```
html = """
<html><body>
<div id="meigen">
  <h1>위키북스 도서</h1>
  <ul class="items">
    <li>유니티 게임 이펙트 입문</li>
    <li>스위프트로 시작하는 아이폰 앱 개발 교과서</li>
    <li>모던 웹사이트 디자인의 정석</li>
  </ul>
</div>
</body></html>
"""
```

In [56]:

```
soup = BeautifulSoup(html, 'html.parser')
```

In [57]:

```
h1 = soup.select_one("div#meigen > h1").string
print(f"h1 = {h1}")
```

h1 = 위키북스 도서

In [58]:

```
li_list = soup.select("div#meigen > ul.items > li")
for li in li_list:
  print(f"li = {li.string}")
```

li = 유니티 게임 이펙트 입문
li = 스위프트로 시작하는 아이폰 앱 개발 교과서
li = 모던 웹사이트 디자인의 정석

4. 활용 예제

- Url과 웹으로 부터 html을 읽어서 진행
- html을 원하는 데이터로 추출하는 과정

In [59]:

```
from bs4 import BeautifulSoup
from urllib import request, parse
```

4.1 네이버 금융 정보 활용

- HTML 불러오기

In [60]:

```
url = "https://finance.naver.com/marketindex/"
res = request.urlopen(url)
```

In [61]:

```
soup = BeautifulSoup(res, "html.parser")
```

In [62]:

```
price = soup.select_one("div.head_info > span.value").string
print("usd/krw =", price)
```

usd/krw = 1,175.00

4.2 기상청 RSS

- 기상청에서 제공하는 XML 데이터를 추출하고 XML 내용을 출력한다.

In [63]:

```
url = "http://www.kma.go.kr/weather/forecast/mid-term-rss3.jsp"

values = { 'stnId':'109'}
params=parse.urlencode(values)
url += "?"+params
print("url=", url)

res = request.urlopen(url)
```

url= http://www.kma.go.kr/weather/forecast/mid-term-rss3.jsp?stnId=109

In [64]:

```
soup = BeautifulSoup(res, "html.parser")
```

- 원하는 데이터로 추출

In [65]:

```
header = soup.find("header")

title = header.find("title").text
wf = header.find("wf").text

print(title)
print(wf)
```

서울,경기도 육상중기예보
○ (강수) 10월 1일(목) 오후~2일(금) 오전에는 비가 내리겠습니다.〈br /〉○ (기온) 이번 예보기간 낮 기온은 20~25도로 오늘(26일, 24~26도)보다 낮겠고, 아침 기온은 9~17도로 선선하겠습니다.〈br /〉특히, 내륙을 중심으로 낮과 밤의 기온차가 10도 내외로 크겠습니다.〈br /〉○ (해상) 서해중부해상의 물결은 0.5~2.0m로 일겠습니다.

4.2 윤동주 작가의 작품 목록을 활용

- #mw-content-text 〉 div 〉 ul:nth-child(6) 〉 li 〉 b 〉 a
- nth-child(n) 은 n 번째 요소를 의미 즉 6번째 요소를 의미, #mw-content-text 내부에 있는 url 태그는 모두 작품과 관련된 태그. 따라서 따로 구분할 필요는 없으며 생략해도 됨. BeautifulSoup는 nth-child 지원하지 않음

In [66]:

```
url = "https://ko.wikisource.org/wiki/%EC%A0%80%EC%9E%90:%EC%9C%A4%EB%8F%99%EC%A3%BC"
res = request.urlopen(url)
soup = BeautifulSoup(res, "html.parser")
```

In [67]:

```
a_list = soup.select("#mw-content-text  ul > li  a")
for a in a_list:
    name = a.string
    print(f"- {name}", )
```

- 하늘과 바람과 별과 시
- 증보판
- 서시
- 자화상
- 소년
- 눈 오는 지도
- 돌아와 보는 밤
- 병원
- 새로운 길
- 간판 없는 거리
- 태초의 아침
- 또 태초의 아침
- 새벽이 올 때까지
- 무서운 시간
- 십자가
- 바람이 불어
- 슬픈 족속
- 눈감고 간다
- 또 다른 고향
- 길
- 별 헤는 밤
- 흰 그림자
- 사랑스런 추억
- 흐르는 거리
- 쉽게 씌어진 시
- 봄
- 참회록
- 간(肝)
- 위로
- 팔복
- 못자는밤
- 달같이
- 고추밭
- 아우의 인상화
- 사랑의 전당
- 이적
- 비오는 밤
- 산골물
- 유언
- 창
- 바다
- 비로봉
- 산협의 오후
- 명상
- 소낙비
- 한난계
- 풍경
- 달밤
- 장
- 밤
- 황혼이 바다가 되어
- 아침
- 빨래
- 꿈은 깨어지고
- 산림
- 이런날
- 산상
- 양지쪽
- 닭
- 가슴 1
- 가슴 2

- 비둘기
- 황혼
- 남쪽 하늘
- 창공
- 거리에서
- 삶과 죽음
- 초한대
- 산울림
- 해바라기 얼굴
- 귀뚜라미와 나와
- 애기의 새벽
- 햇빛·바람
- 반디불
- 둘 다
- 거짓부리
- 눈
- 참새
- 버선본
- 편지
- 봄
- 무얼 먹구 사나
- 굴뚝
- 햇비
- 빗자루
- 기왓장 내외
- 오줌싸개 지도
- 병아리
- 조개껍질
- 겨울
- 트루게네프의 언덕
- 달을 쏘다
- 별똥 떨어진 데
- 화원에 꽃이 핀다
- 종시

네이버 뉴스 헤드라인

- 헤드라인 뉴스 제목을 추출
- css selector를 추가하여 최신 기사의 헤드라인을 스크레이핑하는 코드를 완성하시오. # 네이버로 진행할시, HTTPError: HTTP Error 500: Internal Server Error 이 문구가 계속 괴롭혀서 daum뉴스로 진행했습니다

In [131]:

```
import requests
from bs4 import BeautifulSoup

res = requests.get("https://news.daum.net/digital#1")
soup = BeautifulSoup(res.content, "html.parser")

data = soup.find_all('a', 'link_txt')

for item in data:
    print (item.get_text())
```

'던파'에 드리운 악재..넥슨 3조클럽 적신호?
카카오T 블루로 콜 30% '뚝'..카카오 "42% 늘었다" 누가 맞나?
"메이드 인 인터넷 시대 실현"..'꿈의 공장' 만든 알리바바
배그로 뜨고 지고..크래프톤 IPO '명과 암'
코로나에도 늘어나는 가을철 등산객.. "관절·급성 심질환 주의해야"

LGU+ '초등나라', 보상받는 재미에 인강 1편 '뚝딱'

"보급형 모델인데 이 정도"..갤S20 FE, 코로나 시기에 딱이야

주우면 임자 '로또운석'..그런데 어떻게 알아보지?

'팬'들의 목소리에 귀기울였다..갤럭시S20 팬에디션(FE) 이름처럼

5G 속도 6배 빨라졌지만..고객불만 여전히 LTE 2배

녹아내린 남극 빙상 다시 돌아오지 못한다
'맨해튼 프로젝트' 현장에 가다
도시가 조용해지니 새 소리도 나긋해졌다
중국 화웨이기술 광둥성 실험시설서 대형화재로 3명 숨져
중국 화웨이 R&D; 시설서 큰 불..검은 연기 치솟아
중국 화웨이기술 광둥성 실험시설서 대형화재 발생
'던파'에 드리운 악재..넥슨 3조클럽 적신호?
가상자산 넣은 게임 심의 어쩌나..고민하는 게임위
中 막힌 토종 게임..'외산 무덤' 日시장 정복 나섰다
우주를 보다
혜성에도 신비한 오로라 존재..원자외선 극광 포착
달콤한 사이언스
비타민D가 코로나19 증상 완화시킨다
IT동아 리뷰
"진동 브러쉬로 세안하면 뭐가 좋아?", 비디투 핑거클렌저
사이언스카페
치매로 손상된 신경망, 로봇이 복구한다
아하! 우주
달 형성의 '거대충돌설', 또 다른 증거 발견
NHN 화상화의 툴 '두레이', KISA 보안점검 완료
카카오T 블루로 콜 30% '뚝'..카카오 "42% 늘었다" 누가 맞나?
비타민D가 코로나19 증상 완화시킨다
"요즘 누가 노트북 들고 다니나요?"..코로나19가 부른 탭의 전성기
'던파'에 드리운 악재..넥슨 3조클럽 적신호?
"보급형 모델인데 이 정도"..갤S20 FE, 코로나 시기에 딱이야
'하늘에서 온 로또' 라던 '6년 전 진주 운석'..어디 있을까
미 정부, 중 최대 반도체회사에 수출제한.."중국군 활용 우려"
'아픈 손가락' 이병헌의 '베가'.."아직 살아 있었다!"
"보급형 모델인데 이 정도"..갤S20 FE, 코로나 시기에 딱이야
카카오T 블루로 콜 30% '뚝'..카카오 "42% 늘었다" 누가 맞나?
비타민D가 코로나19 증상 완화시킨다

'인천 초등생 형제' 11일 만에 눈 떴던 형, 오늘은..

주식 3억 원 대주주?..개인 투자자들 "연말 하락장..투기꾼·공매도 세력 득세"

'쌍둥이 배' 타보니 '허리 높이' 난간..유족, 실족 가능성 제기

이재명 "우리 죽지 말고 삽시다" 코로나블루에 SNS 글

北 코로나 포비아.."무조건 사살"

'입학만 해준다면'.. 아이폰 뿌리는 대학, 영업사원 된 교수

7개월 만에 오징어 돌아왔지만, 울릉도 어민들 표정은..

서울 오늘 확진자 최소 31명..도봉구 사우나 여탕서 4명 추가

쌀 나눠줬다고 살해된 경주시민들.. 참혹한 사건

"아빠 살리려" 고3 아들의 간이식..70%가 자녀 기증

신도림역 환경미화원 8명 집단감염..휴게실서 함께 식사·휴식

이해충돌방지법 7년 묵힌 이유는? ..박덕흠 의혹으로 그나마 '희망적'

"추미애 장관은 퇴진하라"..서울 5개 구간서 차량시위

"세계 살리겠다" 中 코로나 백신공장 둘러본 외신들 깜짝

추석에 제주만 30만명? ..의료진은 "설 이후 못 본 아빠, 보고파" 눈물

국민의힘, 北 피격사건 진상규명 청와대 앞 1인 시위 '초강수'

英 퀴즈쇼 스타 "중요치않은 작은 밴드"..BTS 무시했다가 된서리

해경, 선내 CCTV 복원 주력.."실종 전날 아들과 통화"

면허 취소된 뒤 '또 음주운전'..다시 따기 쉬워서?

"옷 달라" 피해자 예리한 눈, KBS 몰카범 '악어 눈물' 밝혔다

휴대용 가스레인지 폭발..제조사 "오래 쓰면 그럴 수도"?

구명조끼 입었는데 시신 사라져? ..의문 여전

셔츠입고 팔굽혀펴기..긴즈버그 20년 지기의 마지막길 배웅법

예방접종 중단 백신 접종한 전남도 31명 '이상 반응 없어'

살아 있던 국정원 '핫라인'..야권 "그때 왜 못 살렸나"

상온 노출된 독감백신, 324명 접종..전날 224명에서 100명 늘어

몸에 흡수되는 칼로리 반으로..더 이상 찬밥신세 아닌 찬밥

방문판매 현장 덮치자, 뻔뻔히 "우린 친목 모임"

뇌사 12개월 남아, 석달간의 연명치료 끝에 장기기증

英서 수천명 코로나 대책 항의..하루 확진자는 7000명 육박

'놀면' 유재석, 눈 앞 제시 엉덩이에 깜짝..이효리 "빵댕이 차별하냐"

정미애 임신 7개월차 "남편, 출산하면 쉬어야 하니까 활동 적극 지원"

'쩐당포' 김창열, 185cm 훈남 아들 공개 "고1인데 나보다 커"

'감기' 배우 박민하, 전국 중고 사격대회 금메달 "연예계 활동도 계속"

'아는 형님' 전인화, 아들 지상 언급 "'슈퍼밴드' 출연 말 안 해"

'비밀의 숲2' 조승우x배두나, 통영 생존자 김동휘..이준혁 납치범으로 의심

'앨리스' 김상호=김희선 죽이려한 선생 수하였다, 주원 진실 알아낼까

고은아 "'전참시' 출연 후 광고 들어왔다, 남동생 미르 좋아해"

'아는 형님' 전인화, "김희애·조용원과 중앙대 3대 미인이라 불렸다"

왜색 논란 구설수 극복 못하고 씁쓸한 퇴장

'살림남2' 박애리 "♥팝핀현준 매년 생일에 리무진→해외여행 이벤트 해줘"

'사생활' 고경표, 오늘 모친상..슬픔 속 빈소

'아는형님' 전인화X황신혜, 원조 여신이 뽑은 예쁜 후배 "송혜교-신민아"

'놀면 뭐하니? ' 김종민 "엄정화, 여자로 좋아하기 보다 지켜주고 싶었다"

'놀면 뭐하니? ' 이효리, 꽃무늬 한복 입고 천꽃선녀님으로 변신 "소름"

'오삼광빌라' 황신혜, 딸 한보름 괴롭힌 진기주 표절 증거 제조

스테파니 "♥ 브래디 앤더슨, 말다툼으로 사랑 시작"

'살림남2' 박애리, 시어머니 생일 축하에 눈물 "오래오래 건강하셨으면.."

'미씽' 폐건물에 감금된 남자 "송건희가 가뒀다" 충격

'살림남2' 윤주만 부부, 냉장고 상태에 충격 "유통기한 2년 지났다"

우리액터스 측 "권민아와 계약해지, 편해지고 싶다고 요청"

'아형' 황신혜 "인천 3대 미녀? 서울서 나 보러 인천에 왔었다"

'아는형님' 황신혜 "유동근과 찍은 '애인', 국정감사 올라갈 정도"

'나는 차였어' 90년대 올드카 캠핑카 가격? "100→800만원에 구입"

'비밀의숲2' 조승우, 선배 김영재 방 뒤지다 현장에서 잡혔다

'오! 삼광빌라! ' 진기주, 황신혜에 인턴 제안받았다..한보름과 과거 악연

'놀면 뭐하니? ' 환불원정대, 활동곡은 '돈 터치 미'..10월10일 발표

김남일♥김보민 13세 아들 등장 "아빠 닮아 입맛 호불호 확실"

'전참시' 제시, 가족사진에 폭풍눈물..센언니 텐션에 숨겨진 비하인드

'앨리스' 곽시양, 주원에 아빠라는 사실 숨겼다 "내가 무슨 자격으로.."

이강인 유무는 무관, 발렌시아 예견된 추락

류현진 WAR 투수 2위..美 매체 "비버, 압도적이지 않네?"

"김광현 불펜, 엉뚱한 선수 차버려.. C-MART 불펜 보내야" STL 담당기자

'이강인 후반 40분 투입' 발렌시아, 우에스카와 1-1 무

답답했던 라이프치히, 황희찬 들어가자 풀렸다

'163km' 토론토 최고 파이어볼러 등장.."몬토요에게 페라리 생겼다"

"류현진 162경기였다면 WAR 7.0" 박찬호-추신수 넘을 기회 놓쳤나

'미친 활약' 손흥민, '더 브라위너-호날두' 넘고 英매체 선정 유럽 랭킹 1위

터키 '코로나 비상'·中 장고 끝 외인 출전 허용, 김연경의 탁월한 선택

'황희찬 후반 교체투입' 라이프치히, 레버쿠젠과 1-1 무승부

솔샤르 논란의 농담, "골대 측정하는 무리뉴 없어 다행"

맨유, '4골대' 행운 속 브라이튼에 3-2 승리

페르난데스 결승골, 9년 만에 '가장 늦게 터진 골'

적수가 없는 아데산야..20연승 전설 쓸까

KCC 전창진 감독 "아무 생각이 없다, 너무 충격이 커서"

KLPGA투어서 하루에 홀인원 3개..역대 두 번째

"X같은 중국인" 파문.. 네이마르, 20경기 출장 정지 처분?

황희찬, 후반전만 뛰고 평점 6.1..슈팅 1회 시도

김기훈, 색다른 6⅓ 롱릴리프 쾌투..선발진 새 옵션될까?

이강인, 2025년까지 재계약 합의..조건은 '올림픽 출전'

마에다, 연봉보다 2배 많은 보너스 "돈 아깝지 않아" 찬사

'답답해서 내가 한다'..체흐 기술고문, 첼시 GK 직접 코칭

"류현진이 쉰다면, 워커 1차전" 캐나다 기자 예상

'7연승' 1위 NC의 막판 스퍼트..키움 점점 힘드네

비길 뻔한 맨유 살려낸 VAR, 종료 휘슬 울린 뒤 PK 판정

미도의 도발.."날 내친 쿠만, 메시한테도 똑같이 해 봐"

'3위 전쟁, 불펜 총력전' LG 좌우놀이, 모두가 잘 막았다

"SON은 사생활보단 프로 의식, 베일은 골프" 토트넘 선수 SNS 분석

MLB 아메리칸리그 포스트시즌 출전 8개팀 확정

"마에다 다저스 시절 PS마다 불펜행, MIN는 1선발 보장" 美 매체

최후의 1인을 가린다! 카카오 배틀그라운드
대체불가 핵앤슬래시! 패스 오브 엑자일
뉴스홈
사회
정치
경제
국제
문화
IT
포토
TV
이슈
언론사별 뉴스
배열이력
전체뉴스
랭킹
연재
1boon

시민의 소리 게시판

- 해당 페이지에 나타난 게시글들의 제목을 수집
- 다음의 코드에 css selector를 추가하여 해당 페이지에서 게시글의 제목을 스크레이핑하는 코드를 완성하시오. 또한 과제 제출시 하단의 추가 내용을 참고하여 수집한 데이터를 csv 형태로 저장하여 해당 csv 파일도 함께 제출하시오.

In [135]:

```
url_head = "https://www.sisul.or.kr"
url_board = url_head + "/open_content/childrenpark/qna/qnaMsgList.do?pgno=1"

res = request.urlopen(url_board)
soup = BeautifulSoup(res, "html.parser")

selector = "#detail_con > div.generalboard > table > tbody > tr > td.left.title > a"
titles = []
links = []
for a in soup.select(selector):
    titles.append(a.text)
    links.append(url_head + a.attrs["href"])

print(titles, links)
```

['관리인 마스크', '어린이 대공원 쓰레기집하장 내 쓰레기 제거 요청 ', '마스크미착용으로 축구 및, 베트민턴 치는 인원이 너무 많아요.', '공원 내 마스크 착용', '청춘핫도그 점장님과 직원분께 감사드립니다', '카드결제를 거부하는 매점을 신고합니다', '참얼굴만큼예쁘고맘씨좋은 여직원을 만나 고마워서 글을남깁니다.', '놀이동산에서 불쾌함을 겪었습니다', '서문 플래카드', '간만에 친절한 아가씨를 만났어요.(놀이동산)'] ['https://www.sisul.or.kr/open_content/childrenpark/qna/qnaMsgDetail.do;jsessionid=DnNWmEotNpVEe1pmrCtO12PpJh8odLBCCp9xCJ4ocg5aHjmVVUJ0x1MU5GhDXNJL.etisw1_servlet_user?qnaid=QNAS20200917000010&pgno=1', 'https://www.sisul.or.kr/open_content/childrenpark/qna/qnaMsgDetail.do;jsessionid=DnNWmEotNpVEe1pmrCtO12PpJh8odLBCCp9xCJ4ocg5aHjmVVUJ0x1MU5GhDXNJL.etisw1_servlet_user?qnaid=QNAS20200902000003&pgno=1', 'https://www.sisul.or.kr/open_content/childrenpark/qna/qnaMsgDetail.do;jsessionid=DnNWmEotNpVEe1pmrCtO12PpJh8odLBCCp9xCJ4ocg5aHjmVVUJ0x1MU5GhDXNJL.etisw1_servlet_user?qnaid=QNAS20200826000002&pgno=1', 'https://www.sisul.or.kr/open_content/childrenpark/qna/qnaMsgDetail.do;jsessionid=DnNWmEotNpVEe1pmrCtO12PpJh8odLBCCp9xCJ4ocg5aHjmVVUJ0x1MU5GhDXNJL.etisw1_servlet_user?qnaid=QNAS20200825000003&pgno=1', 'https://www.sisul.or.kr/open_content/childrenpark/qna/qnaMsgDetail.do;jsessionid=DnNWmEotNpVEe1pmrCtO12PpJh8odLBCCp9xCJ4ocg5aHjmVVUJ0x1MU5GhDXNJL.etisw1_servlet_user?qnaid=QNAS20200818000009&pgno=1', 'https://www.sisul.or.kr/open_content/childrenpark/qna/qnaMsgDetail.do;jsessionid=DnNWmEotNpVEe1pmrCtO12PpJh8odLBCCp9xCJ4ocg5aHjmVVUJ0x1MU5GhDXNJL.etisw1_servlet_user?qnaid=QNAS20200816000002&pgno=1', 'https://www.sisul.or.kr/open_content/childrenpark/qna/qnaMsgDetail.do;jsessionid=DnNWmEotNpVEe1pmrCtO12PpJh8odLBCCp9xCJ4ocg5aHjmVVUJ0x1MU5GhDXNJL.etisw1_servlet_user?qnaid=QNAS20200813000003&pgno=1', 'https://www.sisul.or.kr/open_content/childrenpark/qna/qnaMsgDetail.do;jsessionid=DnNWmEotNpVEe1pmrCtO12PpJh8odLBCCp9xCJ4ocg5aHjmVVUJ0x1MU5GhDXNJL.etisw1_servlet_user?qnaid=QNAS20200813000002&pgno=1', 'https://www.sisul.or.kr/open_content/childrenpark/qna/qnaMsgDetail.do;jsessionid=DnNWmEotNpVEe1pmrCtO12PpJh8odLBCCp9xCJ4ocg5aHjmVVUJ0x1MU5GhDXNJL.etisw1_servlet_user?qnaid=QNAS20200730000004&pgno=1', 'https://www.sisul.or.kr/open_content/childrenpark/qna/qnaMsgDetail.do;jsessionid=DnNWmEotNpVEe1pmrCtO12PpJh8odLBCCp9xCJ4ocg5aHjmVVUJ0x1MU5GhDXNJL.etisw1_servlet_user?qnaid=QNAS20200728000002&pgno=1']

In [137]:

```
import pandas as pd

board_df = pd.DataFrame({"title": titles, "link": links})
board_df.head()
```

Out[137]:

	title	link
0	관리인 마스크	https://www.sisul.or.kr/open_content/childrenp...
1	어린이 대공원 쓰레기집하장 내 쓰레기 제거 요청	https://www.sisul.or.kr/open_content/childrenp...
2	마스크미착용으로 축구 및, 베트민턴 치는 인원이 너무 많아요.	https://www.sisul.or.kr/open_content/childrenp...
3	공원 내 마스크 착용	https://www.sisul.or.kr/open_content/childrenp...
4	청춘핫도그 점장님과 직원분께 감사드립니다	https://www.sisul.or.kr/open_content/childrenp...

In [138]:

```
board_df.to_csv("board.csv", index=False)
```